ZNALOSTI 2006

Sborník příspěvků 5. ročníku konference
Proceedings of the 5th annual conference

Fakulta elektrotechniky a informatiky
VŠB – Technická univerzita Ostrava

ISBN 80–248–1001–8

ZNALOSTI 2006

http://fim.uhk.cz/znalosti/

5. ročník konference, Hradec Králové
1. – 3. února 2006
Sborník příspěvků

5th Annual Conference, Hradec Králové
1st – 3rd February 2006
Proceedings

Pořádající instituce (organized by)
Univerzita Hradec Králové, Fakulta informatiky a managementu
Technická univerzita v Košiciach, FEI, Centrum pre informačné technológie
VŠB - Technická univerzita Ostrava, FEI, Katedra informatiky
VŠE Praha, FIS, Katedra informačního a znalostního inženýrství
ČVUT Praha, FEL, Gerstnerova laboratoř
Masarykova univerzita v Brně, FI, Laboratoř vyhledávání znalostí
Univerzita Karlova v Praze, MFF, Katedra softwarového inženýrství

ZNALOSTI 2006
© Ján Paralič, Jiří Dvorský, Michal Krátký, editors

Všechna práva vyhrazena podle autorského zákona. Jakákoliv, i částečná, reprodukce či publikace pouze se souhlasem editora.

This work is subject to copyright. All rights reserved. Reproduction or publication of this material, even partial, is allowed only with the editor's permission.

Technická redakce:
Jiří Dvorský, jiri.dvorsky@vsb.cz
Michal Krátký, michal.kratky@vsb.cz

Fakulta elektrotechniky a informatiky,
VŠB–Technická univerzita Ostrava

Počet stran: 344
Náklad: 110 ks
Vydání: 1.
Rok vydání: 2006

Pro sazbu a sestavení sborníku byl použit systém LaTeX.
Tisk a knihařské práce provedl TiskServis Jiří Pustina, Petra Křičky 24, Ostrava 1.

Vydala Fakulta elektrotechniky a informatiky, VŠB–Technická univerzita Ostrava.

Předmluva

Je pre mňa veľkou poctou, že môžem uviesť zborník z jubilejného piateho ročníka konferencie ZNALOSTI, ktorá sa tohto roku koná na pôde Fakulty informatiky a managemetu, Univerzita Hradec Králové. Odborný program konferencie už tradične tvoria tutoriály, pozvané prednášky, dlhé a skrátené príspevky, prezentácie výskumných projektov, postre a panelová diskusia.

Z ponúknutých príspevkov bolo nakoniec prijatých 43 v nasledovnej štruktúre: 10 dlhých príspevkov, 10 skrátených príspevkov, 6 prezentácií výskumných projektov a 17 postrov. Prijaté príspevky ktoré nájdete v tomto zborníku sa dotýkajú väčšiny hlavných tém konferencie, najmä však týchto:

- získavanie znalostí z databáz a textov,
- vyhľadávánie informácií,
- metódy zhlukovej analýzy,
- znalostné aspekty gridového počítania,
- spracovanie prirodzeného jazyka,
- analýza webu,
- sémantický web,
- manažment znalostí,
- znalostné ontológie a ďalšie konceptuálne modely,
- multi-agentové technológie.

Na tomto mieste by som chcel poďakovať všetkým, ktorí sa podieľali na príprave konferencie a tohto zborníka. Predovšetkým ďakujem autorom jednotlivých typov zaslaných príspevkov. Som veľmi rád, že moje pozvanie prijali autori veľmi zaujímavých pozvaných prednášok - Peter Dolog, Martin Džbor, Andreas Rauber, Ján Šefránek, ako aj autori tutoriálov - Vladimír Bureš s Pavlom Čechom, Jirka Kosek a v neposlednej rade Zdeněk Molnár, organizátor panelovej diskusie.

Ďakujem všetkým členom riadiaceho a programového výboru za dôsledný a trpezlivý prístup k dvojkolovému recenznému pokračovaniu a cenné rady a pomoc pri príprave tohto ročníka konferencie ZNALOSTI. Mnoho zaujímavých nápadov som obdržal najmä od Vojtěcha Svátka.

Zvláštne poďakovanie si zaslúži organizačný výbor z Fakulty informatiky a managementu, Univerzita Hradec Králové, vedený Petrom Mikuleckým, za organizáciu tohto ročníka konferencie ZNALOSTI, ako aj Michal Krátký a Jiří Dvorský za technickú redakciu tohto zborníka.

Ján Paralič
předseda programového výboru
ZNALOSTI 2006

Preface

It is a great pleasure for me to present the proceedings of the 5th anniversary conference ZNALOSTI (Knowledge), held at the Faculty of informatics and management, University Hradec Králové. The scientific program of this conference traditionally consists of tutorials, invited talks, long and short papers, presentations of large research projects, posters and a panel discussion.

There have been finally accepted 43 contributions in the following structure: 10 long papers, 10 short papers, 6 presentations of research projects and 17 posters. Accepted papers presented in this proceedings cover most of the themes defined for this conference, in particular the following ones:

- knowledge discovery from databases and texts,
- information retrieval,
- cluster analysis,
- knowledge aspects of grid computing,
- natural language processing,
- web analysis,
- semantic web,
- knowledge management,
- ontologies and other conceptual models,
- multi-agent technologies.

I would like to thank everybody who helped to organize this conference and prepare the proceedings. First of all I would like to thank authors of all types of contributions. I am glad that my invitation accepted authors of very interesting and stimulating invited talks - Peter Dolog, Martin Džbor, Andreas Rauber, Ján Šefránek, as well as authors of tutorials - Vladimír Bureš with Pavol Čech, Jirka Kosek and last but not least to Zdeněk Molnár, who organized the panel discussion.

Many thanks also to all members of the steering and program committees for their liable and patient approach to the two-rounds review process and valuable comments to the organization of this year conference ZNALOSTI. In particular, I have got many interesting suggestions from Vojtěch Svátek.

My special thanks go to the organizational committee from the Faculty of informatics and management, University Hradec Králové, lead by Peter Mikulecký, for organization of this year conference ZNALOSTI, as well as to Michal Krátký and Jiří Dvorský for their work with finalizing this proceedings.

Ján Paralič
Program Committee Chair
ZNALOSTI 2006

Organizace (Organization)

Řídící výbor série konferencí Znalosti (Steering Committee)

Předseda (Chair):
Jan Rauch (Vysoká škola ekonomická v Praze)

Členové (Members):
Ján Paralič (Technická univerzita v Košiciach)
Jaroslav Pokorný (MFF, Univerzita Karlova v Praze)
Lubomír Popelínský (FI, Masarykova univerzita v Brně)
Václav Snášel (FEI, VŠB–Technická univerzita Ostrava)
Vojtěch Svátek (Vysoká škola ekonomická v Praze)
Olga Štěpánková (FEL, České vysoké učení technické v Praze)

Programový výbor (Program Committee)

Předseda (Chair):
Ján Paralič (Technická univerzita v Košiciach)

Členové (Members):
Roman Barták (MFF, Univerzita Karlova v Praze)
Mária Bieliková (Slovenská technická univerzita v Bratislave)
Petr Berka (Vysoká škola ekonomická v Praze)
Radim Bělohlávek (UP Olomouc)
Ivan Bruha (McMaster University, Hamilton, Kanada)
Peter Dolog (University of Hannover, Německo)
Jiří Dvorský (VŠB–Technická univerzita Ostrava)
Martin Džbor (Open University, Milton Keynes, UK)
Jan Hajič (Univerzita Karlova v Praze)
Marek Hatala (Simon Fraser University Surrey, Kanada)
Ladislav Hluchý (ÚI SAV Bratislava)
Petr Hujňák (Per Partes Consulting, Praha)
Dušan Húsek (Ústav informatiky, AV ČR, Praha)
Jiří Iványek (Ústav teorie informace a automatizace, AV ČR Praze)
Karel Ježek (Západočeská univerzita v Plzni)
Petr Jirků (Univerzita Karlova v Praze)
Radim Jiroušek (Ústav teorie informace a automatizace, AV ČR, Praha)
Zdeněk Kouba (FEL, České vysoké učení technické v Praze)
Stanislav Krajči (Univerzita Pavla Jozefa Šafárika v Košiciach)
Michal Krátký (VŠB–Technická univerzita Ostrava)
Miroslav Kubát (University of Miami, Coral Gables, Florida)
Marián Mach (Technická univerzita v Košiciach)
Vladimír Mařík (FEL, České vysoké učení technické v Praze)

Peter Mikulecký (Univerzita Hradec Králové)
Ludmila Mládková (Vysoká škola ekonomická v Praze)
Pavol Návrat (Slovenská technická univerzita v Bratislave)
Richard Papík (Univerzita Karlova v Praze)
Michal Pěchouček (FEL, České vysoké učení technické v Praze)
Jaroslav Pokorný (MFF, Univerzita Karlova v Praze)
Daniela Ponce (Univerzita Hradec Králové)
Lubomír Popelínský (FI, Masarykova univerzita v Brně)
Jan Rauch (Vysoká škola ekonomická v Praze)
Tomáš Sabol (Technická univerzita v Košiciach)
Vilém Sklenák (Vysoká škola ekonomická v Praze)
Václav Snášel (VŠB–Technická univerzita Ostrava)
Vojtěch Svátek (Vysoká škola ekonomická v Praze)
Jana Šarmanová (VŠB–Technická univerzita Ostrava)
Jan Šefránek (Univerzita Komenského, Bratislava)
Milan Šimůnek (Vysoká škola ekonomická v Praze)
Olga Štěpánková (FEL, České vysoké učení technické v Praze)
Peter Vojtáš (Univerzita Pavla Jozefa Šafárika v Košiciach)
Zdeněk Zdráhal (Open University, Milton Keynes, Velká Británie)
Jaroslav Zendulka (Vysoké učení technické v Brně)
Jana Zvárová (Ústav informatiky, AV ČR, Praha)
Filip Železný (FEL, České vysoké učení technické v Praze)
Jan Žižka (Masarykova univerzita v Brně)

Externí recenzenti (External Reviewers)

Peter Bednár (Technická univerzita v Košiciach)
Emil Gatial (Ústav informatiky, Slovenská akadémia vied, Bratislava)
Michal Laclavík (Ústav informatiky, Slovenská akadémia vied, Bratislava)
Rastislav Lencses (Univerzita Pavla Jozefa Šafárika v Košiciach)
Kristína Machová (Technická univerzita v Košiciach)
Lenka Nováková (FEL, České vysoké učení technické v Praze)
Pavel Pecina (MFF, Univerzita Karlova v Praze)
Monika Žáková (FEL, České vysoké učení technické v Praze)

Organizační výbor (Organizing Committee)

Předseda (Chair):
Peter Mikulecký (Univerzita Hradec Králové)

Členové (Members):
Vladimír Bureš (Univerzita Hradec Králové)
Pavel Čech (Univerzita Hradec Králové)
Jiří Dvorský (VŠB–Technická univerzita Ostrava)
Václav Freylich (Univerzita Hradec Králové)
Helena Holubičková (Univerzita Hradec Králové)

Michal Krátký (VŠB–Technická univerzita Ostrava)
Kateřina Mišičková (Univerzita Hradec Králové)
Michaela Nováková (Univerzita Hradec Králové)
Kamila Olševičová (Univerzita Hradec Králové)
Daniela Ponce (Univerzita Hradec Králové)
Hana Šrámková (Univerzita Hradec Králové)

Místo konání (Conference Location):

Fakulta informatiky a managementu, Univerzita Hradec Králové
1.–3. února 2006

`http://fim.uhk.cz/znalosti/`
`http://www.cs.vsb.cz/znalosti/`

Obsah (Contents)

Část I (Part I)

Tutoriály (Tutorials)

Knowledge Management at the Organisational Level 1
Vladimír Bureš, Pavel Čech

Topic Maps 5
Jiří Kosek

Zvané přednášky (Invited Papers)

Knowledge Representation and Reasoning in Personalized Web-Based e-Learning Applications 6
Peter Dolog

Evolution of the idea of the Semantic Web and its implications on educational practice 13
Martin Džbor

Mining Data to Extract Semantic Concepts 26
Andreas Rauber

Reprezentácia neúplných a nekonzistentných znalostí 29
Ján Šefránek

Vybrané příspěvky (Contributed Papers)

Dlouhé příspěvky (Long Papers)

Mining in Complex Medical Data – a Case Study 37
Petr Aubrecht, Martin Kejkula, Petr Křemen, Lenka Nováková, Jan Rauch, Milan Šimůnek, Olga Štěpánková, Monika Žáková

Towards Semantic Services Grid in Flood-Forecasting Simulations 48
Marian Babik, Ondrej Habala and Ladislav Hluchy

Using Algorithms of Computational Geometry for Pattern Recognition and Comparison to Support Vector Machine 60
Marek Bundzel and Tomáš Kasanický

Kombinácia redukcie problému a fuzzy FCA prístupu pri tvorbe konceptuálnych modelov z textových dokumentov 71
Peter Butka

Využití shlukování pro kompresi dat 83
Jiří Dvorský, Jan Martinovič, Václav Snášel

Algoritmy na vyhľadávanie najlepších k objektov bez priameho prístupu 95
Peter Gurský

Resolution Based Reasoning in Description Logic 106
Hashim Habiballa

Ferda, nové vizuální prostredí pro dobývání znalostí 118
Michal Kováč, Tomáš Kuchař, Alexander Kuzmin, Martin Ralbovský

Mining situations and actions from news 130
Luboš Popelínský and Jan Blaťák

Báze znalostí a poskytování informací obsluze linky technické podpory 139
Vladimír Sklenář, Martin Radvanský

Kratké příspěvky (Short Papers)

Refinement of case retrieval for Grid service performance prediction through semantic description of input data 151
Zoltán Balogh, Emil Gatial, Michal Laclavík, Ladislav Hluchý

Aktívne učenie pre kategorizáciu textov 159
Peter Bednár

Efektivní marginalizace mnohadimenzionálních kompoziciálních modelů 167
Vladislav Bína

Visualization of cosymmetric association rules using Hasse diagrams and concept lattices 175
Michal Burda

Případové usuzování v expertním systému NEST 183
Vladimír Laš, Petr Berka

Information Retrieval from Web Pages Employing Classification and Clustering Methods 191
Kristína Machová, Valentín Maták, and Peter Bednár

Využití suffixových stromů pro vylepšení vektorového dotazu 199
Jan Martinovič, Tomáš Novosád, Václav Snášel

Práce s tacitními znalostmi v České republice 207
Ludmila Mládková

A comparison of two approaches to temporal granularity modelling 215
Daniela Ponce

A GUHA Style Association Mining Using Prefix Tree Data Structure 223
Lukáš Vlček

Část II (Part II)

Projekty (Projects)

Enabling Grids for E-sciencE – The EU EGEE Project 233
Luděk Matyska

Nástroje pre získavanie, organizovanie a udržovanie znalostí v prostredí heterogénnych informačných zdrojov 237
Pavol Návrat, Mária Bieliková

Workflow in the Digital Mathematics Library Project 243
Petr Sojka

Projekt M-CAST: vývoj vícejazyčného systému agregace informací 248
Vilém Sklenák, Petr Strossa

Automatické generování vědeckých portálů 252
Pavel Smrž

The *Rainbow* Project: Multiway Analysis of Website Content and Structure 256
Vojtěch Svátek

Postery (Posters)

Využitie atribútovo orientovanej indukcie na zlepšenie výsledkov zhlukovej analýzy 260
František Babič

Medicínská data a jejich analýza (data mining) 264
Ladislav Beránek

Znalostní strategie pro znalostní management 268
Vladimír Bureš

Užití grafických struktur při práci s texty 272
Jiří Jelínek

Vizualizace textových dat pomocí grafu 276
Jiří Jelínek, David Kunčar, Jiří Přibil

Possible Interpretations of Binary Factor Analysis 280
Aleš Keprt

Příprava dat pro clickstream analýzu 284
Tomáš Kliegr

Agent Architecture based on Semantic Knowledge Model 288
Michal Laclavík, Zoltán Balogh, Emil Gatial, Ladislav Hluchý

O význame slovných druhov pre klasifikáciu dokumentov v slovenskom jazyku . . 292
Marián Mach

Management znalostí pro inteligenci prostředí ve znalostní ekonomice 296
Peter Mikulecký

Implicitní znalosti v relačních modelech a jejich odkrývání topologickou analýzou pojmových map 300
Karel Mls

TL - Komplexní komunikační jazyk pro MAS 304
Jaroslav Müller

Textmining in the Grid Environment 308
Martin Sarnovský

First considerations for applying AI techniques in LISp-Miner extension 312
Sebastian Schutte

Algebra of Approximate Pattern Matching Problem 316
Václav Snášel, Marian Mindek

OntoFarm: kolekce paralelních ontologií pro experimenty s nástroji sémantického webu 320
Ondřej Šváb, Vojtěch Svátek

Privacy Preserving Data Mining 324
Ondřej Výborný

Rejstřík autorů (Author Index) 328

Část I (Part I)

Knowledge Management at the Organisational Level

Vladimír Bureš, Pavel Čech

Faculty of Informatics and Management, University of Hradec Králové,
Rokitanského 62, 500 03 Hradec Králové

{vladimir.bures, pavel.cech}@uhk.cz

Abstract: At present we can read in the scientific literature about the two similar terms, knowledge management and management of knowledge. Although some authors regard these two terms as synomyms, there are significant differences. The workshop will emphasize some of the basic differences between these two notions and focus on the knowledge management from various perspectives. First perspective is conceptual in which the definitions of knowledge, knowledge management or intellectual capital are explained. Furthermore, some general frameworks of the knowledge management can also be included into the first perspective. The second perspective concentrates on knowledge processes in the organisations, their identification and description. It is followed by the technological perspective aimed at finding an appropriate support in the form of modern ICT. The organisational perspective then focuses on the organisational issues connected with the introduction of knowledge management in the organisation. Very important perspective is the implementation one, which shows the way how knowledge management is implemented into the organisation. The last perspective is the managerial one that is based on the use of appropriate managerial methods and approaches that foster knowledge management in the organisation.

Keywords: Knowledge Management, Perspectives on Knowledge Management, Organisational Level

1 Introduction

Nowadays, knowledge management (KM) is a very popular business concept. The reason is that knowledge, together with information, is an important constituent in decision making activities; both play a vital role when performing complex tasks and when solving problems. Therefore, businesses re-examine the ways how they treat their knowledge assets. The businesses are trying to implement a systematic and formalized approach to dealing with knowledge. However, there is often a narrow or limited view of KM, that causes such efforts to fail. Successful KM in an organisation is conditioned, not only by a process of its implementation, but also by different levels that pay attention to knowledge and its management. While supranational and national levels are above an organisational level, the basic level is below, where fundamental research dealing with knowledge is realised.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 1–4, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2 Levels of work with knowledge

KM can contribute to higher competitiveness only if the lower and higher levels, where the attention to knowledge is paid, are ensured. That is why, the systemic view to this problem is essential. A single phenomenon, including KM, needs to be embedded into the context of a greater whole. In this way, substantial relations at a single level and among particular levels are not only preserved, but also emphasized.

2.1 Supranational level

This level is the broadest and most general one and creates a basic framework for the lower levels. This level does not concern KM itself, but concepts like "knowledge economy" (KE) and "knowledge society" (KS). It is possible to define KE in the following way: "KE consists in a creation of an added value on the basis of knowledge utilisation, not only on manual labour, and there is an increasing significance of education and exploitation of scientific findings from the perspective of a country overall competitiveness" [2]. From CR's perspective, this level is ensured in a proper way. This fact is obvious from strategic documents of the European Union (EU), for example, the Presidency Conclusions from a special meeting of the European Council that was held in Lisbon. The aim of this meeting was "to agree a new strategic goal for the Union in order to strengthen employment, economic reform and social cohesion as part of a knowledge-based economy"[4].

2.2 National level

The effort to illustrate the existence of support of KM at the national level of the CR can again be supported by strategic documents. The best way to visualise the current situation, is to refer for example to the Programme Declaration of the Government of the Czech Republic, the Strategy of the economic growth of the Czech Republic, the Strategy of the human resources development for the Czech Republic or the Strategy of the Government of the Czech Republic in the EU framework.

2.3 Organisational level

An organisational level can be found below the national level. At this level, more concrete and more tangible concepts occur. The real KM is realised here. Nowadays, many organisations, including Czech companies, realise that traditional resources are not the only sources that should be managed during the transition to KE and KS. Knowledge becomes a strategic organisational resource for the 21st century that should ensure stable growth of performance and competitiveness. That is why, KM and related activities are implemented and realised in many organisations, regardless of their main goals or branch of industry. Basic principles of KM are applicable both in business organisations or manufacturing companies, educational institutions and institutions of public administration. At the organisational level, the systemic and

systematic approach to, and work with, knowledge is a relatively new idea. This new approach to business is introduced by the fusion of findings from different fields of human activities. At this level is the possibility to work with knowledge still subject to discussion. Some authors are convinced that it is not possible at this level to work with knowledge, but only to create an appropriate environment for successful knowledge production, sharing, development or usage [3].

The organizational level of KM can be considered from different perspectives. These perspectives create so called multisystem that enables better understanding of an explored object of interest. Examples of such a perspectives are conceptual perspective, process perspective, technological perspective, organizational perspective, implementation perspective, economical perspective, human resources perspective, administrative perspective or managerial perspective. Only examples of selected perspectives follow.

Conceptual perspective is very important, but also very often neglected. It deals with definitions of basic concepts like knowledge, KM or intellectual capital. Furthermore, some general KM frameworks can also be included into this perspective. The process perspective concentrates on knowledge processes in the organisations, their identification and description. These processes mostly reflect the knowledge life cycle. This perspective is followed by the technological perspective aimed at finding an appropriate support in the form of modern ICT, i.e. this perpective is mainly about implementation of different technologies that should support several KM activities. This is the essential distinction between the technological perspective and management of knowledge. The organisational perspective focuses on the organisational issues connected with the introduction of KM in the organisation. It takes into consideration formal and informal organizational structures, team working or assignment of basic responsibility for KM in organization. Very important perspective is the implementation one, which shows the way how KM is implemented into the organisation. It deals for example with methodologies of KM introduction the an enterprise. The last perspective, that is mentioned here, is the managerial one that is based on the use of appropriate managerial methods and approaches that foster KM in the organisation.

2.4 Management of knowledge

Management of knowledge (MK) is a well established technological discipline that represents the lowest, but also the most basic level, where the attention to knowledge is paid. MK is focused, for example, on data mining from databases, knowledge acquisition from an experts, information extraction from text, knowledge systems, multi-agents technologies, mark-up languages, semantic web, knowledge ontology and other conceptual models, natural language processing, etc. Therefore, KM is achieving its own goals by utilising the outputs of MK in its own activities. Managers usually do not know the principles, on which the products of MK are based.

MK is sometimes considered as KM itself. However, MK provides "only" means and tools to exercise KM in a business environment. MK is focused primarily on activities such as knowledge identification, codification, storage and share. These activities, no matter how they are important and sophisticated, cannot guarantee successful exploitation of knowledge. Outputs and products of MK serve only as an other perspective which has to be taken into account but which role must not be prioritized over the other perspectives in the organization level of KM. The reason is that without paing attention to other perspectives MK tends to be overly technological. In a long run, KM needs to be rather people focused to be successful [5].

References

1. Bousdira N. http://www.knowledgemanagement.org.uk, EU Funded Knowledge Management Projects, 2004.
2. Jahn M. et al. http://www.hospodarskastrategie.org, Strategy of the economic growth of the Czech Republic, 2005.
3. Sveiby K. http://www.sveiby.com.au/, Knowledge Management - The Viking way, 2000.
4. The European Council. http://europa.eu.int, Presidency Conclusions - Lisbon European Council, 2000.
5. Wiig K. *People Focused Knowledge Management – How Effective Decision Making Leads to Corporate Success*, Elsevier Butterworth-Heinemann, New York, 2004. ISBN 0-7506-7777-5.

Topic Maps

Jiří Kosek

Katedra informačního a znalostního inženýrství, FIS, VŠE
nám. W. Churchilla 4, 130 67, Praha 3 Žižkov

jirka@kosek.cz

Abstrakt. Cílem tutoriálu je uvést posluchače do problematiky Topic Maps. Topic Maps (mapy témat) jsou mezinárodní standard pro reprezentaci a sdílení znalostí. V praxi se nejlépe uplatňují pro organizování rozsáhlých zdrojů informací a jsou tak technologií, která se nasazuje při implementaci informačího managementu, managementu znalostí nebo v knihovnách. V poslední době, kdy roste potřeba na organizování stále většího množství informací, jsou mapy témat velice dobrou technologií pro zvládnutí problému.
Úvodní část tutoriálu seznámí posluchače se základními pojmy jako jsou témata, vztahy a výskyty témat. Poté se zaměří na problematiku identifikace témat a předmětů, což je důležité zejména při výměně a slučování map témat.
Pozornost bude věnována i vysvětlení jednotlivých standardů z oblasti Topic Maps. Zejména standardního formátu XTM a referenčního (TMRM) a datového modelu (TMDM). Seznámíme se i s jazyky pro dotazování nad mapami témat (TMQL, Tolog) a pro definici omezení nad mapou (TMCL). Zmíněny budou i atraktivní témata jako vizualizace map a jejich slučování v distribuovaném prostředí. Na závěr provedeme stručné srovnání Topic Maps s technologiemi sémantického webu.

Annotation:

Topic Maps Tutorial

Topic Maps is an international standard for knowledge integration. This tutorial is an introduction to the basic concepts of Topic Maps and its usage. Relation between Topic Maps and Semantic Web is discussed as well.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 5–5, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

Knowledge Representation and Reasoning in Personalized Web-Based e-Learning Applications

Peter Dolog

Forschungszentrum L3S, Universität Hannover,
Expo Plaza 1, 30539 Hannover, Germany

dolog@l3s.de

Abstract. Adaptation that is so natural for teaching by humans is a challenging issue for electronic learning tools. Adaptation in classic teaching is based on observations made about students during teaching. Similar idea was employed in user-adapted (personalized) eLearning applications. Knowledge about a user inferred from user interactions with the eLeanrning systems is used to adapt offered learning resources and guide a learner through them. This keynote gives an overview about knowledge and rules taken into account in current adaptive eLearning prototypes when adapting learning instructions. Adaptation is usually based on knowledge about learning resources and users. Rules are used for heuristics to match the learning resources with learners and infer adaptation decisions.

1 Introduction

E-learning applications have been introduced for two reasons:

- to aid learners in the learning process,
- to simplify access to education for people who cannot physically attend classic education

The Internet enabled distance education by providing electronic tools to support the above goals. On the other hand, increasing variability in learning style, culture, knowledge, background of learners has fostered research about methods which addresses these differences in learning.

Traditional teaching is very adaptive. Human teachers adapt to learners when lecturing, tutoring, and guiding learners. They adapt their interaction in more collaborative exercises.

User centered adaptation (or personalization) of electronic learning follows a similar idea. Personalization can be seen as a set of particular decisions about variable information sources fitting the context of learners. In order to allow for making informed decisions, information intensive applications should on one hand collect and maintain knowledge about information resources and context of their usage, and user information on the other hand. Reasoning on the knowledge can then be applied to adapt access, presentation, and navigation in the information resources.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 6–12, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

Electronic learning on the web recently is connected usually with self-directed learning by exploring a learning material. The learning material is accessible as other information sources on the web, i.e. information access on the web is realized through the hypertext paradigm. Hypertext interlinks related pieces of information (pages) and allows the user to browse through the information space. The links are provided either explicitly, encoded by authors of the pages, or they are generated automatically, for example based on the results of a query.

Guidance of a learner and support of his orientation in learning, so natural for teaching, can be at least partly realized through a personalized access. Personalized information access in this context is concerned with user-centered bias of the hyperlinks to support better the current user context. Generating links automatically taking user profiles into account is a very attractive option. However, it creates challenges as well. Solutions based on semantic web techniques to realize personalized access on the web seem promising. Key aspects of these solutions are ontologies and reasoning techniques. Ontologies represent shared and agreed upon conceptual models in a domain, which describe the main concepts of the domain and their relationships. Ontologies can thus serve as reference models for generating links in this domain, and represent hypertext, content and user information. Reasoning techniques can then work on metadata based on these ontologies, and generate links based on content, user context and user background.

2 Knowledge Representation

Semantic web technologies allow to link information objects thus moving from document centric idea of current web to more fine grained semantic structures [2]. The information objects may be media items used in learning. The metadata provides facts about the media items forming a knowledge base used for inferring personalization decisions. However, with the object paradigm we can go even further and see the objects on the semantic web as little agents which act on behalf of information providers. In learning domains these may represent for example peer experts on particular topics, tutors able to advice and train on particular skill or environment agents, where one can acquire a skill through an interactive game.

2.1 Learning Resources and Content

Besides generic metadata known in digital libraries, in learning we should be able to determine more learning oriented attributes such as learning goal of a media item, particular activity presented by a media item, intended audience of the media item, etc. Such attributes from the knowledge representation point of view can be seen as binary predicates which bind particular learning media item with a semantic information about it. The predicates can link the media items as resources also to other resources or complex objects. A set of such predicates can be seen as a story about a media item and serve for reasoning

about its possible usage. Resource Description Format (RDF) [12] is well suited for predicate/property centric metadata linking objects treated as resources on the web.

Several open specifications or standards have been proposed to represent facts about learning objects: Dublin Core and Dublin Core Terms standards [1] and Learning Object Metadata (LOM) [10] among the most used provide vocabularies for the predicates.

Structural properties like composites or learning prerequisites (background required to understand particular learning resource) are also important in inferences for personalization. The composition of learning units can be for example used to chose appropriate granularity for a learner if he realizes difficulty or is to advanced in the study [5, 9].

If a media item provides learner with knowledge on more than one topic, a semantic network of roles (importance of a knowledge item in a learning material) [6] can be used to determine appropriateness of the learning material for the learning goal posed by a learner.

The roles and types can enhance adaptation possibilities for construction of learning sequences based on user profile, annotating the position of a user in the learning resource structure, helping to identify main outcomes of a learning resource based on roles and level of concept coverage, and so on.

Furthermore, additional sequencing or relations on knowledge items can reflect different pedagogical approaches to guide a learner [9].

2.2 User

User interests, user performance in different courses, user goals and preferences, and other user features are important when searching for appropriate learning resources for particular learner and his context.

Similarly to the knowledge representation for media items, user features are represented as predicates connecting learner with semantic information about him or his classification to particular stereotypes. Open specifications serve with vocabulary for the predicates as well (e.g. IEEE PAPI and IMS LIP with some specific extensions for the ELENA project [7, 8, 3]).

The knowledge background and competencies of a learner are indexed similarly to the learning material; i.e. concepts from domain ontologies are used to classify a learner and his knowledge and skills.

3 Reasoning

Reasoning for personalized eLearning systems is based on matching learning material offers described in metadata with the descriptions of a learner. According to the level of matching, it infers whether particular resources are suitable and how they are suitable. The inferences take place at two levels: queries from a learner and queries placed in web guides for the learner. The idea is that learner

first specifies his goal by a query and after selecting particular resource he is guided through the selected learning resource.

Figure 1 describes examples of activities which support the interaction of a learner with an adaptive eLearning system. The user has the possibility to define a learning goal, or the goal is defined implicitly by a lecture as in the Personal Reader [4] system. In the personal learning assistant (PLA) [5], the user needs to define a learning goal by selecting some concepts from an appropriate ontology. The ontology contains competencies, skills, or concepts to be learned.

Query Rewriting. Inferences take place in the phase of query construction. The initial query restrictions are constructed either from user selected concepts or from metadata about a presented resource. The result variables are taken from user preferences about an environment he has been using. After a formal query is constructed from the concepts chosen by a user, additional restrictions are added based on knowledge about a user. This includes usually user preferences but the query can be restricted further by other information from his user profile.

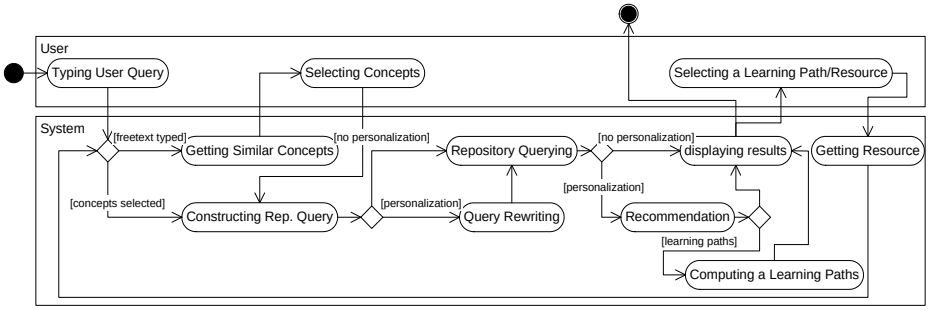


Fig. 1. Activities in adaptive system

Enhancing Query Results. If some knowledge about user was omitted in a query rewriting, it can be considered in inferences on the query results. The inferences are usually based on traversing the ontology graphs and checking whether particular nodes in the graph match to the nodes in the user profile ontology. This for example includes checking prerequisites of resources against learner performance. Language preference partial order can be traversed and used to order returned query results accordingly. Sibling nodes, father nodes, and child nodes of particular nodes in an ontology can be checked against learning performance, portfolios, skills and competencies and generate further levels in learning resources recommended to a user.

The reasoning usually concludes with additional knowledge about learning resources from a personalization point of view. This means for example that a resource is considered as recommended, not recommended or when some prerequisite of a learning resource are missing in user profile they are recommended

for particular user as well. The learning resources can be clustered into groups based on their types; e.g. examples, peer experts, additional material, discussion groups, chat messages related to learning goal of a learner or a resource presented.

Annotation and Ranking of Learning Resources. As the ontologies used on a heterogeneous environment can differ from one provider to another and from one user to another, formal reasoning can be employed also for re-annotating resources with ontologies used in a community. This includes different skill ontologies, concept ontologies, and so on. Such annotator was for example used in integration of global context into personal reader [4]. External context was considered for a JAVA tutorial, linking particular resources with related JAVA API page entries. An ontology of JAVA API was learned from JAVA class descriptions, packages, and relations between them[1]. The ontology for JAVA tutorial was used to re-annotate JAVA API pages and JAVA API ontology was used to re-annotate JAVA tutorial pages. This metadata extensions allowed for more precise querying and better fitting the context of the JAVA tutorial pages.

Similarly, an annotator is used in the HCD-Suite online[2] where different competence ontologies are used to index and classify metadata and content (based on term frequency analysis). Such extended annotations are used for ranking purposes. Search results in HCD-Suite are ranked by two different filter types, which are combined to get a weight for each learning resource to rank: *text*, and *categories*. The specified text is searched in all documents and for each document a weight is calculated. The larger the weight, the better the document meets the search criteria. For category filtering, the distances from the specified classifications (e.g. interests) in the ontology to the entries specified in the subject field from each resource are evaluated.

4 Further Challenges

Further challenges have three aspects: pedagogical, technological, and computational. From pedagogical point of view, procedural knowledge in addition to the propositional knowledge about content and user is important especially in learning at workplaces. The procedures which correspond to problem solving and related to a user activity are used to guide users through the problem in according to real workplace settings and workflows. Learners benefit from such an approach because they construct their knowledge in the context of activities they are going to perform. Several information models have been proposed to represent knowledge about learning activities such as IMS Learning Design [11, 14, 13] or IMS Simple Sequencing[3]. However, the connection between learning and work processes has not been studied yet. New EU projects in technology

[1] http://www.l3s.de/ dolog/laboratory/ss2004.html

[2] HCD-Suite online (http://www.hcm-online.com) is a result of EU/IST ELENA project http://www.elena-project.org

[3] http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_bestv1p0.html

enhanced learning, such as Prolix, TenCompetence, and COOPER, are devoted to such issues.

From technological point of view, heterogeneity of information resource is a big challenge. Information integration and approximation approaches are possibly relevant when searching large collections of heterogeneous information sources.

From the computational point of view, performance of the reasoners is a big issue, especially, when considering the large semantically interconnected collections of objects on the web. For practical applications, the performance issues related with reasoning should be researched.

Acknowledgments

I am grateful to Nicola Henze, Wolfgang Nejdl, and Michael Sintek with whom I cooperated on the issues discussed in this paper. This work was partly supported by EU/IST FP5 project ELENA - Creating Smart Spaces for Learning (IST-2001-37264) and FP6 network of excellence on professional learning PROLEARN (IST-2004-507310).

References

1. The dublin core metadata initiative. http://dublincore.org/.
2. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.
3. Peter Dolog. Identifying relevant fragments of learner profile on the semantic web. In *Proc. of SWEL'2004 — International Workshop on Semantic Web for eLearning, International Semantic Web Conference 2004*, Hiroshima, Japan, November 2004.
4. Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. The personal reader: Personalizing and enriching learning resource using semantic web technologies. In Wolfgang Nejdl and Paul De Bra, editors, *Proc. of AH2004 — International Conference on Adaptive Hypermedia*, volume 3137 of *LNCS*, Einghoven, The Netherlands, August 2004. Springer.
5. Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. Personalization in distributed e-learning environments. In *Proc. of WWW2004 — The Thirteen International World Wide Web Conference*, New Yourk, May 2004. ACM Press.
6. Peter Dolog and Wolfgang Nejdl. Using UML-based feature models and UML collaboration diagrams to information modelling for web-based applications. In *Proc. of UML 2004 — The Unified Modeling Language. Model Languages and Applications. 7th International Conference*, LNCS. Springer, October 2004.
7. Peter Dolog and Michael Schäfer. A framework for browsing, manipulating and maintaining interoperable learner profiles. In Liliana Ardissono, Paul Brna, and Antonija Mitrović, editors, *Proc. User Modeling 2005: 10th International Conference, UM 2005*, volume 3538 of *LNAI*, Edinburgh, Scotland, UK, July 2005. Springer.
8. Peter Dolog and Michael Schäfer. Learner modeling on the semantic web. In *Proc. of PerSWeb-2005 Workshop: Personalization on the Semantic Web 2005: 10th International Conference, UM 2005*, Edinburgh, Scotland, UK, July 2005.

9. Nicola Henze, Peter Dolog, and Wolfgang Nejdl. Towards personalized e-learning in a semantic web. *Educational Technology and Society Journal. Special Issue on Ontologies and the Semantic Web for E-learning*, 7(4), October 2004.
10. IEEE Learning Technology Standards Committee. IEEE standard for learning object metadata (IEEE 1484.12.1–2002). http://ltsc.ieee.org/, July 2002.
11. Rob Koper and Colin Tattersall, editors. *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training Koper*. Springer Verlag, 2005.
12. O. Lassila and R.R. Swick. W3c resource description framework (rdf) model and syntax specification. Available at: http://www.w3.org/TR/REC-rdfsyntax/. Accessed on October 25, 2002.
13. P. Rosmalen and J. Boticario. Using learning design to support design and runtime adaptation. In Rob Koper and Colin Tattersall, editors, *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, pages 291–301. Springer Verlag, 2005.
14. B. Towle and M. Halm. Designing adaptive learning environments with learning design. In Rob Koper and Colin Tattersall, editors, *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, pages 215–226. Springer Verlag, 2005.

Evolution of the idea of the Semantic Web and its implications on educational practice

Martin Džbor

Knowledge Media Institute, The Open University, UK
mdzbor@acm.org

1 Background

Education as many other disciplines attempts to take advantage of the web technology to provide and/or produce learning resources faster, easier and to tailor them to the specific needs of a learner. However, education has always relied on a strong *interpretative component*. In addition to knowledge recall, document search or information retrieval, education needs the analysis and synthesis – both on the level of individual learners and on the level of groups. The interpretation can be looked at from several angles; it comprises the capability to link otherwise independent sources of information, the capability to make statements about information sources and to make inferences using the asserted or emerging relationships. Above all, education is a highly social, interactive activity, which centers on the learners and expects them to actively participate in their education.

As the search technology on the Web becomes more robust, the access to information in general and to learning resources in particular is improving. It is an old cliché that we are now facing information overload and new means are needed to reduce the amount of information we are interacting with. From the experience of The Open University as the largest provider of open and distance education, we learned that the provision of vast libraries to the learners does not improve their learning experience nor does it accelerate the experience building. This could be because most libraries are steeped in the "web is for searching" paradigm, and support for other paradigms (e.g. exploration) is limited to rather cumbersome lists and forms.

If we want to peruse the existing web technologies in the educational domain, we need to make a shift towards the "web is for exploration" paradigm. The view of learning as an exploratory conversation has emerged in late 1990s, e.g. in the work of [19]. More specifically, we see two distinct modes of exploratory learning: (i) *convergent, 'spotlight' browsing* of internal resources, and (ii) *divergent, 'serendipitous' browsing* in an open web space. Before going on with the details about the relationships between the semantic web technologies and the different modalities of exploratory learning, let us highlight the key features of the both the standard, non-semantic Web and the emerging Semantic Web.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 13–25, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2 Briefly about the evolution of web applications

The World Wide Web has a conceptually extremely simple architecture that provides access to large amounts of content. It succeeds because it's scalable, and relies on the simplicity of its mechanisms for locating, browsing and publishing information. A web application for information or resource management is typically seen in a form of a portal [7]. A *knowledge or information portal* represents an early and (still) the most popular way of managing large but essentially, closed, data-intensive and relatively knowledge-poor repositories. As [7] found, portal-based information management is difficult – the effort to maintain the content is centralized and relies on a few administrators, the effort to integrate chunks of information is even more substantial, and the presentation of dynamic information is far from well mastered.

Another, conceptual shortcoming is that each portal reflects the perspective of its designer or a view of corporate or governmental champions. Traditionally, portals are based on "one central repository", whence users or learners retrieve knowledge to make decisions [2]. Recently, portals draw upon advances in ontological engineering [22], which enable them to express richer contextual links in the content using shared referential frameworks (ontologies). Whilst ontologies make the life of information managers easier, they have inherent limitations. The content captured and encoded using a specific ontology is *not* context-free, but supports a particular task, such as gene description in biotechnology.

A natural step in the evolution of the Web is to be achieved by the attachment of such semantic and pragmatic meanings to the distributed information. This capability is envisaged to be enabled first, through a *semantic mark-up* of web resources, and second, through *interpreting this mark-up* in a range of shared referential frames. Semantic mark-up shall not be seen as a goal in its own right. Neither shall mark-up fulfill the role of a solution for problems with interoperability, flexibility and integration we are facing. On the contrary, semantic mark-up is only one of the means that helps us – the users – to address a particular set of problems, issues and needs, and do it faster, more efficiently and more effectively.

The value of semantic mark-up is in using it in (perhaps many) particular contexts to tackle (many different) problems. Out of the two capabilities mentioned earlier, only the interpretation of semantic mark-up in different referential frames adds value to the processes that are available on the current Web. Our challenge is to use and relate the semantic mark-up data to different 'frames' we may have to make sense of a particular problem or situation (see also section 1.1). Web applications, including those for education, shall shift their current emphasis on repositories to the techniques for integrating, making use of, and comprehending diverse sources of information available in the networks on the Web. More research is needed into supporting the interpretative aspects of managing information, rather than mere retrieval.

2.1 What are the interpretative frames?

First, the concept of a 'frame' can be considered from the angle of problem solving in ill-defined situations, such as design. Based on Nakakoji's et al. analysis of design tasks [23], and on Schön's concept of *problem framing* [26], a frame can be defined

as *a point of view, which implies that certain goals exist, certain bodies of domain knowledge are relevant, and certain forms of solution are preferred* [9].

Considering the frame in the knowledge representation domain, this term is often replaced by the term 'ontology'. Hence, a frame against which the diverse semantic data and metadata are interpreted is an explicit (and often formal) specification of domain conceptualization, where conceptualization is again *a view of the world to be represented for some purpose* [13]. Ontology encodes a pragmatic vocabulary for queries and assertions [14], and it essentially circumscribes possible interpretations of the terms that are expressible and thus form 'a language' for a given domain. And languages don't tend to take form of neat hierarchies very often…

Importantly, the consequence of distinguishing between a vocabulary and a language is that sharing vocabularies does not necessarily mean sharing and having access to the same semantic metadata. Also, ontology for a semantically enriched application may be agreed in advance, and sometimes, "ontology developers may be charged to develop the ontologies for describing and indexing the content" [7]. However, ontology together with metadata is a dynamic structure that is subject to adaptation, customization and evolution. The capability to work with multiple ontologies and to evolve the ontologies alongside information retrieval and interpretation has to find its way into application development, if the Semantic Web is to enjoy the same success as its ancestor, the Web did.

2.2 Tensions on the Semantic Web

Let us thus move towards the evolving understanding of the Semantic Web and its implications on practice. We identified two main issues with the semantic mark-up: (i) the design of the ontological vocabularies to facilitate conceptually sound mark-up, and (ii) the acquisition of the actual semantic mark-up that expresses some commitments with respect to a particular domain. Both are due to the fact that unlike the 'rendering' mark-up using HTML, the semantic mark-up languages (e.g. RDF) or ontology definition languages (e.g. OWL [29]) are more complex. The costs of any mark-up are substantial, as librarians and publishers know. In order to benefit from a formal structure and reusability, we usually demand semantic annotation to be done using a particular ontology. However, the overheads for the ordinary users with committing their resources to a selected ontology are simply too great. Ontologies are hard to navigate – especially, when they take form of deep hierarchies, and expressing commitments manually is just too mundane a task.

Thus, there is a well-recognized tension between the semantic mark-up overheads and complexity on one hand side, and the dependence of the Semantic Web on a critical mass of semantic annotations for bootstrapping and sustaining itself.

Furthermore, the majority of resources that are on the Web are web pages, images and other documents, which contain the bulk of information distributed on the public Web. A majority is not semantically annotated in any way; with a small exception of the `<meta/>` tags in the HTML denoting keywords, authors, etc., which have limited semantics. Many existing web resources will never be annotated, because the authors no longer maintain them. Even if the authors wanted to, the web pages they created might be annotated differently for different user audiences.

There are, in principle, two different ways to address this tension. Over the past years, our research group in KMi has actively experimented with both. One is to facilitate the Semantic Web in the absence of pre-existing annotations. This initial solution led (back in 2002) to a suite of tools supporting a 'low-cost' browsing on the Semantic Web. The suite comprised *Magpie*, a plug-in extending the standard web browsers and an underlying infrastructure. The Magpie infrastructure later developed into a framework for developing Semantic Web applications based on automating the recognition of known instances in the browsed text and associating services with the annotated entities, which the user could invoke from the browser.

Whilst Magpie bypassed the need of having semantic annotations to some extent, it did not make the recognized, semantically marked entities available to other, non-Magpie users or agents. Thus another way to redress the same tension was to close the knowledge processing loop, so to say. Instead of making a common assumption of agreeing an ontology and acquiring knowledge once at the beginning, knowledge acquisition has to be an integral part of a user's experience. However, this integration should be hidden from the end user's interfaces – the Magpie-enhanced web browser.

Although the two approaches seem to be disjunctive alternatives, they are causally linked, and the latter approach seamlessly evolved from the former one. Because they use different paradigms to tackle the task – one more from the user interaction perspective, the other from knowledge acquisition perspective – they are in fact complementary. What they capture rather neatly is the evolution of the challenges emerging from the progress of Semantic Web research over last few years. In addition to traditional research topics, such as knowledge representation, efficient reasoning, information extraction, or expert-driven knowledge acquisition, our experience supports a few new trends.

The first generalization from the various generations of Magpie applications, relates to the *automated population of ontologies* and thus creation of semantic metadata. The second one relates to the provision of tools that are *usable by the end users of the Semantic Web*. These two trends can be put in a broader, pragmatic context. The emergence of techniques and tools aimed at end users are signals that Semantic Web research is moving from its early vision and early adopter phase to more pragmatic issues. And one such pragmatic challenge for the Semantic Web future emerged in Kalfoglou's argument: "…when it comes to promoting the Semantic Web idea to web users, as they have accumulated a 10 year experience with the Web, only a truly superior product will win them over" [17].

3 Which? (user interaction with the Semantic Web)

In the previous section, the Magpie framework has been mentioned as an initial solution to the tension between the needs and means of the Semantic Web with an aim to improve the user experience of this technology. However, Magpie is only one of the applications pioneering the emerging Semantic Web to the end users. Other tools emerged to support and assist the users with annotating documents [16]; with browsing semantically enriched data [4, 25]; with managing (semantic) services [1, 27]; or with integrating Semantic Web with other paradigms and processes [11, 15,

18]. These tools are diverse and support many different activities an end user may engage in. Yet, they can be characterized by a shared emphasis on applying realistic, real-world knowledge models, rather than developing the ideal, closed-world models.

Before one starts analyzing the likelihood of Semantic Web tools succeeding in practice, one needs to agree on a reasonably common understanding of what actually constitutes this concept. In fact, the research community is far from united in its views of what interaction with the Semantic Web means, and how the users of the Semantic Web can be supported. One view on the subject of the Semantic Web itself and its different forms have been presented in [20]. These authors see the flavours of the Semantic Web ranging from a more sophisticated infrastructure for information access, through large federated knowledge and data bases, to powerful knowledge navigators, assistants and advisors.

Rather than considering these technical differences, I would focus on what is there for the user who wants to take advantage of the Semantic Web. In the following sections, I will look at the Semantic Web from the user's (and usability) point of view. My rationale is simple – the users' experience of the actual technology depends less on the technology itself than on its presentation and on the interaction with it.

3.1 Role of familiarity and novelty in user interaction

The perspective leading to the first argument considers the relationships between developing novel interactive solutions as opposed to re-using more familiar ones.

The Semantic Web is often seen as a qualitatively different and more efficient way of processing distributed information. It has been implicitly assumed that this qualitative change can only be realized through entirely new tools and metaphors for user interaction. It is indeed true that novel approaches – for instance, new applications, new user interfaces or, most abstractly, new user interaction metaphors – are highly visible. Novelty is always good at 'headline grabbing'; a lot of attention can be generated by a radically new way of processing information.

However, there are two drawbacks of the novel user interfaces: (i) novelty is hard to sustain over a longer period, and (ii) novelty generates various forms of resistance to the proposed change. These are probably the primary reasons why so few research prototypes using semantic web technologies really made it to the market. Many have underestimated the sustainability and resistance factors.

On the contrary, 'old' (read familiar, established) tools and user interaction metaphors may be less glamorous, but they usually have an advantage. They facilitate quick buy-in of their users into novel aspects of the technology. Since new technology is introduced less radically and perhaps less visibly, there is less resistance to the proposed change from the users. Thus, one question one shall ask in connection with developing applications for the Semantic Web is: *How much radical does my application or user interaction has to be?*

I propose to ground this question in the cyclical relationship between the technology itself, the user interaction paradigm and the actual users. As Fig. 1 shows, this relationship forms a triangle, or alternatively, a closed loop. The three nodes of the triangle interact with each other and have impact on the overall user's experience with the applied technology. However, they may have different weight in different

interactive situations. Consider Mozilla web browser's XUL – an XML-based User Interface definition language[1]. This is a kind of (simple) semantic web technology used by developers to specify the user interfaces to their plug-ins or extensions. Thus, a semantic web technology is 'used' by every Mozilla user without them actually visualizing or browsing any RDF or XML specifications. The use experience in this case centers on the bottom nodes of the triangle in Fig. 1; yet it has to be considered as a use experience with a semantic web technology.

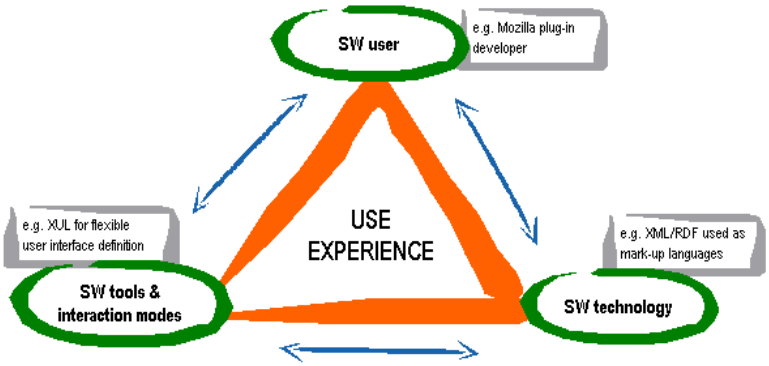


Fig. 1. User – User Interaction Paradigm – Technology triangle

This can be generalized so as to limit the number of unknowns in the triangle, or to introduce the innovations in steps rather than radically. Take the semantic web technology such as RDF for annotating documents that we want to push to the user. That makes one node in the triangle a variable. As the RDF technology is unfamiliar, it brings in uncertainty for the user, who may not have had any experience with it, may not know its benefits, etc. Now, if we decide to realize this technological change in a radically different user interface from what the user is familiar with, we are introducing another unknown into the interactive experience triangle.

Moreover, we often see even bolder assumptions that propose to change the role of the user from a document reader to a document marker. In terms of the interaction experience triangle, such assumptions are opening the third node to uncertainties, too. Such a shift in role demands may seem insignificant for the researcher, but it quickly increases the burden for an ordinary user, and thus reduces the quality of his or her experience with a particular (semantic web) technology.

Thus, uncertainty comes not only from the (semantic web) technology itself but also from the chosen tool and from the tasks we expect people to carry out. People need to learn to use the new interface, to understand it, to develop routines, etc.

In the debate on familiarity versus novelty, I would suggest applying the maxim of "doing more with less". It is well known in other sciences that the more unknowns, parameters or degrees of freedom we introduce into a complex system, the more difficult it becomes to reach the stability. And stability is important, because user routines can only develop when the system they are using is reasonably stable.

[1] See http://developer.mozilla.org/en/docs/Introduction_to_XUL for an overview.

3.2 Bootstrapping new technologies

In the previous section I illustrated the relationships between the user ('who'), the technology ('how') and the user interaction paradigm ('with what'). Another aspect to consider is 'why' – why should the user choose a particular technology and what can s/he achieve with that technology (presumably in terms of deriving some benefits).

Most researchers would agree that one purpose of the Semantic Web is to make information processing more efficient, scalable, comprehensible, transparent, and eventually, possible to automate. The user expectations from the original vision of the Semantic Web [3] are often high. In contrast, the response from the research circles is almost always conditional: "*if there were* enough of semantic mark-up (or reasoning support, etc.)". An implicit prerequisite of the Semantic Web to succeed is the presence of a critical mass of a new kind of data comprehensible to the computers for working and reasoning with.

Well, in reality there is vastly greater number of legacy data, legacy protocols and well-established legacy applications, which are not going to be scrapped any time soon. A lot of effort and resources was invested in creating web applications like digital document repositories. They indeed represent the knowledge that currently is within the organizations and on the Web at large. It is wasteful and risky to simply replace these legacy sources of data and information with something new, untested.

For the foreseeable future, legacy applications and applications using the semantic web technologies have to co-exist. In fact, the best way to show the users what are the benefits of the new paradigm, as also noticed by Kalfouglou in section 2.2, is to apply it in parallel with the old ones. Similarly as MP3 players prove their reliability and storage capacity by co-existing with CD players, the semantic web technologies can only succeed *by building upon the existing legacy.* As I argued earlier, the majority of information on the public Web is in a shape of web pages, and is not be semantically annotated by the authors.

Assuming that the critical mass of semantically marked up resources would just be there one day is not realistic. Our tools need to cope with the lack of annotation, and even more with the imperfect or rather superficial annotation. For example, taxonomies and database schemas are often frowned upon by the ontology design experts. Yet, each company already uses one form or another of some schema to classify, to order, or to warehouse its internal resources. Declaring that in order to be 'compatible' with the Semantic Web, all that needs to be done using ontology formalized in an appropriate language (OWL), is not going to improve the uptake and to help with the users' experience of the new technology.

As before with the familiarity, from our experience is seems that a suitable way to tackle the legacy information is by extending the existing rather than obliterating it. Take Magpie or KIM [24]. Where we cannot get semantic mark-up done by the authors, we may apply a visual 'semantic layer' over the legacy web page. In addition to bootstrapping the technology, we are also bootstrapping the user's experience. Simply by showing how much more can be done and how useful it can be to use the new technology, it is possible to generate a significant motivation. This motivation, in turn, may prompt some users to start tagging their data or expressing it semantically. A nice proof of such an acceptance of the legacy data can be seen in tagging images or bookmarks on sites such as flickr.com or deli.cio.us.

3.3 Managing users' expectations

I mentioned several times the relationship of new technologies and experiences on the expectations the users have or develop from a variety of sources. Visions of the future technologies are important to drive the research and development. However, reading only about these long-term prospects may greatly inflate users' expectations about the emerging technologies. Comparing a vision and promise with the status quo and all its additional requirements and constraints often leads to disillusion. There is a very fine line between the user being disillusioned and the user rejecting or resisting the novel technology, such as the Semantic Web.

We have argued that for the novel tools, applications or user interaction metaphors to succeed, they not only should be familiar, with low overheads and coping with legacy data, but also should provide some element of *reward* [10]. To appreciate the benefit of a new application, the user has to see a quick and immediate effect, improvement or at least difference. Together with [21] we started referring to this range of benefits as an instant reward. If a user decides to mark up a legacy document, s/he wants to see not only the result but mainly the benefit of this action. It is not useful to argue that time spent marking-up will be eventually useful if other people also do it. People want to show the outcomes of their activity.

Consider the classic example of authoring web pages with HTML – a few seconds after generating an image gallery as a web page or after amending the layout of the web page, these changes are available to everybody to see. There is an instant reward in publishing something on the Web – e.g. in a form of nicely rendered and formatted layout. From our experience this instant reward is very important. For example, when creating a semantic layer of annotations over a legacy text, one can 'show off' the annotations in a separate file full of angled brackets, tags and so on, or alternatively, one can superimpose those annotations on the text itself using e.g. colour swatches.

Although the 'angled brackets' approach is more formal and more useful from the long-term perspective of researchers' and developers', it is less rewarding for the users. On the contrary, text highlights expressing the same are friendlier and more rewarding in the short term. This argument aligns with the discussion on familiarity of user interaction metaphors in section 3.1. Familiarity implies faster comprehension and this, in turn, conveys the notion of instant reward much easier.

As a generalization, I suggest to 'downgrade' the visions of the Semantic Web into a series of increasingly bigger instant rewards. What are the features the users need right now? What would make their experience with the tool positive tomorrow? What would prompt the user to adopt the ideas of the Semantic Web? And what is needed to make this adoption voluntary and viral, rather than imposed through standards?

Yet, as much as is the instant reward critical to make that first step or to get the foot in the door, the instant reward is not sufficient. Instant rewards by their very definition are instantaneous and transient moments. They only act temporarily and usually for a very short time. In addition to making an impact instantaneously, the applications for the Semantic Web need to consider to notion of sustainability and so-called delayed reward [28].

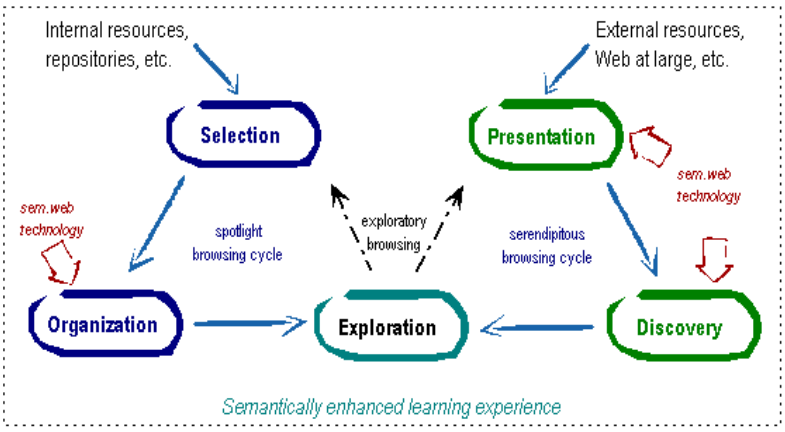


Fig. 2. Realization of exploratory interaction by exploiting semantic web technologies

4 Back to Semantic Web and (educational) practice

I have mentioned the need to see the user interaction with the Semantic Web as more than mere browsing of semantic metadata and suggested an exploratory approach to interacting with learning resources as an example. Let me elaborate on this subject by revisiting the two distinct modes of exploratory learning: (i) *convergent, 'spotlight' browsing* of internal resources [6], and (ii) *divergent, 'serendipitous' browsing* in an open web space [12].

Spotlight browsing approach to supporting exploratory learning is based on a cycle involving the *selection* of a collection of resources, the *organization* of this collection into pedagogically effective presentations, and the *exploration* of those presentations by the learner. It is suitable for the learning situation where the learner has a specific, well-defined objective; e.g. to learn more about cognitive models of memory. The concept in question becomes the center of the spotlight, and the 'shadow' cast by the spotlight leads to semantically close or similar concepts.

The serendipitous approach to exploratory learning relies on the learner engaging in a different cycle: the *presentation* of an arbitrary, serendipitous resource, the *discovery* of domain-specific anchors in this resource, and the *exploration* of a range of other resources starting from the anchors. This model is more suitable for a learner who wants to have an overview of a domain or to learn something about the structure of a particular domain without any specific objectives. Thus, learning objectives when exploring third-party resources are less well defined; e.g. generic news stories or scientific reports may serve as a jumping board to more specialized sources such as digital libraries, courseware or various models and tools supporting learning.

Both models of exploratory learning lend themselves rather well to the emerging semantic web technologies. The spotlight browsing has been piloted in a museum context by the Bletchley Park Text application and in a corporate communication context. The serendipitous browsing has been deployed in a range of domain, among them as a tool enabling learners to make sense of complex climate analyses and

climate related resources. The common ground shared by both models is their *flexibility*, *extensibility*, and *their potential for customization*. Furthermore, both models argue to bring in the semantic web technologies to the user; particularly to the user who does not want to learn deep details of a new formalism or a new query language. Our research is thus focusing on developing tools to hide the complexity of the semantic web technologies and to make them usable by a mass audience.

4.1 Three views on the web of semantics

As illustrated in section 2, Magpie is able of assigning a semantic layer based on a user-selected ontology to an arbitrary web page. There are many ways to characterize Magpie on the conceptual level. One view, emphasized in the early papers [8], is to see Magpie as a tool supporting the *interpretation of web pages*. The automatic recognition of entities in web pages is emphasized, and linking entities to semantic concepts is seen as a way to bring an interpretative context to bear, which, in turn, can help users to make sense of the information presented in a web page. In this role, Magpie occupies similar segment as e.g. SCORE [27]. For instance, in the context of a climate science course, using Magpie can be seen as adopting the viewpoint of an expert (a lecturer) in the field, and use this as an aid for navigating the web.

Another way to look at Magpie is as a *Semantic Web browser*, which can be seen as the second stage in the evolution of Magpie [10]. If we take this view, then Magpie provides an efficient way to integrate semantic and 'standard' (i.e., non-semantic) web browsing, through the automatic association of semantic annotations of a web page with the provision of simple yet effective user-interface support. This allows the user to navigate the Web using both hypertext and semantic links, which, in turn, helps him/her to work with the semantic services appropriate for a given web page. The concept of browsing using semantic relationships, conceptually proposed in the hypertext research community and piloted by COHSE [5], essentially extends a familiar metaphor from the standard Web. It allows the user to move between the chunks of information based on semantic relations or semantic proximity, rather than following physical (mostly author-defined) hyperlinks.

The first notion that influences the capability of a semantic browser to navigate between semantically annotated entities is related to recognizing concepts and entities in the free text of standard, non-annotated web documents. The second functionality of a semantic browser is to provide services that facilitate the navigation through the marked-up space. In the early Magpie applications, both functionalities were statically represented using domain ontologies and essentially did not change over the life of the application without an explicit knowledge acquisition step.

The third viewpoint, and the third evolutionary stage, we can use to characterize Magpie, is as a *framework for developing semantic web applications* [11]. According to this view, the Magpie suite of tools can be seen as a 'shell' for building Semantic Web applications, which provides generic mechanisms to bring together ontologies, web resources and (semantic) web services. The key feature of Magpie in the role of a framework for Semantic Web applications is that it allows developers to focus on the semantic functionalities, i.e., specifying and populating the ontology and defining the services, with no need to identify, let alone annotate web resources. Moreover, much

of the user interaction management is taken care by the Magpie framework, which further simplifies the design and the deployment of a semantic application.

Whatever viewpoint we take on Magpie, the key aspect of this research, and of the Magpie evolution, is that it neatly captures the challenges emerging from the progress of Semantic Web research over last few years. In addition to challenges such as knowledge representation, efficient reasoning, information extraction, annotation or expert-driven knowledge acquisition, we can generalize our experience with evolving Magpie into two new trends. We believe that these two trends are also critical for future evolution, and indeed success, of the Semantic Web. The first emerging trend, generalized from the evolution of Magpie applications, relates to the *automated population of ontologies*. The second trend, relates to a *provision of tools that are suitable and usable for the users of the Semantic Web*.

4.2 Semantic Web technologies in practice

Semantic Web technologies, as I summarized in the earlier parts of this section, are capable of covering a wide range of user activities. An example in Fig. 2 shows them to support the organization of retrieved resources, to facilitate the discovery of concepts and/or resources or to enable the presentation and inclusion of semantic knowledge into non-semantic content. Rather than choosing one of these activities as the context for a Semantic Web killer application, I argue for using the semantic web technologies to support as large sections of the overall information management cycle as is practically possible. In Fig. 2, the sample information management cycle covers a fairly flexible double loop, which was labeled as semantically enhanced learning experience. In other fields, different labels may obviously be used.

Putting the semantic web technologies to bear onto multiple nodes in the information processing cycle and attending to the entire loop rather than to its individual nodes has several useful and important side effects. First, from the user experience viewpoint, the application becomes *more interactive*. The semantically enriched application shall not confine its user to one specific activity or role; however, it shall not force the user to adopt new, unfamiliar roles either. And here is where the principle of exploratory user interaction is helpful. Integrating techniques supporting specific tasks using a shared Semantic Web framework enables the user to link the new with the old, and to appreciate the difference. The user can easily choose which sub-cycle in the overall loop (see Fig. 2) s/he wants to follow and how much of the 'Semantic Web' to allow to come to the surface.

Another side effect of the double loop introduced earlier is the *dynamics* it lends to the integrated application. Ontology-driven solutions are often brittle; they are basically closed worlds that enable reasoning solely about the known concepts. Linking the discovery activity to the presentation activity enables to overcome this brittleness, and also disguises the knowledge acquisition process so that it can be carried out by an ordinary user. The user's task may then change from the acquisition to the validation. And judging by the popularity of numerous review sites[2] and

[2] Such as the 'Which…?' series of shopping guides in the UK, see http://www.which.co.uk

rankings, people seem to be more at home with validating and ranking than with actually formally acquiring and annotating things.

Third, the double loop also shows the importance of *embedding* the semantic web features into the processes and activities, the user normally carry out. Instead of trying to convey the link between the course concepts (e.g. radiation and atmosphere) by visualizing the formal ontology or RDF graph containing these instances, the semantic component was 'layered' on top of standard web pages and rendered into standard HTML. In other words, the semantic relationship between the two or more concepts is not only visualized but, more practically, applied, used and exemplified. This approach makes it possible to show the concrete benefit or a reward in a practical setting rather than artificially imposing an abstract structure of an RDF graph onto the user for whom this method of interacting with information is alien.

Acknowledgements

Martin has been supported by the climate*prediction*.net, Advanced Knowledge Technologies (AKT), Knowledge Web, and NeOn, projects. Climate*prediction*.net was sponsored by the UK Natural Environment Research Council and the Department of Trade. AKT is an Interdisciplinary Research Collaboration (IRC) sponsored by the UK Engineering and Physical Sciences Research Council by grant no. GR/N15764/01. Knowledge Web and NeOn are supported by the IST Framework VI (grants no. FP6-507482 and FP6-027595).

References

[1] Agarwal, S., Handshuh, S., and Staab, S. *Surfing the Service Web*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 211-226.
[2] Bank, D., *Know It Alls.* The Wall Street Journal, 1996(Nov.18): p. R28.
[3] Berners-Lee, T., Hendler, J., and Lassila, O., *The Semantic Web.* Scientific American, 2001. **279**(5): p. 34-43.
[4] Buitelaar, P. and Eigner, T. *Semantic Navigation with VIeWs*. In *UserSWeb: Wksp. on User Aspects of the Semantic Web*. 2005. Crete.
[5] Carr, L., Bechhofer, S., Goble, C.*, et al. Conceptual Linking: Ontology-based Open Hypermedia*. In *10th Intl. WWW Conf.* 2001. Hong-Kong.
[6] Collins, T., Mulholland, P., and Zdrahal, Z. *Semantic Browsing of Digital Collections*. In *Proc. of the 4th Intl. Semantic Web Conf.* 2005. Ireland. p. 127-141.
[7] Corcho, O., Gomez-Perez, A., Lopez-Cima, A.*, et al. ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 802-817.
[8] Domingue, J., Dzbor, M., and Motta, E., *Semantic Layering with Magpie*, In *Handbook on Ontologies in Information Systems*, Staab, S. and Studer, R., Eds. 2003, Springer Verlag.
[9] Dzbor, M. and Zdrahal, Z. *Design as interactions between problem framing and problem solving*. In *15th European Conference on AI (ECAI)*. 2002. Lyon, France. p. 210-214.
[10] Dzbor, M., Domingue, J., and Motta, E. *Magpie: Towards a Semantic Web Browser*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 690-705.

[11] Dzbor, M., Motta, E., and Domingue, J. *Opening Up Magpie via Semantic Services*. In *Proc. of the 3rd Intl. Semantic Web Conf.* 2004. Japan. p. 635-649.
[12] Dzbor, M., Motta, E., and Stutt, A., *Achieving higher-level learning through adaptable Semantic Web aplications.* Int. J. of Knowledge and Learning, 2005. **1**(1/2): p. 25-43.
[13] Gruber, T.R., *A Translation approach to Portable Ontology Specifications.* Knowledge Acquisition, 1993. **5**(2): p. 199-221.
[14] Gruber, T.R., *Towards principles for the design of ontologies used for knowledge sharing.* Intl. Journal of Human-Computer Studies, 1993. **43**(5/6): p. 907-928.
[15] Huynh, D., Mazzocchi, S., and Karger, D.R. *Piggy Bank: Experience the Semantic Web Inside Your Web Browser*. In *Proc. of the 4th Intl. Semantic Web Conf.* 2005. Ireland. p. 413-430.
[16] Kahan, J., Koivunen, M.-R., Prud'Hommeaux, E.*, et al. Annotea: An Open RDF Infrastructure for Shared Web Annotations*. In *10th Intl. WWW Conf.* 2001. Hong-Kong.
[17] Kalfoglou, Y., Alani, H., Schorlemmer, M.*, et al. On the Emergent Semantic Web and Overlooked Issues*. In *Proc. of the 3rd Intl. Semantic Web Conf.* 2004. Japan. p. 576-590.
[18] Lanfranchi, V., Ciravegna, F., and Petrelli, D. *Semantic Web-based Document: Editing and Browsing in AktiveDoc*. In *2nd European Semantic Web Conference*. 2005. Greece.
[19] Laurillard, D., *Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies*. 2nd ed. 2002, London, UK: RoutledgeFarmer.
[20] Marshall, C.C. and Shipman, F.M. *Which Semantic Web?* In *ACM Hypertext*. UK. 2003.
[21] McDowell, L., Etzioni, O., Gribble, S.D.*, et al. Mangrove: Enticing Ordinary People onto the Semantic Web via Instant Gratification*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 754-770.
[22] Motta, E., Buckingham Shum, S., and Domingue, J., *Ontology-Driven Document Enrichment: Principles, Tools and Applications.* Intl Journal of Human-Computer Studies, 2000. **52**(5): p. 1071-1109.
[23] Nakakoji, K., Sumner, T., and Harstad, B. *Perspective-based critiquing: helping designers cope with conflicts among design intentions*. In *AI in Design*. 1994. p. 449-466.
[24] Popov, B., Kiryakov, A., Kirilov, A.*, et al. KIM - Semantic Annotation Platform*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 834-849.
[25] Quan, D., Huynh, D., and Karger, D.R. *Haystack: A Platform for Authoring End User Semantic Web Applications*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 738-753.
[26] Schön, D.A., *Reflective Practitioner - How professionals think in action*. 1983, USA: Basic Books, Inc.
[27] Sheth, A., Bertram, C., Avant, D.*, et al.*, *Managing Semantic Content for the Web.* IEEE Internet Computing, 2002. **6**(4): p. 80-87.
[28] Takeda, H. and Ohmukai, I. *Building semantic web applications as information sharing systems*. In *UserSWeb: Wksp. on User Aspects of the Semantic Web*. 2005. Crete.
[29] van Harmelen, F., Hendler, J., Horrocks, I.*, et al.*, *OWL web ontology language reference.* 2002, http://www.w3.org/TR/owl-ref/.

Mining Data to Extract Semantic Concepts

Andreas Rauber

Department of Software Technology and Interactive Systems
Vienna University of Technology,
Favoritenstrasse 9-11/188, A-1040 Wien, Austria

rauber@ifs.tuwien.ac.at

Abstract. With the abundance of data available in electronic form, sophisticated search techniques have become crucial to find useful pieces of information. Yet, this is now gradually being superseded by the need and desire to do more than merely locate information. The value does not reside primarily within single pieces of data to be located, but in the data collection as a whole. Obtaining an overview of the amount of data available, uncovering structure, semantic concepts and relationships between groups of data allows us to make use of an amount of data as a whole. In this talk we present an overview of recent work in the field of information extraction and visualization that allow us to analyze data, provide an overview, and unveil underlying concepts. Specifically, we will be addressing enhanced visualization techniques for the Self Organizing Map (SOM), a popular unsupervised neural network model, to reveal cluster structures as well as dependencies between data. We will further present applications from the fields of text and music mining, the latter specifically resulting in novel interfaces for music repositories.

1 Introduction

Until some time ago, locating specific pieces of data has been the primary goal of information retrieval research. Especially with the ever increasing volumes of data, high precision retrieval has been gaining importance, and will continue to remain a challenging field for quite some time to come. Yet, with the drastic increases in data volume, and also in high degrees of data redundancy, with the same piece of information being represented multiple times in only slightly different flavors, the focus is increasingly moving from data to actual information retrieval, and continuing via information mining to, ultimately, knowledge retrieval. While we most probably are – in spite of all promising claims and sophisticated research results – still far from really achieving sophisticated, automatic knowledge retrieval in general disciplines, advances in information mining are substantial, allowing us to slowly climb the ladder to higher levels of semantic concepts automatically extractable from data. Core to most of these techniques still is some kind of human involvement, adding, evaluating and/or confirming semantic knowledge to the information elicited from the data by sophisticated algorithms. While this is generally considered a weakness of current technology,

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 26–28, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

the holy grail being a fully automatic knowledge acquisition and evaluation system, the strengths and merits of these systems should not be underestimated, and may actually prove to be more than a mere intermediary step. Similar to the somewhat surprising potential of – by now and by comparison – rather simple data mining algorithms, a range of rather simple approaches and tools to help with in interaction and analysis of the results of data mining algorithms provide flexible and powerful techniques for obtaining more explicit information from data, and eventually move us closer to the levels of knowledge we would like to see emerge from raw data.

In this talk we will take a closer look at several techniques, picking a standard data mining method, namely the self-organizing map (SOM) [2, 3] as a basis. After a brief introduction to the basic technique, and after identifying the strength and weaknesses of the approach, we will briefly review a set of existing visualization techniques, such as the popular U-Matrix [11], as well as extensions thereof [12], clustering of component planes [14], color coding [1], smoothed data histograms (SDH) [5] and) a range of other visualization techniques [13].

We will further present some new visualization concepts, aiming specifically at the visualization of vector fields [6, 7]. In spite of the fact that all of these aim at the same goal, namely identifying structure in the data available, we will see how a combination of different techniques allows us to better interact with the results represented by the SOM. In doing so, we will specifically consider

It will also show how specific visualization and analysis techniques are particularly suitable for specific application domains and target audiences. In doing so we will touch on a range of different application domains, ranging from standard sensor data analysis [8], via text mining [9] to the analysis of music data in MP3 files [10], offering itself both for analysis as well as direct interaction [4].

References

1. J. Himberg. A SOM based cluster visualization and its application for false coloring. In S. Amari, C. L. Giles, M. Gori, and V. Puri, editors, *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, Como, Italy, July 24 – 27 2000. IEEE Computer Society.
2. T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
3. T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, 2001.
4. Robert Neumayer, Thomas Lidy, and Andreas Rauber. Content-based organization of digital audio collections. In *Proceedings of the 5th Open Workshop of MUSICNETWORK*, Vienna, Austria, July 4-5 2005.
5. E. Pampalk, A. Rauber, and D. Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proceedings of the International Conference on Neural Networks (ICANN 2002)*, pages 871–876, Madrid, Spain, August 27-30 2002. Springer.

6. Georg Pölzlbauer, Michael Dittenbach, and Andreas Rauber. Gradient visualization of grouped component planes on the som lattice. In Marie Cottrell, editor, *Proceedings of the Fifth Workshop on Self-Organizing Maps (WSOM'05)*, pages 331–338, Paris, France, September 5–8 2005.
7. Georg Pölzlbauer, Andreas Rauber, and Michael Dittenbach. Advanced visualization techniques for self-organizing maps with graph-based methods. In Zhang Yi Jun Wang, Xiaofeng Liao, editor, *Proceedings of the Second International Symposium on Neural Networks (ISNN'05)*, pages 75–80, Chongqing, China, May 30 – June 1 2005. Springer-Verlag.
8. Georg Pölzlbauer, Andreas Rauber, and Michael Dittenbach. A SOM-view of oilfield data: A novel vector field visualization for self-organizing maps and its applications in the petroleum industry. In Klaus Tochtermann and Hermann Maurer, editors, *Proceedings of the Fifth International Conference on Knowledge Management (I-KNOW'05)*, pages 502–509, Graz, Austria, June 29 – July 1 2005. J.UCS - Journal of Universal Computer Science.
9. A. Rauber and D. Merkl. Text mining in the SOMLib digital library system: The representation of topics and genres. *Applied Intelligence*, 18(3):271–293, May-June 2003.
10. A. Rauber, E. Pampalk, and D. Merkl. The SOM-enhanced JukeBox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research*, 32(2):193–210, June 2003.
11. A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In *Kohonen Maps*, pages 33–46. Elsevier Science, 1999.
12. A. Ultsch. Clustering with SOM: U*C. In *Proceedings Workshop on Self-Organizing Maps (WSOM 2005)*, pages 75–82, Paris, France, September 5-8 2005.
13. J. Vesanto. SOM-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.
14. J. Vesanto and E. Alhoniemi. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000.

Reprezentácia neúplných a nekonzistentných znalostí

Rozšírený abstrakt

Ján Šefránek

Fakulta matematiky, fyziky and informatiky, Univerzita Komenského
sefranek@fmph.uniba.sk

1 Úvod

Prednáška[1] bude venovaná hlavne teoretickému výskumu v oblasti reprezentácie znalostí, založenému na logike. Akcent bude na výpočtoch s neúplnými a nekonzistentnými znalosťami. Odvodzovanie z neúplných (nekonzistentných) znalostí je nepostrádateľná schopnosť netriviálneho znalostného systému, zvlášť ak je navrhovaný pre dynamickú doménu (alebo ak očakávame, že jeho báza znalostí sa bude postupne vyvíjať). Ak sa v báze znalostí vyskytne nekonzistentnosť, usudzujúci stroj by ju mal vedieť lokalizovať, prípadne odstrániť (na rozdiel od klasickej logiky, ktorá z nej odvodí čokoľvek). Všímať si budeme predovšetkým sémantické problémy, ale aj hľadisko vypočítateľnosti.

Súčasná teória reprezentácie znalostí považuje odvodzovanie dôsledkov (usudzovanie, reasoning) za centrálny problém, kľúčovú črtu a úlohu reprezentácie znalostí. Neseparuje odovodzovanie od reprezentácie. Súčasťou definície jazyka na reprezentáciu znalostí je špecifikácia relácie odvodenia. Nástroje na reprezentáciu poznatkov, blízke implementačnej úrovni, údajovým štruktúram nie sú z hľadiska súčasnej teórie zaujímavými a dôležitými objektami výskumu.

V prednáške najprv načrtneme históriu reprezentácie znalostí založenej na logike. Potom sa budeme venovať niektorým formalizáciám, na ktorých ukážeme, ako sa riešia problémy s neúplným a nekonzistentným poznaním. Všimneme si nemonotónne pravidlá, abdukciu, sémantiku stabilných modelov, dynamické logické programovanie. Poukážeme na negatívne zložitostné charakterizácie nemonotónneho usudzovania, načrtneme, aký pozitívny obsah tieto charakterizácie majú. Budeme sa venovať paradigme answer set programming a možnostiam rozsiahlych aplikácií v tejto paradigme.

2 Historická perspektíva

John Mc Carthy [1] podľa všetkého ako prvý prišiel s predstavou reprezentácie znalostí založenej na logike. Vlastnú prácu [1] characterizuje ako “probably the first paper on logical AI[2], i.e. AI in which logic is the method of representing information in computer memory”. Výskumný program, s ktorým prišiel, ilustroval návrhom počítačového programu, zvaného *advice taker* (AT). AT navrhoval tak, aby bol rozumný (a *program with common sense*) v tom zmysle, že “it automatically deduces for itself a sufficiently wide

[1] Práca bola podporená grantom APVV-20-P04805 a grantom VEGA 1/3112/06.
[2] artificial intelligence

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 29–36, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

class of immediate consequences of anything it is told and what it already knows". Jednou z jeho kľúčových myšlienok bola myšlienka deklaratívneho programovania (tak dnes tú predstavu nazývame). Správanie programu je determinované vetami, ktoré mu oznamujeme. Jeho správanie môžeme opraviť, ak mu oznámime iné vety. Tento spôsob špecifikácie správania programu umožňuje modularitu a nezávislosť v nasledujúcom zmysle slova: ak oznamujeme nejaké vety programu AT, nepotrebujeme vedieť nič (alebo potrebujem vedieť iba veľmi málo) o AT a o jeho predchádzajúcich poznatkoch (oveľa neskôr na označenie tejto schopnosti použil Mc Carthy slogan *elaboration tolerance* [4] – ak rozpracujeme, konzistentne doplníme poznatky systému, nemení to jeho správávanie, založené na starších poznatkoch).

Teraz budeme venovať pozornosť inej kľúčovej myšlienke, ktorá stála za projektom AT (za víziou reprezentácie znalostí, založenej na prvorádovej logike). Ide o predstavu, že je potrebné (ale aj postačujúce) zozbierať všetky potrebné poznatky a z nich "jednokrokovou" dedukciou odvodzovať dôsledky, pričom zabezpečíme, aby sa dôsledky odvodzovali iba z relevantných predpokladov. Táto vízia precenila možnosti deduktívnej logiky, ale aj ľudí, ktorých úlohou by bolo zozbierať všetky informácie potrebné a postačujúce na to, aby z nich počítač získal všetky dôsledky, ktoré získava common sense.

Výskum v oblasti umelej inteligencie (a reprezentácie znalostí), založenej na logike, skoro narazil na obmedzenia dedukcie. Ukázalo sa, že je potrebné pracovať s neúplným poznaním a riešiť nekonzistentnosti. To, samozrejme, nie je zvládnuteľné klasickou deduktívnou logikou. Pokusy s prvorádovou logikou narážali aj na výpočtovú zložitosť a nevypočítateľnosť problémov, spätých s korektným a úplným odvodzovaním.

Spomínané okolnosti a skúsenosti viedli k tomu, že priekopníci representácie znalostí, založenej na logike, hľadali iný, vhodnejší logický aparát. Aparát, ktorého vyjadrovacia sila by bola vyššia a ktorého výpočtové vlastnosti priaznivejšie. Aj sám John Mc Carthy prišiel s novou víziou. Charakterizoval ju sloganom "jumping to conclusions". Obsahom tejto predstavy bolo, že z reprezentovaných poznatkov treba vedieť odvodiť aj také dôsledky, ktoré nezískame dedukciou (dôkladnou, ale zdĺhavou). Takéto dôsledky sú vlastne hypotézami, domnienkami, ľudia ich získavajú vďaka common sensu. Logiku, ktorú navrhol Mc Carthy, nazval cirkumskripciou [2, 3]. Predpokladajme, že program dostal špecifikáciu nejakej úlohy (vyjadrenú pomocou logických formúl). Základnou myšlienkou cirkumskripcie bolo minimalizovať výnimočné stavy, odchýľky od špecifikovaného prípadu a tak získať iba dôsledky, zodpovedajúce danej špecifikácii a nezohľadňujúce všetky možné stavy sveta, ktoré by situáciu komplikovali.

Usudzovanie, formalizované pomocou cirkumskripcie je *nemonotónne*: ak sa doplnia nové poznatky, nemusia byť odvoditeľné niektoré dôsledky, ktoré boli odvoditeľné predtým. Ak množina formúl A je nadmnožinou B, potom dôsledky A nemusia byť nadmnožinou dôsledkov B. Ďalšími najznámejšími formalizáciami nemonotónneho usudzovania z konca sedemdesiatych a začiatku osemdesiatych rokov sú defaultová logika [5] a autoepistemická logika [6]. Nemonotónne usudzovanie sa stalo oblasťou horúceho výskumu v reprezentácii znalostí a je v centre pozornosti dodnes. Dosiahli sa hlboké výsledky týkajúce sa sémantických základov, výpočtovej zložitosti. Navrhli a implementovali sa sofistikované algoritmy.

Ďalší dôležitý impulz dostal výskum vreprezentácii znalsotí koncom osemdesiatych a začiatkom deväťdesiatych rokov. Výskumníci z troch oblastí – logického programova-

nia, deduktívnych databáz a nemonotónneho usudzovania – si uvedomili blízkosť, príbuznosť problémov, ktoré riešili. Každá z týchto oblastí pritom vznikla s inou motiváciou a s odlišnými cieľmi. Logické programovanie a deduktívne databázy narazili na problémy s nemonotónnosťou negácie ako konečného zlyhania. Medzi hlavné výzvy patrila uspokojujúca sémantická charakterizácia jazyka s negáciou. Na druhej strane, logické programovanie a deduktívne databázy sa ukazovali byť sľubnými prostrediami pre implementáciu nemonotónnych odvodzovacích strojov. Celkom prirodzene výskum v reprezentácii znalostí vďaka synergii týchto troch oblastí dosiahol hlbšie teoretické výsledky, objavili sa nové a úspešné sémantické konštrukcie, získali sa komplexitárske charakterizácie a vytvoril sa aj priestor pre solídne implementácie. Logické programovanie sa začalo považovať za prostriedok (prinajmenšom teoretický model) reprezentácie znalostí [7].

3 Formalizácie usudzovania s neúplnými a nekonzistentnými znalosťami

Jadro prednášky bude venované niektorým prístupom k formalizácii usudzovania s neúplnými a nekonzistentnými znalosťami, ich sémantikám, zložitostným aspektom, implementáciám. Nasleduje iba veľmi stručný prehľad tém.

3.1 Nemonotónne pravidlá

Neúplnosť (ľudských) znalostí kompenzujeme hypotetickým usudzovaním. Hypotézy (aj tie najelementárnejšie – napríklad o tom, kde je kniha, ktorú práve hľadáme) prijímame, ak ich podporuje naša doterajšia evidencia a naše všeobecnejšie poznatky. Prijímame ich dovtedy, kým ich nevyvráti ďalšia získaná evidencia alebo nejaká solídne podložená teória (i keby sa týkala iba hľadanej knihy).

Takýto druh neúplnosti nášho poznania sa často prejavuje tým, že poznáme iba čiastočne podmienky platnosti nejakého dôsledku, ale vieme identifikovať aspoň niektoré prípady, kedy neplatí. Budeme sa venovať formalizácii tohto druhu hypotetického usudzovania. Príkladom sú defaultové pravidlá [5]. Ich zovšeobecnenie – nemonotónne pravidlá – pochádza od [16]. Nemonotónnne pravidlá sú trojice tvaru $\langle \gamma, X, Y \rangle$, kde γ je dôsledok, X množina predpokladov a Y možno nazvať množinou stráží. Ak niektorý z predpokladov neplatí, dôsledok nemôžeme odvodiť podľa tohto pravidla. Podobne, ak niektorý prvok z Y je vyvrátený (platí jeho opak), pravidlo nemožno aplikovať. Inak je pravidlo aplikovateľné. Vidno, že doplnenie poznatkov môže viesť k vyvráteniu množiny stráží Y.

Prednáška bude venovaná aj sémantickej charakterizácii systémov s nemonotónnymi pravidlami. Použije sa fixpointová konštrukcia. Intuitívny zmysel konštrukcie možno vyjadriť tak, že sa špecifikuje nasýtená množina hypotéz, ktorá konzistentne zúplňuje neúplné poznanie (bázu poznatkov).

3.2 Abdukcia

Ďalším spôsobom usudzovania, ktorým sa vyrovnávame s neúplnosťou nášho poznania, je abdukcia. Pri diagnostickom usudzovaní usudzujeme z pozorovateľných prejavov

(symptómov) na ich skryté príčiny, na stavy, ktoré nie sú priamo pozorovateľné. Naše poznanie týchto vzťahov býva neúplné, odhadované príčiny alebo stavy iba hypotetické. Pri plánovaní postupne odvodzujeme dosiahnuteľnosť vzdialených cieľov z opisu počiatočnej situácie, z predpokladaného splnenia podmienok vykonania nejakých akcií a z ich očakávaných výsledkov. Opäť, dosiahnutie cieľa naplánovanými akciami nie je zaručené, efekty akcií nie sú dokonale predvídateľné, okolnosti, ktoré ich podmieňujú, nie sú spravidla opísateľné do detailov. I plánovanie, i diagnostikovanie možno modelovať abdukciou.

V deduktívnej logike sa abdukcia definuje nasledovne. Majme pozorovania Δ a relevantné poznatky Γ (teóriu v pozadí). Hovoríme, že Φ je abduktívnym vysvetlením pozorovaní Δ, ak

- $\Gamma \not\models \Delta$ (inak by nemalo zmysel hľadať vysvetlenie, naše poznanie domény by bolo z hladiska daných pozorovaní úplné),
- $\Gamma \cup \Phi \models \Delta$,
- $\Gamma \cup \Phi$ je konzistentné (v opačnom prípade by v klasickej logike bola predchádzajúca podmienka splnená triviálne).

Je prirodzené hľadať zovšeobecnenia, ktoré by prekonali obmedzenia klasickej logiky. Klasické vyplývanie nie je celkom vhodný vzťah, pozorovania i hypotézy bývajú niekedy nekonzistentné s dostupným poznaním. Jeden spôsob, ako prekonať obmedzenia, späté s klasickou logikou ukázal [17].

Prístupy k abdukcii, pri ktorých sa konštrukty klasickej logiky nahrádzajú konštruktami z nemonotónnych formalizmov, budú prezentované v prednáške. Dôraz bude na črtách, ktoré prinášajú navyše v porovnaní s deduktívnou logikou, pozri napr. [14, 15].

3.3 Výpočtová zložitosť

Pôvodné očakávania, že nemonotónne logiky umožnia, aby reprezentácie znalostí a odvodzovanie boli rýchlejie, sa nenaplnili. Komplexitárske výskumy dokonca ukázali, že výpočtová zložitosť formalizácií nemonotónneho usudzovania je podstatne vyššia než pri dedukcii. Najzávažnejšie výsledky uvedieme v prednáške, prehľad v [8, 9, 11].

Predpokladalo sa, že hlavnou príčinou negatívnych zložitostných charakterizácií je nutnosť preverovať splniteľnosť (konzistentosť) a vyplývanie (videli sme to na príklade abdukce). Ukázalo sa však, že nejde o jediné príčiny výpočtovej náročnosti hypotetického usudzovania. Ak uvažujeme radikálne limitovaný jazyk (defaultové teórie bez disjunkcie), kde sa problém splňovania a vyplývania redukuje na preverovanie členstva v množine, zložitostné charakteristiky naďalej ukazujú, že ide o ťažké problémy, pozri [10]. Hlavnou príčinou (v prípade problémov, súcisiacich s extenziami defaultových teórií) bývajú interakcie, cyklická závislosť defaultových pravidiel a potenciálny exponenciálny nárast počtu extenzií. Toto pozorovanie možno generalizovať.

3.4 Kompaktnosť reprezentácie

Veľmi zaujímavú otázku si postavili [11]: je výpočtová nezvládnuteľnosť nemonotónneho usudzovania skutočne jeho nedostatkom, nevýhodou? Podarilo sa im ukázať, že tieto

výsledky možno pochopiť ako extrémnu kompaktnosť reprezentácie v nemonotónnych formalizmoch. Použili na to techniku, ktorú nazvali kompilovateľnosťou.

Predpokladajme, že máme inštanciu (alebo viac inštancií) výpočtovo ťažkého (napr. NP-úplného) problému, representovaného v jazyku J_1, ale môžeme si dovoliť preložiť ho off-line (keď nezáleží na čase) do inej reprezentácie v jazyku J_2 a potom, neskôr, keď záleží na čase, riešiť problém v novej reprezentácii (kde riešenie bude on-line a rádovo rýchlejšie).

Príklad: nech T je formula v konjunktívnej normálnej forme, l je literál a problémom je zistiť, či $T \models l$. Trojicu $\langle T \models l; T; l\rangle$ považujme za problém s fixnou (T) a variabilnou (l) časťou. Ak off-line skonštruujeme tabuľku, v ktorej pre každý literál vyskytujúci sa v T zaznamenáme, či vyplýva z T alebo nie, môžeme hovoriť, že problém vyplývania sme skompilovali na problém prehľadávania tabuľky. Navyše, rozsah tabuľky je $\mathcal{O}(n)$, kde n je počet literálov v T. Pri konštrukcii tabuľky riešime $\mathcal{O}(n)$ inštancií co-NP-úplného problému, ale vtedy, keď potrebujeme riešenie, môžeme ho získať v lineárnom čase.

Problém $\langle P, F, V\rangle$, kde F je jeho fixná a V jeho variabilná časť, je *kompilovateľný*, ak existujú dva polynómy p_1 a p_2 a algoritmus $\mathcal{A}$ také, že pre každú inštanciu f fixnej časti F existuje údajová štruktúra D_f taká, že jej rozsah $|D_f|$ nepresahuje hodnotu $p_1(|f|)$ a pre každú inštanciu v variabilnej časti V dáva $\mathcal{A}$ pre vstupy D_f a v presne tú odpoveď ako riešenie pôvodného problému P a čas, ktorý $\mathcal{A}$ na riešenie potrebuje, neprevyšuje $p_2(|v| + |D_f|)$.

Celá táto konštrukcia má zmysel iba vtedy, ak nová reprezentácia je nanajvýš polynomicky rozsiahlejšia než pôvodná. Ukázalo sa, že pre niektoré výpočtovo nezvládnuteľné problémy, späté s nemonotónnym usudzovaním, nie je možná kompilácia do priestoru iba polynomicky rozsiahlejšieho. Interpretácia tohto výsledku je, že nemonotónny formalizmus umožňuje reprezentovať poznatky vo veľmi kompaktnom tvare [11] – iná reprezentácia by vyžadovala exponenciálne väčší priestor.

Poznamenajme na záver, že práca, ocenená ako najlepšia na ostatnej vrcholnej konferencii o reprezentácii znalostí KR 2004 používala techniku kompilácie na reprezentáciu preferencie [13]. Preferencie sú ďalšou dôležitou témou z hľadiska reprezentácie znalostí.

3.5 Answer set programming

Answer set programming je relatívne nová (koniec 20. storočia) paradigma logického programovania (radšej sa vyhnem pokusu o slovenský termín programovanie s odpoveďovými množinami). Sémantickým základom prístupu sú stabilné modely [18], neskôr zovšeobecnené na odpoveďové množiny (answer sets) pre jazyky s explicitnou negáciou [19]. Nasledovali ďalšie zovšeobecnenia pre ďalšie rozšírenia jazyka. Stabilný model si môžeme definovať ako množinu literálov S, ktorá spĺňa nasledujúcu podmienku: nech S^- je množina všetkých negatívnych litetálov z S, P je logický program a $least$ je operátor, ktorý vypočíta najmenší model logického programu bez negácie (na negatívne literály nazeráme ako na nové atómy). Potom S je stabilným modelom programu P práve vtedy, keď $S = least(P \cup S^-)$.

Základná myšlienka tejto novej paradigmy je nasledovná – vypočíta sa stabilný model programu (všetky jeho stabilné modely) a na otázky sa odpovedá (problémy sa

riešia) na základe stabilného modelu (stabilných modelov). Fakt, že môže existovať viac stabilných modelov, zväčšuje vyjadrovaciu silu formalizmu, umoňuje uvažovať alternatívne riešenia, možno rozlíšiť skeptické a dôverčivé riešenia. Samozrejme, z výpočtového hľadiska ide o ťažké problémy, nájdenie stabilného modelu je NP-úplný problém. Napriek tomu existujú sofistikované a optimalizované systémy dlv [25, 26] a smodels [23, 24], ktoré dovoľujú rozsiahle úlohy (rádovo desaťtisíce pravdiel) riešiť v solídnom čase (rádovo desiatky sekúnd).

Answer set programming sa považuje za sľubný nástroj na reprezentáciu poznatkov. Rozvíja sa rôznymi smermi – od fundamentálneho výskumu, vedúcemu k optimalizácii výpočtov až po aplikácie v takých doménach ako usudzovanie o akciách a zmene, usudzovanie o dynamických doménach, plánovanie, diagnostikovanie, integrácia dát z heterogénnych zdrojov, systém na podporu rozhodovania pre NASA, sémantický web, pozri [20–23, 25, 27–31] a mnoho ďalších. Treba zvlášť spomenúť monografiu [22].

3.6 Dynamické logické programovanie

Multidimenzionálny dynamický logický program je množina logických programov, na ktorej je definovaná relácia preferencie. Medzi programami môžu vznikať konflikty, ktoré sa riešia v súlade s reláciou preferencie [32–34]. Poukážeme na niektoré neintuitívnosti sémantiky, založenej iba na riešení konfliktov medzi hlavami pravidiel a načrtneme vlastú sémantiku, založenú na závislostiach, ktorá tieto vady odstraňuje a približuje dynamické logické programovanie ku klasickému výskumu revízií a k preferenčnému logickému programovaniu.

References

1. John Mc Carthy, Programs with common sense. Proceedings of Teddington Conference on the Mechanization of Thought Processes, 1958; http://www-formal.stanford.edu/jmc/mcc59.html
2. John McCarthy, Epistemological problems of artificial intelligence. Proceedings of IJCAI'77; http://www-formal.stanford.edu/jmc/epistemological/epistemological.html
3. John Mc Carthy, Circumscription – a form of nonmonotonic reasoning. Artificial Intelligence Journal, 1980; http://www-formal.stanford.edu/jmc/circumscription.html
4. John Mc Carthy, Elaboration tolerance. Proceedings of Common Sense 98; http://www-formal.stanford.edu/jmc/elaboration.html
5. Ray Reiter, A logic for default reasoning. Artificial Intelligence, 13:81-132, 1980.
6. R.C. Moore, Semantical considerations on non-monotonic logic, Arti cial Intelligence 25, pp. 75-94, 1985.
7. Chitta Baral, Michael Gelfond, Logic programming and knowledge representation. J. Log. Program., 19/20, 1994, 73-148, (1980), http://citeseer.ist.psu.edu/baral94logic.html
8. Georg Gottlob, Complexity results for nonmonotonic logic. Journal of Logic and Computation 2:(3), pp.397-425, 1992
9. Marco Cadoli, Marco Schaerf, A survey of complexity results for nonmonotonic logics. Journal of Logic Programming 17, pp. 127-160, 1993; http://citeseer.ist.psu.edu/cadoli93survey.html
10. Henry A. Kautz, Bart Selman, Hard problems for simple default logics. Artif. Intell., 49, 1-3, 1991, 243-279; http://citeseer.ist.psu.edu/kautz89hard.html

11. Marco Cadoli, Francesco M. Donini, Marco Schaerf: Is intractability of non-monotonic reasoning a real drawback? Artif. Intell. 88(1-2): 215-251 (1996); http://citeseer.csail.mit.edu/20108.html
12. Marco Cadoli, Francesco M. Donini, A survey on knowledge compilation. AI Commun., 10, 3-4, 1997, 137-150; http://citeseer.ist.psu.edu/cadoli98survey.html
13. Sylvie Coste-Marquis, Jerôme Lang, Paolo Liberatore, Pierre Marquis, Expressive power and succintness of propositional languages for preference representation. Principles of Knowledge representation and reasoning: Proceedings of the Ninth International Conference (KR 2004); eds.: D.Dubois, C.Welty, M-A. Williams; AAAI Press 2004.
14. Eiter, Gottlob,Leone, Abduction from logic program: semantics and complexity. Theoretical computer science, Vol. 189 , 1-2 (December 1997); http://citeseer.ist.psu.edu/eiter98abduction.html
15. Thomas Eiter, Georg Gottlob, Nicola Leone: Semantics and complexity of abduction from default theories. Artif. Intell. 90(1-2): 177-223 (1997), pp. 129 - 177, 1997; http://www.kr.tuwien.ac.at/staff/eiter/et-archive/
16. V. Wiktor Marek, Anil Nerode, Jeffrey B. Remmel: A theory of nonmonotonic rule systems I. Ann. Math. Artif. Intell. 1: (1990)
17. Martin Lang, Abdukcia bez podmienok konzistentnosti, Zmalosti 2005, 250-257
18. Michael Gelfond, Vladimir Lifschitz, The stable model semantics for logic programming, ICLP/SLP, 1988, 1070-1080
19. Michael Gelfond, Vladimir Lifschitz, Logic programs with classical negation, ICLP, 1990, 579–597, MIT Press
20. Michael Gelfond, Representing knowledge in A-Prolog, Computational Logic: Logic Programming and Beyond 2002: 413-451; www.cs.ttu.edu/ mgelfond/papers/surv2.p
21. Michael Gelfond, Nicola Leone, Logic programming and knowledge representation – the A-Prolog perspective, Artif. Intell., 138, 1-2, 2002, 3-38
22. Chitta Baral, Knowledge representation, reasoning and declarative problem solving, Cambridge University Press, 2003, ISBN: 0521818028, 544 pp.
23. Ilka Niemela, Patrick Simons, Smodels – an implementation of the stable model and well-founded semantics for normal logic programs. In Proc. LPNMR 1997
24. Smodels, http://www.tcs.hut.fi/Software/smodels/
25. Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Francesco Calimeri, Tina Dell'Armi, Thomas Eiter, Georg Gottlob, Giovambattista Ianni, Giuseppe Ielpa, Christoph Koch, Simona Perri, Axel Polleres: The DLV system. JELIA 2002: 537-540
26. The dlv project, http://www.dbai.tuwien.ac.at/proj/dlv/
27. M. Nogueira, M. Balduccini, M. Gelfond, R. Watson, M. Barry, A-Prolog decision support system for the Space Shuttle. PADL '01: Proceedings of the Third International Symposium on Practical Aspects of Declarative Languages, 2001, 169-183, Springer-Verlag
28. Thomas Eiter, Data Integration and Answer Set Programming. LPNMR 2005: Springer, 13-25
29. Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, Hans Tompits, Combining answer set programming with description logics for the semantic web. Proc. of KR 2004: 141-151
30. Thomas Eiter, Wolfgang Faber, Nicola Leone, Gerald Pfeifer, Axel Polleres: Answer set planning under action costs. J. Artif. Intell. Res. (JAIR) 19: 25-71 (2003)
31. Thomas Eiter, Wolfgang Faber, Nicola Leone, Gerald Pfeifer: The diagnosis frontend of the dlv system. AI Commun. 12(1-2): 99-111 (1999)
32. José Júlio Alferes, João Alexandre Leite, Luís Moniz Pereira, Halina Przymusinska, Teodor C. Przymusinski: Dynamic logic programming. Proc. of KR 1998: 98-111
33. João Alexandre Leite, José Júlio Alferes, Luís Moniz Pereira: Multi-dimensional dynamic knowledge representation. Proc. of LPNMR 2001, Springer, 365-378

34. João Alexandre Leite, Evolving knowledge bases. IOS Press, 2003

Annotation:
Representation of incomplete and inconsistent knowledge

Design and implementation of a non-trivial knowledge-based system tackles a task of mastering (notoriously) incomplete and – maybe – inconsistent knowledge, especially in the case of dynamically evolving domain/knowledge. This talk is devoted to the logic-based research in knowledge representation theory, which is relevant to the topic of reasoning with incomplete (and/or inconsistent) knowledge. Semantic problems are emphasized, but the viewpoint of computability cannot be ignored.

Our attention is devoted to nonmonotonic rule systems, abduction, stable model semantics of logic programs and answer set programming, dynamic logic programming, computational complexity of non-monotonic reasoning and knowledge compilation.

Mining in Complex Medical Data – a Case Study

Petr Aubrecht[1], Martin Kejkula[2], Petr Křemen[1], Lenka Nováková[1],
Jan Rauch [2], Milan Šimůnek[2], Olga Štěpánková[1], Monika Žáková[1]

[1]Czech Technical University in Prague, FEE,
Prague 6, Czech Republic,
{aubrech,kremep1,novakova,step,zakovm1}@fel.cvut.cz

[2]University of Economics, Prague, W. Churchill Sq. 4,
130 67 Praha 3, Czech Republic
{kejkula,rauch,simunek}@vse.cz

Abstract. The paper suggests a technique for searching temporal patterns, which is tested on the hepatitis data set. Relevant preprocessed time series is divided into several periods with respect to the given significant instants (for example biopsy date or start of interferon application), development trends are computed. Then dependencies in form of complex association rules between these groups are searched for. We have focused on the analysis of differences between examination results before and after interferon therapy for hepatitis C patients. To reach this goal two software systems SumatraTT and LISp-Miner are combined – sophisticated data transformations and enhancements are designed and ensured through SumatraTT while LISp-Miner takes care for the search of interesting significant differences in the preprocessed datasets.

1 Introduction

This paper presents technique and results of mining in the hepatitis data set that was offered as a part of *Discovery Challenge* at PKDD 2005. Due to promising results of searching for differences between hepatitis B and C presented in [9] we tried to use similar methodology to discover differences in time series characteristics before and after the interferon therapy for different classifications (fibrosis level, virus activity, etc.). Our procedure consists of two parts. First data preprocessing and statistical analysis was performed using the SumatraTT and R systems. Second the results of the first step were processed using the LISp-Miner system to find association rules.

SumatraTT[3] [8] is a modular system for data preprocessing and data transformations. It offers a number of easy-to-use modules for loading or exporting data from/to different formats, for analysis of individual attributes (elementary statistics, contingency tables, etc.) and for definition of additional derived attributes through sophisticated processing (scripting, integration with SQL databases etc.). The last property represents one of the main advantages of SumatraTT for the considered data-mining task: to characterize the temporal

[3] homepage: http://krizik.felk.cvut.cz/sumatra

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 37–47, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

patterns in the measured attributes of patients, it proved necessary to introduce number of new derived attributes – they have been obtained through transformations and aggregations ensured by SumatraTT. The individual modules can be combined into a project through an intuitive graphical interface which creates automatically a detailed documentation as a by-product. In this way, the SumatraTT project becomes an efficient communication platform for our team during the work on any data-mining task.

LISp-Miner is an academic software system intended to support teaching and research. It consists of six data mining procedures, machine learning procedure KEX and two procedures for data transformation [5, 7] see also `http://lispminer.vse.cz`. All six data mining procedures of LISp-Miner system are *GUHA procedures* in the sense of [1]. The input of GUHA procedure consists of the analyzed data and of a simple definition of relevant (i.e. potentially interesting) patterns. GUHA procedure automatically generates each particular pattern and tests if it is true in the analyzed data. The output of the procedure consists of all *prime patterns*. The pattern is prime if it is true in the analyzed data and if it does not immediately follow from other simpler output patterns [1]. Each GUHA procedure of the LISp-Miner system mines for a particular type of patterns. Its most frequently used procedure is 4ft-Miner, which mines for enhanced association rules [5]. In this paper, we use the procedure SD4ft-Miner [6].

The paper first describes applications of the SumatraTT, which ensure creation of several data matrices suitable for further analysis. Their content is described in section 2. The procedure SD4ft-Miner mines for SD4ft-patterns that are introduced in section 3 together with results of several applications of the SD4ft-Miner. Some concluding remarks are in section 5.

2 Transforming Data by SumatraTT

2.1 Data Understanding – Important Properties of the Dataset

The hepatitis source data is in a form of CSV files with a good documentation. Data was primarily loaded from the text files (`ilab_e030704.csv` etc.) into SQL database. We prepared several steps of data preprocessing. First, the data from both tables ilab and olab (internal and external examinations) was merged and further considered together. In the next step, non-numeric values were either fixed or ommited (values like `+␣1`, `+`, `>3` etc.).

The merged table contains 1060 different types of examinations (mainly due to disparate examination types in olab with 845 diff. examinations). To reduce this excessive number only a few exam types were selected upon specialist's recommendation.

During the data preprocessing we have identified some important properties of the considered dataset which were not explicitly mentioned in the former articles dedicated to this dataset. Some of the properties are summarised in tables 1 and 2.

# of patients	Patients with...
771	all
503	with a biopsy
99	more than 1 biopsy (all have type C)
74	more than 1 biopsy and interferon
123	exactly 1 biopsy and interferon
1	with interferon, no biopsy
306	at least 1 biopsy and no interferon
281	*exactly 1 biopsy and no interferon*
460	at least 1 exam before the first biopsy
3	at least 1 biopsy, no other exam (mid=808, 500, 202)
459	at least 1 exam in hemat table before the first biopsy

Table 1. Important numbers of the hepatitis dataset

# of exams	Description
795.851	all in the merged table (ilab+olab) after cleaning
279.228	exams of patients with application of interferon
101.716	exams before application of interferon
12.202	exams within three months before application of interferon
150.645	exams after application of interferon
9.885	exams within three months after application of interferon
195	patients related to the exams three months before and after application of interferon

Table 2. Numbers of examinations related to interferon therapy

After reviewing several summary numbers in table 1, we decided to deal with differences between temporal parameters within a fixed interval before application of interferon and after the application.

2.2 Temporal Characteristics of the Considered Attributes

The patients are not examined regularly: a period between two examinations can range form one day to several months. Some periods, when a patient is observed frequently, alternate with more restful periods. The highest number of all exams for one patient is 12659 (patient # 321). This irregularity has to be taken into account when choosing the characteristics for description of the temporal properties of the measured values. In order to standardize information provided about individual patients, we have decided to concentrate on data collected during a specific well defined time interval of a fixed length $\Delta\tau$ for each individual patient. The considered interval does not start on a single date for all the patients. On the contrary, it is tightly bound to the state of the individual patient: the considered interval ends (or begins) in a *significant instant*, which can be easily recognized in the available data of the patient, e.g. on the time when a specific treatment was introduced (e.g. interferon). The length of the time

interval is set constant for all patients and it is understood to be a parameter $\Delta\tau$ of the considered project.

The number of measurements for one patient during one year ranges usually between 2-10. To make up for this non-uniformity we have decided to use the following trend characteristics of the considered sequence of time-stamped data: average, number of measurements, gradient (resulting from linear approximation), maximum, minimum, and variance. For purposes of further data mining the results were saved as a matrix with patients in rows and trend characteristics in columns. The rest of the paper tries to prove that this type of derived attributes can depict interesting dependencies in the considered data. For that purpose we have to fix the *significant instant* and $\Delta\tau$.

In our experiments, two intervals were selected and mutually compared. The first interval ends in the *significant instant* set to the time of the beginning of interferon application, the second interval starts at the end of the interferon application. Measurements within the $\Delta\tau$ period (see further) before/after the *significant instant* were selected.

Data selected in the previous steps was then analyzed as sequences and there were calculated upper mentioned trend characteristics. Finally, data about patients was added (sex, age, type of hepatitis, maximum fibrosis and activity).

The data preprocessing resulted in a data matrix, rows of which correspond to particular patients identified by MID. Columns of this data matrix contain various trend characteristics of considered examinations for the corresponding patients (e.g. ALB_avg is an average of the values of ALB exam results).

2.3 Enhanced Datasets

The following data matrices describing behavior of various characteristics before the first biopsy were prepared for data mining:

- HEPAT-BEFORE-3 is a matrix of exam characteristics related to the interval $\Delta\tau = 3$ months ending in the *significant instant* set to the beginning of application of interferon,
- HEPAT-AFTER-3 is a matrix of exam characteristics related to the same interval $\Delta\tau = 3$ months starting in the *significant instant* set to the end of an interferon treatment,
- HEPAT-DIFF-3 is a matrix of differencies (of the corresponding fields) between HEPAT-AFTER-3 and HEPAT-BEFORE-3, i.e. HEPAT-AFTER-3 − HEPAT-BEFORE-3.

Interesting changes in some exam results (e.g. t-bil and i-bil) can be identified in the differencial matrix, this is e.g. the case of the histogram of the changes, see figure 1. The value increased in the case of 43 patients, no change was observed for 8 patients, while for 125 patients the value decreased.

The size of resulting datasets is summarized in table 2. For pilot experiments the $\Delta\tau$ was chosen to be three months.

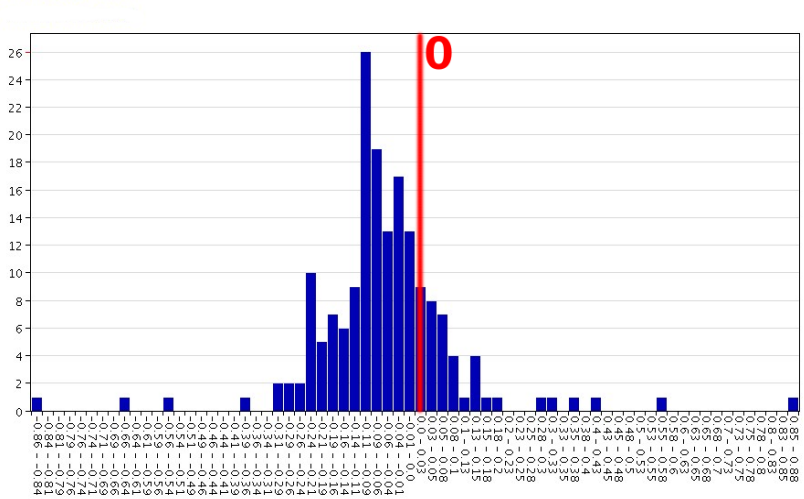


Fig. 1. Difference between i-bil exam after and before application of interferon

3 SD4ft-patterns

We use data matrix *HEPAT-DIFF-3* shown in fig. 2 to introduce the SD4ft-patterns. Each row of *HEPAT-DIFF-3* corresponds to one patient identified

row	Basic attributes			Difference of gradient of in-hospital examinations		
number	MID	Fibrosis	Activity	D_BIL_grad	G_GL_grad	...
1	2	1	1	-1.0	-0.999	...
2	16	4	2	-7.02E-14	-1.005	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱
176	960	1	2	-5.119E-4	-0.995	...

Fig. 2. Data matrix *HEPAT-DIFF–3*

by the value of column MID. Columns Fibrosis and Activity come from `bio_e030704.csv` and they indicate values at the time of the first biopsy. The value -1.0 in the column D_BIL_grad in the row 1 is the difference between the gradients of D_BIL values in the time series of the examination after interferon therapy and before interferon therapy for the patient with MID = 2. Analogously for further patients and columns. The data matrix *HEPAT-DIFF-3* contains further columns with differences of gradients of values concerning 9 specific examinations. The discretization of gradient values was done in *LISp-Miner:* equidistant intervals were used.

The procedure *SD4ft-Miner* mines for SD4ft-patterns of the form

$$\alpha \bowtie \beta : \ \varphi \approx \psi \ / \ \gamma \ .$$

Here α, β, γ, φ, and ψ are Boolean attributes defined from the columns of analyzed data matrix $\mathcal{M}$. The attributes α and β define two subsets of rows (i.e. subsets of patients in our case). The attribute γ defines a condition. The attribute φ is called *antecedent* and the attribute ψ is called *succedent*. All Boolean attributes α, β, γ, φ, and ψ are called *cedents*.

The SD4ft-pattern $\alpha \bowtie \beta : \varphi \approx \psi/\gamma$ means that the subsets of patients meeting the Boolean conditions α and β differ in what concerns the validity of association rule $\varphi \sim \psi$ when the condition given by Boolean attribute γ is satisfied. A measure of difference is defined by the symbol $\approx$ that is called *SD4ft-quantifier*. The association rule $\varphi \sim \psi$ means here a general relation of Boolean attributes φ and ψ in the sense of [5]. An example of the SD4ft-pattern is the pattern

$$\text{Fibrosis}(0) \bowtie \text{Fibrosis}(4) :$$
$$\text{GPT_grad}(\langle -1.5; -0.7\rangle) \Rightarrow^{D}_{0.45} \ \text{GOT_grad}(\langle -1.1; -0.8\rangle)$$

It means that the patients fibrosis level 0 differ from the patients with fibrosis level 4 what concerns relation of Boolean attributes GPT_grad ($\langle$-1.5;-0.7$\rangle$) (i.e. the value of the difference of gradients of the values of the GPT examinations after and before interferon therapy is in the interval$\langle$-1.5;-0.7$\rangle$).

The difference is given by the *SD4ft-quantifier* $\Rightarrow^{D}_{0.45}$. We introduce it using general notation α, β, γ, φ, and ψ. The *SD4ft-quantifier* concerns two four-fold contingency tables (i.e. 4ft-tables) $4ft(\varphi, \psi, \mathcal{M}/(\alpha \wedge \gamma))$ and $4ft(\varphi, \psi, \mathcal{M}/(\beta \wedge \gamma))$, see fig. 3.

$\mathcal{M}/(\alpha \wedge \gamma)$	ψ	$\neg\psi$
φ	$a_{\alpha\wedge\gamma}$	$b_{\alpha\wedge\gamma}$
$\neg\varphi$	$c_{\alpha\wedge\gamma}$	$d_{\alpha\wedge\gamma}$

$4ft(\varphi, \psi, \mathcal{M}/(\alpha \wedge \gamma))$

$\mathcal{M}/(\beta \wedge \gamma)$	ψ	$\neg\psi$
φ	$a_{\beta\wedge\gamma}$	$b_{\beta\wedge\gamma}$
$\neg\varphi$	$c_{\beta\wedge\gamma}$	$d_{\beta\wedge\gamma}$

$4ft(\varphi, \psi, \mathcal{M}/(\beta \wedge \gamma))$

Fig. 3. 4ft-tables $4ft(\varphi, \psi, \mathcal{M}/(\alpha \wedge \gamma))$ and $4ft(\varphi, \psi, \mathcal{M}/(\beta \wedge \gamma))$

The 4ft-table $4ft(\varphi, \psi, \mathcal{M}/(\alpha \wedge \gamma))$ of φ and ψ on $\mathcal{M}/(\alpha \wedge \gamma)$ is the contingency table of φ and ψ on $\mathcal{M}/(\alpha \wedge \gamma)$. The data matrix $\mathcal{M}/(\alpha \wedge \gamma)$ is a data submatrix of $\mathcal{M}$ that consists of exactly all rows of $\mathcal{M}$ satisfying $\alpha \wedge \gamma$. It means that $\mathcal{M}/(\alpha \wedge \gamma)$ corresponds to all objects (i.e. rows) from the set defined by α that satisfy the condition γ. It is $4ft(\varphi, \psi, \mathcal{M}/(\alpha \wedge \gamma)) = \langle a_{\alpha\wedge\gamma}, b_{\alpha\wedge\gamma}, c_{\alpha\wedge\gamma}, d_{\alpha\wedge\gamma}\rangle$ where $a_{\alpha\wedge\gamma}$ is the number of rows of data matrix $\mathcal{M}/(\alpha \wedge \gamma)$ satisfying both φ and ψ, etc. The 4ft-table $4ft(\varphi, \psi, \mathcal{M}/(\beta \wedge \gamma))$ of φ and ψ on $\mathcal{M}/(\beta \wedge \gamma)$ is defined analogously.

The *SD4ft-quantifier* $\Rightarrow^{D}_{0.45}$ is defined by the condition

$$| \frac{a_{\alpha\wedge\gamma}}{a_{\alpha\wedge\gamma}+b_{\alpha\wedge\gamma}} - \frac{a_{\beta\wedge\gamma}}{a_{\beta\wedge\gamma}+b_{\beta\wedge\gamma}} | \geq 0.45$$

This condition means that the difference between the confidence of the "classical" association rule $\varphi \rightarrow \psi$ on data matrix $\mathcal{M}/(\alpha \wedge \gamma))$ and the confidence of this association rule on data matrix $\mathcal{M}/(\beta \wedge \gamma))$ is at least 0.45. The SD4ft-pattern $\alpha \bowtie \beta : \varphi \Rightarrow^{D}_{0.45} \psi \ / \ \gamma$ is *true on data matrix* $\mathcal{M}$ if the condition $| \frac{a_{\alpha\wedge\gamma}}{a_{\alpha\wedge\gamma}+b_{\alpha\wedge\gamma}} - \frac{a_{\beta\wedge\gamma}}{a_{\beta\wedge\gamma}+b_{\beta\wedge\gamma}} | \geq 0.45$ is satisfied.

The SD-4ft pattern

$$\text{Fibrosis}(0) \bowtie \text{Fibrosis}(4) :$$

$$\text{GPT_grad}(\langle -1.5; -0.7 \rangle) \Rightarrow^{D}_{0.45} \text{GOT_grad}(\langle -1.1; -0.8 \rangle)$$

is verified using the 4ft-tables $\mathcal{F}_0$ and $\mathcal{F}_4$ see Fig. 4. Let us note that the sum of

HEPAT-DIFF-3 / Fibrosis(0)	GOT_grad($\langle -1.1; -0.8 \rangle$)	¬ GOT_grad($\langle$-1.1;-0.8$\rangle$)
GPT_grad($\langle$-1.5;-0.7$\rangle$)	8	0
¬ GPT_grad($\langle$-1.5;-0.7$\rangle$)	0	0

$\mathcal{F}_0$ = 4ft(GPT_grad($\langle$-1.5;-0.7$\rangle$),GOT_grad($\langle$-1.1;-0.8$\rangle$), *HEPAT-DIFF-3*

HEPAT-DIFF-3 / Fibrosis(4)	GOT_grad($\langle$ -1.1;-0.8 $\rangle$)	¬ GOT_grad($\langle$-1.1;-0.8$\rangle$)
GPT_grad($\langle$-1.5;-0.7$\rangle$)	11	10
¬ GPT_grad($\langle$-1.5;-0.7$\rangle$)	0	8

$\mathcal{F}_4$ = 4ft(GPT_grad($\langle$-1.5;-0.7$\rangle$),GOT_grad($\langle$-1.1;-0.8$\rangle$), *HEPAT-DIFF-3*

Fig. 4. 4ft-tables $\mathcal{F}_0$ and $\mathcal{F}_4$

all frequencies from 4ft-tables $\mathcal{F}_0$ and $\mathcal{F}_4$ is smaller than 176 because of omitting missing values "X" and patients with fibrosis levels 1, 2 and 3.

It is easy to verify that the condition corresponding to the SD4ft quantifier $\Rightarrow^{D}_{0.45}$ is satisfied.

We can conclude that the SD4ft pattern

$$\text{Fibrosis}(0) \bowtie \text{Fibrosis}(4) :$$

$$\text{GPT_grad}(\langle -1.5; -0.7 \rangle) \Rightarrow^{D}_{0.45} \text{GOT_grad}(\langle -1.1; -0.8 \rangle)$$

is true on the data matrix *HEPAT-DIFF-3*.

4 SD4ft-Miner Application Results

We solved several different tasks. In this paper we introduce mining for very simple SD4ft-patterns (without condition γ). Detailed description of further analyses can be found at `http://screwdriver.felk.cvut.cz/sumatra/articles/05gl-192/05-GL-182.pdf`.

We searched for differences of gradients of time series of several examinations after and before interferon therapy. We analyse how differ gradients among patients with different level of discovered fibrosis.

Table 3 contains some of discovered strong SD4ft patterns, describing differences between group of patients with the level of fibrosis 2 and patients with fibrosis level 4. Selected discovered patterns concerning differences between patients with the level of fibrosis 0 and the group of patients with fibrosis levels 3 or 4 are described in table 4. SD4ft patterns concerning differences between the group of patients with fibrosis level 0 and 1 and the group of patients with fibrosis level 4 are in table 5. Table 6 contains selected discovered SD4patterns describing differences between patients with fibrosis level 0 and the group of patients with fibrosis levels 2, 3 and 4.

rule	fibrosis 2			fibrosis 4			$\text{Conf}_2 - \text{Conf}_4$
	Conf_2	support		Conf_4	support		
		a_2	%		a_4	%	
D-BIL_grad $\langle -1.005; -0.995)$ $\wedge$ G-GL_grad $\langle -1.005; -0.975)$ $\rightarrow$ U-UBG_grad $\langle -0.01; 0.01)$	1.00	13	48	0.53	8	28	0.47
G-GTP_grad$\langle -1.09; -0.54)$ $\rightarrow$ U-UBG_grad $\langle -0.02; 0.01)$	0.95	21	78	0.50	10	34	0.45

Table 3. Differences concerning gradients of examinations. Fibrosis level 2 vs. level 4.

rule	fibrosis 0			fibrosis 3, 4			$\text{Conf}_0 - \text{Conf}_{3,4}$
	Conf_0	support		$\text{Conf}_{3,4}$	support		
		a_0	%		$a_{3,4}$	%	
GPT_grad $\langle -1.2; -0.3)$ $\rightarrow$ GOT_grad $\langle -1.2; -1.0)$	1.00	8	100	0.55	23	42	0.45
T-BIL_grad$\langle -1.006; -0.996)$ $\rightarrow$ GOT_grad $\langle -1.5; -1.0)$	1.00	8	100	0.52	26	47	0.48
T-BIL_grad$\langle -1.006; -0.996)$ $\rightarrow$ GOT_grad $\langle -1.1; -1.0)$	1.00	8	100	0.24	12	22	0.76

Table 4. Differences concerning gradients of examinations. Fibrosis level 0 vs, levels 3 and 4.

rule	fibrosis 0, 1			fibrosis 4			
	$\text{Conf}_{0,1}$	support		Conf_4	support		$\text{Conf}_{0,1}$
		$a_{0,1}$	%		a_4	%	$-\ \text{Conf}_4$
G-GTP_grad$\langle -1.39; -0.84)$ $\wedge$ I-BIL_grad$\langle -1.003; -0.993)$ $\rightarrow$ GOT_grad $\langle -1.1; -0.7)$	0.88	63	67	0.38	9	31	0.50
G-GL_grad$\langle -1.020; -0.995)$ $\rightarrow$ GOT_grad $\langle -1.1; -0.7)$	0.88	56	60	0.44	8	28	0.43
D-BIL_grad(≤ -0.010) $\wedge$ T-BIL_grad$\langle -1.006; -0.996)$ $\rightarrow$ GOT_grad $\langle -1.1; -0.5)$	0.88	49	52	0.44	8	28	0.43

Table 5. Differences concerning gradients of examinations. Fibrosis levels 0 and 1 vs. level 4.

rule	fibrosis 0			fibrosis 2, 3, 4			
	Conf_0	support		Conf_{2-4}	support		Conf_0
		a_0	%		$a_{2\text{-}4}$	%	$-\ \text{Conf}_{2\text{-}4}$
GOT_grad$\langle -1.8; -1.0)\wedge$ T-BIL_grad$\langle -1.006; -0.996)$ $\rightarrow$ GPT_grad $\langle -1.1; -0.8)$	1.00	8	100	0.55	22	27	0.45
G-GL_grad$\langle -1.020; -0.995)$ $\rightarrow$ GOT_grad $\langle -1.1; -0.7)$	0.88	56	60	0.44	8	28	0.43

Table 6. Differences concerning gradients of examinations. Fibrosis level 0 vs. levels 2, 3 and 4.

We tried also to find some conditions under which is the difference of confidences even stronger. We searched for SD4ft-patterns of the form more $\alpha \bowtie \beta : \varphi \approx \psi \ / \ \gamma$ where where the condition γ was created from Sex, Age, Fibrosis and Activity. We were also mining with success for patterns describing differences between groups of patients with hepatitis type B and C. About $6.2 * 10^6$ relevant patterns was verified. The amount of 76 true SD4ft-patterns with condition were found in 2 minutes and 23 seconds. Some examples of the strongest and the most interesting ones are reviewed at `http://screwdriver.felk.cvut.cz/sumatra/articles/05gl-192/05-GL-182.pdf` and [9].

5 Conclusions and Further Work

We have succeeded to find several patterns that indicate differences between time series of examination after and before interferon therapy among groups of patients with different fibrosis levels. The process is far from straightforward.

First, it was necessary to transform original data into suitable data matrix using SumatraTT and then the SD4ft-Miner procedure has been applied several times. There seem to appear some strong rules but interpretation of the obtained results given in tables 3, 4, 5, and 6 is impossible without relevant medical knowledge. Anyway, the considered set of 176 patients is too small - the applied restrictions leading to creation of the considered data matrix do not take optimal advantage of all the available data. All over it, we are satisfied with the obtained results, since they prove that the suggested methodology can lead to meaningful knowledge.

Based on the experience with the present data and results of current data mining efforts, our next challenge is to modify selection criteria for the used preprocessing. We believe that the suggested methodology based on selection of a time window related to some significant instant could prove useful when studying influence of the interferon therapy. Further analysis should work with a new enhanced data set in which two significant instants are considered: one corresponds to the beginning of the interferon therapy, while the other is set several months after that. This setting makes it possible to study changes in time patterns due to the therapy. Moreover, measurements from the table hemat will be included. Results from this new data are under investigation now. All the steps of our approach are easy to repeat and modify – it is mostly a matter of changing of parameters of the used scripts. There are lot of possibilities how to proceed in the search of meaning of medicaly relevant knowledge. Now, it is time to consult medical speciallist, who should identify tasks, that are the most important from his/her point of view.

The project showed, that a cooperation of the both tools, SumatraTT and LISp-Miner, is effective and allows fast data preprocessing and data mining cycle. The whole process can be easily modified and reused for different data mining tasks (influence of interferon) and even to different datasets.

Acknowledgements

The work described here has been supported by the grant 201/05/0325 of the Czech Science Foundation, by the grant 25/05 of the Grant Agency of University of Economics, Prague and the research program No. MSM 6840770012 “Transdisciplinary Research in Biomedical Engineering II” of the CTU in Prague.

References

1. Hájek, P., Havránek, T.: Mechanizing Hypothesis Formation (Mathematical Foundations for a General Theory), Springer–Verlag 1978.
2. Ho T.B. et al: Combining temporal abstraction and data mining to study hepatitis. In Proceedings of the Discovery Chalenge 2004 – A Collaborative Effort in Knowledge Discovery from Databases. Prague: University of Economics, 2004
3. Kléma, J. - Nováková, L. - Karel, F. - Štěpánková, O.: Trend Analysis in Stulong Data. In Proceedings of the Discovery Chalenge 2004 – A Collaborative Effort in Knowledge Discovery from Databases. Prague: University of Economics, 2004, pp. 56–67.

4. Rauch J, Šimůnek M (2000).: Mining for 4ft Association Rules. In: Arikawa S, Morishita (eds) Discovery Science, Springer Verlag, pp. 268–272
5. Rauch J, Šimůnek M (2005) An Alternative Approach to Mining Association Rules. In: Lin T Y, Ohsuga S, Liau C J, and Tsumoto S (eds) Data Mining: Foundations, Methods, and Applications, Springer-Verlag, 2005, pp. 219–238 (*to appear*)
6. Rauch J, Šimůnek M (2005) GUHA Method and Granular Computing. In: HU, Xiaohua, LIU, Qing, SKOWRON, Andrzej, LIN, Tsau Young, YAGER, Ronald R., ZANG, Bo (ed.). Proceedings of IEEE International Conference on Granular Computing. IEEE, 2005, pp. 630–635.
7. Šimůnek M (2003) Academic KDD Project LISp-Miner. In Abraham A et al (eds) Advances in Soft Computing – Intelligent Systems Design and Applications, Springer, Berlin Heidelberg New York
8. Štěpánková O., Aubrecht P., Kouba Z., Mikšovský P. *Preprocessing for Data Mining and Decision Support*, pp. 107–117. Kluwer Academic Publishers, Dordrecht, 2003.
9. Aubrecht, P. - Kejkula, M. - Křemen, P. - Nováková, L. - Rauch, J. - Štěpánková, O.: Mining in Hepatitis Data by LISp-Miner and SumatraTT. In Proceedings of the Discovery Chalenge 2005 – to appear.

Towards Semantic Services Grid in Flood-Forecasting Simulations

Marian Babik, Ondrej Habala and Ladislav Hluchy*

Dept. of Parallel and Distributed Computing, Institute of Informatics,
Slovak Academy of Sciences,

Marian.Babik@saske.sk, {Ondrej.Habala, Ladislav.Hluchy}@savba.sk

Abstract. In this paper we present design and development of the semantic web services for the flood forecasting simulations. We will highlight the corresponding architecture and the process of service discovery and composition in the project K-WfGrid [16]. We will describe in detail the challenges of the flood-forecasting application and corresponding design and development of the service oriented model, which is based on the well known Web Service Resource Framework (WSRF). Semantic descriptions of the WSRF services will be presented as well as the architecture, which exploits semantics in discovery and composition of services. The system provides a unique approach in Semantic Grids by combining the advances of semantic web services and grid architectures.

1 Introduction

Recently, Web service (WS) technologies are gaining importance in the implementation of the distributed systems, especially grids. One such example is the Web Service Resource Framework (WSRF) [4], which extends the current WS technologies by modeling the stateful services. Design and development of the service oriented distributed system is quite common and there are several emerging WS initiatives, which tries to automate the process of discovery, composition and invocation of services. The semantic web services are a typical example, showing the potential of how ontological modeling can improve the shortcomings of the service oriented computing.

In this paper we provide a brief overview of the Web Service Resource Framework specification (WSRF) and Web Ontology for Services (OWL-S) and show how can the stateful service be described in terms of the OWL-S specification (see Sec. 2). Further, we describe a corresponding architecture for discovery, composition and invocation of both stateful and stateless services (see Sec. 4). Finally, we will present the process of design and development of the flood-forecasting simulation services and the process of their translation to the semantic descriptions (see Sec. 6.1).

* Acknowledgments: The research reported in this paper has been partially financed by the EU in the project IST-2004-511385 K-WfGrid and Slovak national projects, VEGA No. 2/3132/23: Effective tools and mechanisms for scalable virtual organizations and Tools for acquisition, organization and maintenance of knowledge in an environment of heterogeneous information resources 2004-2007 SPVV 1025/04.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 48–59, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2 Web Service Resource Framework (WSRF)

WS-Resource Framework (WSRF) is a set of specifications, which are based on the concept of modeling state as stateful resource and codify the relationship between Web Services (WS) and stateful resource in terms of a set of conventions on current (i.e. stateless) WS technologies [4]. A stateful resource is defined as having specific state data expressible as an XML document and a well defined life-cycle. It can be acted upon by one or more stateful services, e.g. files in a file system or rows in a relational database are considered a stateful resource. Fig. 1 shows a sample implementation scenario for the stateful service in the globus toolkit environment (GT4) [5].

In the example a sample meteorological service called MM5 is shown. A sample stateful resource is actually a grid job of MM5 running on some PC farm. It is defined by various resource properties, e.g. jobState, workflowID, MM5DomainInput, MM5TerrainData, MM5PredictionOutput, etc. Submitting a particular MM5 job is performed by contacting a service factory, which creates an instance of the resource (binding of service and particular resource) and return an identifier, i.e. so called EPR (end-point reference). The client can interact with the given resource by calling the methods exposed by the service instance interface, i.e. configureFromProperties, start, abort, etc. The pairing of the service instance and the resource is called a *WS-Resource*. The normal usage of the MM5 WS-Resource is to setup the resource properties, call the configureFromProperties method and then call the start method, which submits the job to a PC farm. The actual resource instances are homed by the Resource home, which can be implemented as either persistent or transient store. Detailed description of the WSRF is beyond the scope of this paper; for additional details see [4].

3 Web Ontology of Services (OWL-S)

OWL-S is an ontology-based approach to the semantic web services [1]. The structure of the ontology consists of a service profile for advertising and discovering services, a process model which supports composition of services, and a service grounding, which associates profile and process concepts with underlying service interfaces. Service profile (OWL-S profile) has functional and non-functional properties. The functional properties describe the inputs, outputs, preconditions and effects (IOPE) of the service. The non-functional properties describe the semi-structured information intended for human users, e.g. service name, service description, and service parameter. Service parameter incorporates further requirements on the service capabilities, e.g. security, quality-of-service, geographical scope, etc. Service grounding (OWL-S grounding) enables the execution of the concrete Web service by binding the abstract concepts of the OWL-S profile and process to concrete messages. Although different message specifications can be supported by OWL-S, the widely accepted Web Service Description Language (WSDL) is preferred [2].

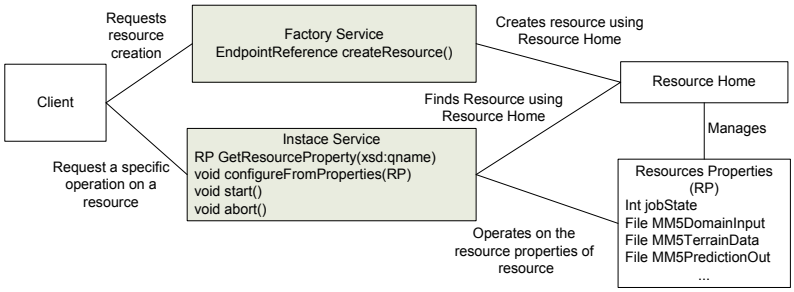


Fig. 1. Sample implementation scenario for the WS-Resource.

4 Adding semantics to stateful services

4.1 Overview

In OWL-S, the service capabilities are described by the corresponding IOPE (inputs, outputs, preconditions and effects). Such description can be partly annotated from the WSDL description of the service and is sufficient for the stateless service. Stateful services, i.e. WS-Resources, however, are composed of a service and a stateful resource. Since resource as well as corresponding service are well defined, it is possible to semi-automatically map the resource description with a domain specific ontology. This can enable automated discovery and execution for the stateful services. A possible ontological mapping supporting OWL-S and WSRF is described in the next section.

4.2 Service Discovery

In our system, the process of service discovery has three parties, client, provider and broker (so called Workflow Composition Tool - WCT). The provider is publishing the profile's functional and non-functional properties, called advertisements, to a broker. The client queries the broker for the given functional and non-functional properties and receives descriptions of all the relevant services. Service registry is comparing the request to the advertisements. This is called matchmaking of services. Since requests usually do not match exactly the advertisements, it is necessary to derive the corresponding match based on the subsumption, e.g. service reserving places in Cantonese restaurant should also match queries for Chinese restaurant. There are different ways of performing semantic matching of services [17]; however, in our case we will only consider hierarchy of classes, properties and explicitly defined equivalence. In the implementation we have defined the degrees of matching by subsequent matching rules. The rules are based on the subsumption of classes, properties and also consider explicitly stated equivalence, i.e. the OWL construct owl:sameAs.

We have designed a simple ontology, which describes WS-Resource, i.e. resource, resource properties and its association with instance service and service factory, see Fig. 2. A WSRFServiceProfile was defined, extending the already existing OWL-S Profile. In order to describe the resource properties, additional object property was introduced called hasResourceProperty, which points to one or many instances of class ResourceProperty. As with Inputs, Outputs, Preconditions and Effects, ResourceProperty is described by the identification, datatype property and a relevant ontological concept. Further, since ResourceProperties can act as both Inputs and Outputs, the corresponding Input, Output concepts are introduced pointing to the already existing ResourceProperty instance. Since resource properties are already defined as an XML Schema, the ontological description can be created semi-automatically. The datatype mappings for the WS-Resource is described in the next section. It should be noted that both ontological and datatype mappings describe only the resource properties types, the actual values of resource properties are implementation dependent.

In the process of implementation, there were several well known issues in the OWL-S profile, which were related to the OWL language and the limitations of the description logics in the description of services. A good survey of the OWL-S shortcomings is described in [8]. One of the major issues in the profile description is the conditional model. Since conditions refer to the concrete parameter instances, which are not known before execution, it is necessary to define the variables.

The latest version of OWL-S is addressing the issue and specifies the conditions by providing a class *Expression*. *Expression* is defined by the language used, e.g. SWRL [11], RDF [3], KIF [6], and the actual expression. Based on the language, the expression is incorporated as either XML Literal or String Literal. In our case we have used KIF for expressing conditions, therefore it was necessary to execute the given KIF expression on the instances of the service profiles.

4.3 Service Invocation

Service grounding for the stateful service differs in the description of the resource properties document, which is specified in XML Schema [10]. It is thus essential to define a datatype mapping between XML Schema and OWL-S. Datatype mapping in OWL-S grounding is based on the RDFS and eliminates previous shortcomings, which were based on the XSLT approach. Currently, a mapping between XML schema datatypes and OWL, is based on semantics and uses a sample RDFS ontology, as described in [8]. The mapping is depicted in Fig. 2. It can be seen that the mapping reflects the nested data structures of the XML Schema. XML Schema SimpleType is mapped to a corresponding OWL type, which is bound to the datatype property of the OWL-S grounding. Together with complex and array types, a mapping through RDF property is used to map the elements to XSDTypes.

The basic WS-Resource has a well defined interface for accessing and modifying its resource properties and their values. This interface has three main meth-

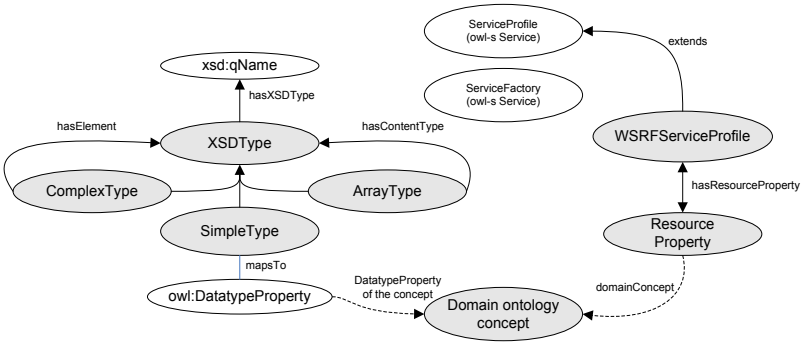


Fig. 2. A simplified schema of the ontological model (right) and datatype mapping (left) for WSRF services.

ods, namely, getResourceProperty, setResourceProperty and getMultipleResourceProperties. The datatype mapping for the stateful services is based on mapping the inputs and outputs of these methods to a datatype property of the domain concept. The actual mapping is based on the RDFS mapping schema, which is complemented by the ontological mapping to the *Resource* class. Schematic representation of the process is shown in Fig 2.

5 Service Discovery and Execution Architecture

Fig. 3 presents a UML deployment diagram of the system components of the workflow orchestration and execution environment and its interfaces. The main user interface for developing semantic-based Grid applications is the Grid Workflow User Interface (GWUI). The Grid Workflow User Interface contacts the Grid Workflow Execution Service (GWES) that manages the interactive process of composing and executing the services. The GWES is deployed as a standard Web Service. The automated semantic service composition is partly delegated to the Automatic Application Builder (AAB), the Workflow Composition Tool (WCT), and the user (by means of the GWUI). The Automatic Application Builder and the Workflow Composition Tool are knowledge-based semi-automatic modeling services, able to propose known solutions to problems solved in the past by selecting appropriate services and interconnection patterns of interest. When parts of the workflow are ready to be executed on the Computing Grid, the Grid Workflow Execution Service asks the Scheduler for the optimal resource, due to some user-defined metrics. Then, the corresponding Web Service operation is invoked remotely on the Grid middleware using WSRF protocols. The Scheduler works together with the Grid Performance Analysis Service (PAS) that aggregates the monitoring information derived from instrumentation and monitoring services. The events triggered by the workflow orchestration and execution will be pub-

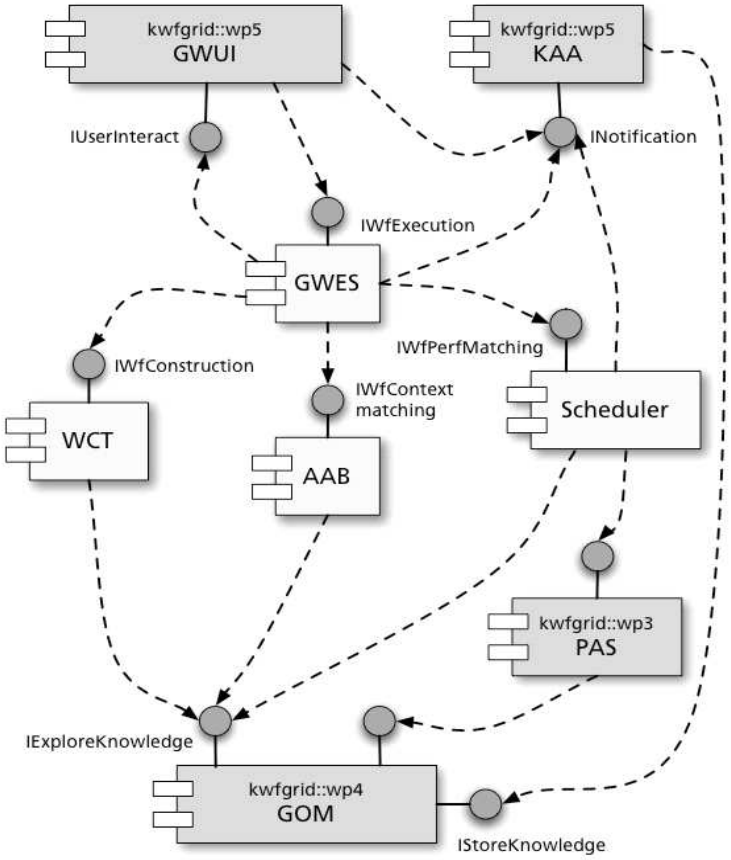


Fig. 3. A simplified schema of the service discovery and execution architecture.

lished by means of an event system. The Knowledge Assimilation Agents (KAA) consume these events and generate knowledge that is stored in the Grid Organizational Memory (GOM). This knowledge will later be reused by the components of the workflow orchestration and execution environment.

Automated service composition and creation of the abstract workflow is the task of the The Workflow Composition Tool (WCT). WCT provides the functionality of composing abstract workflows of Grid applications from simple user requirements. It employs semantic reasoning techniques over OWL-S descriptions (i.e. subsumption, classification) and it tries to propose a solution to the user's problem by using provided descriptions of available resources. Such a solution is delivered in form of an abstract workflow instance composed of service operations (methods). The main input to the WCT is a description of data (results) which is to be produced by the target application and, optionally, a set

of user-provided data (input) to be used by the composed application. It is also possible to upload an incomplete workflow as an input in order to complete it automatically. The main output of the composition process is a refined description of the abstract workflow. During its operation the WCT extensively uses the external registry, which provides the descriptions of available elements that build the abstract workflow. Such elements are called service classes - each abstract workflow consists of a number of identified service classes, links between them and other constructs needed to further refine and (finally) execute the workflow. The WCT contacts the registry and queries it in order to obtain the required descriptions of the service classes.

6 Service based Flood-forecasting Simulation Cascade

The flood prediction application (FFSC) is based on a a network of loosely coupled, cooperating but independent services. The K-WfGrid's flood application became instantiated in several servers across the testbed, with users calling the interface methods. The overview of the architecture of the FFSC is shown in Figure 4 [21]. It can be seen, that it is a set of loosely coupled services, with several possible execution scenarios. The entities displayed in this figure are described in the following section.

6.1 Components of the Application

Figure 4 contains several entities, each of them having its role in our application. At the top of the figure is depicted our main data provider, the Slovak Hydrometeorological Institute (SHMI). The SHMI provides us with input data for the first stage of our application, the *Meteorology*. In this stage, we employ two models *ALADIN* and *MM5* for weather predictions, with the latter having three distinct operation modes (simple, one-way nested and two-way nested). The predicted weather conditions are used in the *Watershed integration* stage to compute water runoff into the target river. This result is then further processed in the *Hydrology* stage, where two models - *HSPF* and *NLC* compute river levels for selected geographical points. These levels are then used to model water flow in the last, *Hydraulic* stage of the application. Concurrently all important results are optionally visualized, packaged and displayed to the user - if he requires it.

Apart from the simulation models, preprocessor and associated tools, the data flow contains also several *job packagers* and a *User Proxy Service*. These services implement our approach towards interactive grid jobs.

The flood application was chosen for the CROSSGRID project [15], because of its interactive nature. At several points during its execution, the user may choose to preview intermediate results and take action based on them (he may for example decide to stop the application, if there is no rain to produce a flood). While in the CROSSGRID project this problem was dealt with in the heavily customized application portal, we decided to use a more generic approach in K-WfGrid. Each user will be represented in the cyberspace of the grid by his/hers

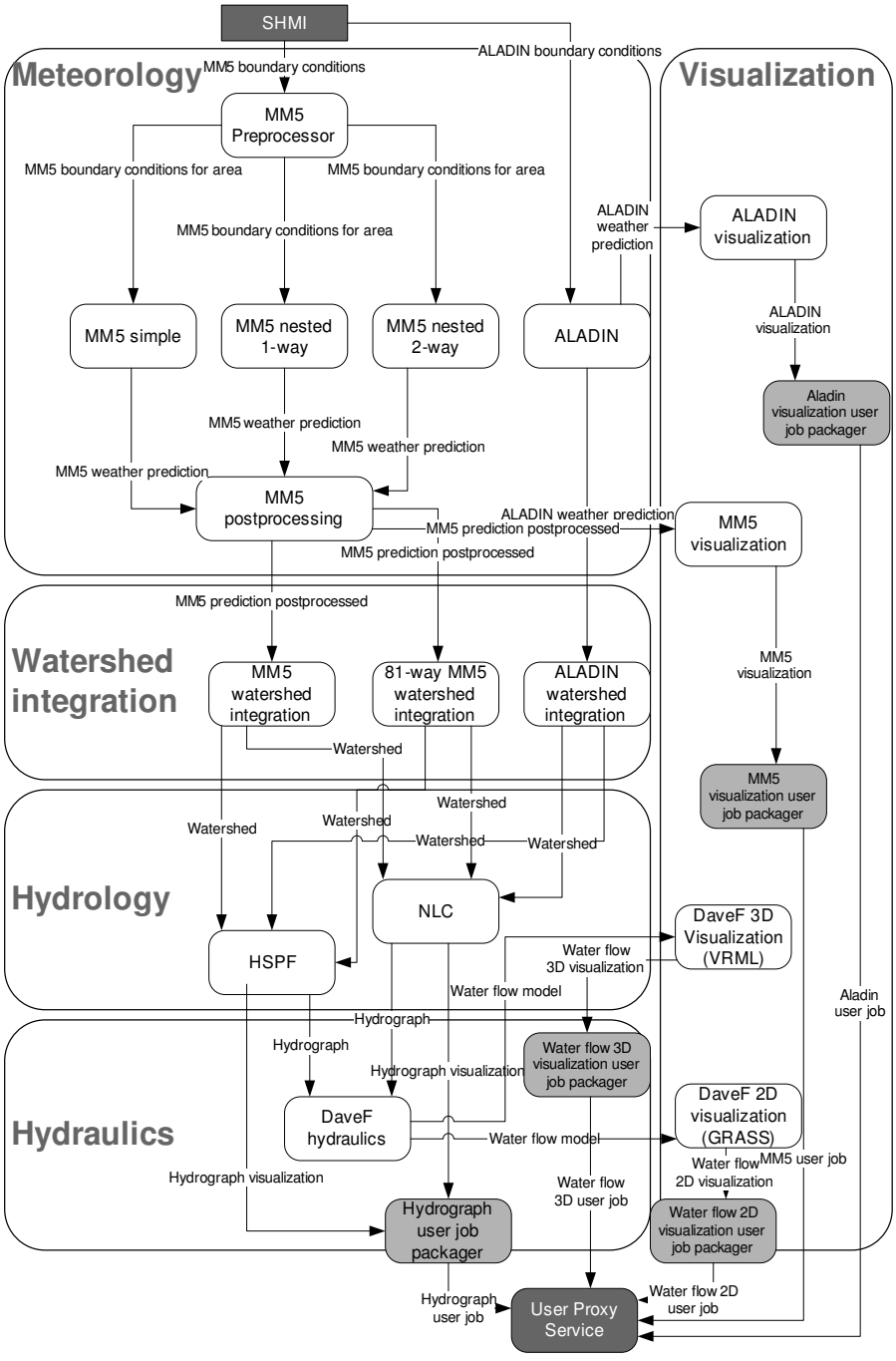


Fig. 4. A complete data flow diagram of the K-WfGrid's flood application

User Proxy Service. This service will be able to accept jobs for the user, contact the user and present the job to him/her for processing. The job will be a package consisting of one or more HTML documents (with forms for user's input), pictures and other web items. The package itself will be highly application-independent, displayable in any web browser. But since all the presented data is just the opposite, it has to be prepared by a customized tool. This tool is called *job packager* and it is basically a service, able to download one type of data and process it into a HTML-based form presentable to a human. There will be a distinct job packager for each type of data which will have to be displayed to the user.

Apart from data presentation, the *User Proxy Service* can also enable the user to be a part of the workflow. Although it is not shown in the data flow in Figure 4, the user's inputs may be further processed in the workflow by other services. Also another notable feature of this approach is its asynchronous nature. Grid workflows running for several days don't have to be periodically monitored for outputs, the User Proxy Service notifies its user when it receives any inputs (by e-mail, instant message or SMS, depending on its implementation).

6.2 Flood application implementation

To actually implement the application components, we have first created a UML design of their respective interfaces. These interfaces were then described in WSDL documents ([2]). Apart from the currently standardized WSDL elements, these documents contain also some elements pertinent to the WSRF framework - for example resource properties definitions. From these WSDL documents were created (using automated tools supplied by Globus Toolkit) interface modules (so-called *stubs*), which allow clients to connect to the remote services and these services to process client requests and send responses. The final implementation stage consisted of development of several classes, which provide access to the legacy executables of the flood application. The existing WSDL descriptions were translated into the corresponding OWL-S ontology, creating an OWL-S description for each WSDL method. We have used the possibility to annotate the WSDL XML Schema, describing the resource properties, with a documentation tag pointing to the existing ontological domain concept, i.e. resource property MM5DomainInput is translated into OWL-S Profile's Input and mapped to the concepts of the FFSCDataOntology. Such mapping then facilitates the service discovery by simple subsumption and classification of ontological domain concepts. Analogical process is done for the OWL-S Service Groundings, which then maps the Profile's concepts into a WSDL messages and types. The deployment of the modules was tested with both Globus Toolkit's grid service container and the Jakarta Tomcat servlet container coupled with the Axis web service container.

6.3 Example Application Scenario

To better illustrate the way the application will execute in K-WfGrid, we present here a simple workflow - a sub-flow of the data flow in Figure 4. In this case, let us

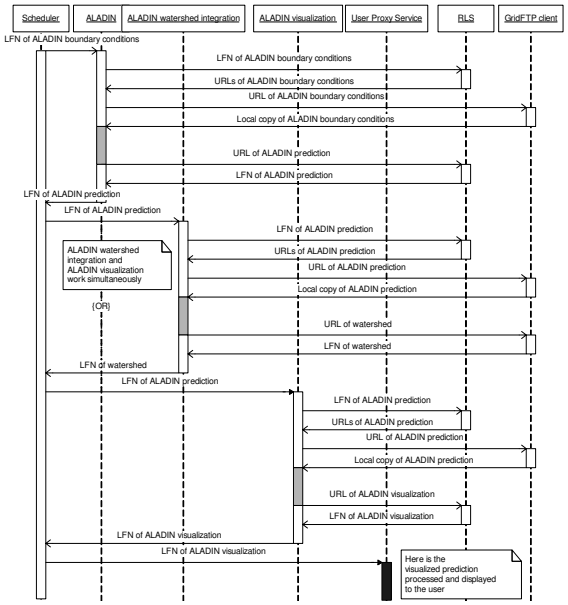


Fig. 5. Sequence diagram of the example workflow execution

assume that the user chose to use the ALADIN model for meteorology, the NLC model for hydrology, and he/she would like to see weather prediction and the final water flow, not the hydrograph produces in the hydrology stage. A partial sequence diagram (with the meteorological prediction details) is in Figure 5. The sequence diagram shows how the scheduler provides the LFN (*Logical File Name*) of the input data to each of the employed services, the service locates the file, downloads it and processes it. After producing results, the service then registers them with the RLS (*Replica Location Service*) and returns the obtained LFN back to the scheduler. Please note that the User Proxy Service is handled the same way as all other services in the example and the user is actually integrated into the workflow.

7 Related work

7.1 Semantic Web Services

One of the challenges of the loosely coupled distributed systems is the ability to dynamically discover and integrate the services needed by the applications. Interoperability among services is especially important in the distributed environments hosting a large number of services, i.e. grids. Semantic descriptions facilitates the process by expressing the characteristics of the service, which can

later be used for their discovery, composition and invocation. The rich semantics needed for such descriptions can be provided by the ontologies.

OWL-S is an ontology-based approach to the semantic web services [1]. The structure of the ontology consists of a service profile for advertising and discovering services, a process model which supports composition of services, and service grounding, which associates profile and process concepts with underlying service interfaces. Service profile (OWL-S profile) has functional and non-functional properties. The functional properties describe the inputs, outputs, preconditions and effects (IOPE) of the service. The non-functional properties describe the semi-structured information intended for human users, e.g. service name, service description, and service parameter. Service parameter incorporates further requirements on the service capabilities, e.g. security, quality-of-service, geographical scope, etc. Service grounding (OWL-S grounding) enables the execution of the concrete Web service by binding the abstract concepts of the OWL-S profile and process to concrete messages. Although different message specifications can be supported by OWL-S, the widely accepted Web Service Description Language (WSDL) is preferred [2].

The Web Service Modeling Framework (WSMF)[18] is an industry scale framework for semantic web service discovery, execution and composition. It is a joint effort of the European research projects on the Semantic Web and Semantic Web Services. It has three development areas concerning conceptual model (WSMO), the representation language (WSML) and the execution framework (WSMX). Although WSMF approach is much more profound and shows many significant contributions to modeling semantic web services, the level of implementation and the development support at the time of evaluation was unacceptable for our purposes. However, since WSMF has gained an enormous momentum in the last year, it will be considered in our future work.

The Internet Reasoning Service (IRS-II)[20] is a Java framework for publishing, locating, composing and executing semantic web services. IRS-II is modeling services based on the tasks, that needs to be fulfilled and the problem-solving-methods (PSM), that can be used to solve specific tasks. It utilizes a formal language called Operational Conceptual Modeling Language (OCML). It supports the specification and operationalization of functions, relations, classes, instances and rules. This proves to be more suitable for procedural knowledge representation than OWL. The process of assigning services the corresponding tasks has to be done manually, which is unscalable and introduces many possibilities for failure. METEOR-S [19] attempts to add semantics to the basic web service descriptions by adding semantics to current industry standards. It is an effort of the LSDIS Lab of the University of Georgia. Although the framework presents an interesting bottom-up approach to the semantic descriptions, the service annotation and WSDL-S, semantic extension of the WSDL, are not scalable in terms of functionality to the level of semantic information provided by the previous approaches.

References

1. A. Ankolekar et.al, OWL-S: Semantic Markup for Web Service, 2003, http://www.daml.org/services/owl-s/1.1
2. E. Christensen, F. Cubera, G. Meredith, S. Weerawarana, Web Services Description Language (WSDL) 1.1, Technical report, WWW Consortium, 2001, http://www.w3c.org/TR/wsdl
3. Resource Description Framework, http://www.w3.org/RDF/
4. Web Service Resource Framework, http://www.globus.org/wsrf/
5. Globus Toolkit, http://www-unix.globus.org/toolkit/
6. Knowledge Interchange Format, draft to American National Standard, 1998, http://logic.stanford.edu/kif/dpans.html
7. Universal Description, Discovery and Integration Protocol Specification, UDDI Consortium, 2001, http://www.uddi.org/specification.html
8. Pitfalls of OWL-S – A practical Semantic Web Use Case, S. Balzer and T. Liebig and M. Wagner, Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC004), 2004
9. JTP: A System Architecture and Component Library for Hybrid Reasoning, R. Fikes, J. Jenkins, and G. Frank, Proceedings of the Seventh World Multiconference on Systemics, Cybernetics, and Informatics. Orlando, Florida, USA, 2003
10. Extensible Markup Language (XML), XML Protocol Working Group, 2004, http://www.w3.org/TR/2004/REC-xml11-20040204/
11. A semantic web rule language combining OWL and RuleML (SWRL), Ian Horrocks et.al., 2003, http://www.daml.org/2003/11/swrl/
12. R. Fikes, P. Hayes, I. Horrocks, OWL-QL: A Language for Deductive Query Answering on the Semantic Web, KSL Technical Report 03-14, 2003
13. N. Srinivasan, M. Paolucci and K. Sycara, Adding OWL-S to UDDI, implementation and throughput, First International Workshop on Semantic Web Services and Web Process Composition, (SWSWPC 2004) 6-9, 2004, San Diego, California, USA
14. M. Paolucci, J. Soudry, N. Srinivasan, and K. Sycara, Untangling the Broker Paradox in OWL-S, Proceedings of AAAI 2004 Spring Symposium
15. CrossGrid consortium, CrossGrid Technical Annex, 2004, http://www.crossgrid.org
16. The Knowledge-based Workflow System for Grid Applications FP6 IST project. http://www.kwfgrid.net
17. L. Li and I. Horrocks, A software framework for matchmaking based on semantic web technology. In Proc. of the Twelfth Inl. Word Wide Web Conference (WWW 2003), pp. 331-339, ACM, 2003
18. Fensel D. and Bussler C., The Web Service Modeling Framework WSMF, Eletronic Commerce: Research and Applications, 1, 2002
19. P. Rajasekaran and J. Miller and K. Verma and A. Sheth, Enhancing Web Services Description and Discovery to Facilitate Composition, International Workshop on Semantic Web Services and Web Process Composition, 2004
20. Motta E. and Domingue J. and Cabral L. and Gaspari M., IRS-II: A Framework and Infrastructure for Semantic Web Services, 2nd International Semantic Web Conference (ISWC2003), Sundial Resort, Sanibel Island, Florida, USA, 2003
21. Hluchy, L., Tran, V.D., Habala, O., Simo, B., Gatial, E., Astalos, J., Dobrucky, M.: Flood Forecasting in CrossGrid project. In: Grid Computing, 2nd European Across Grids Conference, Nicosia, Cyprus, January 28-30, 2004, LNCS 3165, Springer-Verlag, 2004, pp. 51-60, ISSN 0302-9743, ISBN 3-540-22888-8.

Using Algorithms of Computational Geometry for Pattern Recognition and Comparison to Support Vector Machine

Marek Bundzel[1] and Tomáš Kasanický[1,2]

[1]Technical University of Košice,
Faculty of Electrical Engineering and Informatics,
Department of Cybernetics and Artificial Intelligence,
Letná 9, Košice, 04001, Slovak Republic
bundzel@neuron.tuke.sk
[2]Slovak Academy of Sciences, Institute of Informatics,
Dúbravská cesta 9, 845 07 Bratislava 45, Slovak Republic
kasanicky@neuron.tuke.sk

Abstract. Selected computational geometry algorithms based on Delaunay graph (triangulation) and Support Vector Machines used in pattern recognition applications divide feature space into regions surrounding patterns representing separate classes. The approaches to the feature space division are analogous, the results are usually similar and the classification processes differ. The selected algorithms were variously pre-set and analyzed in order to evaluate the measure of similarity among them. The purpose was to identify possible ways of hybridization while a synergic effect was to be achieved.
Properties of the individual algorithms were compared using several criteria. Artificial and real world data were used. Differences regarding the mechanism of selection and location of the support vectors were described. Optimality of the resulting decision boundaries was assessed. Robustness and accuracy of the algorithms were compared as well. The computational geometry based algorithms were compared to Support Vector Machine, eventually.

1 Motivation

Pattern recognition is based on correct determination of membership of particular examples to individual classes (sets). Classification techniques use several ways of encoding of the reference examples. The type of the encoding influences bias robustness and accuracy of the method. Mathematic description of a decision curve separating individual classes represents another interesting approach. The decision curve can be also represented differently. Support Vector Machines (SVMs) are well known and respected for their generalization ability. SVMs search for a decision boundary in a form of a hyperplane and non-linearity of the solution is achieved using a virtual transformation of the training examples into higher–dimensional space. However, SVMs are quite demanding and therefore alternatives are being developed.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 60–70, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

The pattern recognition algorithm proposed in this paper is based on determination of the reference examples – support vectors – using computational geometry algorithms, specifically Delaunay triangulation, Gabriel graph and relative neighborhood graph. The application of the mentioned methods combined with gradient optimization represents a new approach.

The paper is thematically divided into three parts. Firstly, computational geometry (CG) algorithms and SVMs are briefly described. The role of the CG algorithms in the proposed design is explained. Secondly, the proposed approach is outlined. Finally, information on the experiments performed, comparison to SVM and conclusions are reported.

2 Background

2.1 CG Algorithms

In computer science the computational geometry studies algorithms solving problems formulated in terms of mathematic geometry. Some purely geometrical problems arise out of the study of computational geometric algorithms and the study of such problems is considered to be part of computational geometry as well. Delaunay triangulation, Gabriel graph and relative neighborhood graph were selected to be applied in the proposed approach. The algorithms are mainly used in computer graphics, CAD systems, cartography and visualizations in GIS. Most implementations work in 2D or 3D but generalizations for N-dimensional space exist for all of the selected algorithms.

Delaunay Triangulation: Triangulation is a method of building triangles with the irregularly sampled points as the triangles apexes. This particular type of triangulation was invented by Boris Delaunay in 1934 ([1]). The Delaunay triangulation for a set P of points in the plane is the triangulation $DT(P)$ of P such that no point in P is inside the circumcircle of any triangle in $DT(P)$. Figure 1 a. shows an example of Delaunay triangulation. Complexity O of the calculation depends on the count n of the points and dimension d of the space of the points, $O(n^{\frac{d}{2}})$.

Gabriel graph: The Gabriel graph $GG(P)$ ([2]) is the set of edges l_{ij} subset of $DT(P)$, for which the open circle (ball) with diameter $[P_iP_j]$ contains no other points than P_i and P_j:
$GG(P) = \{l_{ij} \subset P \mid \forall P_k \in P, (P_k - P_i)^2 + (P_k - P_j)^2 > (P_i - P_j)^2\}$
The situation is documented in Figure 1 b. The brute force algorithm to compute $GG(P)$ is straight forward from the above definition. The condition for an edge to be created is stricter than by Delaunay triangulation, therefore fewer edges arise. Average complexity of Gabriel graph is $O(dn^3)$ ([6]).

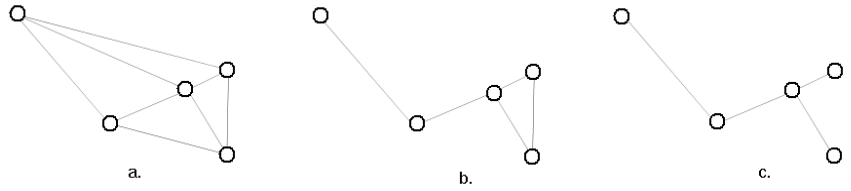


Fig. 1. Edges created by CG methods – a. Delaunay triangulation, b. Gabriel graph, c. relative neighborhood graph.

Relative neighborhood graph: The relative neighborhood graph of P, the $RNG(P)$, is a neighborhood graph, where two points P_i and P_j are called relative neighbours, if all other points P_x in the set satisfy:
$distance(P_i, P_j) < max\{distance(P_i, P_x), distance(P_j, P_x)\}$.
The situation is documented in Figure 1 c.

One of the first works discussing the possible application of CG for the purposes of pattern recognition was [8]. A pioneer work dealing with comparison of SVM with the graph building algorithms was [6]. The author points out the possibility to use CG algorithms for identification of support vectors. Complexity and performance of the algorithms were compared. Nevertheless, the author does not consider the idea of further application of support vectors together with a kernel function. This is the base of our method, increasing the accuracy and the generalization performance eventually.

2.2 Support Vector Machine

SVM is a technique used for various estimation problems including pattern recognition, developed by Vladimir N. Vapnik and his group at AT&T Bell Labs. SVM features high generalization ability. The factors responsible for future generalization were extracted and bounds on generalization that depends on these factors were obtained and lastly a technique that constructively minimizes these bounds was developed. The algorithm is based on statistical learning theory, the fundamentals of which have been laid in 70's ([3], [4]).

Let us have the training data in the form: $\{\boldsymbol{x}_i, y_i\}, i = 1, ..., l,\ y_i\ \in \{-1, 1\}$, $\boldsymbol{x}_i \in\ R^d$. Let us assume there is a hyperplane separating the positive examples from the negative examples (separating hyperplane). Points lying on the hyperplane satisfy $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$, where w is normal to the hyperplane and $|b|/||\boldsymbol{w}||$ is a perpendicular distance from the hyperplane to the origin of the coordinate system. Let d_+ and d_- be the shortest distance from the separating hyperplane to the closest positive (negative) example. Let us define the 'margin' of the separating hyperplane to be $d_+ + d_-$. For linearly separable case the algorithm searches for the separating hyperplane with the largest margin.

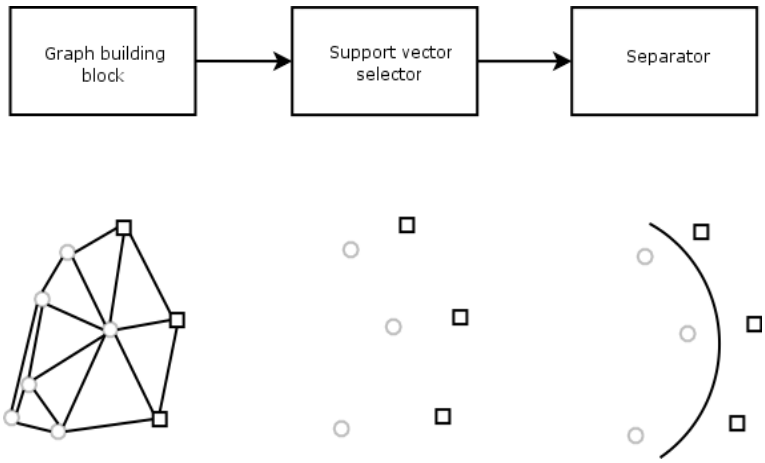


Fig. 2. Outline of the proposed system.

The Karush-Kuhn-Tucker theorem for convex optimization is used. The training data appear only in the form of the dot product in the process. This allows generalization of the procedure to nonlinear case.

Let us assume a mapping $\Phi : R^d \mapsto H$ where R^d denotes the space of features and H a transformed space. Let us introduce a kernel function K so that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$. Now it is possible to replace the standard dot products by K everywhere in the algorithm and work with the features transformed nonlinearly. Φ is not defined explicitly.

Trained SVM has the form:

$$f(\boldsymbol{x}_t) = \sum_{i=1}^{N_s} \alpha_i y_i K(\boldsymbol{s}_i, \boldsymbol{x}_t) + b \tag{1}$$

where $\boldsymbol{x}_t$ is the evaluated vector N_s is the number of support vectors, $\boldsymbol{s}_i$ are the support vectors, α_i denotes Lagrangian multipliers. $\boldsymbol{s}_i$ and α_i were determined during the optimization process.

3 Design

3.1 Outline

The proposed system and its process chart are depicted in Figures 2 and 3.

Graph building block creates a graph in the space of features (training data) using one of the above described CG algorithms. Support vector selector extracts support vectors form the training data. Support vectors are defined as points connected with an edge and belonging to different classes. These examples are stored. Using the set of support vectors, membership of any example can be determined using a simple algorithm of one–nearest–neighbor (Winner Takes

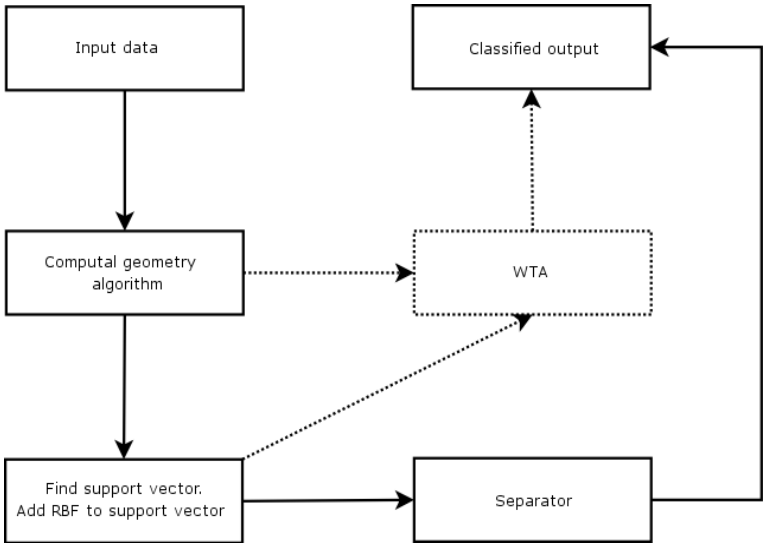


Fig. 3. Description of the processes in the proposed system.

All – WTA), i.e. the example will belong to the same class as the support vector nearest to it. Dotted lines illustrate this process branch in Figure 3. Another possibility is to apply Radial Basis Function (RBF) on the extracted support vectors and finalize the classifier to the form described in Eq. 1, with K defined as Gaussian RBF in the form:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2} \tag{2}$$

The training data are non–linearly transformed. The parameter γ of the Gaussian kernel is set by the user. In this case the final block of the system, the Separator, searches for the parameters of the hyperplane separating the transformed examples linearly.

3.2 Optimal Linear Separation

The method described here was originally introduced by RNDr. Ján Buša, CSc., from Technical University of Košice (unpublished) and modified by the authors for the purposes of another work. It can be used as a base for calculation of an index of separability determining a measure of ability to linearly separate a set of features. It also represents a freestanding linear classifier.

Let us consider two sets

$$\mathcal{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{N_\mathcal{A}}\}, \qquad \boldsymbol{a}_i \in \mathbb{R}^n, \quad i = 1, 2, \ldots, N_\mathcal{A} \tag{3}$$

and

$$\mathcal{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{N_\mathcal{B}}\}, \qquad \boldsymbol{b}_i \in \mathbb{R}^n, \quad i = 1, 2, \ldots, N_\mathcal{B} \tag{4}$$

Next let us consider normal vector $\boldsymbol{v} \in \mathbb{R}^n$, with unity norm

$$\boldsymbol{v} = \{v_1, v_2, \ldots, v_n\}, \qquad \|\boldsymbol{v}\|_2 = \left[\sum_{i=1}^{n} v_i^2\right]^{1/2} = 1. \tag{5}$$

For the hyperplanes with the normal vector $\boldsymbol{v}$ we consider the next criterial function which describes the dichotomous classification separation quality, achieved with direction choice:

$$Q_{\boldsymbol{v}} = \max\left([\min_{\boldsymbol{a}\in\mathcal{A}}(\boldsymbol{a}\cdot\boldsymbol{v}) - \max_{\boldsymbol{b}\in\mathcal{B}}(\boldsymbol{b}\cdot\boldsymbol{v})]; [\min_{\boldsymbol{b}\in\mathcal{B}}(\boldsymbol{b}\cdot\boldsymbol{v}) - \max_{\boldsymbol{a}\in\mathcal{A}}(\boldsymbol{a}\cdot\boldsymbol{v})]\right). \tag{6}$$

Then the best separation direction will be that, which maximizes this criterial function. The best separation value is given by:

$$\mathcal{Q} = \max_{\|\boldsymbol{v}\|_2=1} Q_{\boldsymbol{v}}. \tag{7}$$

Then the Best Separation Problem may be formulated as:

Find the direction $\boldsymbol{v}^*$, for which

$$Q_{\boldsymbol{v}^*} = \max_{\|\boldsymbol{v}\|_2=1} Q_{\boldsymbol{v}}. \tag{8}$$

The 'optimal' direction, which maximizes the criterial function on the set of unity vectors $\boldsymbol{v}$, is computed using numerical iterative method. If the data are linearly separable, the separation hyperplane will have the widest margin. The separation hyperplane will have the most narrow margin to the closest misclassified examples otherwise.

4 Experiments

4.1 Data Used and Methods of Evaluation

Artificial and real world data were used. The typical 'Double Spiral' classification problem was used for testing. The entire bitmap was 1000x1000 pixels. 1250 randomly chosen examples were used for training (554 positive examples, 696 negative examples), the rest (8750, 3757 positive examples, 4993 negative examples) was used for testing. This dataset was selected because it represents a relatively difficult classification problem often used as a benchmark. The real world data used are described bellow.

The experimental results were evaluated graphically and statistically. Overall accuracy was calculated and the results are reported in Table 1. The measure of similarity between the results of the proposed algorithms and SVM was assessed, mainly regarding the shape of the decision boundary.

Method	Accuracy -Train (%)	Accuracy -Test (%)
Delaunay +WTA	100.0	87.92
Gabriel Graph +WTA	100.0	87.74
Relat. Neighbors +WTA	99.84	86.54
Delaunay +OLS	98.48	91.17
Gabriel Graph +OLS	98.32	90.38
Relat. Neighbors +OLS	96.48	88.32
SVM	100.0	92.1

Table 1. Experimental results. OLS = Optimal Linear Separation.

4.2 Experiments with WTA Separation Method

In this series of experiments, support vectors were extracted using the three mentioned CG algorithms and the class of the fact of an evaluated example was determined as equal to the class of the fact of the nearest support vector. Figure 4 depicts the results.

4.3 Experiments with Optimal Linear Separation Method

In this series of experiments support vectors were extracted using the three mentioned CG algorithms. Gaussian RBF was applied to the support vectors and parameters of the separating hyperplane were determined using the Optimal Linear Separation method as described in Section 3. Parameter γ of the RBF was set to 95. Figure 5 depicts the results of the proposed algorithms and results of SVM for comparison.

SVM experiment were performed on SVM_{light} implementation of T. Joachims ([5]). Parameter γ of the RBF was set to 95 and parameter C of trade-off between training error and margin was set to 10000. Experimental results are in Fig. 5 d.

4.4 Experiments on Real World Data

The dataset represented multispectral, remotely sensed image data. The image containing 775x475 pixels was taken over the eastern Slovakia region. The image was sensed from Landsat satellite in 6 spectral bands. The main goal was land use identification. There were seven classes of interest picked up for the classification procedure. More details on the data are provided in ([7]). This application required extending the above described method for multiclass classification but this was trivial. The experimental results are provided in Table 2.

5 Conclusions

The results presented let us conclude the proposed method achieves comparable accuracy on the selected dataset as SVM with Gaussian RBF kernel. The shape

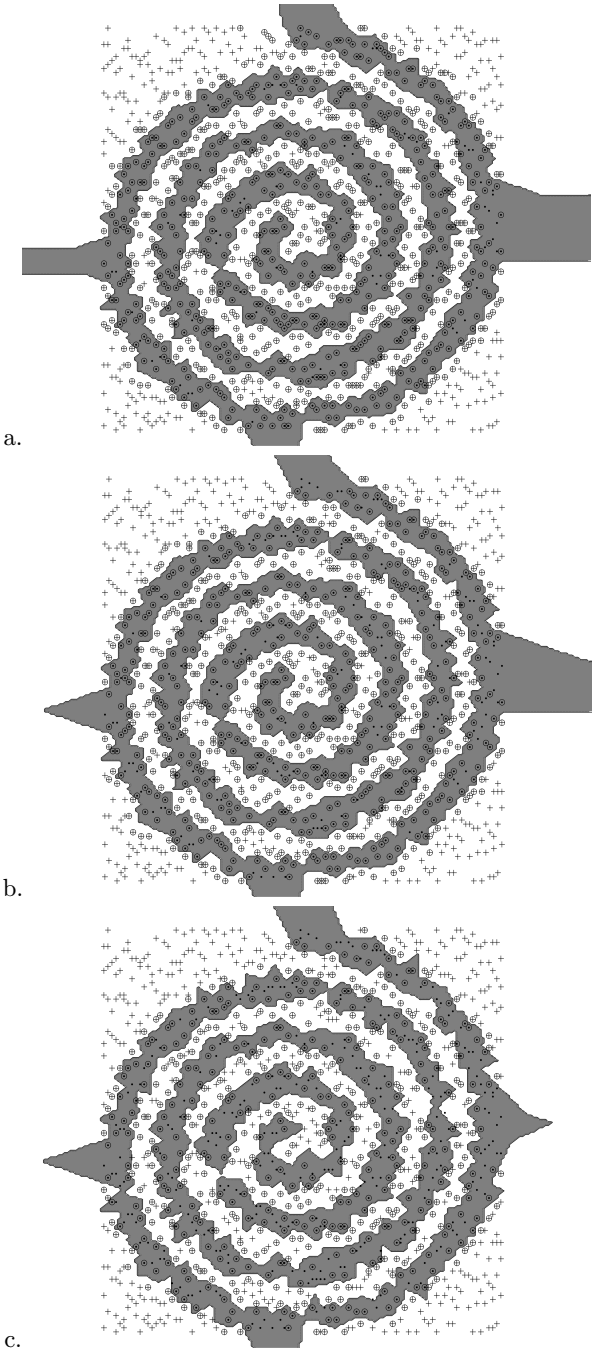


Fig. 4. WTA. Support vectors(SV) extracted by CG methods – a. Delaunay triangulation (910 SV), b. Gabriel graph (805 SV), c. relative neighborhood graph (576 SV). Support vectors are circled.

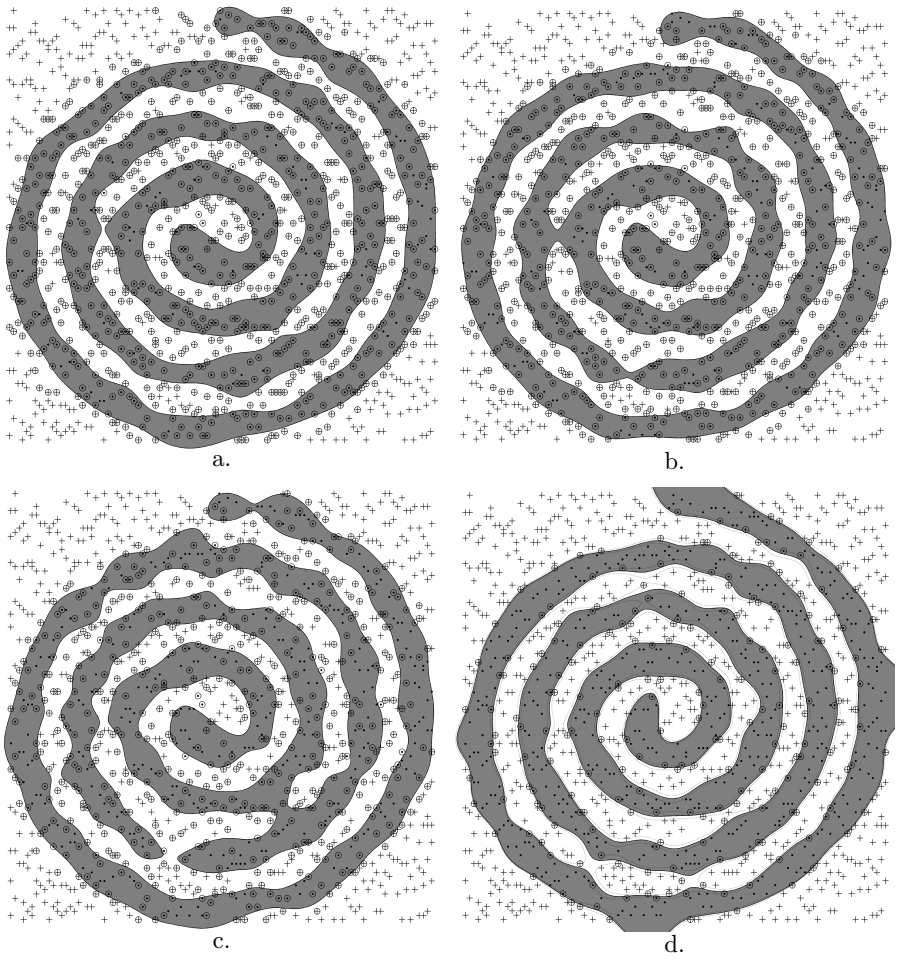


Fig. 5. Optimal Linear Separation. Support vectors extracted by CG methods – a. Delaunay triangulation, b. Gabriel graph, c. relative neighborhood graph. d. Gaussian RBF Support Vector Machine (213 SV). Support vectors are circled.

Method	A (%)	B (%)	C (%)	D (%)	E (%)	F (%)	G (%)	Overall(%)
Gabriel Graph +WTA	80.77	82.23	100.00	75.94	62.57	99.57	85.14	86.96
Gabriel Graph +OLS	75.64	87.81	100.00	92.57	49.12	99.32	71.62	91.16
SVM	83.33	79.96	100.00	94.57	60.82	99.40	72.97	91.6

Table 2. Experimental results on real world data. Overall accuracy is calculated on the testing set, percentual accuracies were calculated for each class (A,B,C ...).

of the decision boundary of both methods is similar and smooth. The differences observed are caused by different criterial functions used by SVM and Optimal Linear Separation methods and different criteria of the graph producing (CG) algorithms. Differently large sets of the support vectors are selected. This is in accordance with the statement that edges of relative neighborhood and Gabriel graphs form subsets of Delaunay triangulation. The extracted sets of support vectors can be used as prototypes for classification directly using the WTA method. In this case, however, is the decision boundary edgy which can cause high bias in the points of abrupt changes. According to the results presented in Table 1 smoothing of the decision boundary improved generalization. Introduction of Gaussian RBF on support vectors and Optimal Linear Separation method improved results but increased the computational complexity. It is interesting that support vectors selected by SVM (set up as described, all training points classified correctly) form a subset of Delaunay triangulation.

Consequently to the fact that SVM searches for the support vectors in the transformed space optimality of the decision boundary projected in the original space of features can be put in question as the margin is not necessarily equidistant to it anymore. The proposed method reduces this because the support vectors are selected directly in the space of features. The advantage of the proposed method over SVM is not in its accuracy so far. But the CG identified support vectors can be possibly used – with fewer difficulties – for description of the modeled system and for interpretation of the relationships between the examples of different classes in multiclass applications. Moreover, the WTA implementation of the proposed method enables incremental learning. It is important to note that any linear separator (e.g. Rosenblats perceptron etc.) can be used as a Separator in the algorithm. This opens a door for the future incremental implementations of the method with kernel functions.

The experiment on real world data documents the performance of the proposed method. Interesting insight provides the accuracy calculated for the class E represented largely by noisy and/or conflicting examples. Sensitivity of the CG support vector identification to the noise is a problem which is to be solved. A possible solution is to consider contribution of several examples near to a support vector (weighted averaging) and production of more typical support vectors.

Future work includes search for alternative methods of linear separation (e.g. hill–climbing, enhanced simple perceptron), development of methods reducing the amount of the extracted support vectors and development of the methods reducing sensitivity of the algorithm to noise.

This work is partially supported by Scientific Grant Agency of the Ministry of Education of Slovak Republic and the Slovak Academy of Sciences under the contract No. VEGA 2/4148/24.

References

1. B. Delaunay, Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793-800, 1934.
2. Gabriel, K.R. & Sokal, R.R.(1969) A new statistical approach to geographic variation analysis. Syst. Zoology,18:259-278.
3. V. Vapnik and A. Chevornenkis, Theory of Pattern Recognition (in Russian), 1974, Nauka, Moscow, German Translation: W. Wapnik and A. Tscherwonenkis, Theorie der Zeichenerkennung, Akademie-Verlag, Berlin, 1979.
4. Vladimir N. Vapnik, The Nature of Statistical Learning Theory, 1995, isbn 0-387-94559-8, Springer-Verlag New York, Inc.
5. T. Joachims, 11 in: Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schlkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.
6. Wan Zhang and Irwin King, A Study of the Relationship Between Support Vector Machine and Gabriel Graph, Proc. of IEEE Int. Joint Conf. on Neural Networks, IJCNN2002.
7. Marek Bundzel, Structural and Parametrical Adaptation of Artificial Neural Networks Using Principles of Support Vector Machines, Ph.D. Thesis, Technical University of Kosice, Slovakia, Faculty of Electrical Engineering and Informatics, Department of Cybernetics and Artificial Intelligence, 2005, http://neuron.tuke.sk/ bundzel/Ph.D-thesis/main.pdf.
8. Urquhart, R Graph Theoretical Clustering Based on Limited Neighborhood Sets., Pattern Recognition Vol. 15, no. 3, pp. 173-187. 1982

Kombinácia redukcie problému a fuzzy FCA prístupu pri tvorbe konceptuálnych modelov z textových dokumentov

Peter Butka

Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky, Technická Univerzita v Košiciach, Letná 9/B, 041 20, Košice, Slovenská republika
Peter.Butka@tuke.sk

Abstrakt. Formálna konceptová analýza (FCA) poskytuje formálny rozbor tabuľky dát a umožňuje identifikáciu zhlukov podobných objektov (reprezentujúcich koncepty). Výpočtová zložitosť tvorby formálnych konceptových zväzov s veľkých vstupných kontextov je značná, preto má zmysel hľadať metódy pre efektívnejšie generovanie konceptov, napr. kombináciou s inými algoritmami a použitím heuristík. Rozšírením klasickej FCA je fuzzy verzia pracujúca s reálnymi hodnotami v objekt-atribútovej tabuľke, ktorú predstavuje v prípade textových dokumentov dokument-term matica vytvorená pomocou tf-idf váhovania. Táto práca sa venuje štúdiu určitých aspektov využitia princípu redukcie problému pri tvorbe konceptuálnych modelov nad textovými dokumentmi (potenciálne s výhľadom na tvorbu ontológií), t.j. rozdeleniu pôvodnej úlohy vytvorenia modelu použitím FCA priamo z celej množiny dokumentov na množinu sub-problémov tvorby častí modelu z menších podmnožín pôvodného korpusu. Pri experimentoch bola ako metóda redukcie použitá rozdeľujúca zhlukovacia metóda k-stredov, ako testovacia množina bola zvolená menšia množina anglicky písaných dokumentov.

1 Úvod

Jednou z možností spracovania tabuľkových dát je použitie formálnej konceptovej analýzy (FCA). Táto poskytuje formálny rozbor tabuľky dát a umožňuje identifikáciu zhlukov podobných objektov (reprezentujúcich koncepty) z formálneho konceptuálneho pohľadu. Výpočtová zložitosť tvorby formálnych konceptových zväzov s veľkých vstupných kontextov je značná, preto má zmysel hľadať metódy pre efektívnejšie generovanie konceptov, napr. kombináciou s inými algoritmami a použitím heuristík.

Zhlukovanie a formálna konceptová analýza (FCA) sa používajú pri hľadaní vzorov v dátach. Zhlukovanie zoskupuje objekty s navzájom podobnou charakteristikou do klastrov (zhlukov). FCA identifikuje podobnosť medzi množinami objektov na základe ich spoločných atribútov. Problémom v zhlukovaní je často výber vhodnej podobnostnej metriky, najvýraznejším problémom konceptovej analýzy je obrovské množstvo konceptov a časovo náročné spracovanie.

Jednoduché prepojenie klasickej FCA a zhlukovania nad binárnymi dátami, tak ako ju zaviedli Ganter a Wille v [6], popísali v [13] Rice a Siff. Definovali špeciálnu vzdialenostnú (pseudo)metriku pre prípad klasického "crisp" konceptového zväzu na binárnej objekt-atribútovej tabuľke dát. Navrhnutú metriku použili pre aglomeratívne

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 71–82, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

zhlukovanie konceptov, čím sa značne redukuje počet konceptov (zhlukov). V [7] je možné nájsť aplikáciu tohto prístupu pre určitý prípad jednostranného fuzzy zovšeobecnenia konceptového zväzu. Definíciu všeobecného prípadu fuzzy konceptového zväzu je možné nájsť v [3]. Vstupom je tabuľka s reálnymi hodnotami, autor upresňuje postup pri fuzzifikácii takéhoto prípadu (tzv. jednostranná fuzzifikácia), definuje potrebné zobrazenia a vzdialenostnú funkciu pre tento prípad.

Na týchto základoch boli postavené naše prvé experimenty s množinou textových dokumentov [4], ktorou bolo 420 článkov z anglického denníka Times z obdobia 60-tych rokov minulého storočia. Aglomeratívnym zhlukovacím algoritmom nájdené zhluky (koncepty) sme sa snažili pre zlepšenie interpretácie popísať množinou charakterizujúcich termov (slov) aplikáciou metódy LabelSOM [12]. Pre prácu z dokumentmi bola použitá nami navrhnutá knižnica JBOWL [1] pre spracovanie a reprezentáciu textových dokumentov.

V tejto práci sa taktiež zaoberáme štúdiom určitých aspektov využitia princípu redukcie problému pri tvorbe konceptuálnych modelov nad textovými dokumentmi (potenciálne s výhľadom na tvorbu ontológií), t.j. rozdeleniu pôvodnej úlohy vytvorenia modelu použitím FCA priamo z celej množiny dokumentov na množinu sub-problémov tvorby častí modelu z menších podmnožín pôvodného korpusu. Táto vízia sa objavila ako čiastočný dôsledok predchádzajúcich experimentov a ako odpoveď na požiadavku riešenia problému budovania ontológiu, najmä pre potreby zlepšenia vyhľadávania informácií, s dôrazom na sémanticky neznáme množiny textových dokumentov. Pre základné experimentovanie bola ako metóda redukcie použitá rozdeľujúca zhlukovacia metóda k-stredov, ako testovacia množina bola zvolená rovnaká množina anglicky písaných dokumentov, t.j. články denníka Times.

V kapitole 2 je stručne charakterizovaný uvažovaný prípad jednostranne fuzzifikovaného konceptového zväzu, ako aj prípad použitej zhlukovacej metódy, vzdialenostnej funkcie pre priame generovanie konceptov a metóda pre popis termami. V ďalšej kapitole sú popísané určité skúsenosti s použitým postupom. Na ich základe je načrtnutá možnosť využitia redukcie problému pri budovaní modelu, súčasťou je aj experimentálne štúdium niektorých aspektov takéhoto riešenia. Krátke zhrnutie a myšlienky pre ďalšiu prácu sa nachádzajú v závere príspevku.

2 Aplikácia FCA v doméne textových dokumentov

Táto kapitola popisuje jeden prípad jednostrannej fuzzifikácie konceptového zväzu a jeho budovanie pomocou aglomeratívnej zhlukovacej metódy [7], ako aj príklad aplikácie a doplnenia tohto postupu pri popise konceptov v doméne textových dokumentov charakterizujúcimi termami [4].

2.1 Jednostranne fuzzy konceptový zväz

Majme neprázdnu množinu atribútov A a množinu objektov B a fuzzy reláciu R, pričom R: $A \times B \rightarrow [0,1]$. Túto reláciu potom reprezentuje tabuľka dát, kde $R(b,a)$

vyjadruje hodnotu atribútu *b* objektu *a*. V našom prípade objekt je dokument a atribút je term, resp. miera vyjadrujúca výskyt daného termu (slova) v dokumente.

Definujme zobrazenie τ: $P(B) \rightarrow {}^{A}[0,1]$, ktoré priradí každej množine X prvkov z B nejakú funkciu $\tau(X)$, hodnotou ktorej v bode $a \in A$ je:

$$\tau(X)(a) = \min\{R(a,b) : b \in X\}, \tag{1}$$

t.j. táto funkcia priradí každému atribútu aspoň takú hodnotu, akú minimálne dosahujú všetky objekty z X.

Definujme ďalej zobrazenie σ: ${}^{A}[0,1] \rightarrow P(B)$, ktoré potom nasledovne priradí každej funkcii f: $A \rightarrow [0,1]$ množinu:

$$\sigma(f) = \{b \in B : \ (\forall a \in A) \ \ R(a,b) \geq f(a)\}, \tag{2}$$

t.j. objekty ktoré dominujú danej funkcii f v každom atribúte.

Tieto zobrazenia majú nasledovné vlastnosti:

$$X_1 \subseteq X_2 \rightarrow \tau(X_1) \geq \tau(X_2) \tag{3}$$

$$f_1 \leq f_2 \rightarrow \sigma(f_1) \supseteq \sigma(f_2) \tag{4}$$

$$X \subseteq \sigma(\tau(X)) \tag{5}$$

$$f \leq \tau(\sigma(f)) \tag{6}$$

Potom dvojica zobrazení $<\tau,\sigma>$, ktorá spĺňa tieto podmienky, tvorí tzv. Galoisovu konexiu.

Definujme zobrazenie cl: $P(B) \rightarrow P(B)$ ako kompozíciu zobrazení τ and σ, t.j. nech pre každú množinu $X \in B$ je $cl(X) = \sigma(\tau(X))$. Z vlastností spájaných zobrazení (3) až (6) je možné ľahko vidieť, že cl je uzáverovým operátorom, t.j. spĺňa nasledovné podmienky:

$$X \subseteq cl(X) \tag{7}$$

$$X_1 \subseteq X_2 \rightarrow cl(X_1) \subseteq cl(X_2) \tag{8}$$

$$cl(X) = cl(cl(X)). \tag{9}$$

Tak ako v prípade binárnej tabuľky dát, pre identifikáciu konceptov sú dôležité podmnožiny objektov $X \subseteq B$, pre ktoré je $X = cl(X)$. Potom pár $<X,\tau(X)>$ sa nazýva *fuzzy koncept*, pričom X je extentom a $\tau(X)$ je intentom daného konceptu. Následne, množina L všetkých extentov, t.j.

$$L = \{X \in P(B) : \ \ X = cl(X)\}, \tag{10}$$

usporiadaná inklúziou je zväz, v ktorom pre operácie infima a suprema platí:

$$X_1 \wedge X_2 = X_1 \cap X_2 \tag{11}$$

$$X_1 \vee X_2 = cl(X_1 \cup X_2) \tag{12}$$

Tento zväz potom nazývame *jednostranne fuzzy konceptovým zväzom*.

2.2 Vzdialenostná funkcia a zhlukovací algoritmus

Pre potreby zhlukovania je potrebné definovať vzdialenostnú funkciu (metriku), podľa ktorej je možné určiť navzájom blízke koncepty. Autor v [7] popisuje funkciu, ktorá zovšeobecňuje (pseudo)metriku dvojice autorov Rice a Siff pre fuzzy prípad:

$$\rho(X_1, X_2) = 1 - \frac{\sum_{a \in A} \min\{\tau(X_1)(a), \tau(X_2)(a)\}}{\sum_{a \in A} \max\{\tau(X_1)(a), \tau(X_2)(a)\}}. \quad (13)$$

Vzdialenostná funkcia ρ nadobúda hodnoty z intervalu [0,1], je pseudometrikou na $P(B)$ a metrikou na L (dôkaz napr. v [8]). Aglomeratívny zhlukovací algoritmus je ten istý, ktorý použili Rice a Siff v [13], akurát bola použitá uvedená funkcia ρ pre fuzzy prípad. Pseudokód algoritmu vyzerá nasledovne:

```
C ← D ← {cl({b}): b∈B}
while(|D|>1) do
{
    m ← min{ρ(X1,X2): X1,X2∈D,X1=X2}
    E ← {<X1,X2>∈D×D: ρ(X1,X2)=m}
    V ← {X1∈D: (∃ X2∈D)<X1,X2>∈E}
    N ← {X1∨X2: <X1,X2>∈E}
    D ← (D\V) ∪ N
    C ← C ∪ N
}
```

Premenná D je množina konceptov (zhlukov) v aktuálnej iterácii, C je zjednotenie všetkých množín D od začiatku po aktuálnu iteráciu. Premenná m je minimálna vzdialenosť dvojíc z D. Potom množina E obsahuje "hrany", t.j. dvojice klastrov z D, ktorých vzdialenosť je m, V obsahuje "vrcholy" (konce hrán). Prvky v N sú nové zhluky, nová iterácia nahradzuje v D prvky z V prvkami z N.

2.3 Predspracovanie textových dokumentov

Pre naše experimenty sme vybrali kolekciu 420 článkov v anglickom jazyku z denníka Times zo 60-tych rokov minulého storočia. Jej predspracovanie v našom prípade pozostávalo z nasledujúcich krokov:

- Tokenizácia
- Eliminácia neplnovýznamových slov
- Úprava slov na základný tvar (stemming)
- Výber termov na základe ich frekvencie výskytu

V prvom kroku (tokenizácia) bol vstupný text transformovaný na tokeny. Tokenizácia je nevyhnutnou súčasťou predspracovania, ostatné kroky sú voliteľné (používajú sa kvôli redukcii priestoru termov). Získané tokeny predstavujú základný slovník spracovávanej množiny dokumentov.

V nasledujúcom kroku boli z tejto množiny odstránené neplnovýznamové slová („stopwords“ - spojky, častice, zámená, ...) porovnaním so zoznamom slov v tzv. stop-liste.

V ďalšom kroku boli zostávajúce tokeny transformované na ich základné tvary - stemy (ako napríklad „wins“, „win“, sú transformované na jeden koreň – „win“). Dosiahneme tak opätovné zmenšenie slovníka (pre rôzne morfologické tvary zostane

len akýsi koreň slova – „stem“). Pre naše účely sme pri stemovaní použili pre anglický text známy Porterov algoritmus [10], založený na orezávaní sufixov.

Posledný krok na základe frekvencie výskytu slov takto upraveného slovníka odstráni tie slová, ktoré sa vyskytovali v texte v príliš malom počte dokumentov. Preto sa ponechajú len tie tokeny (termy), ktorých frekvencia výskytu dosiahla aspoň minimálnu zvolenú hodnotu frekvencie výskytu. Skúmaná kolekcia dokumentov je potom reprezentovaná pomocou dokument-term matice. Pre váhovanie termov sme použili *tf-idf schému*, ktorá priradzuje každej dvojici dokument-term reálne číslo (príslušnú váhu) nasledovne:

$$w_{ij} = tfidf_{ij} = tf_{ij} \log\left(\frac{ndocs}{df_i}\right), \tag{14}$$

kde tf_{ij} je frekvencia výskytu *j*-tého termu v *i*-tom dokumente, *ndocs* je počet dokumentov v kolekcii a df_i je počet dokumentov obsahujúcich daný *j*-tý term (dokumentová frekvencia). Vektory boli následne normalizované na jednotkovú dĺžku. Takto sme s vektorov vytvorili vstupnú objekt-atribútovú (dokument-term) maticu (formálny fuzzy kontext), pričom jej reprezentácia je vo forme množiny "riedkych" vektorov. Minimálny prah pre frekvenciu bol výskyt termu aspoň v troch dokumentoch (po orezaní zostalo 5779 termov), maximálna hodnota prahu bola 40 (po orezaní 405 termov). Pre indexovanie, váhovanie a reprezentáciu dokumentov, ako aj pre ich spracovanie a algoritmizáciu metód bola použitá nami navrhnutá knižnica JBOWL [1].

2.4 Použitie metódy LabelSOM na vygenerované koncepty

Metóda LabelSOM [12] je založená na pozorovaní, že po naučení SOM-u (SOM – samoorganizujúce sa mapy) sa zložky vektora váh neurónu čo najviac podobajú odpovedajúcim zložkám vstupného vektora vo všetkých vstupných signáloch, ktoré boli namapované na tento jeden neurón, ako aj na niektoré zložky vektorov susediacich neurónov.

Miera relevancie jednotlivých príznakov je založená na kvadratickej odchýlke príkladov daného neurónu od centra. Vyberú sa tie zložky vektora, ktorých hodnota chyby je blízka nule. Kvadratická odchýlka vektora sa vypočíta pre každý neurón *i* ako priemerná vzdialenosť medzi zložkami vektorov všetkých vstupných dát namapovaných na neurón *i* a zložkami vektora váh tohto neurónu. Aplikácia tohto prístupu v našom prípade je jednoduchá, príklady určitého konceptu ("neurónu") sú priamo jemu prislúchajúce objekty a váhový vektor konceptu (centroidu, "neurónu") je napr. aritmetický priemer váhových vektorov daných objektov konceptu.

Nech C_i je množina objektov x_j konceptu *i*. Vypočítaním vzdialeností pre všetky zložky vektora *k* cez všetky vektory x_j ($x_j \in C_i$) dostaneme kvadratickú chybu vektora q_i pre všetky koncepty *i* (q_{ik} – vypočítaná odchýlka pre *k*-ty atribút v rámci *i*-tého konceptu, m_{ik} – *k*-tá zložka vektora váh „centra“ *i*-tého konceptu, x_{jk} – *k*-tá zložka vektora váh objektu x_j patriaceho k *i*-tému konceptu):

$$q_{ik} = \sum_{x_j \in C_i} \sqrt{\left(m_{ik} - x_{jk}\right)^2}, \quad k = 1,\ldots,n \tag{15}$$

Vybraním tých zložiek vektora, ktorých hodnota chyby je blízka nule sa získa zoznam atribútov, ktoré sa nachádzajú vo veľkom počte vstupných vektorov (dokumentov) namapovaných na tento zhluk. Tieto atribúty slúžia ako názvy resp. popisy (labels) jednotlivých konceptov.

Problémom v prípade textov je fakt, že sa vyskytuje veľký počet zložiek vstupného vektora, ktorých hodnota je nulová (existuje veľa termov - slov, ktoré nie sú obsiahnuté v danej skupine dokumentov). Pri bežnom použití LabelSOM-u by neurón bol prakticky popísaný len termami, ktoré sa v daných dokumentoch ani nevyskytujú (samozrejme vzhľadom na daný zhluk). Riešenie je určiť tie zložky váhového vektora, ktoré na jednej strane preukazujú približne tie isté hodnoty pre všetky objekty špecifického konceptu a na druhej strane majú vysokú celkovú hodnotu váhového vektora, čo indikuje ich dôležitosť. Na dosiahnutie toho sa používa prah t a vybrané sú iba atribúty s veľmi nízkou hodnotou kvadratickej chyby, ale ktoré vykazujú hodnotu váhového vektora nad prahom t. V tomto prípade je t vlastne minimálna hodnota atribútu potrebná na to, aby bol vybraný pre popis zhluku.

Po aplikácii celého procesu (prezentovaného v tejto kapitole) je výsledkom množina konceptov vygenerovaných algoritmom, ktoré sú potom popísané charakterizujúcimi slovami, ktoré z konceptov (zhlukov) vyextrahoval LabelSOM (príklad viď. Obr. 1).

Concept (467):
IntSet 467(|467| = 2) = {6 59}
Labels (467):
indian 0.0849, provinc 0.125887, negoti 0.127356, nehru 0.14293, kashmir 0.149645, india 0.169807, pakistan 0.200657

Obr. 1. Príklad popisu vygenerovaného konceptu. V každom popise je ID konceptu (467), počet jeho dokumentov (|467|=2) a zoznam ich identifikátorov (6 a 59). Poslednú časť tvorí zoznam popisných termov (získaných aplikáciou metódy LabelSOM) a ich váhy, vyjadrujúca aj mieru kvadratickej odchýlky daného atribútu voči stredu (menšia odchýlka znamená významnejší term pre popis). Tieto dokumenty popisujú pokusy o negociáciu mierových zmlúv medzi Pakistanom a Indiou v sporoch o Kašmír (Nehru bol vtedajší premiér v Indii).

Pri našich experimentoch sa ukázalo, že aglomeratívne zhlukovanie konceptov je z hľadiska budovania spodných vrstiev hierarchie konceptov zaujímavé, pri veľkých kontextoch je (okrem časovej náročnosti) však postupné posúvanie v hierarchii problematické a neprehľadné, zhoršuje sa aj možnosť interpretácie. Použitie algoritmu LabelSOM síce uľahčilo proces interpretácie, stále je však tento proces pri veľkom vstupnom kontexte značne subjektívny a nejednoznačný. Navyše vlastnosti klasickej vektorovej reprezentácie textov spôsobujú rovnaké problémy, ako pri použití klasických metód (ani min/max metrika sa nevyhla problémom, ktoré majú aj iné metriky na vektorovej reprezentácii textov).

3 Možnosti redukcie problému pomocou zhlukovania

Jedným z dôsledkov skúseností s generovaním konceptov pomocou doteraz popísaného postupu je myšlienka redukcie pôvodného problému na sub-problémy, najmä za predpokladu použitia väčších množín dokumentov v budúcnosti. Logicky prvou ideou je potom klasický redukčný prístup rozdelenia vstupnej množiny na prijateľné podmnožiny, ich spracovanie a následná vzájomná konfrontácia vytvorených sub-modelov, s cieľom vytvoriť model pre celú množinu.

Využitie podobného postupu pri generovaní konceptov je možné nájsť v rôznych zdrojoch. Formálny prístup využitia rozdelenia množiny, tvorby menších častí zväzu a jeho následné spájanie je možné nájsť v [14], resp. [15]. Prístup k redukcii priamo veľkosti fuzzy konceptového zväzu použitím SVD dekompozície je možné nájsť napr. v [2].

3.1 Postupnosť procesov pri redukcii na sub-problémy

V našom ponímaní je možné využitie FCA pri tvorbe hierarchie konceptov a relácií medzi nimi. Veľkosť problému v doméne textových dokumentov nás privádza k myšlienke redukcie na sub-problémy. Jednou z možností je zakomponovanie redukcie pomocou rozdeľujúcej zhlukovacej procedúry. Potom by riešenie budovania konceptuálneho modelu mohlo pozostávať z nasledujúcich krokov:

- *Zhlukovacia fáza* – predzhlukovanie vstupnej množiny dokumentov, nájdenie zaujímavých podskupín podobných dokumentov.
- *Redukčná fáza* – rozdelenie do podmnožín podľa predchádzajúceho kroku, príprava týchto podmnožín na ďalšie spracovanie, redukcia atribútov na základe rozdelenia do zhlukov (t.j. filtrácia termov pre jednotlivé zhluky podľa konkrétnych objektov v nich).
- *Fáza budovania lokálnych modelov* – každý zhluk je nezávislou trénovacou množinou (prípadne s regulovanou kardinalitou váhových vektorov), pre každý z nich sa vytvorí lokálny konceptuálny model využitím napr. fuzzy FCA prístupu.
- *Fáza spájania* – množina lokálnych modelov sa v tomto kroku spája do jedného veľkého modelu, vzťahujúceho sa k celej testovanej množine.

Z hľadiska budúcnosti sémantického webu si pod konceptuálnym modelom môžeme predstaviť ontológiu. Potom je postup v podstate zhodný a vykresľuje ho Obr.2.

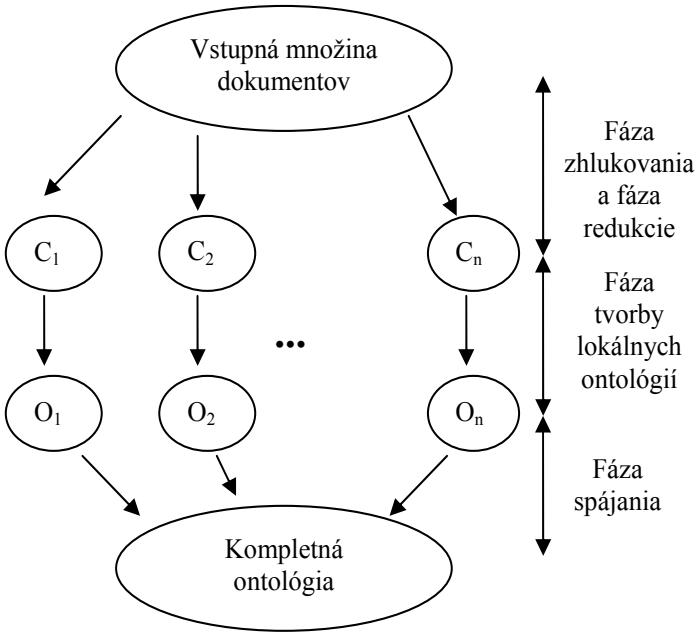
Vo fáze zhlukovania sa ako najskôr použiteľné zdajú byť rozdeľujúce metódy. Zaujímavé by mohlo byť použitie hierarchického algoritmu na báze samo-organizujúcich sa máp (SOM). Úlohou by bolo rozdeľovať postupne množinu podľa určitých kritérií, napr. identifikovať skupiny podobných dokumentov do určitého počtu prvkov v každom klastri (pre naše jednoduché experimenty s rozdelením sme však zatiaľ použili len algoritmus k-stredov).

Redukčnou fázou sa myslí priamo časť prerozdelenia pôvodnej množiny a príprava trénovacích dátových tabuliek pre ďalšiu fázu. Na tomto mieste sa otvára priestor pre redukciu príznakového priestoru, t.j. ku každému zhluku (množine jeho

objektov) sa vyberie určitá podmnožina termov, a to podľa ich „lokálneho“ výskytu v dokumentoch klastra.

Budovanie lokálnych modelov je veľmi dôležitá časť celého procesu. Existuje viacero rôznych známych prípadov využitia FCA pri budovaní ontológie, napr. autori v [5] navrhli algoritmus pre učenie hierarchií z dokumentov (proces predspracovania však bol na lingvistickej úrovni, našou snahou je zistiť do akej miery je možné využiť pomerne jednoduchú vektorovú reprezentáciu), zaujímavá je aj úprava konceptuálneho zhlukovania využitím myšlienok fuzzy FCA pre automatické generovanie hierarchií prezentovaná v [11]. V rámci prezentovaného procesu však nie sme príliš viazaný na charakter algoritmu, ktorý sa pri budovaní lokálnych modelov použije, viazanie sa môže prejaviť skôr v procese spájania.

Hierarchický SOM môže byť zaujímavý taktiež v procese spájania, relácie medzi jednotlivými časťami modelu môžu byť objavené a testované s dôrazom na pozíciu klastrov v mape a v rámci celej hierarchie vytvorenej vo fáze zhlukovania.

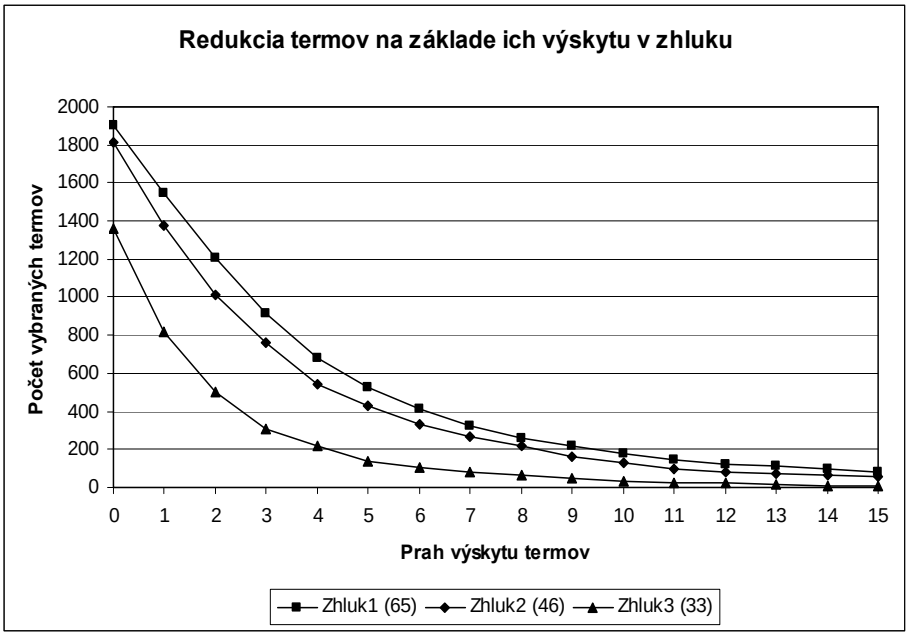


Obr. 2. Všeobecná schéma procesu tvorby ontológie (resp. iného konceptuálneho modelu) založenom na rozdeľujúcej redukcii („rozdeľuj a panuj“). V zhlukovacej fáze je vstupná množina rozdelená na menšie trénovacie podmnožiny (s redukovanými váhovými vektormi). Následne je ku každému zhluku C_i napr. pomocou fuzzy FCA prístupu vytvorená časť ontológie O_i (lokálny model zhluku). Nakoniec sú jednotlivé časti pospájané do celkovej ontológie vo fáze spájania.

3.2 Rozdelenie množiny a redukcia atribútov

Spomínané rozdelenie pôvodného vstupného korpusu má vplyv na ďalšie časti celého procesu. Jedným z prvých je príprava podmnožín na spracovanie v ďalšom kroku. Samozrejme počet riadkov podtabuľky je daný dokumentmi zhluku, objavuje sa tu však možnosť, ako vybrať atribúty a ako pristúpiť k ich hodnotám v tabuľke (t.j. či sa hodnoty nejakým spôsobom prepočítajú, alebo sa len pre vybrané atribúty skopírujú hodnoty z pôvodnej tabuľky).

Ak algoritmom ktorý buduje lokálny model je metóda založená na fuzzy FCA prístupe podobnom tomu, aký je popísaný v druhej kapitole, obyčajné skopírovanie hodnôt atribútov by mohlo byť postačujúce. Dôvod je jednoduchý, FCA prístup generovania konceptov je založený priamo na zobrazeniach, ktoré pracujú nad podmnožinami potenčnej množiny, t.j. sú to podobné výbery riadkov z tabuľky dát. Naše prerozdelenie pomocu zhlukovania je v podstate len jedným možným rozkladom na disjunktné podmnožiny v rámci celej potenčnej množiny nad objektami v tabuľke dát (je to veľmi podobné s použitím zhlukovania nad konceptami, viď. kapitola 2.2, kde redukcia konceptov je založená na preferencii spojenia dvoch podmnožín, t.j. konceptov, do väčšej množiny, t.j. konceptu vyššej kardinality).



Obr. 3. Graf redukcie termov na základe ich výskytu v dokumentoch zhluku. Graf zobrazuje počet vybraných termov (prah výskytu termov *tf* znamená, že výskyt termu dosiahol najmenej hodnotu *tf+1*). Graf prezentuje tri zhluky (viď. legenda grafu), v zátvorke je počet dokumentov zhluku.

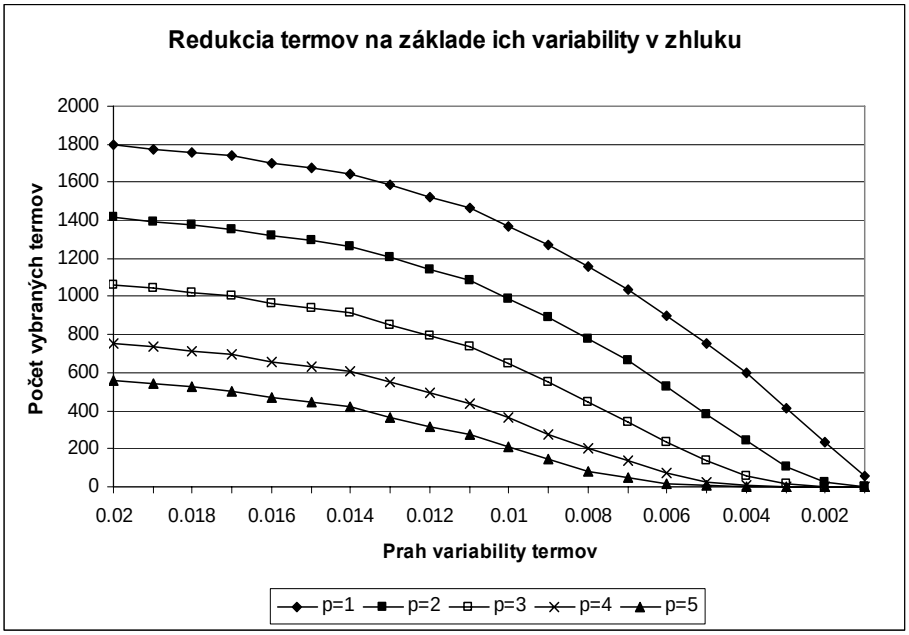
Výber samotných atribútov je opäť možné uskutočniť viacerými spôsobmi. Pri našich experimentoch s k-stredovým algoritmom sme vyskúšali (t.j. k akej redukcii je

možné sa dostať, použili sme pritom 420 dokumentov a 2015 atribútov, t.j. term sa musel nachádzať aspoň v 10 dokumentoch celej kolekcie) dve možnosti:

- *Redukcia zistením lokálnych početností*
- *Redukcia na základe variability v rámci zhluku*

V prvom prípade sa v rámci objektov daného zhluku vyberú tie atribúty, pri ktorých je dosiahnutá určitá početnosť nenulových prvkov (term sa v dokumentoch zhluku vyskytol aspoň *tf*-krát, *tf* je potom vlastne prahom orezania). Na grafe k tomuto prípadu (Obr.3) je možné vidieť aký vplyv má prah na počet vybraných atribútov.

V druhom prípade sme použili prístup založený na rovnakom princípe, na akom pracuje metóda LabelSOM. Atribúty sú vybrané na základe ich variability v rámci zhluku. Čím má atribút menší rozptyl voči stredu zhluku, tým väčšiu šancu má, že bude vybraný. Vzhľadom k vlastnostiam vektorovej reprezentácie je aj tu požiadavka na minimálny počet výskytov v dokumentoch (inak by stačila nulová resp. jedna hodnota). Na nasledujúcom grafe (Obr. 4.) je možné vidieť, akú redukcii dosiahneme pri sledovaní variability termov (ak je menšia ako prah, term je vybraný), za predpokladu určitého minimálneho výskytu daného termu v dokumentoch zhluku.



Obr. 4. Graf redukcie termov na základe ich variability v rámci zhluku. Graf zobrazuje počet vybraných termov v závislosti od miery variability. Atribúty s variabilitou nižšou ako prah variability boli vybrané, ak dosiahli požadovaný prah výskytu (*p*). Každá krivka teda odpovedá výberu termov na základe variability, avšak pri určitej minimálnej povolenej hodnote výskytu termu v dokumentoch zhluku (všetky krivky v tomto grafe sa týkajú jedného zhluku, ktorý obsahoval 66 dokumentov).

Ďalšie kroky v budúcnosti by mali viesť k realizácii celého procesu. Z hľadiska sémantického webu sa chceme zamerať na vytváranie ontológií, pričom tento postup je v takom prípade aplikovateľný (je to taktiež konceptuálny model). Ako sa ukazuje, ontológie môžu prispieť k zlepšeniu výsledkov vyhľadávania informácií [9], naša vízia je využiť automaticky vytvorené ontológie pre tento účel nad množinami (zo sémantického hľadiska) neznámych dokumentov, a prispieť tak k budovaniu sémantického webu.

4 Záver

Cieľom tejto práce bolo popísať predstavu fuzzy FCA prístupu k budovaniu konceptuálnych modelov z množín textových dokumentov, prezentovať niektoré naše počiatočné experimenty s menšou testovacou množinou, niektoré naše skúsenosti potom vyústili do predstavy kombinácie princípu redukcie problému a fuzzy FCA prístupu. Náš prístup k redukcii problému spočíva v rozdelení vstupného korpusu množiny rozdeľujúcim zhlukovacím algoritmom na zhluky. Pre každý klaster sa následne vytvorí „lokálny" model. Množina takýchto modelov (častí ontológií) sa vo fáze spájania skombinuje do podoby výsledného modelu (celá ontológia). V tomto článku sme sa venovali len štúdiu niektorých aspektov tohto procesu, pre základné experimentovanie bol použitý zhlukovací algoritmus k-stredov. Testovacou množinou bolo 420 článkov denníka Times z obdobia 60-tych rokov minulého storočia.

V budúcnosti by sme chceli funkčne implementovať a otestovať prezentovaný postup v celej škále, pričom zámerom je vybudovanie ontológií nad množinami sémanticky nepopísaných textových dokumentov a ich aplikácia za účelom zlepšenia vyhľadávania nad takouto množinou.

Práca prezentovaná v tomto príspevku vznikla za podpory Vedeckej grantovej agentúry Ministerstva školstva SR a Slovenskej akadémie vied (VEGA) v rámci projektu č.1/1060/04 s názvom "Klasifikácia a anotácia dokumentov pre sémantický web" a za podpory nemecko-slovenského vedeckého projektu DAAD č.8/2004 "Využitie objavovania znalostí v textoch pre extrakciu metadát a sémantické vyhľadávanie".

Referencie

1. Bednar, P., Butka, P., Paralic, J. Java Library for Support of Text Mining and Retrieval. In Proceedings of Znalosti 2005, pp. 162-169, Stará Lesná, 2005.
2. Bělohlávek, R., Gajdoš, P., Sklenář, V., Snášel, V. Reducing the size of fuzzy concept lattices by SVD. In Znalosti 2005, sborník posterů, pp.1-4, Stará Lesná, 2005.
3. Bělohlávek, R., Vychodil, V. What is a fuzzy concept lattice? In Proceedings of CLA 2005, pp. 34-45, Olomouc, 2005.
4. Butka, P. Aplikácia zhlukovania fuzzy konceptov v doméne textových dokumentov. Prijaté na konferenciu ITAT 2005, Račkova Dolina, 2005.

5. Cimiano, P., Hotho, A., Staab, S. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. In Journal of Artificial Intelligence Research 24:2005, pp. 1-15, 2005.
6. Ganter, B., Wille, R. Formal Concept Analysis. Mathematical Foundations, Springer-Verlag, 1997.
7. Krajči, S. Clustering Algorithm Via Fuzzy Concepts. In Proceedings of DATESO 2003, pp. 94-100, Desná, 2003.
8. Krajči, S.: Vektorová min/max metrika. In Proceedings of ITAT 2002, pp. 119-123, Malinô Brdo, 2002.
9. Paralič, J., Koštiaľ, I. Ontology-based Information Retrieval. In Proceedings of IIS 2003, 14th International Conference on Information and Intelligent Systems, Varaždín, 2003.
10. Porter, M.F. An algorithm for suffix stripping. In V Program, 14(3), 130-137, 1980.
11. Quan,T.T., Hui, S.C., Cao,T.H. A Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data. In Proceedings of CLA 2004, pp. 1–12, Ostrava, 2004.
12. Rauber, A. On the labeling of self-organizing maps. In Proceedings of IJCNN'99, Washington, 1999.
13. Rice, M.D., Siff, M. Clusters, Concepts, and Pseudometrics. In Proceedings of First Irish Conference on Mathematical Foundations of Computer Science and Information Technology, 2001.
14. Valtchev, P., Missaoui, R., Lebrun, P. A partition-based approach towards constructing Galois (concept) lattices. Discrete Mathematics 256(3):801-829, 2002.
15. Valtchev, P., Duquenne, V. Towards scalable divide-and-conquer methods for computing concepts and implications. In Proceedings of JIM'03: Knowledge Discovery and Discrete Mathematics, pp. 3-15, Metz, 2003.

Annotation:

Combination of problem reduction techniques and fuzzy FCA approach for building of conceptual models from textual documents

Formal Concept Analysis (FCA) can be used for formal analysis of data tables and identification of similar objects clusters (concepts). Computational complexity of creation of concept lattices from large context is considerable. For that reason methods for effective generation of concepts are important, e.g. combination with other algorithms and use of heuristics. Extension of classic FCA (binary table data) is one-sided fuzzy version that works with real values in the object-attribute table (document-term matrix in case of vector representation of textual documents). This paper describes some aspects of combination of problem reduction techniques (e.g. clustering) and "fuzzy FCA" approach in conceptual models building (with the vision of ontology building) from textual documents. We used k-means algorithm in our experiments as problem reduction method and small set of English documents as test dataset.

Využití shlukování pro kompresi dat*

Jiří Dvorský, Jan Martinovič, Václav Snášel

Katedra informatiky, FEI, VŠB - Technická Univerzita Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba
{jiri.dvorsky,jan.martinovic,vaclav.snasel}@vsb.cz

Abstrakt. V průběhu tvorby textové databáze je s dokumenty na vstupu prováděna řada činností. Mezi ně patří lingvistická analýza, indexování, shlukování, komprese atd. V našem příspěvku se věnujeme využití informací o podobnosti dokumentů získaných aglomerativním shlukovacím algoritmem pro kompresi plných textů těchto dokumentů. Ukazuje se, že tímto způsobem je možné zlepšit výsledky komprese. V závěru příspěvku budou popsány experimentální výsledky.

Klíčová slova: aglomerativní shlukování, komprese dat, slovní komprese

1 Úvod

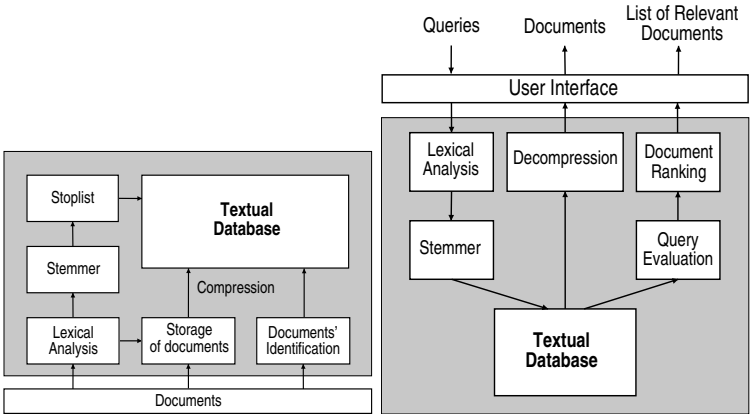
Moderní informační společnost produkuje nesmírné množství informací v textové podobě. Efektivní ukládání textů a vyhledávání potřebných informací v uložených textech je úkolem *dokumentografických informačních systémů* (DIS). Uživatel přistupující k DIS má potřebu získat z tohoto systému informaci, která je pro něj *relevantní*. Při vyhledávání informace v DIS uživatel systému zadává svůj požadavek, *dotaz*, a očekává výsledek ve formě množiny dokumentů, které systém vybírá jako odpovídající požadavku uživatele, jako splňující jeho dotaz. Uživatel klade dotazy pomocí nějakého dotazovacího jazyka, přičemž je podstatná jak logická struktura dotazu, tak i *termy*, které uživatel při formulaci svého dotazu použije. Obecným požadavkem, kladeným na DIS, je poskytnout uživateli systému relevantní dokumenty (tj. ty, které ho zajímají), a to pokud možno všechny (které jsou v systému k dispozici), a právě jen ty (relevantní). Algoritmus realizující výběr relevantních dokumentů předpokládá vhodné předzpracování vstupních informací o dokumentech a uložení do vhodných datových struktur. Architektura typického DIS je znázorněna na obrázcích 1(a) a 1(b).

Je zřejmé, že velikost DIS neustále roste. Jediné co lze očekávat do budoucnosti je ještě rychlejší nárůst požadavků na kapacitu pamětí, úložných kapacit disků, DVD disků etc. Informační explozi je možno čelit v podstatě dvěma způsoby:

1. extenzívním – opětovným nákupem pamětí s vyšší kapacitou nebo
2. intezívním – lepším uložením dat v pamětech.

* Tento příspěvek vznikl za podpory grantu GA ČR No. 201/05/P145.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 83–94, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.



(a) Indexování dokumentů (b) Vyhodnocení dotazu

Obr. 1. Architektura DIS

Řešení první je z výzkumného hlediska nezajímavé. Klíčem k druhému řešení je *komprese dat*. Datovou základnu typického DIS představuje *textová databáze* (viz obr. 1(a)), která uchovává všechny potřebné informace pro provoz DIS. Textová databáze se typicky skládá z následujících částí:

- plné texty dokumentů tvořící kolekci dokumentů,
- datové struktury určené k vyhledávání dokumentů,
- seznam identifikátorů dokumentů, jejich atributů a další pomocné struktury.

Haskin v [10] uvádí, že velikost pomocných struktur textové databáze (tj. mimo vlastní texty dokumentů) představuje 50% až 300% procent velikosti původních dokumentů. Z toho je vidět, že textová databáze představuje vhodný materiál ke kompresi. Stačí-lo by tedy použít některou z bezeztrátových kompresních metod a ušetřit více, či méně prostoru. Problém komprese v DIS ovšem není tak jednoduchý, jak se na první pohled zdá. Komprese na jedné straně uspoří prostor pro data, na druhé straně může znamenat jistou režii v provozu, tj. přidání jistého času k ceně přístupu k datům. Také úspora prostoru musí být významná, aby měla smysl.

Praxe ukazuje, že při ukládání rozsáhlých textových databází má smysl uvažovat, analyzovat a navrhovat *úzce specializované metody komprese*, vhodnými pouze pro určitý typ dat v DIS. I ušetření např. jednoho bitu v datových strukturách určených k vyhledávání, zlepšení kompresního poměru textu v DIS o jedno procento vede k úsporám desítek megabytů. Mezi významné faktory ovlivňující výběr určité kompresní metody patří (viz také [20]):

- kompresní poměr,
- vysoká rychlost dekomprese,
- možnost dekomprese daného dokumentu,
- přijatelná rychlost komprese,
- integrace s vyhledáváním.

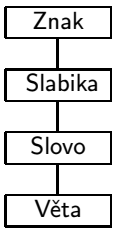
Naším cílem je vyjít z existujících slovních kompresních algoritmů a pomocí shlukování se pokusit dosáhnout zlepšení výkonu těchto kompresních algoritmů.

Příspěvek je organizován následujícím způsobem. Nejdříve je v sekci 2 osvětlen pojem slovní komprese. V sekci 4 je ve stručnosti nastíněn vektorový model DIS. Následuje sekce 5 se stručným popisem shlukové analýzy využité při kompresi. Sekce 6 pojednává o možnosti využití informací získaných shlukovou analýzou při kompresi. Experimentální výsledky jsou uvedeny v sekci 7. Sekce 8 obsahuje závěrečné zhodnocení dosažených výsledků.

2 Slovní komprese

Kompresní algoritmus transformuje obecně vstupní data, která obsahují jistou redundanci, na data výstupní, u nichž je redundance snížena na minimum. Za vstup a výstup kompresních algoritmů se obecně považují řetězce symbolů určité abecedy. Na abecedu nejsou kladeny žádné požadavky. Výběr abecedy je tedy věcí volby, při které se uplatňují různá hlediska.

Při hledání vhodné abecedy vyjdeme ze syntaktické struktury přirozených jazyků. Zamyslíme-li se nad touto strukturou, která je ve velice zjednodušené podobě znázorněna na obrázku 2, můžeme objevit korespondenci mezi strukturou jazyka a možnými abecedami. Každá úroveň představuje jednu potenciální abecedu. První úroveň představuje abeceda založená na znacích[1]. Další možnou úrovní je abeceda slabik. Zde narážíme na problém, kterým je identifikace slabik v textu [13]. Daleko větší možnosti nabízí úroveň třetí – abeceda založená na *slovech*.



Obr. 2. Syntaktická struktura přirozeného jazyka

Kompresní metoda založená na abecedě slov, nazvěme ji *slovní metodou komprese* [11, 20, 5–7], považuje text za sekvenci slov daného jazyka. Při kompresi se pak předpokládá, využití nerovnoměrné distribuce pravděpodobností výskytů

[1] Pro kompresi textů je samozřejmě možné aplikovat libovolný, bezeztrátový, kompresní algoritmus, který bude považovat text za sekvenci znaků – bytů. Tato volba je naprosto regulérní a v současné době je i maximálně využívána. A to jednak z důvodu své univerzality, tak i pro minimální systémové nároky a snadnou implementaci. Je možné využít řady dostupných knihoven kompresních algoritmů.

jednotlivých slov, u statistických metod komprese, nebo shlukování slov do syntaktických struktur jazyka, u metod slovníkových. Předpokládáme totiž, že se strukturou jazyka neřídí jen znaky, ale také slova. Máme tím na mysli:

- ustálené fráze např. "Dobrý den",
- slovní vazby dané obsahem textu např. v této práci se často opakuje spojení "komprese dat", "slovní komprese".

Dále předpokládáme, že se tato slovní spojení opakují a že je možné na základě tohoto opakování v textu dosáhnout komprese. Nepředpokládáme, že by text byl složen pouze z hapax legomenon[2] – i když i toho lze využít.

Shrneme-li předchozí poznatky o volbě abecedy, vidíme, že slovní abeceda má několik významných vlastností (další jsou uvedeny v [7]):

1. nezávislost na použitém kódování (ASCII/UNICODE),
2. minimální mohutnost (obsahuje jen použité symboly nikoliv všechny potenciálně možné),
3. vystihuje charakteristické shluky písmen v textu.

3 Slovní kompresní algoritmy

Slovní abecedu pro kompresi textů využívá několik kompresních algoritmů. Jednak jde o agloritmus HuffWord [20]. Komprese je založena na tzv. Huffmanově kanonickém kódu. Kanonický kód se od běžného Huffmanova kódu liší tvarem kódových slov, která produkuje.

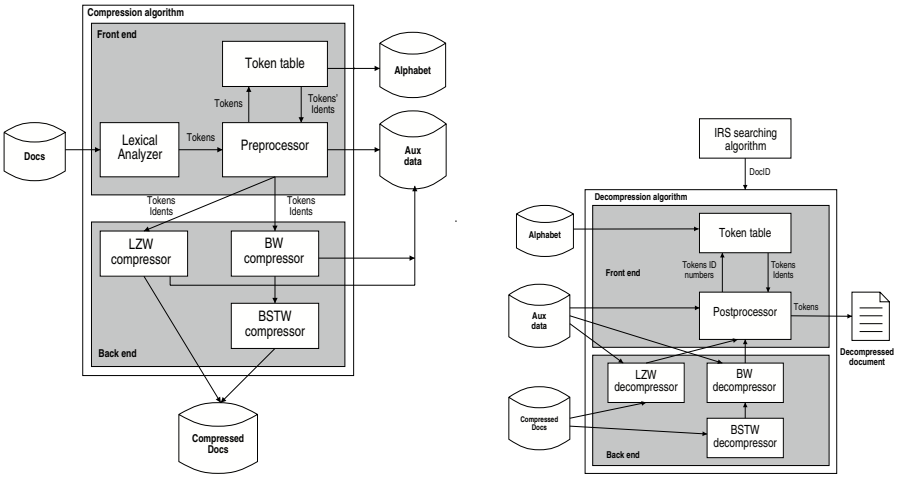
Dále jsou tu metoda WLZW a WBW [7]. Metoda WLZW je založena na algoritmu LZW [19], metoda WBW je pak založena na Burrows-Wheelerově transformaci [4].

Na obrázku 3(a) je zachycena schematická struktura kompresního algoritmu WLZW resp. WBW[3]. Jak je z obrázku patrné, je možné obě kompresní metody rozdělit na přibližně dvě části, nazvané *front end* a *back end*. Obě kompresní metody zpracovávají texty dokumentů ve dvou průchodech. Dělení kompresní metody na dvě části koresponduje s těmito průchody. V jednotlivých průchodech některé části buď vůbec nepracují nebo se jejich činnost liší. Podle pořadí průchodu texty dokumentů můžeme v obou kompresních algoritmech odlišit dvě fáze algoritmu:

- *první fáze* – odpovídá *prvnímu průchodu* kompresního algoritmu. V této fázi dochází ke tvorbě slovní abecedy. Jednotlivé tokeny jsou extrahovány z dokumentů procesem lexikální analýzy, který je realizován částí front end. Tato fáze je sdílena s indexací dokumentů v textové databázi.
- *druhá fáze* – odpovídá *druhému průchodu* kompresního algoritmu. Po ukončení první fáze máme k dispozici kompletní slovní abecedu a může začít vlastní komprese dokumentů. Je opět prováděna lexikální analýza a vznikající sekvence tokenů je komprimována zvoleným algoritmem. V této fázi již pracují obě dvě části kompresního algoritmu, jak front end tak i back end.

[2] Hapax legomenon – slovo s jediným výskytem ve zkoumaném textu.
[3] V našem příspěvku se budeme zabývat metodou WLZW.

Rozdělení kompresního algoritmu na dvě relativně nezávislé části umožnilo od sebe oddělit dvě různé fáze kompresního algoritmu tj. tvorbu slovní abecedy a vlastní kompresi. Pochopitelně toto oddělení zjednodušilo návrh algoritmů, zpřehlednilo implementaci etc.



(a) Schéma kompresního algoritmu WLZW (WBW)

(b) Schéma dekompresního algoritmu WLZW(WBW)

Obr. 3.

Na obrázku 3(b) je zachycena struktura dekompresního algoritmu metod WLZW a WBW. Z obrázku je patrné, že oba dekompresní algoritmy můžeme, stejně jako u komprese, rozdělit na dvě části – *front end* a *back end*. Metody WLZW a WBW jsou konstruovány jako asymetrické, s vyšší složitostí komprese. Z toho plyne že:

- Má-li být dekomprese musí jednodušší než komprese, pak všechna činnost, kterou může vykonat kompresní algoritmus je do tohoto algoritmu přesunuta, tak aby dekompresní algoritmus vykonával pouze nejnutnější.
- Dekomprese má jen jednu fázi. K dekompresi dokumentu je třeba jen jednoho průchodu komprimovaným textem. Všechny objekty zachycené ve schématu 3(b) jsou tudíž v průběhu dekomprese aktivní a průběh dekomprese má „průtokový“ charakter.

Rozdělení dekompresního algoritmu na dvě části umožnilo od sebe separovat samotnou dekompresi a následnou rekonstrukci textu dokumentu z posloupnosti identifikátorů tokenů. Dále toto dělení umožnilo chápat kompresní i dekompresní algoritmus jako dva duální algoritmy s podobnou vnitřní strukturou.

4 Vektorový model DIS

V polovině 20. století Lunh navrhl systém pro vyhledávání informací v textových kolekcích. Systém byl založen na porovnávání informací mezi uloženými dokumenty a informacemi v *dotazu* od uživatele. Ve vektorovém modelu jsou dokument a dotaz reprezentovány *vektorem* [17]. Jednou z výhod tohoto modelu je možnost uspořád podle klesající podobnosti k dotazu dokumenty různých formátů (jakými jsou například textové dokumenty, obrázky, video) [2, 17].

Uvažujme vektorový DIS obsahující údaje o n *dokumentech*, indexovaných pomocí množiny m *termů*. Každý dokument je v indexu reprezentován m-složkovým vektorem, kde každá z m složek odpovídá *váze*, důležitosti jednoho z termů pro identifikaci dokumentu. Formálně:

$$d_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,m}) \in \langle 0,1 \rangle^m$$

Indexový soubor vektorového modelu potom představuje matici

$$D = \begin{pmatrix} w_{1,1} & \ldots & w_{1,m} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \ldots & w_{n,m} \end{pmatrix} \in \langle 0,1 \rangle^{n \times m}$$

kde k-tý řádek odpovídá k-tému dokumentu a l-tý sloupec odpovídá l-tému termu.

Podobnost dvou dokumentů pak definujeme pomocí *kosinové míry*:

$$sim(d_k, d_l) = \frac{\sum_{i=1}^{m} w_{ki} w_{li}}{\sqrt{\sum_{i=1}^{m} w_{li}^2} \sqrt{\sum_{i=1}^{m} w_{ki}^2}}$$

Matici podobnosti $C_{n \times n}$ lze definovat jako:

$$C = \begin{pmatrix} sim(d_1, d_1) & \ldots & sim(d_1, d_n) \\ \vdots & \ddots & \vdots \\ sim(d_n, d_1) & \ldots & sim(d_n, d_n) \end{pmatrix}$$

5 Shluková analýza

Výše popsaná matice podobností dokumentů C představuje v podstatě velké množství čísel, které je možné nějakým způsobem interpretovat. Nabízí se zde možnost sdružit dohromady ty dokumenty, které mají k sobě blízko (ve smyslu koeficientu podobnosti).

Úkolem shlukové analýzy [12, 3] je zjistit, zda mezi objekty existují skupiny objektů se stejnými nebo podobnými vlastnostmi. Tyto skupiny objektů se nazývají *shluky*. Shluk je tedy tvořen množinou navzájem si podobných objektů.

Významným nástrojem shlukové analýzy jsou *hierarchické shlukovací metody*. U těchto shlukovacích metod se vytváří hierarchie rozkladů zadaných objektů. V průběhu výpočtu se vznikají shlukovací hladiny, na kterých jsou body spojovány do shluků. Hierarchické metody se dají rozdělit do dvou skupin:

aglomerativní – Na počátku je každý objekt brán jako jeden shluk, postupně se shluky spojují dohromady. Výpočet je ukončen v momentě, kdy všechny objekty tvoří dohromady jediný shluk.

divizivní – Pracují opačně než aglomerativní metody. Na startu těchto metod tvoří všechny objekty jeden shluk. Shluky se postupně rozpadají až do chvíle, kdy je každý bod samostatným shlukem.

Pro naše účely jsme využili aglomerativní shlukovací metodu, kterou je možno popsat algoritmem uvedeným na obrázku 4.

1. Vytvoříme matici podobnosti dokumentů.
2. Na počátku shlukování, každý dokument představuje jeden shluk, tj. na počátku máme tolik shluků, kolik je dokumentů. Postupně, jak se jednotlivé shluky slučují, shluků ubývá a na konci zůstane pouze jeden shluk.
3. Nalezneme dva nejvíce si podobné shluky p a q a označíme tuto podobnost s_{pq}.
4. Redukujeme počet shluků sloučením shluků p a q. Nový shluk označíme t (shluk s menší podobností vzhledem k nově vznikajícímu shluku t budeme považovat za levého potomka shluku t) a přepočítají se vzdálenosti s_{tr} nově vzniklého shluku t k ostatním shlukům r pomocí průměrové metody:

$$s_{tr} = \frac{N_p s_{pr} + N_q s_{qr}}{N_p + N_q}$$

   Dále označíme l_{pq} jako vzdálenost, na které byly shluky p a q sloučeny. Tato vzdálenost je u této metode rovna hodnotě s_{pq}.
5. Opakujeme předchozí dva kroky, dokud nezůstane pouze jediný shluk.

Obr. 4. Aglomerativní shlukovací metoda

6 Komprese s využitím shlukové analýzy

Při popisu slovních kompresních algoritmů v části 3 jsme dosud nevěnovali pozornost tomu v jakém pořadí dokumenty do kompresního algoritmu vstupují - kompresní metoda pracuje korektně pro libovolné pořadí vstupujících dokumentů. Asi nejjednodušší možností je nechat je vstupovat v tom pořadí v jakém jsou do textové databáze přidávány. Vzhledem k tomu, že kompresní algoritmy jsou založeny na vyhledávání opakujících se úseků textů, je zřejmé, že toto pořadí dokumentů není nutně nejvhodnější. Zlepšení výkonu komprese by bylo možné dosáhnout vhodným přeuspořádáním kolekce dokumentů, aby do kompresního algoritmu vstupovaly za sebou dokumenty si navzájem podobné. Míru podobnosti dokumentů můžeme chápat tak, že v podobných dokumentech budou s vysokou pravděpodobností vyskytovat stejná slova. K ohodnoceni podobnosti

dokumentů můžeme využít například kosinovou mírou, viz odstavec 4. Následně můžeme podobné dokumenty sdružit využitím shlukové analýzy. Pochopitelně nemá cenu provádět shlukování jen pro zvýšení výkonu komprese, protože shlukování většího počtu dokumentů je časově náročná operace. Jelikož ale pracujeme na kompresní metodě pro DIS, výsledky provedeného shlukování dokumentů lze využít při vyhledávání [8, 14]. A naopak, shluky původně určené pro zlepšení vyhledávání lze využít i pro kompresi dokumentů.

Pro shlukování dokumentů jsme využili aglomerativního shlukovacího algoritmu popsaného v části 5. Zůstává však otázkou, jakým způsobem převést hierarchickou, stromovou strukturu na lineární seznam dokumentů. Tento úkol řeší algoritmus TOPIC-CA [14]. Vstupem algoritmu TOPIC-CA je dokument, výstupem pak seznam podobných dokumentů seřazených podle míry podobnosti, ať už klesající nebo rostoucí. Algoritmus TOPIC-CA můžeme stručně popsat způsobem uvedeným na obrázku 5.

1. Označíme výsledný vývoj tématu jako seznam dokumentů S_T.
2. Určíme počet dokumentů, který má seznam S_T obsahovat - $level$.
3. Nalezneme listový shluk obsahující označený relevantní dokument a označíme jej c_t.
4. Dáme dokument v c_t na konec S_T.
5. Shluk c_t nahradíme shlukem o úroven výše v hierarchii shluků a postupně projdeme všechny jeho potomky až k listovým shlukům. Přitom dodržujeme tyto podmínky:
   (a) nejprve vybíráme shluk, který má hodnotu hladiny, na které se spojili jeho potomci, vuči rodiči větší,
   (b) pokud dojdeme k listovému shluku, pak přidáme dokument, který tento shluk obsahuje, do S_T,
   (c) pokud $|S_T| = level$ ukončíme prohledávání a jdeme na bod 7.
6. Pokračujeme bodem 5.
7. Výsledný seznam je C_T.

Obr. 5. Algoritmus TOPIC-CA

Pro přeuspořádání kolekce dokumentů vstupujících do komprese jsme použili dvě strategie:

Most Similar Left (MSL) – provedeme algoritmus TOPIC-CA pro nejlevější dokument v hierarchii shluků. Požadovaný počet dokumentů bude roven celkovému počtu dokumentu v kolekci.

Most Similar Right (MSR) – provedeme algoritmus TOPIC-CA pro nejpravější dokument v hierarchii shluků. Požadovaný počet dokumentů bude roven celkovému počtu dokumentu v kolekci. Dokumenty jsou tak uspořádány v opačném pořadí než v případě strategie MSL.

7 Experimentální výsledky

V experimentální části jsme se zaměřili na praktické ověření vlivu shlukovaní na slovní kompresi textu. Jako experimentální materiál jsme použili dva textové soubory: soubor v bible.txt z Cantenbury Compression corpusu[4] [1] a soubor latimes.txt pocházející z korpusu TREC [9]. Při kompresi byla použita kompresní metoda WLZW [7]. Výsledky komprese touto metodou pro oba soubory jsou uvedeny v tabulce 1. Jednotlivé dokumenty byly komprimovány v tom pořadí jak je obvyklé - u souboru bible.txt obsahující anglickou bibli je uspořádání dáno pořadím veršů a knih daných biblickým kánonem, u souboru latimes.txt, kde se jedná o dva ročníky deníku Los Angeles Times, jsou dokumenty (články) seřazeny podle data vydání.

Nejdříve jsme se zaměřili na soubor bible.txt. Tento soubor je poměrně krátký, jednotlivé dokumenty představovaly jednotlivé biblické verše (v tomto souboru uloženy jako jednotlivé řádky). Z tabulky[5] 2 je patrné, že shlukování bohužel nemá žádoucí vliv na kompresi, ba naopak dochází ke zhoršení výsledků. Při indexování jsme používali stoplist [18, 16] a množina termů vstupujících do shlukovacího procesu byla řízena Zipfovým zákonem. Výsledek se nezměnil ani po té co jsme vyřadili stoplist a následně i eliminaci Zipfovým zákonem [21].

Tabulka 1. Výsledky komprese bez shlukování

	bible.txt	latimes.txt
Původní velikost [byty]	4047392	498360166
Komprimovaná velikost [byty]	885908	158017940
Kompresní poměr [%]	21,888	31,708

Dále jsme provedli celkem 10 náhodných permutací dokumentů. Z tabulky 3 je sice patrné, že náhodné permutace přinášejí ještě horší výsledky než shlukování, ale rozdíl je u takto malého souboru zanedbatelný. Neúspěch shlukování lze v tomto případě přičíst na vrub faktu, že pro dokumenty tvořené jedním veršem lze velice obtížně spočíst vzájemnou podobnost, jinak řečeno míra podobnosti nemá takovou vypovídací schopnost jako u dokumentů větších.

Další série experimentů byla zaměřena na soubor latimes.txt. Tento soubor je už poměrně rozsáhlý. Dokumenty v tomto případě představovaly jednotlivé novinové články, čili texty o rozsahu od dvou do osmi kilobytů. Opět jsme provedli kompresi se shlukováním a tentokrát jen pět náhodných permutací dokumentů a jejich kompresi. Z tabulky 4 je jasně patrné, že u takto velkého souboru shlukování již přináší výsledky. Velikost komprimovaného souboru je o téměř 4% menší a kompresní poměr se zlepšil o přibližně 1,2%[6]. V tabulce 5 je pak možné

[4] Korpus je možné si stáhnout z `http://corpus.canterbury.ac.nz`.

[5] KV znamená komprimovaná velikost souboru, KP kompresní poměr

[6] Na tomto místě je třeba si uvědomit, že postupným vylepšováním dosavadních kompresních metod nelze očekávat průlomový zlom o „desítky“ procent a zlepšení se již budou pohybovat v jednotkách procent.

vidět výsledky komprese téhož souboru s náhodně permutovanými dokumenty. Zde dokonce došlo ke zhoršení komprese i proti původním výsledkům bez použití shlukování. Z toho je patrné, že shlukování má u takto velkého souboru dokumentů pozitivní vliv.

Tabulka 2. Soubor bible.txt: vliv shlukování na výsledky komprese

	Shlukování	KV [byty]	Δ_{KV} [byty]	Δ_{KV} [%]	KP [%]	Δ_{KP} [%]
(a)	MSL	888592	2684	0,303	21,955	0,067
(b)	MSR	888596	2688	0,303	21,955	0,067
(c)	(a), bez stoplistu	887980	2072	0,234	21,94	0,052
(d)	(b), bez stoplistu	889024	3116	0,352	21,965	0,077
(e)	(c), bez Zipfova zákona	887936	2028	0,229	21,938	0,05
(f)	(d), bez Zipfova zákona	888932	3024	0,341	21,963	0,075

Tabulka 3. Soubor bible.txt: vliv náhodné permutace na kompresi

Permutace	KV [byty]	Δ_{KV} [byty]	Δ_{KV} [%]	KP [%]	Δ_{KP} [%]
1	889596	3688	0,416	21,979	0,091
2	889280	3372	0,381	21,972	0,084
3	889792	3884	0,438	21,984	0,096
4	889596	3688	0,416	21,979	0,091
5	890036	4128	0,466	21,990	0,102
6	889696	3788	0,428	21,982	0,094
7	890160	4252	0,480	21,993	0,105
8	889984	4076	0,460	21,989	0,101
9	889880	3972	0,448	21,987	0,099
10	890144	4236	0,478	21,993	0,105
Průměr	889816	3908	0,441	21,985	0,097

8 Závěr

V příspěvku jsme prezentovali možnost využití informací získaných shlukováním pro zlepšení komprese plných textů dokumentů v DIS. Shlukování proběhlo pomocí aglomerativní shlukovací algoritmus a následovala komprese pomocí slovního kompresního algoritmu. Dosažené výsledky ukazují na zlepšený kompresní poměr získaný touto cestou.

V další práci se chceme zaměřit na experimenty s rozsáhlými českými texty, které mají charakteristiku částečně odlišnou od textů anglických. A dále na

Tabulka 4. Soubor latimes.txt: vliv shlukování na výsledky komprese

Shlukování	KV [byty]	Δ_{KV} [byty]	Δ_{KV} [%]	KP [%]	Δ_{KP} [%]
MSL	151869588	-6148352	-3,891	30,474	-1,234
MSR	151973800	-6044140	-3,825	30,495	-1,213

Tabulka 5. Soubor latimes.txt: vliv náhodné permutace na kompresi

Permutace	KV [byty]	Δ_{KV} [byty]	Δ_{KV} [%]	KP [%]	Δ_{KP} [%]
1	160417812	2399872	1,519	32,189	0,481
2	160456620	2438680	1,543	32,197	0,489
3	160448056	2430116	1,538	32,195	0,487
4	160456564	2438624	1,543	32,197	0,489
5	160475324	2457384	1,555	32,201	0,493
Průměr	160450875	2432935	1,54	32,196	0,488

experimenty se metodou WBW. Možnost dalšího výzkumu spatřujeme neustále ve zvyšování výkonnosti shlukovacího algoritmu.

Reference

1. R. Arnold and T. Bell. A corpus for the evaluation of lossless compression algorithms. In J. A. Storer and M. Cohn, editors, *Proc. 1997 IEEE Data Compression Conference*, pages 201–210. IEEE Computer Society Press, Los Alamitos, California, Mar. 1997.
2. M. W. Berry. *Survey of Text Mining.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
3. M. W. Berry and M. Browne. *Understanding Search Engines.* SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1999.
4. M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, Digital Systems Research Center Research Report 124, 1994.
5. J. Dvorský, J. Pokorný, and V. Snášel. Word-based compression methods and indexing for text retrieval systems. In J. Eder, I. Rozman, and T. Welzer, editors, *Proceedings of ADBIS 99*, number 1691 in Lecture Notes in Computer Science, pages 75–84. Springer-Verlag, Berlin, 1999.
6. J. Dvorský, V. Snášel, and J. Pokorný. Word-based compression methods for large text documents. In *Data Compression Conference - DCC '99*, page 523, Snowbird, Utah, USA, 1999.
7. J. Dvorský. *Word-based Compression Methods for Information Retrieval Systems.* Phd thesis, Charles University Prague, 2004.
8. J. Dvorský, J. Martinovič, and V. Snášel. Query expansion and evolution of topic in information retrieval systems. In V. Snášel, J. Pokorný, and K. Richta, editors, *DATESO*, volume 98 of *CEUR Workshop Proceedings*, pages 117–127. CEUR-WS.org, 2004.
9. D. Harman, editor. *The Forth REtrieval Conference (TREC-4).* National Inst. of Standards and Technology, Gaithersburg, USA, 1997.

10. R. L. Haskin. Special-purpose processors for text retrieval. *Database Engineering*, 4(1):16–29, 1981.
11. R. N. Horspool and G. V. Cormack. Constructing word-based text compression algorithms. In J. A. Storer and M. Cohn, editors, *Proc. 1992 IEEE Data Compression Conference*, pages 62–71. IEEE Computer Society Press, Los Alamitos, California, Mar. 1992.
12. A. Lukasová and J. Šarmanová. *Metody shlukové analýzy.* SNTL, Praha, 1985.
13. J. Lánský and M. Žemlička. Text compression: Syllables. In Richta et al. [15], pages 32–45.
14. J. Martinovič and P. Gajdoš. Vector model improvement by fca and topic evolution. In Richta et al. [15], pages 46–57.
15. K. Richta, V. Snásel, and J. Pokorný, editors. *Proceedings of the Dateso 2005 Annual International Workshop on DAtabases, TExts, Specifications and Objects, Desna, Czech Republic, April 13-15, 2005*, volume 129 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.
16. G. Salton. *Automatic Text Processing.* Addison-Wesley Publishing Company, 1989.
17. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
18. C. van Rijsbergen. *Information Retrieval (second ed.).* London, Butterworths, 1979.
19. T. A. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, 1984.
20. I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Van Nostrand Reinhold, 1994.
21. G. K. Zipf. *Human Behaviour and the Principle of Least Effort: an Introduction to Human Ecology.* Addison-Wesley, 1949.

Annotation

Compression with clustering analysis support

Word-based compression method is very suitable compression of documents in information retrieval systems. To enhance performance of searching cluster analysis of indexed documents is usually done. But information comes from clustering can be very helpful in compression. Word-based compression using information about cluster hierarchy is presented in this paper. Some experimental result are given at the end of the paper.

Algoritmy na vyhľadávanie najlepších k objektov bez priameho prístupu[1]

Peter Gurský

Ústav informatiky, Univerzita Pavla Jozefa Šafárika,
Jesenná 5, 041 54 Košice

gursky@upjs.sk

Abstrakt. V tomto článku popisujeme vlastnosti súčasných algoritmov na vyhľadávanie najlepších k objektov z distribuovaných zdrojov bez priameho prístupu. Zároveň ponúkame nový algoritmus a heuristiky na riešenie tohto problému. Výkonnosť algoritmov porovnávame na testovacích dátach.

Kľúčové slová: integrácia distribuovaných zdrojov, agregácia, najlepších k objektov, sekvenčný prístup, NRA

1 Úvod

V súčasnosti sa často stretneme s potrebou spojiť veľké množstvo informácií o jednotlivých objektoch a na základe nich vybrať také objekty, ktoré sú najlepšie vzhľadom na užívateľské požiadavky.

Objektmi môžu byť typicky pracovné miesto, hľadaný dokument, web-stránka, obrázok, kniha, prezentácia, konferencia, hotel, dovolenková destinácia a iné.

Takéto objekty majú typicky niekoľko vlastností (atribútov), na základe ktorých sa rozhodujeme, ktoré objekty sú pre nás tie najlepšie. Rozhodovanie na základe viacerých vlastností objektov nazývame multikriteriálne rozhodovanie.

Vlastnosti takýchto objektov sú typicky troch typov. Prvým typom je rozhodnutie o tom, či objekt danú vlastnosť má, alebo nemá. Príkladom môže byť požiadavka ovládania MS Windows, výskyt dokumentu na slovenskej web-stránke, kniha napísaná v slovenčine, hotel so sociálnym zariadením na izbe, pri dovolenke poskytnutý sprievodca a podobne. Druhým typom sú informácie, ktoré sú nejakým spôsobom stupňované. Príkladom môže byť plat, vzdialenosť, relevantnosť slova v dokumente danej kolekcie, cena, rok publikovania, dĺžka pobytu atď. Posledný typ atribútu, reálne číslo, je typický pre multimediálne databázy alebo záznamy fyzikálnych experimentov. Posledné dva typy sú typické fuzzy atribúty.

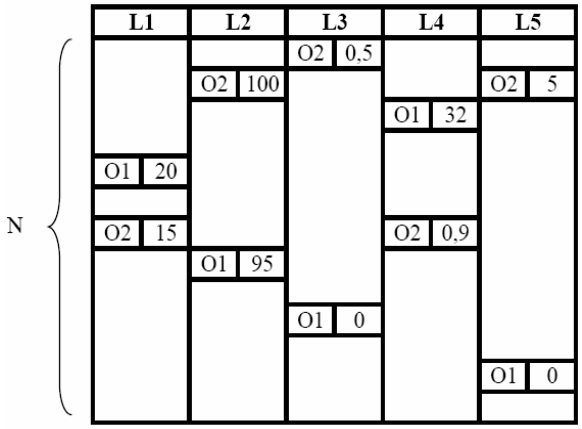
Informácie o jednotlivých atribútoch môžu byť distribuované. Možnosť distribuovať jednotlivé vlastnosti má niekoľko výhod. Napríklad na zistenie dopravných spojení je výhodnejšie využívať servery dopravných spoločností. Na

[1] Táto práca je čiastočne podporovaná grantom VEGA 1/0385/03 a projektom NAZOU (Nástroje pre získavanie, organizovanie a udržovanie znalostí v prostredí heterogénnych informačných zdrojov) prierezového štátneho programu „Budovanie informačnej spoločnosti“.’

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 95–105, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

zistenie toho, aké má daný hotel voľné kapacity sa môže využiť nejaký server, ktorý dané informácie zbiera a aktualizuje. Na druhej strane môžeme hovoriť o distribuovaných zdrojoch aj v prípade, ak jednotlivé atribúty máme uložené síce na jednom počítači, ale v rôznych formách ako sú súbory, relačná databáza, RDF databáza a podobne.

Podľa [2,3] predpokladajme, že máme konečnú množinu objektov rovnakého druhu. Mohutnosť tejto množiny je N. Každý objekt x má m atribútov (vlastností) $x_1,...,x_m$. Hodnota i-teho atribútu je uchovávaná v usporiadanom zozname L_i vo forme objekt-hodnota (binárny objekt-atribút model). Zoznamy sú zotriedené podľa hodnoty vlastnosti. Zoznamy môžu byť distribuované. V nasledujúcej tabuľke môžme vidieť príklad výskytu hodnôt jednotlivých vlastností v zoznamoch pre dva objekty *O1* a *O2*.



Obr. 1. Príklad výskytu hodnôt piatich atribútov pre dva objekty.

Nech $s_i(x)$ je hodnota i-tej vlastnosti objektu x. Napríklad $s_4(O2)$ z tejto tabuľky je rovné 0,9. Majme m-árnu neklesajúcu agregačnú funkciu F. Potom celkové ohodnotenie objektu x agregačnou funkciou F bude $S(x)=F(s_1(x),...,s_m(x))$. Užívateľ pomocou agregačnej funkcie určí relevantnosť (dôležitosť) jednotlivých vlastností objektov v celkovom hodnotení. Našou úlohou je nájsť spomedzi týchto objektov k takých, že sú pre daného užívateľa najlepšie vzhľadom na túto agregačnú funkciu.

K zoznamom sa obvykle dá pristupovať dvoma možnými spôsobmi [2]. Prvým spôsobom je sekvenčný prístup, ktorý zistí, ktorý objekt má danú vlastnosť h-tú najlepšiu spomedzi všetkých objektov aj s jej hodnotou pre dané h z intervalu [1,N]. V praxi sa sekvenčný prístup používa tak, že vracia dvojice „objekt - hodnota atribútu“ postupne od najväčších po najmenšie. Druhý spôsob je priamy prístup, ktorý vráti hodnotu vlastnosti pre daný objekt, bez ohľadu na poradie v danom zozname.

V tomto článku sa zameriame na prípady, keď priamy prístup k zoznamom nie je možný. V článku [3] sa uvádza ako príklad fulltextový vyhľadávač, pri ktorom nevieme pre danú otázku a dokument dostať interné ohodnotenie tohto dokumentu. Ďalším príkladom môže byť použitie zoznamov ako „vstupných prúdov“, napríklad v prípade výstupov iných vyhľadávaní.

Prípadom, kedy uvažujeme iba sekvenčný prístup, sa zaoberali Fagin, Lotem a Naor v [3]. Uvádzajú algoritmus NRA (= „no random access", v preklade „bez priameho prístupu"). Tomuto algoritmu sa budeme podrobnejšie venovať v kapitole 2.

Myšlienku NRA využíva Ihab L. Ilyas [4, 5] pri tvorbe algoritmu KRJN (= key rank join), ktorý integruje priamo do databázového stroja. Tento algoritmus je schopný pracovať vždy iba s dvoma zoznamami a v prípade viacerých zoznamov, je potrebné výstupy kombinovať viacnásobným použitím KRJN algoritmu. Tento prístup síce umožňuje paralelné spracovanie, ale vyžaduje aj schopnosť „rozdeliť" agregačnú funkciu na podfunkcie pre dvojice vstupných atribútov. Podľa testov v [4] takéto zreťazenia vždy vyžadujú väčší počet prístupov ako algoritmus NRA.

Posledný známy algoritmus využívajúci iba sekvenčné prístupy je Stream-combine [1]. Tomuto algoritmu sa je venovaná kapitola 3.

Okrem spomínaných troch algoritmov bol v [3] prezentovaný aj algoritmus „combined algorithm", ktorý priamy prístup síce používa, ale výrazne ho obmedzuje. Toto obmedzenie vyplýva z toho, že priamy prístup je často omnoho časovo náročnejší ako sekvenčný. V tomto článku sa však budeme venovať iba takým algoritmom, ktoré priamy prístup nepoužívajú vôbec.

Tento článok je ďalej organizovaný nasledovne. V kapitolách 2 a 3 uvádzame doteraz známe algoritmy NRA a „Stream-combine". V kapitole 4 popíšeme náš nový algoritmus 3P-NRA a dokážeme niekoľko jeho vlastností. V 5. kapitole uvádzame popis použitých heuristík. Kapitola 6 zahŕňa výsledky testov na umelých aj reálnych dátach. V kapitole 7 sumarizujeme dosiahnuté výsledky a diskutujeme o možnej ďalšej práci.

2 Algoritmus NRA

Algoritmus NRA (= no random access) [3] je najlepším doteraz publikovaným algoritmom, ktorý sa venuje multikriteriálnemu vyhľadávaniu najlepších k objektov z viacerých distribuovaných zdrojov bez priameho prístupu.

Nech $z=(z_1,...,z_m)$ je vektor pozícií v jednotlivých zoznamoch (nazývaný aj vektor zlezenia), do ktorých sa algoritmus dostal pomocou sekvenčného prístupu. Ďalej nech $r_i(j)$ je j-tý najlepší objekt v zozname L_i, alebo inak povedané objekt na j-tej pozícii v i-tom zozname. Pre každý usporiadaný zoznam L_i nech $u_i = s_i(r_i(z_i))$. Hodnotu $\tau = F(u_1, \ldots, u_m)$ budeme nazývať threshold.

Pre daný objekt x a množinu známych hodnôt jeho atribútov $V(x)=\{i_1, \ldots, i_n\}\subseteq \{1,\ldots,m\}$ s hodnotami $x_{i1},...,x_{in}$ označme $W_V(x)$ (skrátene $W(x)$ ak V je známa z kontextu) ako minimálnu (*worst*) hodnotu agregačnej funkcie F pre objekt x. Keďže predpokladáme, že F je neklesajúca kladná funkcia, môžme túto hodnotu vypočítať tak, že dosadíme za všetky neznáme atribúty $i \in \{1,\ldots,m\}\setminus V$ nulu. Napríklad ak $V(x)=\{1,\ldots,g\}$ potom $W_V(x)=F(x_1,...,x_g,0,...,0)$.

Analogicky označíme $B_V(x)$ (skrátene $B(x)$ ak V je známa z kontextu) ako maximálnu (*best*) hodnotu agregačnej funkcie F pre objekt x. Keďže vieme, že hodnoty sú v zoznamoch usporiadané zostupne, môžeme dosadiť namiesto chýbajúcich hodnôt atribútov hodnoty pozdĺž aktuálneho vektora zlezenia z. Napríklad ak $V(x)=\{1,\ldots,g\}$ potom $W_V(x)=F(x_1,...,x_g,u_{g+1},...,u_m)$.

Reálna hodnota objektu x je teda $W(x) \leq S(x) \leq B(x)$. Poznamenajme, že každý objekt, u ktorého nepoznáme žiadne atribúty má hodnotu $B(x) = \tau = F(u_1, \ldots, u_m)$ a $W(x) = F(0,...,0)$. Na druhej strane, ak poznáme hodnoty všetkých atribútov, tak $W(x) = B(x) = S(x) = \mathrm{F}(x_1,...,x_m)$.

Algoritmus NRA vyzerá nasledovne:

1. Pomocou sekvenčného prístupu do všetkých zoznamov L_i, získaj nové hodnoty $u_1, \ldots, u_m$
   - Pre každý objekt x s videnými atribútmi $V(x) \subseteq \{1,\ldots,\mathrm{m}\}$ vypočítaj $W(x)$ a $B(x)$
   - Nech T_k, aktuálny zoznam najlepších k objektov usporiadaný zostupne podľa *worst* hodnôt, obsahuje k objektov s najväčšími aktuálnymi *worst* hodnotami. Ak majú dva objekty rovnakú *worst* hodnotu, sú zotriedené podľa *best* hodnoty. Nech M_k je k-ta najväčšia *worst* hodnota tohto zoznamu.
2. Objekt x nazveme *relevantný* ak $B(x) > M_k$. Skonči a vráť zoznam T_k, ak sme videli aspoň k rôznych objektov a už neexistujú žiadne relevantné objekty mimo T_k, teda $B(x) \leq \mathrm{M}_k$ pre každé $\mathrm{x} \notin \mathrm{T}_k$. Inak choď na bod 1.

Algoritmus 1. NRA

Tento algoritmus korektne[3] nájde najlepších k objektov vzhľadom ku konkrétnej agregačnej funkcii. Poznamenajme, že neposkytuje presnú hodnotu alebo poradie každého objektu.

Nanešťasie počas behu algoritmu potrebujeme v každom kroku prerátavať *best* hodnoty všetkých relevantných objektov a porovnávať ich s hodnotou M_k, čo v nejakej hĺbke h môže znamenať až $m.h$ prepočtov pre každé $h \leq N$.

3 Algoritmus Stream-combine

Algoritmus „Stream-combine“ [1] je historicky starší, ako NRA. Pracuje podobne ako NRA, no autori v ňom uvažujú iba hornú (*best*) hranicu hodnoty objektu. Tým pádom je, počas behu programu, aktuálny zoznam najlepších k objektov usporiadaný podľa *best* hodnôt objektov. Keďže sa *best* hodnoty všetkých videných objektov menia skoro s každým prístupom, je potrebné zakaždým celý tento zoznam preusporiadávať. Takéto prepočítavanie hodnôt a preusporiadávanie objektov je veľmi časovo náročné.

Ďalšou nevýhodou toho, že sa využívajú iba *best* hodnoty, je, že sa zoznam videných objektov neskracuje. Inými slovami všetky objekty sú relevantné až do ukončenia algoritmu. Zoznam objektov, uchovávaný v pamäti, je teda veľmi veľký, čo znovu zhoršuje časovú zložitosť algoritmu.

Autori v tomto článku síce popisujú heuristiky, ktoré by podľa nich mali tento prístup zrýchliť, neuvádzajú však žiadne testy. Tento algoritmus sme implementovali. Podľa očakávania bol, žiaľ, nie len rádovo 100 krát pomalší ako NRA, ale vykonával aj niekoľko násobne viac prístupov. Vzhľadom na tieto skúsenosti sme náš nový algoritmus 3P-NRA porovnávali už iba s algoritmom NRA.

4 Algoritmus 3P-NRA (3-phased no random access)

Tento algoritmus je „kombináciou“ algoritmu NRA a algoritmov popísaných v [6]. Algoritmus 3P-NRA sa snaží minimalizovať počet prepočtov *best* hodnôt a zároveň poskytuje priestor pre implementáciu viacerých heuristík na sekvenčné prístupy do zoznamov.

Nech H je heuristika, ktorá vyberie, do ktorých zoznamov sa má algoritmus pozrieť pomocou sekvenčného prístupu v ďalšom kroku. Formálnejšie nech H je taká, že pre každé $j \leq m$ platí $H(z)_j = z_j$ alebo $H(z)_j = z_j + 1$ a exituje aspoň jedno také $i \leq m$ že $H(z)_i = z_i + 1$. Poznamenajme, že H sa môže v priebehu výpočtu meniť. Množinu $\{i \leq m: H(z)_i = z_i + 1\}$ nazveme kandidátskou množinou pre nasledujúci sekvenčný prístup. Jej prvky nazveme kandidátmi. Niekoľko heuristík uvádzame v ďalšej kapitole.

I. Zliezanie s thresholdom a heuristikou *H1*.
   1. $z := (0,...,0)$
   2. Nastav heuristiku *H1* a paraleným sekvenčným prístupom choď do všetkých kandidátskych zoznamov. Nastav $z_i := H1(z)_i$ pre každé i.
   3. Pre každý objekt, ktorý sme dostali niektorým sekvenčným prístupom v kroku 2 rob: Označme objekt ako x. Prepočítaj $W(x)$ a $B(x)$ a zaraď objekt x na správne miesto v zozname najlepších objektov T zoradenom podľa worst hodnoty. Ak je $B(x)$ hodnota menšia ako *worst* hodnota k-teho objektu v T, vyhoď objekt x zo zoznamu.
   4. Ak už máme nájdených aspoň k objektov, ktoré majú väčšiu *worst* hodnotu ako τ tak choď na fázu II, inak skoč na krok 2 fázy I.

II. Odstraňovanie nerelevantných objektov
   Prechádzaj zoznam T od $(k+1)$-ho objektu po posledný a prepočítavaj *best* hodnoty týchto objektov. Ak je táto hodnota menšia ako *worst* hodnota k-teho objektu v T, vyhoď objekt x zo zoznamu. Ak ostalo v T viac ako k objektov choď na fázu III, inak vráť zoznam T a skonči.

III. Zliezanie s heuristikou *H2*.
   1. Nastav heuristiku *H2* a paraleným sekvenčným prístupom choď do všetkých kandidátskych zoznamov. Nastav $z_i := H2(z)_i$ pre každé i.
   2. Pre každý objekt, ktorý sme dostali niektorým sekvenčným prístupom v kroku 1 rob: Označme objekt ako x. Ak $x \in T$, prepočítaj $W(x)$ a $B(x)$ a zaraď objekt x na správne miesto v T. Ak je $B(x)$ hodnota menšia alebo rovná ako *worst* hodnota k-teho objektu v T, vyhoď objekt x zo zoznamu.
   3. Ak $|T| = k$, vráť zoznam T a skonči.
   4. Ak sa presunom objektu x zmenila *worst* hodnota na k-tom mieste v T (presunul sa tam iný objekt s väčšou *worst* hodnotou, alebo x sa stal k-tym objektom s novou *worst* hodnotou), alebo ak klesla hodnota τ a už prešlo b cyklov fázy III bez vykonania fázy II pre nejaké kladné b, skoč na fázu II, inak skoč na krok 1 fázy III.

Algoritmus 2. 3P-NRA

Ukážme si, že náš nový algoritmus 3P-NRA je korektný.

Veta 2.1: Ak F je neklesajúca kladná agregačná funkcia, tak algoritmus 3P-NRA korektne nájde najlepších k objektov.

Dôkaz: Predpokladajme, že 3P-NRA zastane na pozícii $z=(z_1,...,z_m)$ a vráti objekty $X_1,...,X_k$. Nech $M_k = \min\{W(X_1),...,W(X_k)\}$ a $X \notin \{X_1,...,X_k\}$. Ukážeme, že $B(X) \leq M_k$.

Keďže *worst* hodnota *k*-teho objektu v *T* nemôže klesať, a my sme v celom algoritme vyhadzovali z *T* len také objekty, ktoré majú hodnotu *best* menšiu alebo rovnú ako aktuálna *worst* hodnota *k*-teho objektu, tak pre každý takýto objekt *X* platí, že jeho *best* hodnota je menšia ako posledná *worst* hodnota *k*-teho objektu. Teda pre všetky vyhodené objekty platí, že $S(X) \leq B(X) \leq M_k$.

Ešte treba uvažovať tie objekty, ktoré ignorujeme v druhom kroku tretej fázy. Ide o také objekty *x*, ktoré nie sú v *T*. Sú to buď tie objekty, ktoré sme vyhodili kvôli nízkej *best* hodnote (to sme už vyriešili), alebo objekty, na ktoré sme nenarazili ani v jednom zozname v prvej fáze. Nech je to objekt *X* s hodnotami atribútov $x_1,...,x_m$. Nech $u_1,\ldots,u_m$ sú hodnoty prislúchajúce vektoru zlezenia na konci fázy I. Platí, že $x_i \leq u_i$ pre každé $i \in \{1,\ldots,m\}$. Z monotónnosti agregačnej funkcie vyplýva $S(X) = F(x_1,...,x_m) \leq B(X) = F(u_1,...,u_m) = \tau$. Predposledná rovnosť vyplýva z toho, že ak nepoznáme hodnoty atribútov pre *best*, nahradzujeme ich hodnotami pozdĺž vektora zlezenia. Z podmienky ukončenia fázy I však vieme, že $\tau \leq W(X_j)$ pre každé $j \in \{1,\ldots,k\}$ a teda $B(X) \leq M_k$. □

Ako je vidieť, algoritmus 3P-NRA potrebuje oveľa menej prepočtov hodnôt *best* ako NRA. Kým v NRA sa vypočítava *best* hodnota po každom sekvenčnom prístupe pre všetky objekty v *T*, 3P-NRA vypočítava v prvej a tretej fáze iba jednu best hodnotu pre každý sekvenčný prístup. Jedinou výnimkou je druhá fáza, ktorá sa vykonáva až po skončení prvej fázy a v tretej fáze v prípade zmeny *worst* hodnoty *k*-teho prvku. Keby sme fázu II vynechali, zbytočne by sme uchovávali nerelevantné objekty v zozname až pokiaľ by sme na nich nenarazili pri niektorom sekvenčnom prístupe. Zároveň je pravdepodobne zbytočné túto fázu robiť príliš často. Preto v 4. kroku fázy III uvádzame ako druhú podmienku prechodu do fázy II pokles thresholdu a zároveň opakovanie *b* cyklov tretej fázy po sebe. Experimentovanie s číslom *b* je súčasťou budúceho skúmania. V prípade, že b = 1, môžeme dokázať nasledujúcu vetu.

Veta 2.2: Majme neklesajúcu kladnú agregačnú funkciu *F*. Nech *H1* a *H2* sú také heuristiky, ktoré vyberú v každom kroku za kandidátov všetky zoznamy, a nech $b = 1$. Potom algoritmus 3P-NRA vykoná rovnaký počet sekvenčných prístupov ako algoritmus NRA.

Dôkaz: V prípade, že $b = 1$, je vidieť, že fáza II a fáza III dohromady pracujú úplne rovnako ako algoritmus NRA. Stačí nám teda ukázať, že algoritmus NRA nemôže skončiť skôr, ako sa 3P-NRA dostane z fázy I. Algoritmus NRA končí, iba ak už neexistujú žiadne relevantné objekty mimo T_k, teda $B(x) \leq M_k$ pre každé $x \notin T_k$. Fáza I algoritmu 3P-NRA končí ak $\tau \leq M_k$. Nakoľko však τ je *best* hodnotou všetkých doposiaľ nevidených objektov, tak musíme počkať, kým threshold klesne natoľko, že

$\tau \leq M_k$ aby sme vedeli s istotou povedať, že už neexistuje objekt, ktorý má *best* hodnotu väčšiu ako M_k. □

Na základe vlastnosti „instance optimal“ pre algoritmus NRA [3] a vety 2.2 môžeme vysloviť nasledujúci dôsledok.

Dôsledok 2.3: Majme neklesajúcu kladnú agregačnú funkciu F. Nech *H1* a *H2* sú také heuristiky, ktoré vyberú v každom kroku za kandidátov všetky zoznamy, a nech $b = 1$. Potom algoritmus 3P-NRA je „instance optimal“.

Vlastnosť „instance optimal“ zaručuje efektívnosť algoritmov bez priameho prístupu tak, že počet prístupov oproti optimálnemu počtu vykonaných prístupov je maximálne počet zoznamov vynásobený optimálnym počtom prístupov. V tomto prípade ak s je počet sekvenčných prístupov algoritmu, s_o je optimálny počet prístupov a m je počet zoznamov tak platí: $s \leq (s_o.m)$.

5 Heuristiky

Podobne ako v [6] ukážeme, že nie je možné uspokojiť sa s tým, že algoritmus je „instance optimal“. Platí to najmä v prípade, že m je veľké, ale už pre m=2 sa môže počet prístupov až zdvojnásobiť. Ukazuje sa, že vhodné heuristiky na výber zoznamov na sekvenčný prístup výrazne zvyšujú výkonnosť tohto druhu algoritmov.

V našich testoch sme použili ako heuristiky H1 všetky heuristiky použité v článku [6]. Na podrobnejšie vysvetlenie fungovania týchto heuristík preto odporúčame práve tento článok. Uveďme si aspoň hlavné myšlienky týchto heuristík.

Najjednoduchšou heuristikou je heuristika, ktorú využíva aj algoritmus NRA. Historicky bola prvý krát použitá v algoritme „Threshold algorithm“ [2]. Budeme ju označovať ***TA***. Táto heuristika vyberá vo všetkých prípadoch za kandidátov všetky zoznamy. V ľubovoľnom kroku algoritmu sú teda všetky zoznamy zlezené rovnako hlboko.

Heuristika, ktorú budeme označovať ***δF/δx(Δx)***, použitá ako súčasť algoritmu „Quick combine“[7], vyberá za kandidátov tie stĺpce, ktorým najrýchlejšie klesá hodnota objektov za posledných p prístupov vynásobená váhou stĺpca v agregačnej funkcii. Váha stĺpca v agregačnej funkcii sa počíta ako ľavá derivácia agregačnej funkcie.

Heuristika označená ***δF/δx(x)*** [6] vyberá za kandidátov zoznamy, ktorých je najväčšia posledná videná hodnota prenásobená váhou stĺpca v agregačnej funkcii. Ide teda o hodnotu, ktorá má najväčší prírastok k celkovej hodnote agregačnej funkcie.

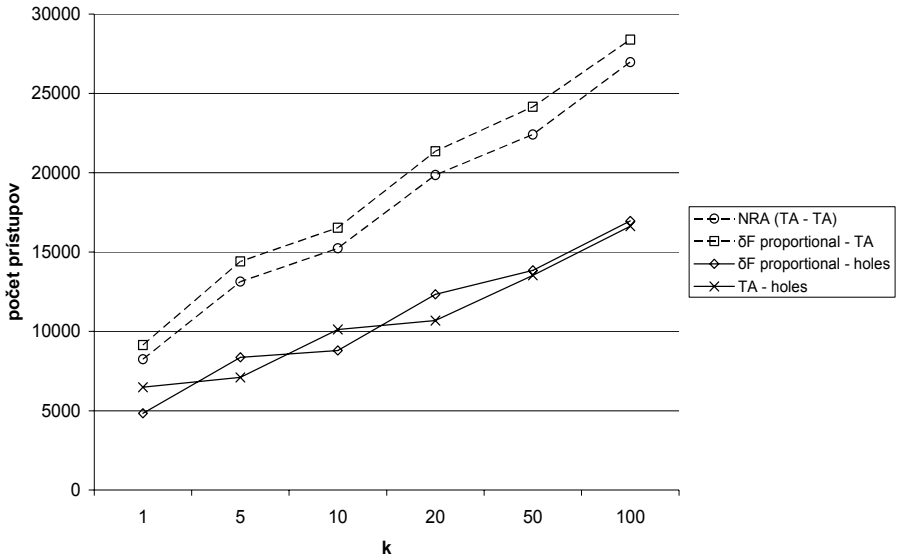
Heuristika ***x/Δx switch*** [6] robí to, že v raz sa rozhoduje ako heuristika *δF/δx(Δx)* a raz ako heuristika *δF/δx(x)*.

Heuristika ***δF proportional*** postupuje v každom zozname rôznou rýchlosťou. Rýchlosť je určená hodnotou váhy stĺpca v agregačnej funkcii.

Zameriame sa na výber heuristiky H2. Vyššie spomínané heuristiky sa ako heuristika H2 ukázali neúčinné. Vyvinuli sme teda heuristiku ***holes*** (= diery), ktorej použitie porovnávame s heuristikou použitou v NRA, teda heuristikou *TA*.

Tabuľka 1. Priemerný počet sekvenčných prístupov (menej je lepšie) – umelé dáta

		k					
H1	H2	1	5	10	20	50	100
TA (NRA)	TA	8247	13126	15232	19851	22402	26968
δF proportional	TA	9136	14412	16532	21343	24147	28384
δF/δx(x)	TA	11541	17377	19199	22789	24837	27980
δF/δx(Δx)	TA	9459	14833	17175	22062	24698	28201
x/Δx switch	TA	9709	15881	18141	22760	25010	28134
TA	holes	6482	**7098**	10116	**10682**	**13528**	**16630**
δF proportional	holes	**4832**	8367	**8796**	12329	13847	16963
δF/δx(x)	holes	8320	12166	12213	14069	16140	18755
δF/δx(Δx)	holes	5810	7908	9829	12210	14286	17405
x/Δx switch	holes	5856	9749	11063	12216	14691	18054

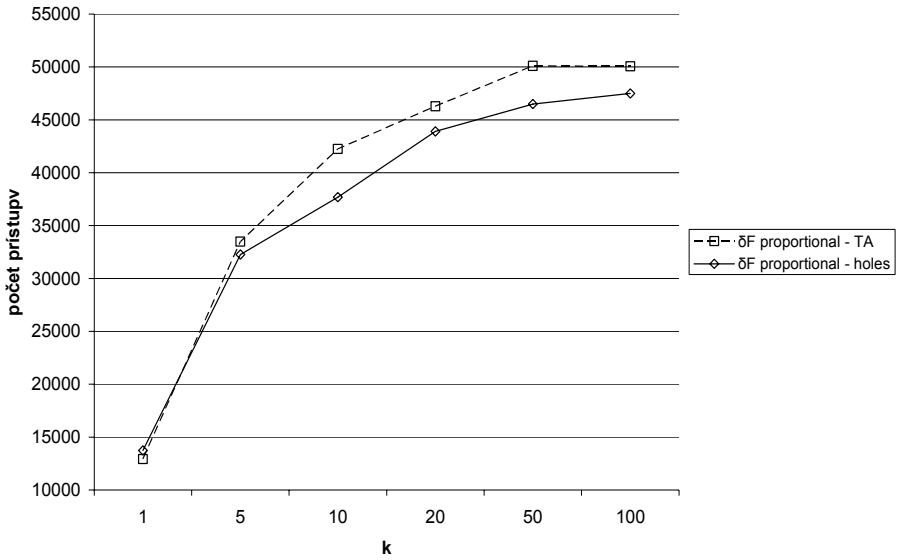


Obr. 2. Priemerný počet sekvenčných prístupov (menej je lepšie) – umelé dáta. Dva najlepšie pre oba typy heuristiky H2.

Heuristika *holes* vyberá za kandidátov tie zoznamy pre ktorých atribúty máme v zozname T najviac nevidených hodnôt. Inými slovami, táto heuristika sa pozrie na zoznam relevantných objektov ako na tabuľku a snaží sa vypĺňať ten stĺpec, v ktorom je najviac neznámych hodnôt.

Tabuľka 2. Priemerný počet sekvenčných prístupov (menej je lepšie) – reálne dáta

		K					
H1	H2	1	5	10	20	50	100
TA (NRA)	TA	13548	33810	42259	46330	50095	50048
δF proportional	TA	**12924**	33475	42249	46284	50095	50048
δF/δx(x)	TA	17202	34194	44430	47532	50416	50500
δF/δx(Δx)	TA	13658	33802	42224	46317	50095	50048
x/Δx switch	TA	14216	32718	42606	46487	50500	50500
TA	holes	14293	32885	**37634**	**43900**	**46485**	**47492**
δF proportional	holes	13737	**32261**	**37684**	**43900**	**46485**	**47492**
δF/δx(x)	holes	17053	33891	41832	45574	48207	48849
δF/δx(Δx)	holes	13866	32794	37634	43900	46485	**47492**
x/Δx switch	holes	13700	32657	37890	44619	47060	47944



Obr. 3. Priemerný počet sekvenčných prístupov (menej je lepšie) – reálne dáta. Najlepšie pre oba typy heuristiky H2.

6 Testy

Na testovanie pre spojité dáta sme použili umelo vytvorené zoznamy s rôznou distribúciou hodnôt. Vytvorili sme 2 exponenciálne a 2 logaritmické distribúcie s 10000 objektami a 6 typov agregačných funkcií. Rôznou kombináciou vstupných zoznamov a agregačných funkcií sme získali 16 rôznych vstupov pre algoritmus. Pri

testovaní sme skúšali všetky spomínané heuristiky použité ako heuristika H1 v kombinácii s heuristikami *TA* a *holes*, ktoré boli použité ako heuristika H2. Vo výsledkoch uvádzame priemerné hodnoty z týchto testov. V tabuľke 1 si môžeme všimnúť, že pri použití heuristiky *holes* sa veľmi výrazne znížil počet prístupov. Na druhej strane môžeme konštatovať, že výber heuristiky H1 neovplyvňuje výsledky až tak výrazne. Na obrázku 2 sú tieto výsledky zobrazené graficky pre dve najlepšie kombinácie pre obe heuristiky použité ako H2.

Druhou testovacou kolekciou boli dáta z oblasti information retrieval. Použitím rôznych kombinácií lokálnych a globálnych váh [8] pri ohodnocovaní výskytov termov v dokumentoch bolo získaných 6 rôznych množín dát. Každá obsahuje viac ako 25 000 objektov (dokumentov) s 50 rôznymi atribútmi (termami). Vo výsledkoch, ktoré môžeme vidieť v tabuľke 2 a na obrázku 3, porovnávame priemerné hodnoty počtov prístupov, pričom sme použili náhodne generované váhy v agregačnej funkcii. Distribúcie týchto dát sú exponenciálne, obsahujú teda málo hodnôt s veľkými hodnotami a veľa hodnôt s malými hodnotami. Vo všetkých zoznamoch je časté opakovanie hodnôt, čiže majú charakter diskrétnych dát. Práve to môže byť dôvodom menej výraznej prevahy heuristiky holes, než tomu bolo pri spojitých dátach.

V ďalšom výskume sa budeme snažiť vytvoriť heuristiky, ktoré by boli vhodnejšie pre diskrétne dáta.

7 Záver

V tomto článku sme sa venovali multikriteriálnemu vyhľadávaniu najlepších *k* objektov iba za pomoci sekvenčných prístupov k distribuovaným zoznamom. Uviedli sme známe algoritmy NRA a „Stream combine". Navrhli sme nový algoritmus 3P-NRA, o ktorom sme dokázali, že je korektný a že, pri použití rovnakej heuristiky ako v NRA, potrebuje rovnaký počet sekvenčných prístupov čím spĺňa vlastnosť „instance optimal". 3P-NRA zároveň výrazne obmedzuje drahé prepočítavanie hodnôt všetkých objektov počas svojho behu. Na záver sme navrhli novú heuristiku *holes*, ktorá v našich testoch dokázala výrazne znížiť počet sekvenčných prístupov.

V budúcnosti je potrebné venovať pozornosť heuristikám, ktoré by boli vhodnejšie v prípade diskrétnych dát. Ďalej by sme sa mali venovať prípadom, keď jednotlivé zoznamy nie sú dostupné v rovnakom čase. Je potrebné analyzovať nakoľko je potrebné zohľadňovať také zdroje, ktoré poskytujú informácie v skupinách („batches"). Zaujímavé môže byť študovanie prípadu, že vieme histogramy distribúcie dát v zoznamoch alebo využitie predchádzajúcich hľadaní s inými agregačnými funkciami.

Poďakovanie

Chcel by som sa poďakovať Martinovi Šumákovi za pomoc pri implementácii algoritmov, Rasťovi Lencsesovi za poskytnutie testovacích dát a anonymným recenzentom za veľmi plodné pripomienky.

Literatúra

1. Güntzer, U., Balke, W., Kiessling, W.: Towards efficient multi-feature queries in heterogeneous enviroments. Proceedings of International Symposium on Information Technology (ITCC, Las Vegas, Nevada) pages 622 – 628. IEEE Computer society, 2001
2. Fagin, R.: Combining fuzzy information from multiple systems. J. Comput. System Sci., 58:83-99, 1999
3. Fagin, R., Lotem, A., Naor, M.: Optimal Aggregation Algorithms for Middleware. In Proc. 20th ACM Symposium on Principles of Database Systems, pages 102-113, 2001
4. Ihab F. Ilyas: Rank – aware query processing and optimization. PhD Thesis, Purdue University, august 2004
5. Ihab F. Ilyas, Walid G. Aref, Ahmed K Elmagarmid: Supporting top-k join queries in relational database, Proceedings of VLDB, pages 754 – 765, Berlin, 2003
6. Gurský, P., Lencses, R.: Aspects of integration of ranked distributed data. In proc. Datakon, 2004
7. Güntzer, U., Balke, W., Kiessling, W.: Optimizing Multi-Feature Queries for Image Databases, Proceedings of the 26th VLDB Conference, Cairo, Egypt, 2000
8. Lencses, R.: Dopytovanie v systéme zameranom na získavanie informácií s podporou relačnej databázy, Zborník Datakon 04, Brno, Česká republika, Masarykova univerzita, Brno, Október 2004, 271-280

Annotation:

Algorithms for retrieving top k objects without random access

In this paper we present a quality of contemporary algorithms for user dependent integration of ranked distributed information without random access. We propose a new algorithm 3P-NRA and a new heuristic "*holes"*. We prove that this algorithm is correct and also "instance optimal" and we show that it needs less computation of objects values than NRA algorithm. We test our new approach on artificial and benchmark data.

Resolution Based Reasoning in Description Logic

Hashim Habiballa

Institute for Research and Applications of Fuzzy Modeling
and Department of Computer Science
University of Ostrava, 30. dubna 22, Ostrava, Czech Republic
Hashim.Habiballa@osu.cz

Abstract. The article presents refutational resolution theorem proving system for Description Logic (DL) based on general (non-clausal) resolution rule. There is also presented unification algorithm handling existentiality without the need of skolemization in first-order logic. Its idea follows from general resolution with existentiality for first-order logic. With this system existing resolution strategies may be used.
keywords: Automated theorem proving, non-clausal resolution, general resolution, unification, description logic.

1 Introduction

Description logic (DL) is an intensively studied formalism at present time [Bd03]. Its expressive power is sufficient for knowledge representation even it forms a subclass of First-Order Logic (FOL). Deduction as an important task in knowledge representation could be performed by several principles derived from FOL. We can state the tableau algorithm [Lu05] or clausal resolution [Hu00]. The second mentioned approach provides a promising formal framework benefiting from well-studied inference strategies in FOL. However the clausal form resolution has also some disadvantages when we consider the potential production of big amount of clauses with skolem functors. The conversion to the clausal form destroys the original structure of a formula and requires removal of existentiality. Therefore it could be interesting to search for more general rules. We will present a refutational resolution theorem proving system for an enriched ALC-like DL ($RRTP_{DL}$) based on a general (non-clausal) resolution principle in FOL [Ba97]. The prover handles formulas without any transformation, it requires more complex unification algorithm based on polarity criteria and quantifier mapping. The presented idea should be used in two-valued logic but it is more attractive for many-valued logics where often the clausal form is not obtainable at all.

2 Background from first-order logic

2.1 General resolution - brief overview

For the purposes of ($RRTP_{DL}$) we will use generalized principle of resolution defined in [Ba01].

⋆ Supported by research grant of Czech Ministry of Education - MSM 6198898701

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 106–117, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

General resolution

$$\frac{F[G] \quad F'[G]}{F[G/\bot] \vee F'[G/\top]} \tag{1}$$

where the first-order formulas F and F' are the premises of inference and G is an occurrence of a subformula of both F and F'. The expression $F[G/\bot] \vee F'[G/\top]$ is the resolvent of the premises on G. Every occurrence of G is replaced by false in the first formula and by true in the second one. It is also called F the positive, F' the negative premise, and G the resolved subformula.

The proof of the soundness of the rule is similar to clausal resolution rule proof. Suppose the Interpretation I in which both premises are valid. In I, G is either true or false. If G ($\neg G$) is true in I, so is $F[G/\top]$ ($F[G/\bot]$). Further we can observe the behavior of the rule within the frame of clausal form resolution. Consider following table showing various cases of resolution on clauses.

Premise1	Premise2	Resolvent	Simplified	Comments
$a \vee b$	$b \vee c$	$(a \vee \bot) \vee (\top \vee c)$	$\top$	no complemental pair
$a \vee \neg b$	$b \vee c$	$(a \vee \top) \vee (\top \vee c)$	$\top$	wrong selection of premises
$a \vee b$	$\neg b \vee c$	$(a \vee \bot) \vee (\bot \vee c)$	$a \vee c$	right clausal resolution
$a \vee \neg b$	$\neg b \vee c$	$(a \vee \top) \vee (\bot \vee c)$	$\top$	no complemental pair

Example 1. **General resolution with equivalence**
1. $a \wedge c \leftrightarrow b \wedge d$ (axiom), 2. $a \wedge c$ (axiom), 3.$\neg[b \wedge d]$ (axiom) - negated goal
4. $[a \wedge \bot] \vee [a \wedge \top]$ (resolvent from (2), (2) on c) $\Rightarrow$ a
5. $[a \wedge \bot] \vee [a \wedge \top \leftrightarrow b \wedge d]$ ((2), (1) on c) $\Rightarrow a \leftrightarrow b \wedge d$
6. $\bot \vee [\top \leftrightarrow b \wedge d]$ ((4), (5) on a) $\Rightarrow b \wedge d$
7. $\bot \wedge d \vee \top \wedge d$ ((6), (6) on b) $\Rightarrow$ d
8. $b \wedge \bot \vee b \wedge \top$ ((6), (6) on d) $\Rightarrow b$
9. $\bot \vee \neg[\top \wedge d]$ ((8), (3) on b) $\Rightarrow \neg d$
10. $\bot \vee \neg\top$ ((7), (9) on d) $\Rightarrow \bot$ (refutation)

When trying to refine the general resolution rule for first-order logic and description logic, it is important to devise a sound and complete unification algorithm. Standard unification algorithms require variables to be treated only as universally quantified ones. We will present a more general unification algorithm, which can deal with existentially quantified variables without the need for those variables be eliminated by skolemization. It should be stated that the following unification process doesn't allow an occurrence of the equivalence connective. It is needed to remove equivalence by rewrite rule: $A \leftrightarrow B \Leftrightarrow [A \rightarrow B] \wedge [B \rightarrow A]$.

We assume that the language and semantics of FOL is standard. We use terms - individuals $(a, b, c, ...)$, functions (with n arguments) $(f, g, h, ...)$, variables $(X, Y, Z, ...)$, predicates(with n arguments) $(p, q, r, ...)$, logical connectives $(\wedge, \vee, \rightarrow, \neg)$, quantifiers $(\exists, \forall)$ and logical constants $(\bot, \top)$. We also work with standard notions of logical and special axioms (sets LAx, SAx), logical consequence, consistency etc. as they are used in mathematical logic.

Definition 1. ***Structural notions of a FOL formula***
Let F be a formula of FOL then the structural mappings Sub (subformula), Sup (superformula), Pol (polarity) and Lev (level) are defined as follows:

$F = G \wedge H$ *or* $F = G \vee H$	$Sub(F) = \{G, H\}$, $Sup(G) = F$, $Sup(H) = F$ $Pol(G) = Pol(F)$, $Pol(H) = Pol(F)$
$F = G \rightarrow H$	$Sub(F) = \{G, H\}$, $Sup(G) = F$, $Sup(H) = F$ $Pol(G) = -Pol(F)$, $Pol(H) = Pol(F)$
$F = \neg G$	$Sub(F) = \{G\}$, $Sup(G) = F$ $Pol(G) = -Pol(F)$
$F = \exists \alpha G$ *or* $F = \forall \alpha G$ *(*α *is a variable)*	$Sub(F) = \{G\}$, $Sup(G) = F$ $Pol(G) = Pol(F)$

$Sup(F) = \emptyset \Rightarrow Lev(F) = 0, Pol(F) = 1,$
$Sup(F) \neq \emptyset \Rightarrow Lev(F) = Lev(Sup(F)) + 1$
For mappings Sub and Sup reflexive and transitive closures Sub^* *and* Sup^* *are defined recursively as follows:*
1. $Sub^*(F) \supseteq \{F\}$, $Sup^*(F) \supseteq \{F\}$
2. $Sub^*(F) \supseteq \{H | G \in Sub^*(F) \wedge H \in Sub(G)\}$, $Sup^*(F) \supseteq \{H | G \in Sup^*(F) \wedge H \in Sup(G)\}$

Example: $A \rightarrow B$ - $Pol(A) = -1$, $Pol(B) = 1$, $Lev(A) = 1$

These structural mappings provide framework for assignment of quantifiers to variable occurrences. It is needed for the correct simulation of skolemization (the information about a variable quantification in the prenex form). Subformula and superformula mappings and its closures encapsulate essential hierarchical information of a formula structure. Level gives the ordering with respect to the scope of variables (which is also essential for skolemization simulation - unification is restricted for existential variables). Polarity enables to decide the global meaning of a variable (e.g. globally an existential variable is universal if its quantification subformula has negative polarity). Sound unification requires further definitions on variable quantification. We will introduce notions of the corresponding quantifier for a variable occurrence, substitution mapping and significance mapping (we have to distinguish between original variables occurring in special axioms and newly introduced ones in the proof sequence).

Definition 2. ***Variable assignment, substitution and significance***
Let F *be a formula of FOL,* $G = p(t_1, ..., t_n) \in Sub^*(F)$ *atom in* F *and* α *a variable occurring in* t_i*. Variable mappings Qnt(quantifier assignment), Sbt (variable substitution) and Sig(significance) are defined as follows:*

$Qnt(\alpha) = Q\alpha H, where Q = \exists \vee Q = \forall, H, I \in Sub^*(F), Q\alpha H \in Sup^*(G),$
$\forall Q\alpha I \in Sup^*(G) \Rightarrow Lev(Q\alpha I) < Lev(Q\alpha H).$
$F[\alpha/t']$ *is a substitution of term* t' *into* α *in* $F \Rightarrow Sbt(\alpha) = t'.$
A variable α *occurring in* $F \in LAx \cup SAx$ *is significant w.r.t. existential substitution,* $Sig(\alpha) = 1$ *iff variable is significant,* $Sig(\alpha) = 0$ *otherwise.*

Example: $\forall x(\forall x A(x) \rightarrow B(x))$ - $Qnt(x) = \forall x A(x)$, for x in $A(x)$ and $Qnt(x) = \forall x(\forall x A(x) \rightarrow B(x))$, for x in $B(x)$.

Note that with Qnt mapping (assignment of first name matching quantifier variable in a formula hierarchy from bottom) we are able to distinguish between variables of the same name and there is no need to rename any variable. Sbt

mapping holds substituted terms in a quantifier and there is no need to rewrite all occurrences of a variable when working with this mapping within unification. It is also clear that if $Qnt(\alpha) = \emptyset$ then α is a free variable. These variables could be simply avoided by introducing new universal quantifiers to F. Significance mapping is important for differentiating between original formula universal variables and newly introduced ones during proof search (an existential variable can't be bounded with it).

Before we can introduce the standard unification algorithm, we should formulate the notion of global universal and global existential variable (it simulates conversion into prenex normal form).

Definition 3. ***Global quantification***
Let F be a formula without free variables and α be a variable occurrence in a term of F.

1. *α is a global universal variable ($\alpha \in Var_{\forall}(F)$) iff ($Qnt(\alpha) = \forall \alpha H \wedge Pol(Qnt(\alpha)) = 1$) or ($Qnt(\alpha) = \exists \alpha H \wedge Pol(Qnt(\alpha)) = -1$)*
2. *α is a global existential variable ($\alpha \in Var_{\exists}(F)$) iff ($Qnt(\alpha) = \exists \alpha H \wedge Pol(Qnt(\alpha)) = 1$) or ($Qnt(\alpha) = \forall \alpha H \wedge Pol(Qnt(\alpha)) = -1$)*

$Var_{\forall}(F)$ and $Var_{\exists}(F)$ are sets of global universal and existential variables.

Example: $F = \forall y(\forall x A(x) \rightarrow B(y))$ - x is a global existential variable, y is a global universal variable.

It is clear w.r.t. skolemization technique that an existential variable can be substituted into an universal one only if all global universal variables over the scope of the existential one have been already substituted by a term. Skolem functors function in the same way. Now we can define the most general unification algorithm based on recursive conditions (extended unification in contrast to standard MGU [Lt93]).

Definition 4. ***Most general unifier algorithm***
Let $G = p(t_1, ..., t_n)$ and $G' = r(u_1, ..., u_n)$ be atoms. Most general unifier (substitution mapping) MGU(G, G') = σ is obtained by following atom and term unification steps or the algorithm returns fail-state for unification. For the purposes of the algorithm we define the Variable Unification Restriction (VUR).

Variable Unification Restriction

Let F_1 be a formula and α be a variable occurring in F_1, F_2 be a formula, t be a term occurring in F_2 and β be a variable occurring in F_2. Variable Unification Restriction (VUR) for (α,t) holds if one of the conditions 1. and 2. holds:

1. *α is a global universal variable and $t \neq \beta$, where β is a global existential variable and α not occurring in t (non-existential substitution)*
2. *α is a global universal variable and $t = \beta$, where β is a global existential variable and $\forall F \in Sup^*(Qnt(\beta))$, $F = Q\gamma G$, $Q \in \{\forall, \exists\}$, γ is a global universal variable, $Sig(\gamma) = 1 \Rightarrow (Sbt(\gamma) = r') \in \sigma$, r' is a term (existential substitution).*

Atom unification

1. *if $n = 0$ and $p = r$ then $\sigma = \emptyset$ and the unifier exists (success-state).*
2. *if $n > 0$ and $p = r$ then perform term unification for pairs $(t_1, u_1), \ldots, (t_n, u_n)$; If for every pair unifier exists then $MGU(G, G') = \sigma$ obtained during term unification (success state).*
3. *In any other case unifier doesn't exist (fail-state).*

Term unification (t', u')

1. *if $u' = \alpha$, $t' = \beta$ are variables and $Qnt(\alpha) = Qnt(\beta)$ then unifier exists for (t', u') (success-state) (occurrence of the same variable).*
2. *if $t' = \alpha$ is a variable and $(Sbt(\alpha) = v') \in \sigma$ then perform term unification for (v', u'); The unifier for (t', u') exists iff it exists for (v', u') (success-state for an already substituted variable).*
3. *if $u' = \alpha$ is a variable and $(Sbt(\alpha) = v') \in \sigma$ then perform term unification for (t', v'); The unifier for (t', u') exists iff it exists for (t', v') (success-state for an already substituted variable).*
4. *if $t' = a$, $u' = b$ are individual constants and $a = b$ then for (t',u') unifier exists (success-state).*
5. *if $t' = f(t'_1, ..., t'_m)$, $u' = g(u'_1, ..., u'_n)$ are function symbols with arguments and $f = g$ then unifier for (t', u') exists iff unifier exists for every pair (t'_1, u'_1), ..., (t'_n, u'_n) (success-state).*
6. *if $t' = \alpha$ is a variable and VUR for (t', u') holds then unifier exists for (t', u') holds and $\sigma = \sigma \cup (Sbt(\alpha) = u')$ (success-state).*
7. *if $u' = \alpha$ is a variable and VUR for (u', t') holds then unifier exists for (t', u') holds and $\sigma = \sigma \cup (Sbt(\alpha) = t')$ (success-state).*
8. *In any other case unifier doesn't exist (fail-state).*

$MGU(A) = \sigma$ for a set of atoms $A = \{G_1, \ldots, G_k\}$ is computed by the atom unification for $(G_1, G_i), \sigma_i = MGU(G_1, G_i), \forall i, \sigma_0 = \emptyset$, where before every atom unification (G_1, G_i), σ is set to σ_{i-1}.

With above defined notions it is simple to state the general resolution rule for FOL (without the equivalence connective). It conforms to the definition from [Ba97].

Definition 5. ***General resolution for first-order logic*** (GR_{FOL})

$$\frac{F[G_1, , ..., G_k] \quad F'[G'_1, ..., G'_n]}{F\sigma[G/\bot] \vee F'\sigma[G/\top]} \tag{2}$$

where $\sigma = MGU(A)$ is the most general unifier (MGU) of the set of the atoms $A = \{G_1, \ldots, G_k, G'_1, \ldots, G'_n\}$, $G = G_1\sigma$. For every variable α in F or F', $(Sbt(\gamma) = \alpha) \cap \sigma = \emptyset \Rightarrow Sig(\alpha) = 1$ in F or F' iff $Sig(\alpha) = 1$ in $F\sigma[G/\bot] \vee F'\sigma[G/\top]$. F is called positive and F' is called negative premise, G represents an occurrence of an atom. The expression $F\sigma[G/\bot] \vee F'\sigma[G/\top]$ is the resolvent of the premises on G.

Note that with Qnt mapping we are able to distinguish variables not only by its name (which may not be unique), but also with this mapping (it is unique). Sig property enables to separate variables, which were not originally in the scope of an existential variable. When utilizing the rule it should be set the Sig mapping for every variable in axioms and negated goal to 1. We present a very simple example of existential variable unification before we introduce the refutational theorem prover for FOL.

Example 2. Variable Unification Restriction
We would try to prove if $\forall X \exists Y p(X,Y) \vdash \exists Y \forall X p(X,Y)$? We will use refutational proving and therefore we will construct a special axiom from the first formula and negation of the second formula:
$F_0 : \forall X \exists Y p(X,Y)$. $F_1(\neg query) : \neg \exists Y \forall X p(X,Y)$.
There are 2 trivial and 2 non-trivial combinations how to resolve F_0 and F_1 (combinations with the same formula as the positive and the negative premise could not lead to refutation since they are consistent):
Trivial cases: $R[F_1 \& F_1] : \bot \vee \top$ and $R[F_0 \& F_0] : \bot \vee \top$. Both of them lead to $\top$ and the atoms are simply unifiable since the variables are the same.
Non-trivial cases:$[F_1 \& F_0]$: no resolution is possible.
$Y \in Var_\forall(F_1)$ and $Y \in Var_\exists(F_0)$ can't unify since VUR for (Y,Y) doesn't hold - there is a variable $X \in Sup^*(Qnt(Y))$(over the scope), $X \in Var_\forall(F_0), Sbt(X) = \emptyset$); the case with variable X is identical.
$[F_0 \& F_1]$: no resolution is possible (the same reason as above).
No refutation could be derived from F_0 and F_1 due to VUR.

Further we would like to prove $\exists Y \forall X p(X,Y) \vdash \forall X \exists Y p(X,Y)$.
F0 : $\exists Y \forall X p(X,Y)$. F1 ($\neg$query) : $\neg \forall X \exists Y p(X,Y)$
In this case we can simply derive a refutation:
$R[F_1 \& F_0] : \bot \vee \neg \top (refutation)$
$X \in Var_\forall(F_0)$ and $X \in Var_\exists(F_1)$ can unify since VUR for (X,X) holds - there is no global universal variable over the scope of X in F_1; $Sbt(X) = X$ and $Sbt(Y) = Y$.

Now we entail above presented definitions by introduction of refutational theorem proving with the rule GR_{FOL}. We assume standard notions and theorems stated in [Ba01].

Definition 6. ***Refutational resolution theorem prover for FOL***
Refutational non-clausal resolution theorem prover for FOL ($RRTP_{FOL}$) is the inference system with the inference rule GR_{FOL} and sound auxiliary simplification rules for $\bot$, $\top$ (standard equivalencies for logical constants). A refutational proof of the goal G from the set of formulas (special axioms) $U = \{A_1, ..., A_m\}$ is a sequence of formulas $F_1, F_2, ..., F_n, \bot$, where F_i is an axiom from N, $\neg G$ or a resolvent from premises F_k and F_l $(k, l < i)$, where simplification rules may be applied to the resolvent. If there is a refutational proof of G from U then we write the $\bot$ is provable $U \cup \{\neg G\} \vdash \bot$). It is assumed that $Sig(\alpha) = 1$ for $\forall \alpha$ in $F \in N \cup \neg G$ formula, every formula in a proof has no free variable and has no quantifier for a variable not occurring in the formula.

Theorem 1. *$RRTP_{FOL}$ system is a sound inference system.*

Proof. The proof is based on standard properties of General resolution [Ba97], which is sound and complete rule in propositional logic (ground case in FOL) and could be lifted also to non-ground cases without existentiality (MGU works with the concept of universal variables and skolem functors). By this means we are able to translate the soundness problem into clausal normal form logic, where soundness of resolution is preserved. Every formula of FOL could be transformed into formula in clausal normal form preserving satisfiability. The notion of global universal and existential variable simulates the conversion into prenex normal form (i.e. a variable in original formula is universal in prenex normal iff it is global universal and contrary). Then the skolem functors (and constants as a special case) are introduced. Functors have arguments being universal variables prefixing the quantifier for the original existential variable. Unification in $RRTP_{FOL}$ has the same meaning. At first it is clear that two different global existential variables can't be unified like two different skolem functors. Further an existential variable α can't be substituted into universal one if it has any universal variable γ in $Sup^*(Qnt(\alpha))$, where $Sbt(\gamma) = \emptyset$.

Theorem 2. *$RRTP_{FOL}$ system is a complete inference system.*

Proof. The proof can be done by the similar way like the previous. We can translate the problem into clausal normal form formulas. For every proof in the $RRTP_{FOL}$ there is equivalent proof in clausal normal form. It could be done by induction on the Sup mapping. It is clear that general resolution encapsulates several steps of clausal form resolution and existentiality could be again simulated by skolem functors and vice-versa.

Proofs for both the properties are rather intuitive and should be done exactly. We presented only its essential ideas since the main intent of the article is to present RRTP for Description Logic. The non-clausal theorem prover for FOL is already implemented in the form of an experimental application [Ha00], [Ha05].

3 General resolution for description logic

In this section we will naturally build up the Refutational Resolution Theorem Prover for Description Logic $RRTP_{DL}$ based on previously presented general resolution rule for FOL. The approach is based on mapping DL formulas into FOL formulas and hence the properties of interpretation and resolution are preserved. The paper works with standard interpretation structures and rules of FOL [Lt93] and also for ALC-like DL [Bd03]. We use atomic concepts to be bound with unary predicates $A_1, ..., A_n$ and atomic roles are bound with binary predicates $R_1, ..., R_m$. Terms are constants $a_1, ..., a_l$ or variables $x_1, ..., x_k$. There is also top and bottom concepts $(\top, \bot)$. Concepts may be constructed using negation $\neg$, conjunction $\sqcap$, disjunction $\sqcup$, implication $\rightarrow$ (introduction of implication will not affect the expressive power), restricted quantification $\forall R.C$

bounded with $\forall y(R(x,y) \to C(y))$, $\exists R.C$ bounded with $\exists y(R(x,y) \wedge C(y)))$. Instances of concepts could be written in the form $C(t)$, where t is a term. When speaking about instances of quantified concepts its meaning in FOL is equivalent to replacing the occurrence of x variable by t. For the purposes of the prover we extend the notion of roles. At first we enable to use the atomic role negation $\neg R$, which is equivalent to its FOL representation and in semantic meaning it relates to complement operation on the extent of R. We also enable to use negation, conjunction and disjunction of roles $(\neg, \sqcap, \sqcup)$ and we introduce notion of full role $(\top_R)$ and empty role $(\bot_R)$. Interpretation of newly introduced full and empty roles is $\top_R^I = (\Delta^I \times \Delta^I)$ and $\bot_R^I = \emptyset$ (where $I = (\Delta^I, ^{\bullet I})$). We call all the formulas of DL - DL formulas and atomic concepts and roles - atoms.

Definition 7. ***Embedding of structural notions into FOL***
Let D be a DL formula. We call a formula F of FOL equivalent embedding formula by the following mapping $Emb(D) = F$ (embedding). All the above defined notions for FOL holds also for $Emb(D)$. Additionally we define a mapping Ext (extension) for D, which returns an explicit form of DL-formula with terms assigned to atomic concepts and roles.

Let C, D be concepts, R, S be roles, t, t_1, t_2 be terms, x, y be variables:
Recursive definiton of Ext:
$Ext(C) = C(x)$, $Ext(R) = R(x,y)$, $Ext(C(t)) = C(t)$, $Ext(R(t_1,t_2)) = R(t_1,t_2)$ *(for atomic concepts and roles)*

For every $\bullet \in \{\sqcap, \sqcup, \to\}$ it holds:
$Ext(C \bullet D) = (C \bullet D)$, $Ext(R \bullet S) = (R \bullet S)$
$Ext(\neg C) = \neg C$, $Ext(\neg R) = \neg R$

For every $Q \in \{\forall, \exists\}$ it holds:
$Ext(QR.C) = QR.C$,

Recursive definiton of Emb:
Let C, D be concepts and R, S be roles:
$Emb(C(t)) = C(t)$, $Emb(R(t_1,t_2) = R(t_1,t_2))$, $Emb(A) = A(x)$, $Emb(R) = R(x,y))$ *(atomic concept and role)*

$Emb(C \sqcap D) = C \wedge D$, $Emb(C \sqcup D) = C \vee D$, $Emb(C \to D) = C \to D$, $Emb(\neg C) = \neg C$, $Emb(R \sqcap S) = R \wedge S$, $Emb(R \sqcup S) = R \vee S$, $Emb(R \to S) = R \to S$, $Emb(\neg R) = \neg R$,

$Emb(\forall R.C) = \forall y(X)$, *where* $Emb(R.C) = X = R(x,y) \to C(y)$,
$Emb(\exists R.C) = \exists y(X)$, *where* $Emb(R.C) = X = R(x,y) \wedge C(y)$,
$Emb(\top) = \top$, $Emb(\bot) = \bot$, $Emb(\top_R) = \top$, $Emb(\bot_R) = \bot$.

For every DL formula D or DL formula part $R.C = D$, a variable y occurring in D or $R.C$, where $Emb(D) = F$, and formula mappings $Map \in \{Sub, Sup, Sub^*, Sup^*, Pol, Lev\}$, it holds $Map(D) = Map(F)$. For a variable y in an atomic concept or role (C or R) and F it holds $Qnt(y) = Qnt(\alpha)$, where α is a variable occurring in $Emb(C)$ or $Emb(R)$. Sig and Sbt mappings could be defined subsequently. If not stated otherwise $Sig(\alpha) = 0$. No free variables should occur in $Emb(D)$.

Example 3. **Examples of structural mappings**
Let $C \rightarrow \exists R.C'$ be a DL formula D. Then there are some notions (not all possible) holding for D:
$Ext(D) = C(x) \rightarrow \exists R(x,y).C'(y)$ and
$Pol(D) = 0$, $Pol(C) = -1$, $Pol(\exists R.C') = 1$, $Pol(C') = 1$, $Lev(D) = 0$, $Lev(\exists R.C') = 1$, $Lev(R.C) = 2$, $Lev(R) = 3$, $Sub(\exists R.C') = R(x,y) \rightarrow C'(y)$, $Sub^*(\exists R(x,y).C'(y)) = \{R(x,y) \rightarrow C'(y), R(x,y), C'(y)\}$,
$Sub(D) = \{C(x), \exists y(R(x,y) \rightarrow C'(y))\}$, $Sup(C'(y)) = R(x,y) \rightarrow C(y)$
For y in $C'(y)$ $Qnt(y) = \exists y(R(x,y) \rightarrow C(y))$

Definition 8. ***Simplification rules for*** $RRTP_{DL}$
Let C be a concept, R be a role and D be a DL formula. Auxiliary simplification rules follow (with additional use of the commutative and associative properties of the connectives):

$D \sqcap \bot \Rightarrow \bot$, $D \sqcap \top \Rightarrow D$, $D \sqcup \bot \Rightarrow D$, $D \sqcup \top \Rightarrow \top$, $D \rightarrow \bot \Rightarrow \neg D$, $D \rightarrow \top \Rightarrow \top$, $\bot \rightarrow D \Rightarrow \top$, $\top \rightarrow D \Rightarrow D$, $\neg\neg D \Rightarrow D$

$\forall \top_R.C \Rightarrow C$, $\forall \bot_R.C \Rightarrow \top$, $\forall R.\top \Rightarrow \top$, $\forall \top_R.\bot \Rightarrow \bot$, $\forall \bot_R.\bot \Rightarrow \top$
$\exists \top_R.\bot \Rightarrow \bot$,$\exists \top_R.\top \Rightarrow \top$, $\exists \bot_R.C \Rightarrow \bot$, $\bot_R \Leftrightarrow \bot$, $\neg\bot \Leftrightarrow \top$, $\neg\top \Leftrightarrow \bot$

Definition 9. ***General resolution for Description Logic*** (GR_{DL})
Let D and D' be DL formulas projected by Ext, G_i, G'_i be an atomic concept or role - atom (a subformula of D, D').

$$\frac{D[G_1,,...,G_k] \quad D'[G'_1,...,G'_n]}{D\sigma[G/\circ] \sqcup D'\sigma[G/\bullet]} \tag{3}$$

where $\sigma = MGU(A)$ is the most general unifier (MGU) of the set of the atoms, $A = \{G_1, \ldots, G_k, G'_1, \ldots, G'_n\}$, $G = G_1\sigma$. For G being an atomic concept $\circ = \bot$, $\bullet = \top$, for G being an atomic role $\circ = \bot_R$, $\bullet = \top_R$. For every variable α in D or D', $(Sbt(\gamma) = \alpha) \cap \sigma = \emptyset \Rightarrow (Sig(\alpha) = 1$ in D or D' iff $Sig(\alpha) = 1$ in $D\sigma[G/\circ] \vee D'\sigma[G/\bullet]$.

Definition 10. ***Refutational resolution theorem prover for DL***
Refutational non-clausal resolution theorem prover for DL ($RRTP_{DL}$) is the inference system with the inference rule GR_{DL} and sound auxiliary simplification rules.

A refutational proof of the goal G from the set of formulas (special axioms) $U = \{A_1, ..., A_m\}$ is a sequence of DL formulas projected by Ext $Ext(D_1)$, $Ext(D_2)$, ... , $Ext(D_n)$, $\bot$, where D_i is an axiom from N, $\neg G$ or a resolvent from premises D_k and D_l $(k, l < i)$; simplification rules may be applied to the resolvent. If there is a refutational proof of G from U then we write the $\bot$ is provable $U \cup \{\neg G\} \vdash \bot$). It is assumed that $Sig(\alpha) = 1$ $\forall \alpha$ variable in $N \cup \neg G$.

The proof of soundness and completeness of the $RRTP_{DL}$ is straightforward. Since the FOL embedding is semantically equivalent to the interpretation of DL formulas, all the properties of $RRTP_{FOL}$ holding for FOL should also hold for $RRTP_{DL}$.

4 Advantages of non-clausal approach and examples

One can argue that there are well-proved inference systems based on the clausal resolution and therefore the above presented prover is too general formalism to be successful in practice. This argument doesn't hold for many-valued logics where the problem of clausal form is very difficult to solve (there are some attempts to approximate normal forms). But also for two-valued DL $RRTP_{DL}$ forms an original solution of structure preserving inference mechanism. An important argument could be found concerning $\exists R.C$ and $\forall R.C$ constructs. Their structure and sense is quite destroyed by normal form transformation as could be seen by the following example.

Example 4. **Quantification in DL and normal forms**
Consider the following DL-formula:
$D = \forall T.C \sqcap \exists R.(\forall S.B)$.
$Emb(D) = \forall Y(T(X,Y) \rightarrow C(Y)) \wedge \exists Y(R(X,Y) \wedge (\forall Y(S(X,Y) \rightarrow B(Y))))$.
Prenex form:
$\forall X \exists Y_2 \forall Y_1 \forall Y_3((T(X,Y_1) \rightarrow C(Y_1)) \wedge R(X,Y_2) \wedge (S(X,Y_3) \rightarrow B(Y_3)))$.
Clausal form (s is a skolem functor):
$(T(X,Y) \rightarrow C(Y)) \wedge R(X,s(X)) \wedge (S(X,Y_3) \rightarrow B(Y_3))$.
$(\neg T(X,Y) \vee C(Y)) \wedge R(X,s(X)) \wedge (\neg S(X,Y_3) \vee B(Y_3))$.
Result (4 clauses):
$C1 : \neg T(X,Y) \vee R(X,s(X)) \vee \neg S(X,Y_3)$.
$C2 : \neg T(X,Y) \vee R(X,s(X)) \vee B(Y_3)$.
$C3 : C(Y) \vee R(X,s(X)) \vee \neg S(X,Y_3)$.
$C4 : C(Y) \vee R(X,s(X)) \vee B(Y_3)$.

In the above example the resulting clauses are significantly larger than the source DL-formula. The structure of the formula is completely lost and the resolution-based inference will potentially need to work with all the four clauses (which will cause unnecessary time and space complexity). Even when one will perform the physical transformation of DL-formula into equivalent FOL-formula the complexity will be significantly reduced. Now we present a simple example of a proof in the prover. Since the $RRTP_{FOL}$ is already implemented in the form of an experimental computer application GERDS [Ha05] it was used for proof generation in $RRTP_{DL}$ (with hand rewriting into DL).

Example 5. **Non-clausal resolution in** $RRTP_{DL}$
Suppose we have the following concepts:
person, male, female and *happy_child*
and *role is_child_of* representing the role of being X child of Y.
We would like to express the knowledge that:
If person is child of at least one male and one female, then it is a happy child (since has both mother and father).
It can be expressed in ALC-like DL with implication like:
$(person \sqcap \exists is_child_of.female \sqcap \exists is_child_of.male) \rightarrow happy_child$
Then we will suppose the following instances:

is_child_of(johana, hashim). is_child_of(johana, lucie). male(hashim). female(lucie). person(johana).

Source formulas (axioms) :

$F0:$ $(person(X) \sqcap \exists is_child_of(X,Y).female(Y)$
$\sqcap \exists is_child_of(X,Y).male(Y)) \rightarrow happy_child(X)$

$F1:$ $is_child_of(johana, hashim).$

$F2:$ $is_child_of(johana, lucie).$

$F3:$ $male(hashim).$

$F4:$ $female(lucie).$

$F5:$ $person(johana).$

$F6:$ $\neg happy_child(johana).(\neg$ query$)$

$[F6\&F0]:$ $\bot \sqcup (person(johana) \sqcap \exists is_child_of(johana,Y).female(Y)$
$\sqcap \exists is_child_of(johana,Y).male(Y) \rightarrow \bot).$

$\Rightarrow R0:$ $\neg(person(johana) \sqcap \exists is_child_of(johana,Y).female(Y)$
$\sqcap \exists is_child_of(johana,Y).male(Y)).$

$[R0\&F5]:$ $\neg(\top \sqcap \exists is_child_of(johana,Y).female(Y)$
$\sqcap \exists is_child_of(johana,Y).male(Y)) \sqcup \bot.$

$\Rightarrow R1:$ $\neg(\exists is_child_of(johana,Y).female(Y)$
$\sqcap \exists is_child_of(johana,Y).male(Y)).$

$[R1\&F4]:$ $\neg(\exists is_child_of(johana,lucie).\top \sqcap \exists is_child_of(johana,Y).male(Y))$
$\sqcup \bot.$

$\Rightarrow R2:$ $\neg(\exists is_child_of(johana,lucie).\top \sqcap \exists is_child_of(johana,Y).male(Y)).$

$[R2\&F3]:$ $\neg(\exists is_child_of(johana,lucie).\top \sqcap \exists is_child_of(johana,hashim).\top)$
$\sqcup \bot.$

$\Rightarrow R3:$ $\neg(\exists is_child_of(johana,lucie).\top \sqcap \exists is_child_of(johana,hashim).\top).$

$[R3\&F2]:$ $\neg(\exists \top_R.\top \sqcap \exists is_child_of(johana,hashim).\top).$

$\Rightarrow R4:$ $\neg\exists is_child_of(johana,hashim).\top.$

$[R4\&F1]:$ $\neg(\exists \top_R.\top) \sqcup \bot.$

$\Rightarrow:$ $\bot$ (refutation)

Non-clausal resolution brings several significant advantages when trying to implement the above presented ideas. It really doesn't require any transformation of a DL-formula. When we construct the Embedding mapping for every subformula, it may be done before the inference process since it can be assigned to special axioms and then it can be copied during the resolution. We suggest to enrich the hierarchy of a formula on every level by the pointer to the *Emb* representation. Then it is possible to work with a tree-based structure of the DL-formula and FOL-representation simultaneously. The second alternative (preferred) is to hold only a DL-formula and to compute the structural notions of every subformula required for the sound unification from the current *Emb* representation (it requires establishment of "virtual" variables for quantifiers). Thus the structural mappings should be only recursively refined and *Emb* need not to be physically created and stored. The third way is to completely rewrite a DL-formula into FOL through *Emb* representation. This way is the simplest but may be perceived to stand out of the Description Logic.

5 Conclusions and further research

The above presented prover represents an alternative deductive system for DL. Its advantage is based on wide usage of resolution principle in FOL, where a lot of implementations and strategies exist. Additionally the structure of original DL-formulas is preserved during theorem proving. We mentioned an application GERDS, which utilizes General resolution in FOL and it seems to be natural to extend GERDS also for Description Logic. Another interesting and natural extension also arise in conjunction with fuzzy logic, where normal form transformations are hard to create. Fuzzy Description Logic forms useful formalization since concepts are often "fuzzy" in real situations [Hj05]. The problem of automated theorem proving in fuzzy predicate logic is much more complicated in comparison with two-valued logic due to clausal normal form conversion. The author also has proposed the Refutational Resolution Theorem Prover for Fuzzy Predicate Logic [Ha05a]. Its implementation is being currently developing together with effective inference strategies.

References

[Ba97] Bachmair L., Ganzinger H. A theory of resolution. Technical report: Max-Planck-Institut für Informatik, 1997.

[Ba01] Bachmair, L., Ganzinger, H. Resolution theorem proving. in Handbook of Automated Reasoning, MIT Press, 2001.

[Bd03] Baader et col. (eds.). The Description Logic Handbook - Theory, Interpretation and Applications. Cambridge Univ. Press, 2003.

[Ha00] Habiballa, H. Non-clausal resolution - theory and practice. Research report: University of Ostrava, 2000, http://www.volny.cz/habiballa/files/gerds.pdf

[Ha05] Habiballa, H. Non-clausal Resolution Theorem Prover. Research report, No.64: University of Ostrava, 2005, http://ac030.osu.cz/irafm/ps/rep64.ps.gz

[Ha05a] Habiballa, H. Non-clausal Resolution Theorem Proving for Fuzzy Predicate Logic. Research report, No.70: University of Ostrava, 2005, http://ac030.osu.cz/irafm/ps/rep70.ps.gz

[Hj05] Hájek, P. Making fuzzy description logic more general. Fuzzy Sets and Systems 154(2005),pp. 1-15

[Hu00] Hustadt, U., Schmidt, R.A. Issues of Decidability for Description Logics in the Framework of Resolution. In Automated Deduction in Classical and Non-Classical Logics, pp. 192-206. LNAI 1761, Springer 2000.

[Lt93] Letz, R. First-order calculi and proof procedures for automated deduction. Ph.D. thesis, TH Darmstadt, 1993

[Lu05] Lukasová, A. Reasoning in Description Logic with semantic tableau binary trees. Research report: Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, 2005.

Ferda, nové vizuální prostredí pro dobývání znalostí

Michal Kováč, Tomáš Kuchař, Alexander Kuzmin, Martin Ralbovský

Univerzita Karlova v Praze, Matematicko-fyzikální fakulta
Ke Karlovu 3, 121 16 Praha 2

ferda-users@lists.sourceforge.net

Abstrakt. Jedním z cílů návrhu programu Ferda bylo vytvořit aplikaci, která nahradí stávající uživatelské prostředí systému na dobývání znalostí LISp-Miner (lispminer.vse.cz) novým, vizuálně a uživatelsky přívětivějším pracovním prostředím. Ferda přináší pohled na zadání úlohy dobývání znalostí jako na zapsání funkce pomocí vizuálních objektů – krabiček. Krabičky mají parametry a vyhodnocují své funkce nad hodnotami těchto parametrů. Program je navržen modulárně tak, aby jej bylo možné snadno rozšiřovat o další krabičky. Podařilo se vytvořit prostředí, které zpřehledňuje a zjednodušuje uživatelskou práci se systémem LISp-Miner a nabízí další možnosti výzkumu a rozšiřování.

Klíčová slova: Ferda, LISp-Miner, vizuální programování, krabička, zásuvka

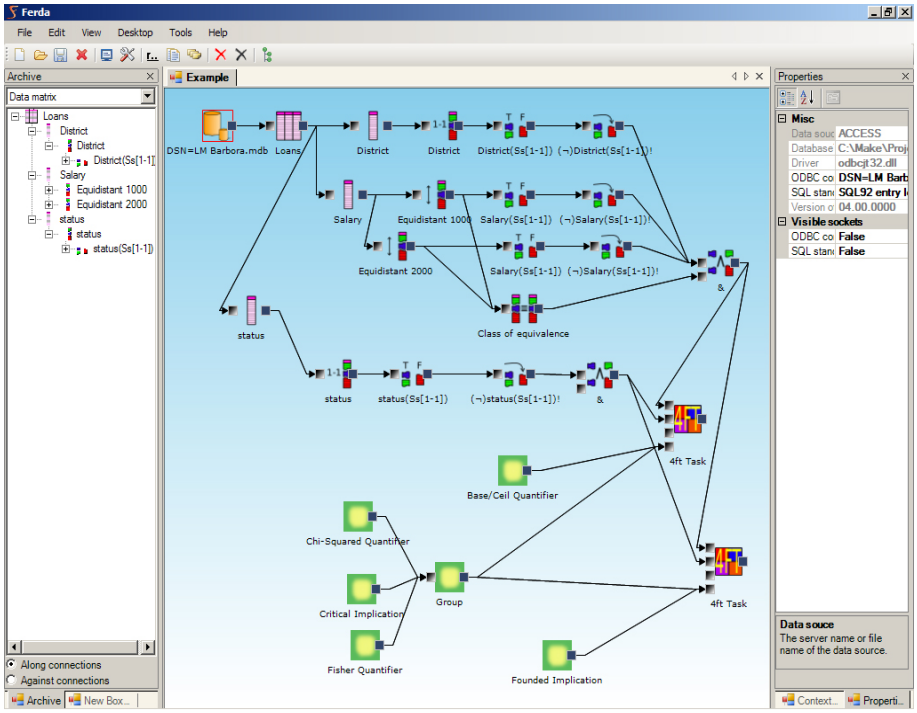
1 Úvod

Historie systému LISp-Miner[1][2] sahá do roku 1996, kdy byl navržen. Jde o akademický softwarový systém pro výuku (zejména FIS VŠE a MFF UK) a výzkum v oblasti dobývání znalostí z databází. Systém obsahuje řadu modulů, které umožňují potřebným způsobem transformovat prvotní data, provádět vlastní analýzy a vhodným způsobem prezentovat výsledky. Systém je stále ve vývoji, který pokrývá jak implementaci nových modulů – v posledních dvou letech byl rozšířen o nové procedury – tak i prohlubování a rozšiřování modulů již implementovaných. Vedlejším důsledkem tohoto vývoje jsou stále větší nároky na uživatele systému, pokud chce využít všech možností systému LISp-Miner. Pro snadné porozumění jistým částem článku předpokládáme, že je čtenář již seznámen se systémem LISp-Miner.

Ferda vzniká jako softwarový projekt na Matematicko-fyzikální fakultě Univerzity Karlovy v Praze pod vedením doc. RNDr. Jana Raucha, CSc. (Vysoká škola ekonomická v Praze). Projekt má za úkol zpřehlednit a zjednodušit postup používání existujících a pokud možno i budoucích modulů systému LISp-Miner při dobývání znalostí. Stávající postup se skládá z postupného spouštění různých aplikací a nastavování parametrů, mezi nimiž nejsou na první pohled vidět souvislosti. V důsledku toho se proces získávání dat stává pro běžné uživatele nejasným a nepřehledným.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 118–129, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

Ferda představuje společné vizuální prostředí pro následující procedury systému LISp-Miner: *4ft-Miner*, *SD4ft-Miner*, *KL-Miner*, *SDKL-Miner*, *CF-Miner* a *SDCF-Miner*.



Obr. 1. Prostředí Ferda

Ferda přináší vizuální model postupu dobývání znalostí podobný svou zásadou *plocha jako pracovní nástroj dobývání znalostí* (obrázek 1) systémům SPSS Clementine[8], SAS Enterprise Miner[10] a Weka[9]. Srovnání těchto systému s Ferdou naleznete v sekci 7.

Ferda není pouze prostředí, které pracuje nad systémem LISp-Miner, ale jedná se o samostatnou jednoduše rozšiřitelnou aplikaci, která může být využita pro různé typy úloh.

V následujících odstavcích nejdříve popíšeme základní ideu hlavních objektů prostředí Ferda (krabiček). Poté nastíníme základy zásad implementace při nichž poukážeme na rozšiřitelnost, modularitu a distribuovatelnost prostředí Ferda. Bude následovat popis prvků prostředí Ferda, na kterých bude ukázáno jak uživateli zjednoduší práci. K závěru bude popsán příklad dobývání znalostí pomocí prostředí Ferda.

2 Krabičky

Ferda přináší pohled na zadání úlohy dobývání znalostí jako na zapsání funkce pomocí jistých vizuálních objektů – tzv. krabiček. Každá krabička zastává také roli funkce nebo několika funkcí. Krabičky mají parametry a vyhodnocují své funkce nad hodnotami těchto parametrů. Parametrům se u krabiček říká zásuvky. Do zásuvek se zapojují jiné krabičky. Krabičky jsou rozděleny podle typů a do každé zásuvky lze zapojit pouze některé typy krabiček.

Pomocí řady jednoduchých krabiček je uživateli umožněno sestavovat vlastní algoritmy. Ferda tedy rovněž představuje něco jako uživatelský programovací jazyk. Na krabičky se lze však také dívat jako na konstrukce logiky či sémantiky, ke které existuje lambda-kategoriální gramatika.

2.1 Krabičky formálně

Krabička je dvojice $\langle S, F \rangle$, kde S je množina zásuvek a F je množina funkcí. Zásuvka je dvojice $\langle n, T \rangle$, kde n je název zásuvky a T je množina typů krabiček, zapojitelných do této zásuvky.

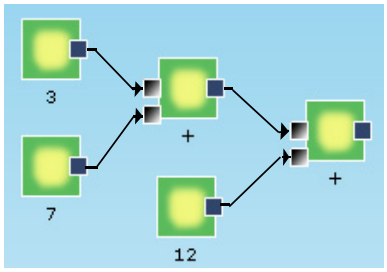
2.2 Ukázka možností krabiček

Ukažme si krabičky na příkladu. Nebude však z oblasti dobývání znalostí, jak by se dalo očekávat. Místo toho budou krabičky předvedeny na jednoduché funkci sčítání, která je výhodná tím, že na ní lze ukázat několik možností krabiček. Bude to také názornější pro čtenáře, kteří neznají systém LISp-Miner.

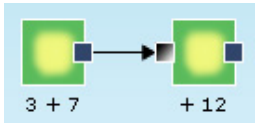
Mějme krabičku *sčítání*. Funkcí této krabičky je součet dvou čísel. Pro každé z těchto čísel bude mít krabička *sčítání* zásuvku, do které bude možné zapojit krabičku typu *číslo*. Sama krabička *sčítání* je však také typu *číslo*. Nyní řekněme, že chceme spočítat $3 + 7 + 12$.

Za tím účelem definujme krabičku typu číslo, jejíž funkcí je číselná konstanta. Krabičky kromě zásuvek mohou mít vlastnosti. Vlastnosti jsou, stejně jako zásuvky, různých typů, například *řetězec*, *čísla* (ta jsou v programu rozdělena podle velikostí a toho, zda jsou s desetinou čárkou) či *datum*. Jedinou vlastností výše definované krabičky bude *hodnota číselné konstanty*. Výsledná zapojení pro výpočet $3 + 7 + 12$ je na obrázku 2.

Vlastnosti a zásuvky představují vstupní parametry pro funkce, které krabička reprezentuje. Hodnoty zásuvek se nastavují vždy pomocí připojení jiné krabičky, zatím co hodnoty vlastností se mohou nastavovat pomocí speciální komponenty v pracovním prostředím – panelu vlastností. U vlastností lze přepínat mezi nastavováním hodnoty pomocí krabiček a pomocí panelu vlastností. Vlastnostem totiž odpovídají zásuvky stejného názvu a typu, které jsou však ve výchozím nastavení uživateli skryty, ale je možné si je nechat zobrazit. Díky tomu, může mít naše krabička sčítání namísto dvou zásuvek dvě vlastnosti typu číslo a tak náš vzorec $3+7+12$ lze zapsat, jak je znázorněno na obrázku 3, místo pomocí pěti krabiček pouze pomocí krabiček dvou.

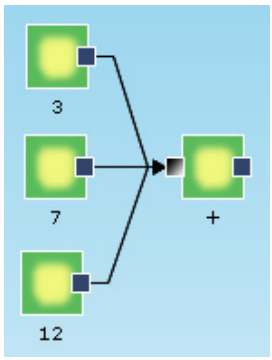


Obr. 2. Jednoduché sčítání



Obr. 3. Sčítání pomocí dvou krabiček

Do některých zásuvek je možné zapojit více než jednu krabičku. Proto lze vytvořit krabičku sčítání, která má jednu zásuvku a sečte všechna čísla reprezentovaná krabičkami, které jsou zapojeny do této zásuvky, viz. obrázek 4.



Obr. 4. Krabička pro multisčítání

V definici krabičky stálo, že krabička může reprezentovat více funkcí. Demonstrujme tuto schopnost krabičky již na příkladu z dobývání znalostí. Ferda obsahuje pro dobývání znalostí mimo jiné tři základní krabičky – *Databáze*, *Tabulka* a *Sloupec*. Krabička *Databáze* zastává roli několika funkcí. Jedna vrací seznam tabulek v databázi, jiná vrací způsob, jakým se má připojovat k dato-

vému zdroji. Tento způsob nastavuje uživatel ve vlastnosti *Připojovací řetězec* této krabičky.

3 Zásady implementace

Ferda je psán pod licencí GNU General Public License (GPL). Licence umožňuje program zdarma používat, zaručuje, že jsou k dispozici zdrojové soubory, a umožňuje rozšiřování programu někým jiným, pokud výsledný kód je také pod licencí GPL.

Většina současného kódu je napsána v druhé verzi jazyka C#. Výsledný program běží nad prostředím .NET Framework. Některé části také je možné spustit nad prostředím Mono[7], které běží nejen na operačním systému Windows, ale například i na systému Linux či Mac OS. Autoři by chtěli docílit toho, aby všechny nynější části prostředí Ferda bylo možné spustit nad prostředím Mono.

Program je navržen modulárně tak, aby jej bylo možné relativně snadno rozšiřovat o další krabičky. Krabička může být samostatný program napsaný v jednom z mnoha programovacích jazyků. Je možné nechat distribuovat výpočet po síti, například každá krabička může běžet na jiném stroji. To vše lze díky tomu, že je Ferda postaven nad technologií Internet Communications Engine (Ice)[6]. Jedná se o rychlý objektový otevřený middleware podobný uznávanému standardu v této oblasti – CORBA.

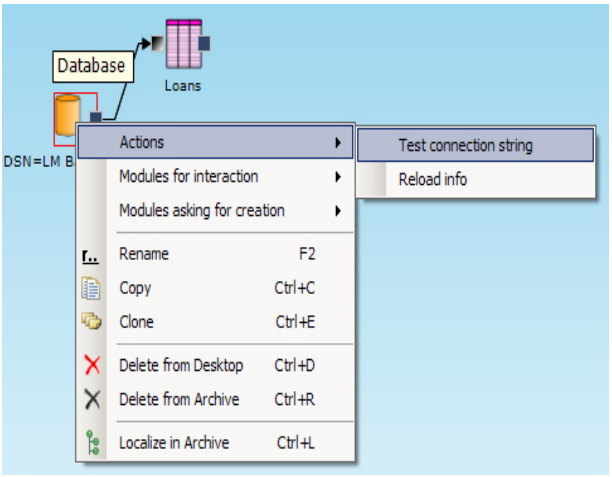
4 Pracovní prostředí

Při návrhu pracovního prostředí systému Ferda se tvůrci snažili držet se co nejvíce standardů v oblasti programování a vizuálního programování. Proto byly pro prostředí zvoleny standardní komponenty (části pracovního prostředí), na které je uživatel zvyklý z různých jiných nástrojů a nebude mu dělat problémy začít je používat i ve Ferdovi. Jedním z nich je mechanizmus dokování oken. Všechny komponenty, u kterých to má smysl, je možné pomocí tohoto mechanizmu přesouvat po pracovní ploše, či skrývat. Dokování je standard používaný v uživatelských prostředích mnohých produktů, za všechny jmenujme Microsoft Visual Studio.

Vraťme se ke krabičkám. Pro uživatele je krabička grafický objekt na pracovní ploše. Krabička je malý čtverec s ikonou, který má na stranách menší čtverce – zásuvky. Jednotlivé zásuvky propojuje uživatel na ploše kliknutím a přetáhnutím myši.

V předchozích odstavcích byl čtenář seznámen s funkcemi krabiček. V pracovním prostředí lze funkce vyvolat pomocí položky *Akce* kontextového menu nad krabičkou, jak je vidět na obrázku 5.

Krabička je někdy tak složitá, že samotné nastavování vlastností a propojování zásuvek nestačí na pohodlnou uživatelskou práci. Proto může mít nadefinované moduly pro interakci. Modul pro interakci znamená volání akce, kterou zajišťuje jiný modul než samotná krabička. Často to může být část pracovního prostředí: dialogy, tabulky grafy....



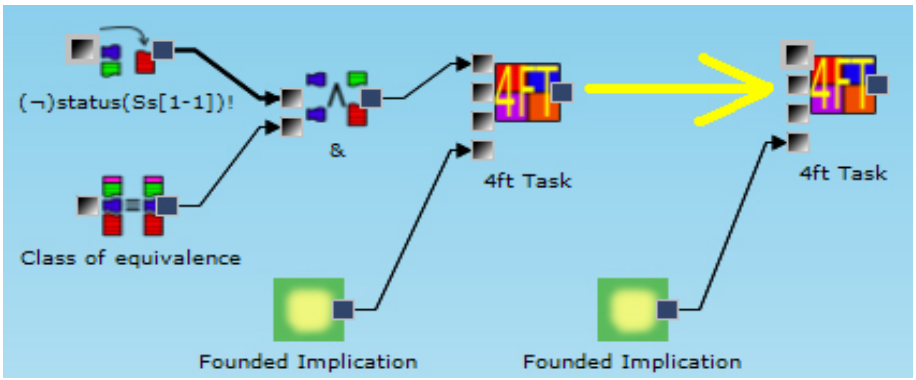
Obr. 5. Krabičky a kontextové menu

Krabička často nabízí v kontextovém menu nebo v hlavním menu aplikace seznam krabiček, které se mohou automaticky vytvořit s přednastavenými hodnotami vlastností a zapojení. Nejlépe to lze vysvětlit na příkladu. Mějme krabičku *Databáze*, ta navrhuje vytvoření krabičky *Tabulka*. V kontextovém menu této krabičky bude tedy pod položkou *Navrhované krabičky na vytvoření* seznam názvů tabulek v databázi. Jestliže uživatel klikne, na některou tabulku, vytvoří se vedle krabičky *Databáze* krabička *Tabulka*, nastaví se u ní vlastnost *Jméno tabulky v databázi* na jméno, které si uživatel vybral. Dále se krabička *Databáze* propojí s touto novou krabičkou *Tabulka*.

Jestliže srovnáme výše popsané způsoby uživatelské práce s Ferdou se způsobem, jakým se formulují úlohy v původním uživatelském rozhraní systému LISp-Miner, vidíme, že je Ferda v mnoha ohledech přívětivější. Zatímco v původním rozhraní, bylo k zadání celé úlohy a k procházení jejích výsledků zapotřebí použití tří různých programů (například pro úlohu pro 4ft-Miner to jsou programy *LMDataSource*, *4ftTask* a *4ftResult*), ve Ferdovi je vše integrováno do jednoho nástroje. Je však nutné počítat s tím, že poměrně velké množství krabiček může být pro uživatele začátečníka matoucí a i zkušený uživatel starého systému může v počátcích tápat. Stávající zkušenosti ukazují, že *Navrhované krabičky na vytvoření* velmi usnadňují začátečníkům práci ve Ferdovi.

Komplikovanější úlohy dobývání znalostí z dat vyžadují, aby na ploše bylo velké množství krabiček zapojených do sebe. Uživatel, který pracuje s touto úlohou, často nechce znát celé komplikované zapojení všech krabiček a zajímá ho jenom výsledek. Pro tyto případy Ferda nabízí systém zabalování a rozbalování krabiček. Uživatel klikne na zásuvku, kterou chce zabalit, popřípadě na celou krabičku, jestliže chce zabalit všechny zásuvky krabičky, a zvolí patřičnou akci v kontextovém menu. Poté z pracovní plochy zmizí krabičky a na zásuvce se

objeví symbol, že je v ní něco zabaleno viz. obrázek 6. Stejným způsobem potom může krabičky rozbalit.



Obr. 6. Zabalování krabiček

Uživatel není v pracovním prostředí Ferda omezen jednou pracovní plochou. Může tvořit a ukládat libovolný počet pracovních ploch. Toto pomáhá rozdělit si komplikované úlohy na menší podčásti a pracovat na nich odděleně.

4.1 Archiv krabiček

Archiv je vedle pracovní plochy další způsob, jak uživateli usnadnit a především zefektivnit jeho práci. Všechny krabičky, které uživatel vytvořil, jsou zaznamenány v archivu. Uživatel se může v archivu lehce vracet k dřívějším postupům a dále je používat. Při výchozím nastavení je panel s archivem umístěn v levé části pracovního prostředí, jak je vidět na obrázku 1.

Uživatel může procházet archiv horizontálně nebo vertikálně. Při vertikálním prohlížení uživatel přepíná mezi typy krabiček, které má v archivu. Systém potom zobrazí pod sebou všechny krabičky v archivu, které jsou daného typu. Horizontální prohlížení archivu znamená to, že uživatel zkoumá archiv po nebo proti směru zapojení šipek propojujících jednotlivé zásuvky krabiček.

Pro propojení archivu a pracovní plochy slouží funkce pracovní plochy nazvaná *Lokalizovat v archivu*. Při této akci kontextového menu nad krabičkou na pracovní ploše se vyhledá daná krabička v archivu a označí se. Tato funkce se opět hodí když uživatel ztrácí přehled nad množstvím krabiček, které vytvořil.

Všechny uživatelem vytvořené krabičky jsou automaticky ukládány do archivu, odkud je může uživatel kdykoliv vyzvedávat, upravovat jejich nastaveni čí používat je v nových kontextech. Archiv naplňuje požadavky dlouholetých uživatelů systému LISp-Miner na jednotné úložiště kroků dobývání znalostí jako například zadání cedentů, zadání literálů nebo zadání tříd ekvivalence.

4.2 Panel vlastností

Panel vlastností je ovládací prvek, pomocí něhož nastavujeme vlastnosti všech krabiček. Jednoduché vlastnosti krabiček se nastavují přímo v něm, pro složitější panel volá dialogy, které vlastnost nastaví. Panel vlastností (PropertyGrid) je standardizovaná komponenta .NET Framework a používá se v mnoha jiných programech. Panel vlastností přináší důležité zlepšení oproti původní verzi uživatelského rozhraní systému LISp-Miner - jednotný způsob nastavování vlastností všech objektů.

4.3 Kontextová nápověda

Kontextová nápověda má za úkol jednoduše seznámit uživatele s funkčností krabičky právě označené na pracovní ploše. Při označení krabičky na pracovní ploše či v archivu se pro krabičku vygeneruje dynamická nápověda v komponentě Kontextová nápověda. Obsahuje základní informace o funkci krabičky, jejích vlastnostech a zásuvkách. Nápověda může obsahovat i odkazy na stránky v externích dokumentech v různých formátech, za všechny jmenujme alespoň HTML či PDF.

5 Implementované krabičky

Ve Ferdovi jsou implementovány i krabičky, které rozšiřují možnosti systému.

5.1 Krabičky jednoduchých datových typů

Jsou vytvořeny krabičky pro nastavování jednoduchých datových typů (textový řetězec, číslo, datum, ...), které pak lze zapojit do zásuvek nastavujících tyto vlastnosti. Například krabička *Zadání booleovského dílčího cedentu* má mimo jiné vlastnosti pro nastavení minimálního a maximálního počtu literálů v cedentu. Tyto délky se standardně nastavují v panelu vlastností krabičky, lze z nich však vytvořit zásuvky akceptující krabičky stejného datového typu. To například umožňuje analytikovi nastavit výše zmíněnou maximální délku v krabičce *Číslo*, kterou pak připojí do patřičné zásuvky. Totéž může udělat u více vlastností různých krabiček a pak pouze jednou změnou hodnoty v krabičce *Číslo* změnit nastavení mnoha krabiček najednou.

5.2 Krabička skupina

Krabička *Skupina* je dalším praktickým pomocníkem analytika při práci s Ferdou. Příklad: Analytik připraví několik krabiček různých 4ft-kvantifikátorů, které zapojí do krabičky *4ft-Task*. Nyní, bude-li chtít použít stejné kvantifikátory u jiné 4ft úlohy, musí znovu vytvořit pro všechny tyto kvantifikátory spojení s novou krabičkou *4ft-Task*. Tuto neefektivitu práce řeší Ferda krabičkou skupina. *Skupina* umožňuje seskupování krabiček jakéhokoli typu, tuto skupinu pak může analytik snadno znovu použít na jiném místě v jiném kontextu. Sama o sobě

Skupina neposkytuje žádné funkce, při jejím zapojování do nějaké zásuvky vystupuje stejně jako krabičky do ní zapojené tj. jestliže zásuvka vyžaduje nějaké funkce, *Skupina* do něj nepůjde zapojit, pokud každá krabička uvnitř *Skupiny* neposkytuje tyto funkce. Rovněž platí, že pokud lze do zásuvek zapojit jen omezený počet krabiček, nebude možné do něj zapojit *Skupinu* pokud do ní není zapojen stejně omezený počet krabiček. *Skupina* je tak pro uživatele vedle archivu dalším snadno použitelným pomocníkem pro vytváření a uchovávání pracovních postupů.

5.3 Dynamické a nedynamické atributy

V modulu *LMDataSource* systému LISp-Miner existuje několik možností pro vytvoření atributu. Uživatel může ručně vytvořit seznamy hodnot či intervalů hodnot, tvořící jednotlivé kategorie, nebo při jejich vytváření může použít tři vestavěné algoritmy pro automatické generování kategorií a tyto pak ručně upravit. Jde o algoritmy:

Each value one category pro každou hodnotu z domény sloupce vytvoří právě jednu kategorii.

Equidistant intervals generuje intervaly stejné délky od zadané počáteční hodnoty prvního intervalu, po poslední hodnotu v doméně sloupce.

Equifrequency intervals generuje zadaný počet kategorií s přibližně stejným počtem objektů v každé kategorii.

Pokud se však data pro analýzu změní, je často třeba pro uchování aktuální struktury vytvořených atributů aplikovat tyto algoritmy znovu, protože již vytvořené struktury zpětně nereflektují změny v datech. Ve Ferdovi byla tato nepříjemná vlastnost odstraněna tím, že byly vytvořeny krabičky tzv. dynamických atributů, které ve chvíli kdy od něj jiná krabička vyžaduje množinu kategorií, ověří zda nedošlo ke změně dat a pokud ano, automaticky vygenerují struktury atributů odpovídající aktuálním datům. Tato vlastnost, mimo jiné, umožní analytikovi relativně snadno zkoumat rozdíly ve výsledcích dobývání nad databází, jejíž data se v průběhu času mění. Uživateli je však i nadále ponechána možnost ručně vytvářet kategorie a nechat si při tom pomáhat těmito algoritmy (krabičkami dynamických atributů).

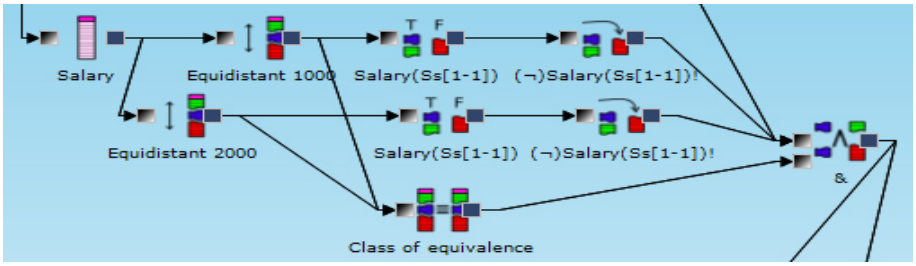
6 Příklad řešení úlohy

Na obrázku 1 je naznačeno zadání dvou úloh pro *4ft-Miner*, které mají téměř stejné parametry, liší se pouze zadanými kvantifikátory. Z obrázku je patrné, že již nebylo nutné znovu vytvářet zadání cedentů pro obě úlohy zvlášť, tak jak tomu je v původním uživatelském rozhraní systému LISp-Miner, ale zde již uživatel snadno opakovaně použil dříve nadefinované – archivované – zadání cedentů.

Další práci, která by vznikala v důsledku četného vytváření spojení mezi krabičkami, si může analytik ušetřit pomocí dříve zmíněné krabičky *Skupina*.

Například obě úlohy na obrázku mají definovány čtyři kvantifikátory, z nichž tři jsou společné, ale jeden kvantifikátor má každá úloha jiný. Analytik vytvořil a nastavil tři společné kvantifikátory, ty umístil do krabičky *Skupina*, tu zapojil do příslušné zásuvky obou 4ft-úloh a poté už jen vytvořil krabičky lišících se kvantifikátorů a také obě zapojil do zásuvek příslušných krabiček.

Na tomto příkladu je rovněž naznačeno používání krabiček dynamických atributů. Takto formulovaná úloha totiž umožňuje pozdější spuštění s tím, že se, bude-li si to uživatel přát, u dynamických atributů znovu dynamicky vygenerují množiny kategorií, což v případě, že se v databázi změnily domény či frekvence u sloupců (případně odvozených), z nichž jsou odvozeny tyto atributy, ušetří analytikovi mnoho jeho práce, kterou by v dřívější implementaci grafického rozhraní systému LISp-Miner musel udělat tj. znovu ručně u každého atributu provést generování množiny kategorií.



Obr. 7. Vylepšení třídy ekvivalence

Čtenář, který již má zkušenosti se třídami ekvivalence v systému LISp-Miner, si může povšimnout, že další změny oproti původnímu rozhraní systému LISp-Miner doznala *Třída ekvivalence*, do níž šlo původně zařazovat pouze zadání literálů. Ve Ferdovi je možné navíc vytvářet už třídy ekvivalence atributů. To znamená, že všechna zadání literálů, která budou odvozena z těchto atributů jsou ve stejné třídě ekvivalence. V tomto příkladě jsou z krabičky *Odvozený sloupec* pojmenované *Salary – payments* odvozeny dva dynamické atributy typu *Equidistantní intervaly* pojmenované *Equidistant 1000* a *Equidistant 2000* (číslo v názvu označuje délku intervalů). Zřejmě nemá smysl, aby v cedentech vystupovali zároveň literály odvozené z obou těchto atributů, proto jsou oba atributy zařazeny do jedné třídy ekvivalence.

7 Porovnání s podobnými systémy

Porovnáme-li pohled na úlohu dobývání znalostí ve Ferdovi a v konkurenčních aplikacích – za všechny jmenujme například komerční produkty Clementine od SPSS a SAS Enterprise Miner, a nekomerční program Weka – ve všech těchto

aplikacích se pracuje s podobnými objekty jako jsou krabičky. Zatímco ve Ferdovi krabička reprezentuje funkce, v ostatních aplikacích je to spíše část procesu dobývání znalostí, proto pojem zásuvka jak jej o známe z Ferdy nemá v těchto aplikacích smysl. Jejich krabičky mají pouze jeden vstupní bod a veliké množství parametrů se nastavuje pomocí různých dialogů, to má za následek menší opakovatelnost použití a větší složitost aplikace z hlediska uživatele. Na druhou stranu zpracování úlohy ve Ferdovi vede k většímu množství typů krabiček, kterým by měl uživatel rozumět, i k většímu množství krabiček na pracovní ploše, které může vést k obtížnější orientaci uživatele. Tomu však byla věnována pozornost a byla vyvinuta řada mechanismů, jenž by měla tyto problémy řešit.

8 Závěr

Ačkoli prostředí Ferda je teprve v beta verzi, naše první praktické zkušenosti s ním přivedly uspokojivé výsledky. Vizuální prostředí Ferdy, které převažuje v moderních systémech na dobývání znalostí, integruje dříve roztříštěný postup v systému LISp-Miner. Možnosti pracovního prostředí popsané v tomto článku v čele s archivem krabiček, pracovní plochou nebo zabalováním a rozbalováním krabiček, pravděpodobně zjednoduší a zefektivní práci analytika při dobývání znalostí. Osvědčily se i nové nápady v čele s krabičkou *Skupina* a dynamickými atributy popsané ve shodně pojmenovaných částech tohoto článku, které rozšiřují záběr stávajícího systému.

Za největší výhodu prostředí však jeho tvůrci považují jeho modularitu a rozšiřitelnost. Například není příliš těžké vytvořit krabičku, která bude sekvenčně měnit nastavení mezí dynamických atributů a bude si tak sama simulovat data měnící se v čase. Otevírá se také prostor pro vytváření jednoduchých krabiček heuristicky simulujících práci analytika. Například jednoduché krabičky dynamicky upravující nastavení zadání koeficientů či kvantifikátorů úlohy tak, aby byly vlastnosti i počet výsledných hypotéz uspokojující pro uživatele.

V budoucnu by Ferda měl být rozšířen o nové procedury dobývání znalostí. Za zmínku stojí procedura RelMiner[5], která počítá asociační pravidla nad hvězdicovitým relačním schématem. Ferda je také vhodným kandidátem na prostředí pro projekt EverMiner[3].

Reference

1. J. Rauch, M. Šimůnek. Systém LISp-Miner *ZNALOSTI 2003, 2. ročník konference* Ostrava 2003. ISBN 80-248-0229-5.
2. Seznam publikací týkajících se systému LISp-Miner. http://lispminer.vse.cz/references.html.
3. J. Rauch, M. Šimůnek. *GUHA Method and Granular Computing.* Beijing 25.07.2005 - 27.07.2005. In: HU, Xiaohua, LIU, Qing, SKOWRON, Andrzej, LIN, Tsau Young, YAGER, Ronald R., ZANG, Bo (ed.). IEEE 2005. Piscataway: IEEE, 2005, s. 630-635.

4. P. Aubrecht, M. Kejkula, P. Kremen, L. Novakova, J. Rauch, M. Simunek, O. Stepankova. *Mining in Hepatitis Data by LISp-Miner and SumatraTT*. Accepted for publication in proceedings of Discovery Challenge 2005 see http://lisp.vse.cz/challenge/ecmlpkdd2005/chall2005.htm.
5. Martin Ducháček. Diplomová práce 2005 *Nástroj pro správu databází s využitím pro multi-relační DataMining*.
6. ZeroC. http://www.zeroc.com, Internet Communications Engine (Ice).
7. Mono Project. http://www.mono-project.com.
8. Clementine. http://www.spss.com/es/noticias/npclementine.htm. Clementine Data Mining System.
9. Weka. http://www.cs.waikato.ac.nz/∼ml/weka. Weka - Java Programs for Machine Learning, University of Waikato.
10. SAS Enterprise Miner. http://www.sas.com/technologies/analytics/datamining/miner,

Annotation:

Ferda, new visual environment for data mining

The goal of project Ferda was to create an application which will replace the current user environment for the data mining system LISp-Miner (lispminer.vse.cz) by a new, more visually and user friendly working environment. Ferda brings in the conception of data mining task as the record of the function using visual objects – boxes. Boxes have their parameters and evaluate their functions based on the values of the parameters. The program is modular and is designed to make the addition of new boxes easy. We have created the environment that organizes and simplifies user interaction with the LISp-Miner system and offers further possibilities for research and extensions.

Mining situations and actions from news

Luboš Popelínský and Jan Blaťák

KD Group, Faculty of Informatics, Masaryk University in Brno,
Botanická 68a, 602 00, Brno, Czech Republic
{popel,xblatak}@fi.muni.cz

Abstract. The definite goal of this project is building tools for automatic filtering news reports on flood, for extracting information from such reports and the use of this extracted information for finding important terms (single words, name/verb phrases, named entities etc.) and eventually building a domain ontology. In this paper we focus on classifying sentences that describe either a situation or an action by means of machine learning. The results of various learning algorithms are discussed. We also bring first results on term extraction.

Keywords: text filtering, information extraction, term extraction

1 Understanding news reports on flood

News reports on flood, like the example below,

> *In the Czech Republic the capital Prague is bracing for a major flood, just days after storms in the south of the country killed six people. "The forecast is bad," said Josef Novotny of the Prague crisis committee, warning that the Vltava river could burst its banks overnight. Floods affected some parts of Prague on Friday, but Mr Novotny said twice as much water was now bearing down on the city. Several southern towns are already cut off by water, and some have been evacuated. "Trains are not running, because bridges have fallen, and buses are not running, because roads are damaged," the mayor of the southern town of Prachatice, Jan Bauer, told Czech radio. Officials called on residents of the UNESCO-protected town of Cesky Krumlov – the second most popular tourist destination in the country – to leave.*
>
> *(Radio BBC Archive)*

usually contain two kinds of information. The first one concerns description of the current situation, the other describes an action performed, e.g. by an emergency unit. For instance the sentence

> *In the Czech Republic the capital Prague is bracing for a major flood, just days after storms in the south of the country killed six people.*

describes a situation whilst the sentence

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 130–138, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

Officials called on residents of the UNESCO-protected town of Cesky Krumlov – the second most popular tourist destination in the country – to leave.

an action. It is evident that a sentence (or more generally, a part of the message) can concern both, or be irrelevant. Then the goal of a classification can be defined as an assigning a label from the set {`SITUATION, ACTION, BOTH, IRRELEVANT`} to each part of the given news report. The class `BOTH` contains sentences that concern both the current situation and the action performed. Then the label `IRELEVANT` is assigned to all sentences that cannot be classified to none of these classes because the sentence brings no information relevant to a situation or to an action.

This work is the necessary step for fully understanding news reports. If we know that a sentence concerns, e.g., a situation, a goal of the next step is understanding this situation, e.g., learning the trend of the flood. Tools for information extraction can be employed here. This extracted information can be used directly for decision support. In this research project we aim also at building a set of terms (or ontology) for flood. We must not forget for the first step – information retrieval. To summarize, this research work can be split into four steps

1. ***News report filtering*** – finding news that are relevant
2. ***Finding text fragments*** – that describe either a situation or an action
3. ***Information and term extraction*** – extracting terms from these fragments that are important for situations or actions or extracting information in this fragment
4. ***Building domain ontology for flood*** – adding relations between these terms

In this paper we focus mainly on the second step, finding relevant fragments. In Section 2 we describe data used in this work. Section 3 brings an evidence that the learning set used is representative enough for discrimination between description of situations and actions. In Section 4 we bring results for sentence labeling. Term extraction is mentioned in Section 6. Plans for future work are discussed in Section 7.

2 Data

As a learning set we used the summary report on 10 days of flood in Central Europe in August 2001 [1]. For each day there are two paragraphs, one describing the situation in the region affected with flood and the other referring about actions performed. The part of the description of the first day of the flood follows.

9 August 2002
Situation
Unusually heavy rains falling over a broad area of Central Europe have resulted in widespread flooding. In Austria, Bulgaria, the Czech Republic and Romania the floods have been particularly severe. The weather forecast for the next few days threatens even more rain. A rain dense and very slow moving front is lingering over the area, heading towards the Black Sea
...
Actions
In Austria, the Red Cross has been working together with the fire brigade and the military to aid those affected by the floods. A 24 hour around the clock operation helped to ensure that those at risk were rescued. While efforts are continuing, it is believed that all of those who were in immediate danger have now been assisted. However, water levels remain dangerously high, with the risk of more rain at any moment. The Red Cross also organized mobile kitchens, providing hot food and drinks to those affected.

This report was manually collected from texts on web – BBC, CNN, France Press, Reuters, Deutsche Welle, The Associated Pres Situation reports of OCHA (United Nations Office for the Coordination of Humanitarian Affairs), ReliefWeb, Emergency appeals and reports of humanitarian organizations: Salvation Army, Red Cross, a report of ENVIS – the Prague Information System on the Environment and an event report of RMS – Risk Management Solutions, Inc.

For testing we retrieved all documents from BBC archive `http://www.bbc.co.uk/` from the same period that contained the word `flood`.

3 Learning discrimination between situations and actions

As preliminary step we checked how different the texts about situations and reports on actions actually were. We used the summary report [1] and built a

Table 1. Discrimination between situations and actions

	Accuracy
baseline	50.0%
Naive Bayes	91.4%
Decision tree	63.8%
Support Vector Machines	71.6%

learning set that contained 10 examples for each class represented as a bag of

words. Each example contained whole part of the text on the daily `SITUATION` or `ACTION`. As in further experiments we used Weka [6], namely Naive Bayes [4], decision tree J4.8 (an implementation of the Quinlan's decision tree algorithm C4.5 [5]) and SMO (Sequential Minimal Optimization Support Vector Machine) [3] and used boolean features – a feature was equal 1 if the corresponding word appeared in the text, otherwise was equal to 0. All results have been obtained with 10-fold cross validation.

Naive Bayes classifier displayed the best accuracy 91.4% (see Table 1) followed by support vector machines SMO and J48 decision tree. It means that the summary report contains enough information for learning the classifier.

4 Sentence labeling

Then we focused on the main task, recognition of two classes of sentences – those describing the situation and those referring about actions.

4.1 Method

For learning we used the same summary report [1] that was split into sentences. According to the appearance of a sentence in the report the sentence was classified into the class `SITUATION` or `ACTION`. As a test set we used news reports from BBC Archive. Each report was again split into sentences. We took all the 1777 words that appeared in the learning set as boolean features. Similarly as in the previous experiment, the value of a feature was set to 1 if the corresponding word appeared in the document, otherwise it was equal to 0. We employed again the Naive Bayes classifier, J48 and SMO.

4.2 Visualization

We implemented a system that enables easily interpret results of sentence classification. This system displays for each classified sentence the sentence itself, the class label assigned by the classifier and the confidence (estimated probability of the predicted class). If the value of the confidence is greater then a given threshold the sentence is printed in italics.

Example 1. Five examples of sentences and associated values obtained by utilizing the Naive Bayes classifier are listed bellow.

Sentence 1: In the firing line Algiers has grown at a breakneck pace in recent years, as the proportion of Algeria's population living in cities has surged from 31% in 1996 to 60% in 2000.

Classified as: `ACTION`
Confidence: 52.65%

Sentence 2: *The World Bank said one of its worries was that the frequent natural disasters were diverting resources away from long-term development needs to finance the recovery efforts.*

Classified as: `ACTION`
Confidence: 99.44%

Sentence 3: *Rain was still falling heavily and some rivers had reached danger levels, a provincial official said.*

Classified as: `SITUATION`
Confidence: 99.70%

Sentence 4: *Emergency teams have been sent to affected areas to help rescue efforts.*

Classified as: `ACTION`
Confidence: 99.99%

Sentence 5: Meanwhile more than 5,000 firefighters and forest rangers have been battling 10 forest fires in the Inner Mongolia region, five of which have been extinguished.

Classified as: `SITUATION`
Confidence: 69.68%

As we can see (sentences 1 and 5), the news in the test set contain also information which is not related to floods in Europe. Because of specificity of the learning set the confidence of the prediction is small in both cases. The sentences 3 and 4 which are clearly related to actions and situations were marked with right labels with very high confidence. The class `ACTION` was assigned to the sentence 2, although it is not clear whether describes action or situation.

5 Results

5.1 Classification

We analyzed first 50 sentences. 13 of them has been classified by human as `IRRELEVANT`, 5 sentences contained information `BOTH` about situation and about action. It mean that only 32 sentences has been clearly recognized by human. The results for this CLEARly classified sentences are in columns CLEAR in Table 2. For each learner, first column contains the number of examples correctly

Table 2. Accuracy of sentence labeling

	Naive Bayes			J48			SMO		
		ALL	CLEAR		ALL	CLEAR		ALL	CLEAR
TRUE	23	46%	71.9%	22	44%	68.7%	29	58%	90.6%
FALSE	9	18%	28.1%	10	20%	31.3%	3	6%	9.4%

(TRUE) or incorrectly classified. The second and third column brings accuracy (from all examples, from examples CLEARly recognized by human). Support Vector Machines SMO reached the highest accuracy 90.6%.

5.2 Exploiting the confidence

Because both Naive Bayes and J48 computes also a confidence of the label we explored also examples classified with high confidence. We again took first 50 examples, now with confidence higher than 95%. Results are in Table 3. It can

Table 3. Labels with a high confidence

	Naive Bayes			J48		
		ALL	CLEAR		ALL	CLEAR
TRUE	30	60%	78.9%	34	68%	80.1%
FALSE	8	16%	21.1%	8	16%	19.0%

be seen that for sentences classified with a high confidence accuracy increased from 71.9% to 78.9% for Naive Bayes and from 68.7% to 80.1% for J48. It must be said that in the case of J48 7 out of 8 false classifications concerned actions classified as situations.

5.3 Committee of classifiers

For dealing with examples classified with a low level of confidence we developed a tool which uses a committee of classifiers. We again employed the same triple of classifiers, J48, Naive Bayes and Support Vector Machines. This tool outputs a list of sentences which were not classified as belonging to the same class by all three classifiers. The result of this tool serves for easy assigning the correct label by hand.

Example 2. Committee of classifiers used for processing uncertainly classified sentence from Example 1:

Sentence 5: Meanwhile more than 5,000 firefighters and forest rangers have been battling 10 forest fires in the Inner Mongolia region, five of which have been extinguished.

NaiveBayes: `SITUATION` (conf. 69.68%)
SMO: `ACTION` (conf. 100%)
J48: `ACTION` (conf. 93.62%)

We can see that although this sentence is not related to floods the most confident label given by the committee of three classifiers is the right class label ACTION.

6 Term extraction

For finding important terms we used the feature selection method that explores the learned hypothesis – the decision tree in the case of J48, or parameters of the linear function learned with Support Vector Machine SMO. Among all the words we took in account only nouns and verbs.

The example of the decision tree is below. Conditions (e.g. `reported = 0`) are connected with conjunction written here as `|`.

Each path in a decision tree is then a condition or a conjunction of conditions. Each list in a decision tree contains an information about the number of learning examples covered by the path ending in this list. E.g `actions (40.0/1.0)` means that 40 examples from the class `ACTION` is covered and only 1 example from the class `SITUATION` is. For the decision tree below, the most expressive terms for the class `SITUATION` in this tree are words `reported` and `metres` because the conditions `reported=1` and `metres=1` holds for 8 and 6 learning examples.

```
J48 pruned tree
------------------

reported = 0
|   metres = 0
|   |   situation = 0
|   |   |   afternoon = 0
|   |   |   |   five = 0
|   |   |   |   |   moving = 0
|   |   |   |   |   |   greenpeace = 0
|   |   |   |   |   |   |   nationwide = 0
|   |   |   |   |   |   |   |   posed = 0
|   |   |   |   |   |   |   |   |   there = 0: actions (40.0/1.0)
|   |   |   |   |   |   |   |   |   there = 1: situations (4.0/1.0)
|   |   |   |   |   |   |   |   posed = 1: situations (2.0)
|   |   |   |   |   |   |   nationwide = 1: situations (2.0)
|   |   |   |   |   |   greenpeace = 1: situations (2.0)
|   |   |   |   |   moving = 1: situations (3.0)
|   |   |   |   five = 1: situations (3.0)
|   |   |   afternoon = 1: situations (3.0)
|   |   situation = 1: situations (5.0)
|   metres = 1: situations (6.0)
reported = 1: situations (8.0)

Number of Leaves  :     11

Size of the tree :      21
```

For better exploitation of knowledge in a decision tree we first define a *strong path* as a path from the root to a leaf that hold for at least N examples. Here N is equal to 4. Then for the class `SITUATION`, the most significant words, i.e. those that appeared on a strong path in the decision tree learned with J48, are

length units – `metres, centimetres`
situation
concerning rain – `fall, down, rain`, and
words with no specific meaning – `reported, braced, parts`

```
|  |  |  |  |  |  |  |  |  |  |  |  |  meters = 1: situations (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  greenpeace = 1: situations (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  expect = 1: situations (3.0)
|  |  |  |  |  |  |  |  |  |  |  rains = 1: situations (4.0)
|  |  |  |  |  |  |  |  |  |  parts = 1: situations (4.0)
|  |  |  |  |  |  |  |  |  particularly = 1: situations (4.0)
|  |  |  |  |  |  |  |  five = 1: situations (4.0)
|  |  |  |  |  |  |  fall = 1: situations (4.0)
|  |  |  |  |  |  down = 1: situations (4.0)
|  |  |  |  |  braced = 1: situations (4.0)
|  |  |  |  centimetres = 1: situations (5.0)
|  |  |  reported = 1: situations (9.0)
|  |  situation = 1: situations (11.0)
|  metres = 1: situations (16.0)
```

When analyzing the linear function learned with SVM, we take all words that have coefficient greater or equal to 0.5 in absolute value. For the class `ACTION` the following terms are supposed to be important:

evacuation* – `evacuations`, `evacuation`
other nouns – `hospitals`, `soldiers`, `pumps`, `tunnels`

For the class `SITUATION` they are

```
camps
expect
```

The linear function is below. The coefficient has been ordered.

```
-0.7024 hospitals          0.5453 camps
-0.6461 soldiers           0.5374 particularly
-0.6048 evacuations        0.5142 expect
-0.5583 pumps              0.4586 summer
-0.5461 evacuation         0.4515 flooded
-0.5368 tunnels            0.4365 situation
-0.4996 work               0.4338 reported
-0.4809 emergency          0.4325 town
...                        0.4246 fall
-0.3951 evacuated          ...
-0.3936 boats              0.3848 rains
-0.3713 firefighters       ...
-0.3659 memory
-0.3659 living
-0.3594 pumping
```

7 Conclusion and future work

In future we plan to use also feature selection methods. We already performed first experiments. Similarly as in Section 6 we used the report [1] and assigned each part on situation (action) the right label. Then we employed feature selection based on information gain (IG) heuristics [2] and took only 350 words with the highest IG measure. The winner was Naive Bayes classifier with 100% accuracy (10-fold cross validation), followed by SMO (95%) and J48 (70%).

It looks necessary to split text into fragments even shorter than one sentence. sentence splitting and shallow parsing will be employed for finding clauses and name/verb phrases.

For obtaining more reliable term extraction method we will use – besides lemmatization and synonyms recognition – also ensemble-based methods for feature selection.

Acknowledgement.

We thanks Natalia Andrienko for providing the report on flood. This work has been partially supported by the Faculty of Informatics, Masaryk University in Brno and by the Grant Agency of the Czech Republic under the Grant No. MSM0021622418 Dynamic Geo-visualization in Crisis Management.

References

1. Andrienko, N., A report on flood in Central Europe 2001. Manuscript
2. Forman, G. Choose your words carefully. In Elomaa, T., Mannila, H., and Toivonen, H., editors, *Proceedings of PKDD 2002, Helsinki, Finland, August 19–23, 2002*, LNCS, volume 2431
3. Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., and Murthy, K. R. K. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.
4. Mitchell, T. M. *Machine Learning*. McGraw Hill, 1997.
5. Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
6. Witten, I. H. and Frank, E, Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufman 1999

Báze znalostí a poskytování informací obsluze linky technické podpory

Vladimír Sklenář, Martin Radvanský

Katedra informatiky, PřF, Univerzita Palackého Olomouc,
Tomkova 40, 779 00, Olomouc-Hejčín

Sklenar@inf.upol.cz, Radvansky@radvansky.net

Abstrakt. Pracovníci, kteří odpovídají zákazníkům na lince technické podpory (Hotline), často potřebují informace o tom, zda právě řešený (nebo analogický) problém již nebyl někdy dříve řešen a popsán. Rychlé vyhledání analogických problémů a jejich řešení může mít značný význam zejména pro méně zkušené nebo zastupující pracovníky. Ne vždy jsou také jasné všechny charakteristiky hlášeného problému a je nutné potřebné informace od zákazníka získat dalším dotazováním. V tomto příspěvku popíšeme systém umožňující vyhledávání dříve zaznamenaných problémů, které vykazují podobné charakteristiky jako právě hlášený problém a poskytující návod na dotazy pro zákazníka, které umožní zpřesnit a doplnit informace o hlášeném problému. Teoretickým základem systému je formální konceptuální analýza.

Klíčová slova: FCA, hotline, kontext

1 Úvod

Linka technické podpory software (hotline) má specifickou úlohu při podpoře aplikace. Pro její správnou a pro uživatele přínosnou funkci je nutné, aby uživatel dostal pokud možno co nejpřesnější odpovědi na své dotazy. Pro uspokojivou odpověď je především nutné správně identifikovat problém na základě informací, které zákazník poskytne. Na typické lince technické podpory tedy sedí pracovníci, jejichž znalosti musí být na vysoké úrovni nejen po stránce podporovaného produktu, ale hlavně musí být schopni získat potřebné informace od zákazníka, zobecňovat jednotlivé hlášené problémy a nacházet jejich řešení. Díky těmto specifikům je vyškolení pracovníka (operátora) pro technickou podporu náročný a časově dlouhodobý úkol. Nemalým problémem je také zastupitelnost těchto pracovníků. Situaci by mohl usnadnit informační systém, který by shromažďoval informace o jednotlivých problémech (incidentech) a jejich řešeních a napomáhal pracovníkům na lince technické podpory správně identifikovat hlášený problém.

2 Potřeby operátorů na lince technické podpory

Požadavky, které jsou kladeny na takový systém lze shrnout do několika bodů:

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 139–150, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

- **Rychlé a jednotné zaznamenávání charakteristik nahlášených incidentů**
  Pro pracovníka na lince technické podpory je důležité, aby zaznamenání charakteristik hlášeného incidentu proběhlo v co nejkratším čase. Je třeba si uvědomit, že převážná část komunikace se zákazníkem se děje on-line tedy za přímého hovoru a jakékoliv zdržení při záznamenávání problému nebo vyhledávání není možné, protože operátor neustále komunikuje se zákazníkem a upřesňuje problém. Způsob zaznamenávání by měl zajistit, že různí operátoři zaznamenají analogické incidenty různým způsobem.

- **Rychlé nalezení analogických incidentů a jejich řešení**
  Množství hlášených incidentů narůstá s počtem uživatelů produktů a je tedy časem vytvořena rozsáhlá znalostní databáze obsahující hlášené incidenty opatřené jejich charakteristikami a řešením. Některé problémy se týkají více zákazníků, jiné se v určitých obdobích opakují (při přechodu na novou verzi, roční uzávěrky, atd.). Je proto pravděpodobné, že právě řešený (nebo analogický) problém již byl někdy v minulosti zaznamenán a v databázi tedy existuje záznam obsahující jeho řešení. Opět je nutné, aby takové záznamy byly nalezeny v co nejkratším čase a s co nejmenšími nároky na obsluhu.

- **Pomoc při analýze incidentu**
  V případě, že linka technické podpory je poskytována pro více produktů, je na oddělení technické podpory více pracovníků, kteří se věnují převážně jednomu produktu. Pokud je takto specializovaný pracovník zastupován kolegou, který se věnuje jinému produktu, je poskytovaná služba degradována omezenými znalostmi zastupujícího operátora a řešení incidentu se může zbytečně protahovat. To má negativní vliv na zákazníka.

- **Nabízení charakteristik podle již vyřešených incidentů**
  Systém by měl navrhovat na základě doposud zjištěných charakteristik problému další postup dotazování zákazníka vedoucí k získání přesnějších informací o hlášeném incidentu. Návrh by měl být založen na podobnosti hlášeného incidentu s dříve zaznamenanými problémy. Operátor by získal seznam vlastností (charakteristik), které se u analogických problémů vyskytly. Je třeba, aby tyto charakteristiky byly nějakým způsobem seřazeny pro jejich snadné nalezení a použití. Nejčastěji používaným pohledem je seřazení navržených charakteristik podle jejich četnosti v incidentech. Dalším pohledem může být priorita jednotlivých charakteristik, protože charakteristiky jsou samozřejmě méně či více důležité v daném kontextu incidentu.

- **Vyhodnocení a analýza práce na lince technické podpory**
  Samozřejmostí je možnost získávání souhrnných informací o práci na lince technické podpory a to jak z pohledu vykonané práce jednotlivými pracovníky, tak i z pohledu incidentů. To znamená mimo jiné zjištění nejčastěji opakovaných problémů, závislostí mezi vlastnostmi incidentů a dalších informací umožňujících získat celkový přehled o stavu produktu.

- **Schopnost samospřesňování (vylepšování dříve uložených informací)**
  Znalostní databáze je periodicky procházena operátory a vývojáři, kteří řeší dosud nevyřešené incidenty. V případě, že při hlášení incidentu nebo jeho řešení je nalezena charakteristická vlastnost spojená s incidentem, která není u doposud zaznamenaných incidentech uvedena, je třeba ji doplnit do systému a promítnout do všech analogických incidentů.

- **Nalezení stejných často se opakujících incidentů**
  Celá řada problémů se na lince technické podpory opakuje. Je proto vhodné nalezení skupin incidentů spojených se stejným problémem a vytvoření tzv. typizovaného řešení.

2.1 Možné přístupy k řešení

Protože přínos informačního systému podporujícího linku technické podpory pro zkvalitnění její práce je zřejmý, byla již dříve vytvořena aplikace založená na fulltextovém přístupu. Každý operátor v okamžiku incidentu zaznamenával v textové podobě dotaz zákazníka a odpověď byla vyhledávána buď z často se opakujících incidentů, nebo z již vyřešených incidentů. Vyhledání bylo provedeno na základě fulltextového dotazu. Tato metoda byla v případě malého množství incidentů dostačující. V okamžiku, kdy množství incidentů začalo narůstat, objevil se problém s rychlostí pořizování dat a s množstvím vyhledaných řešení. Dalším problémem v takto pojatém systému byla zastupitelnost operátora. Systém založený na fulltextovém vyhledávání je náročný na dodržování stejných postupů jak při zadávání incidentů, tak při získávání odpovědí. Každý z operátorů má jiný slovník a problém analyzuje a popisuje svým specifickým způsobem. Docházelo tedy k narůstání počtu shodných incidentů, které byly ovšem jinak formálně popsány. V důsledku to pak znamenalo, že se nepovedlo najít řešení hlášeného incidentu i přesto, že analogické incidenty byly již dříve zaznamenány (ale jinou formou). Doba, za kterou zákazník dostal odpověď na svůj dotaz se tak prodlužovala. Z výše uvedeného vyplynulo, že **použití fulltextového vyhledávání není pro tento typ aplikace vhodným řešením** a je tedy potřeba nalézt jiný mechanizmus nacházení analogických problémů.
Další metodou, která se používá, je case-based reasoning (CBR), viz [1]. CBR využívá znalostí získaných při zkoumání předchozích konkrétních problémových situací. Nový problém je řešen nalezením podobného dříve zaznamenaného problému a znovupoužitím jeho řešení v nové problémové situaci.
My jsme pro splnění našich požadavků na funkčnost systému použili formální konceptuální analýzu (FCA). V další kapitole zavedeme základní pojmy spojené s touto metodou a poté ukážeme způsob jejího použití.

3 Formální konceptuální analýza

3.1 Základní pojmy

Formální konceptuální analýza pracuje se vstupními daty ve formě tabulky s řádky, které reprezentují objekty (v našem případě hlášené incidenty a jejich řešení) a sloupci, které reprezentují jejich atributy (vlastnosti, charakteristiky). V tabulce je zaznamenán vztah mezi jednotlivými objekty a atributy. Objekt příslušný atribut buď obsahuje, nebo neobsahuje. Tabulka je formálně reprezentována tzv. *formálním kontextem*, což je trojice $\langle X,Y,I\rangle$, kde X je množina objektů, Y je množina atributů a I je binární relace mezi X a Y. Fakt, že $\langle x,y\rangle \in I$ znamená, že objekt x má atribut y. Pro každou množinu objektů $A \subseteq X$ a množinu atributů $B \subseteq Y$ značíme pomocí $A^{\uparrow}$ a $B^{\downarrow}$ množiny

$$A^{\uparrow} = \{ y \mid \text{pro každé } x \in A : \langle x,y\rangle \in I \}, \qquad B^{\downarrow} = \{ x \mid \text{pro každé } y \in B: \langle x,y\rangle \in I \}.$$

Jinak řečeno, $A^{\uparrow}$ je množina všech atributů z Y, které jsou společné všem objektům z A. Podobně $B^{\downarrow}$ je množina všech objektů z X, které společně sdílí všechny atributy z B. *Formální koncept* v $\langle X,Y,I\rangle$ je dvojice $\langle A,B\rangle$, kde $A \subseteq X$ a $B \subseteq Y$, pro níž platí, že $A^{\uparrow} = B$ a $B^{\downarrow} = A$. Formální koncept se tedy skládá z množiny objektů A, kterou nazýváme *extent* a množiny atributů B, kterou nazýváme *intent*, pro než platí, že A obsahuje všechny objekty, které mají všechny atributy z B a B obsahuje všechny atributy. které se společně vyskytují u všech objektů z A.

Na množině $\beta(X,Y,I) = \{\langle A,B\rangle \mid A^{\uparrow} = B, B^{\downarrow} = A\}$ všech formálních konceptů můžeme definovat relaci uspořádání $\leq$ pro níž platí

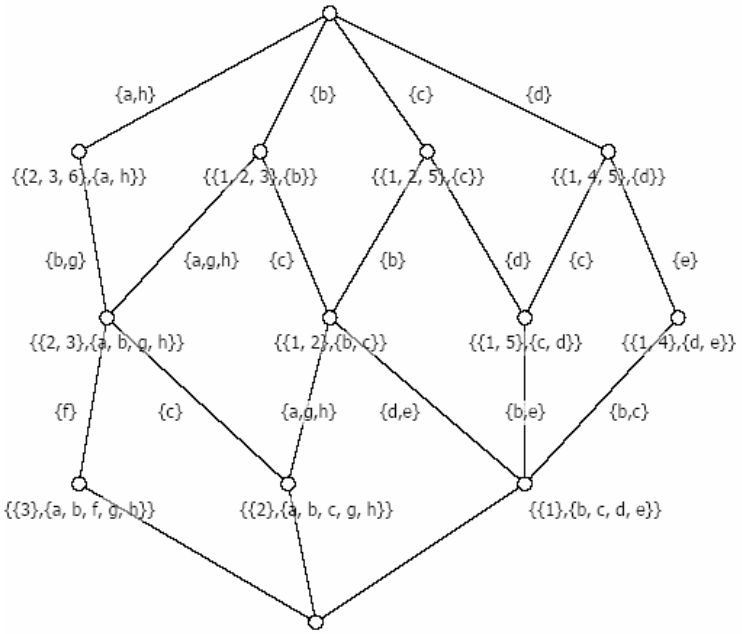
$\langle A_1,B_1\rangle \leq \langle A_2,B_2\rangle$ iff $A_1 \subseteq A_2$ (nebo, ekvivalentně, $B_2 \subseteq B_1$).

Fakt, že $\langle A_1,B_1\rangle \leq \langle A_2,B_2\rangle$ znamená, že každý objekt z A_1 je obsažen také v A_2 (nebo ekvivalentně, každý atribut z B_2 je obsažen v B_1). Relace $\leq$ tedy vyjadřuje hierarchii mezi koncepty (vztah subkoncept – superkoncept).
Množina všech formálních konceptů $\beta(X,Y,I)$ spolu s relací $\leq$ tvoří úplný svaz, který nazýváme *konceptuální svaz*. Podrobněji viz [2] [3].
Pro názornost uvedeme příklad kontextu, který obsahuje 6 objektů a 8 atributů. Relace incidence mezi objekty a atributy je zaznamenána v tabulce 1.

Tabulka 1. Kontext

	a	b	c	d	e	f	g	h
1		x	x	x	x			
2	x	x	x				x	x
3	x					x	x	x
4				x	x			
5			x	x				
6	x							x

Na obrázku 1 je pak zobrazen pomocí Haseova diagramu odpovídající konceptuální svaz. Jeho uzly reprezentují jednotlivé koncepty (popisek obsahuje výčet položek exentu i intentu). Hrany reprezentují uspořádání a jejich popisek obsahuje atributy, které při pohybu po příslušné hraně přibudou (ubudou) v intentu.



Obr. 1. Konceptuální svaz

3.2 Aplikování

Formální konceptuální analýza poskytuje aparát, který umožňuje řešit většinu z požadavků uvedených v kapitole 2. Jako objekty budou vystupovat jednotlivé incidenty a řešení. Prvotním krokem při jejím nasazení je volba atributů (charakteristik), které se budou přidělovat jednotlivým hlášeným incidentům. Jejich stanovení má podstatný vliv na úspěšnost použití. Je proto nutné, aby jejich volbu provedli experti, a aby i případné zařazení dalších atributů podléhalo určitým pravidlům. Ty by měly zabránit zadání dvou atributů s různým názvem, ale se stejným významem. Ukázalo se, že pro zkušené pracovníky to nebyl zásadní problém. Atributy tvoří stromovou hierarchii. Zaznamenávání charakteristik hlášeného problému je pak možné realizovat pouhým výběrem z nabídky všech existujících atributů. Podstatně je tak omezena možnost toho, že různí operátoři popíší analogické problémy různým způsobem.

Problém nalezení analogických incidentů je pak vlastně problém nalezení nejmenšího konceptu, který obsahuje všechny doposud zadané atributy. To znamená, že k dané množině B všech doposud zadaných atributů nalezneme koncept $\langle B^{\uparrow}, B^{\uparrow\downarrow}\rangle$. Množina

$B^{\uparrow}$ obsahuje incidenty, které mají všechny zadané atributy. Tato množina může být při malém počtu atributů poměrně rozsáhlá. Postupným přidáváním atributů, které hlášený problém charakterizují se její mohutnost zmenšuje. V podstatě se v konceptuálním svazu pohybujeme od vrcholu směrem dolů. Po zadání všech atributů pak $B^{\uparrow}$ obsahuje pouze ty incidenty, které jsou s hlášeným problémem analogické nebo je to prázdná množina, což znamená, že se jedná o nový problém, který doposud není zaznamenaný. Pro $B^{\uparrow\downarrow}$ platí vztah $B \subseteq B^{\uparrow\downarrow}$. Může nastat situace, že $B \neq B^{\uparrow\downarrow}$. Pak to znamená, že u doposud zaznamenaných incidentů jsou společně se zadanými atributy vždy uvedeny také všechny atributy patřící do množiny ($B^{\uparrow\downarrow}$ - B). Pokud operátor usoudí (případně po konzultaci se zákazníkem), že hlášený problém tyto atributy také má, může je převzít a urychlit tak zadávání charakteristik hlášeného problému. Operátor má k dispozici i opačný postup. Je-li mu po zadání několika málo atributů jasné o jaký problém jde, může zvolit jeden z nabízených zaznamenaných incidentů. Všechny jeho atributy se pak automaticky přiřadí k hlášenému problému a ten by pak měl být úplně popsán. Pokud se přesto doplní další atribut, doplní se také k vybraným incidentům v databázi. Tyto operace zvyšují přesnost informací v databázi.
Nalezený nejmenší koncept $\langle B^{\uparrow}, B^{\uparrow\downarrow}\rangle$. určuje podsvaz. Při dalším doplňování charakteristik incidentu lze přejít pouze k některému konceptu v tomto podsvazu. Předpokládáme-li, že v bázi dat již nějaký analogický incident existuje, pak pro jeho další popis lze použít pouze ty atributy, které se v podsvazu vyskytují. Tyto "možné" atributy patří do sjednocení množin atributů přiřazených jednotlivým objektům v $B^{\uparrow}$. Pokud tedy ve výše uvedeném příkladu zadáme atribut {a} získáme koncept $\langle\{2,3,6\},\{a,h\}\rangle$. Nyní je možné přidat atributy b, c, f nebo g. Přidání atributu d nebo e by znamenalo, že se jedná o nový problém, který doposud není zaznamenaný.

4 Realizace řešení

S využitím aparátu formální konceptuální analýzy byl vytvořen informační systém Hotline, který plní většinu požadavků uvedených v kapitole 2. Systém byl vytvořen na platformě .NET framework, je založen na datovém úložišti MS SQL Serveru 2000 a je navržen v třívrstvé architektuře.

Jednotlivé požadavky na systém jsou z hlediska operátora realizovány následovně:

- **Rychlé a jednotné zaznamenávání charakteristik nahlášených incidentů**
  Operátor vybírá charakteristické vlastnosti z nabízeného stromu dostupných charakteristik(atributů), který byl naplněn operátory při zavedení systému a který lze na základě nově zjištěných charakteristik průběžně doplňovat.

- **Rychlé nalezení analogických incidentů a jejich řešení**
  Po zadání odpovídajících charakteristik systém nabízí obsluze dva samostatné seznamy s možnými řešeními incidentu. Je to seznam již vyřešených incidentů, které mají všechny doposud zadané charakteristiky a

seznam typizovaných řešení situací odpovídajících zadaným charekteristikám. Operátor může k právě hlášenému incidentu přiřadit řešení z nabízených incidentů nebo řešení se shodnými charakteristikami.

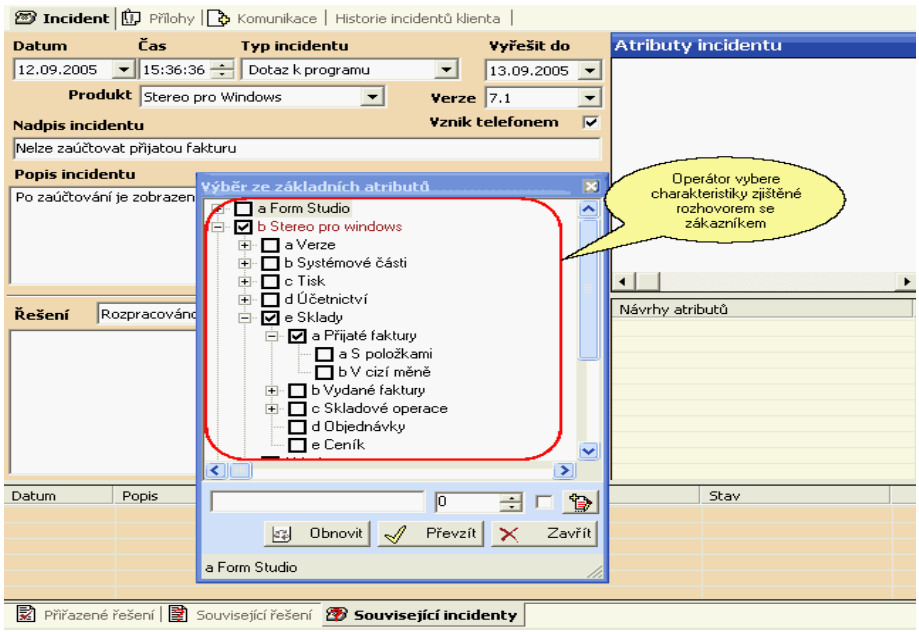
- **Pomoc při analýze incidentu srovnáním s existujícími a nabízení charakteristik podle již vyřešených incidentů**
  Systém reaguje na postupné zadávání charakteristik zobrazením seznamu s dalšími charakteristikami, které se u dříve zaznamenaných incidentů a řešení vyskytují společně s již zadanými charakteristikami. Operátor tedy může při komunikaci se zákazníkem využít těchto navržených charakteristik a ověřit, zda nejsou s incidentem také spojeny. Tímto způsobem je realizováno navádění obsluhy k pokládání cílených dotazů směřujících k získání dalších charakteristik a tím k zpřesnění informací o hlášeném problému. Vzhledem k tomu, že konceptuální svaz tvoří graf není důležité pořadí, ve kterém se charakteristiky zadávají.

- **Vyhodnocení a analýza práce na lince technické podpory**
  Systém podporuje poskytování informací o počtu a struktuře incidentů spojených s vybranými charakteristikami (např. problémy s tiském, problémy spojené s konkrétní verzí produktu, …). Z pohledu FCA jde vlastně o výběr podvazu. Dále je v systému zabudován mechanismus vyhledání často se opakujících incidentů. Odpovědný operátor (vývojář) k nim pak může vytvořit typizované řešení.

- **Schopnost samospřesňování (vylepšování dříve uložených informací)**
  V okamžiku, kdy operátor vybere jako řešení rozpracovaného incidentu nabízené nebo opakované řešení, jsou systémem porovnány doposud zaznamenané charakteristiky hlášeného incidentu a vybraného řešení a je nabídnuta synchronizace. Operátor tedy může provádět zpřesnění charakteristik u incidentu podle vybraného řešení nebo naopak zpřesnit řešení o nově zjištěné charakteristiky. Při doplnění charakteristik, u incidentů, které již mají přiděleno řešení je možné promítnutí těchto charakteristik do řešení. Tato metoda je využívána zejména v případě, kdy vývojáři zjistí charakteristiku incidentu, která nebyla operátorem zadána, přesto hraje důležitou úlohu při rozpoznání incidentu.

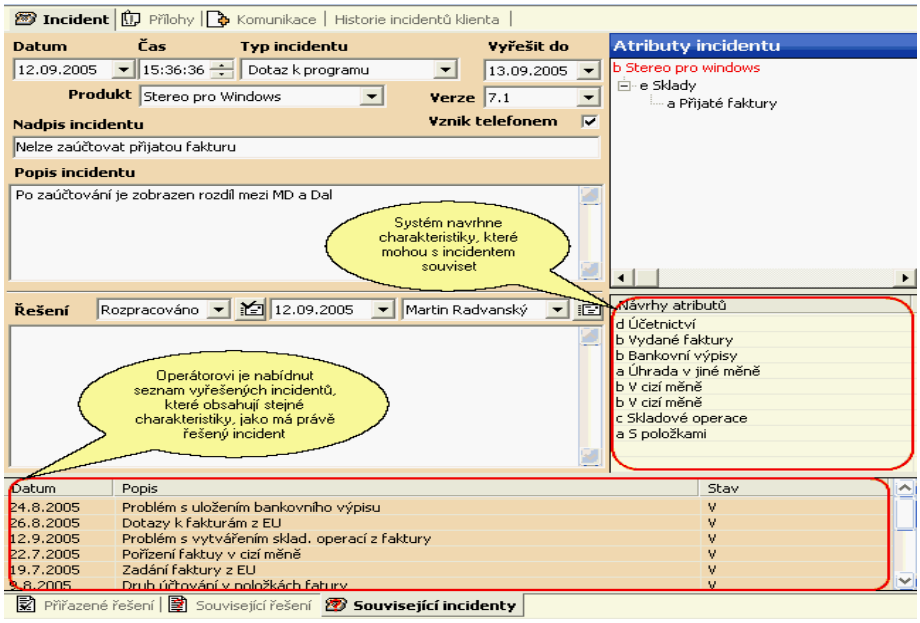
4.1 Ukázka funkce systému

1. Po přijetí hovoru, operátor na lince technické podpory vybere rozpoznané charakteristiky zaznamenávaného incidentu z nabídky dostupných charakteristik obsažených v systému.



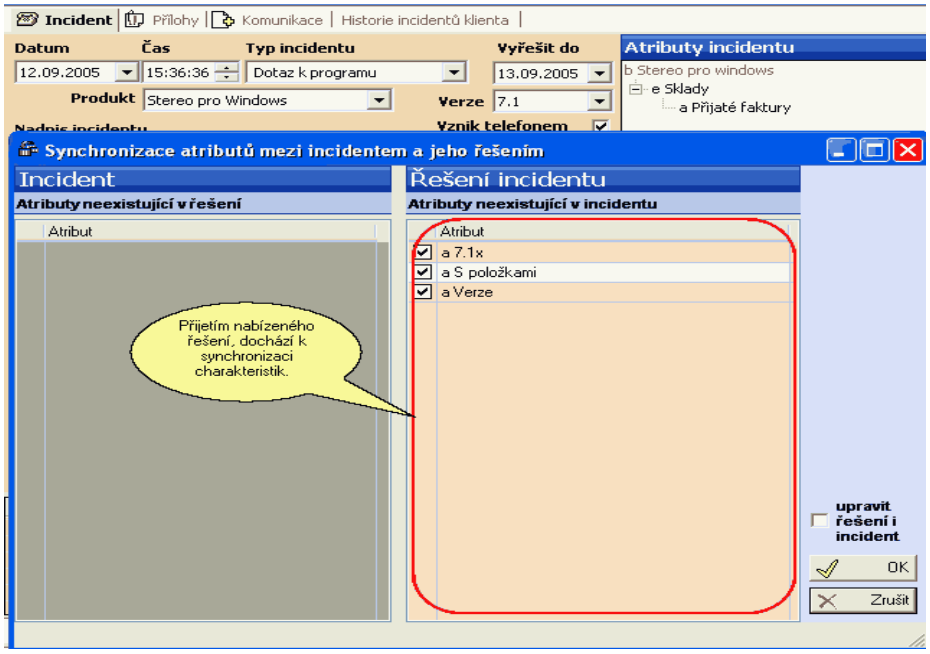
Obr. 3 Výběr vlastností incidentu

2. Systém navrhne charakteristiky, které se vyskytují společně s zadanými charakteristikami. Dále je operátorovi nabídnuta množina dříve vyřešených incidentů, které obsahují zadané charakteristiky.



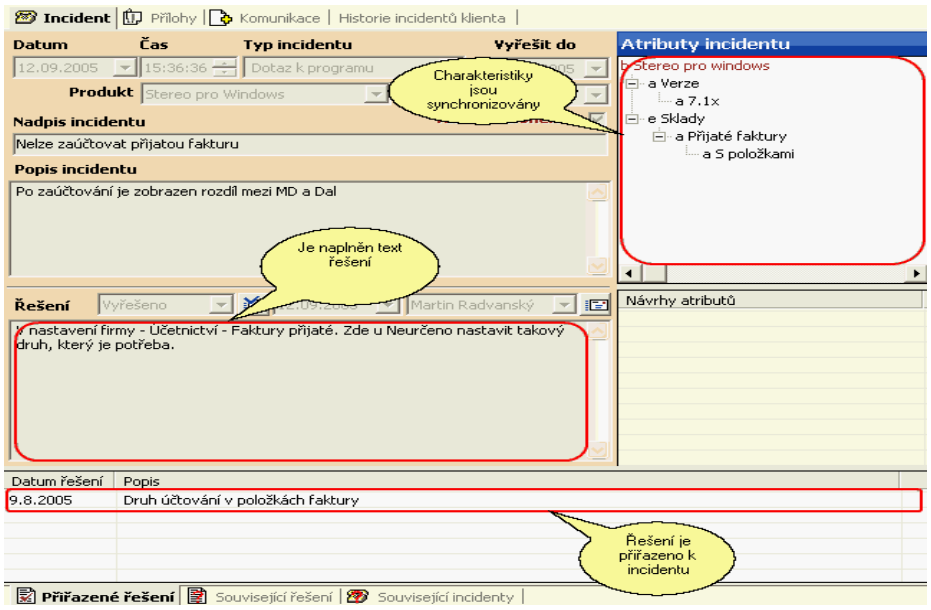
Obr. 4 Systém navrhne operátorovi další postup

3. Pokud operátor rozpozná analogický dříve zaznamenaný incident, přiřadí ho jako řešení aktuálně zpracovávaného incidentu. Systém reaguje vyvoláním synchronizace charakteristik. Tzn., navrhne doplnění charakteristik k incidentu podle vybraného řešení. V případě, že by naopak řešený incident obsahoval charakteristiku, která není uvedena u vybraného vyřešeného incidentu bude nabídnuta opačná synchronizace, tj. přidání nové charakteristiky k zaznamenaným incidentům a řešením.



Obr. 5 Synchronizace atributů incidentu

4. Po synchronizaci je doplněn text řešení a vybraný incident je přiřazen jako řešení k rozpracovanému incidentu. V tomto okamžiku může operátor předat řešení zákazníkovi a ukončit práci s incidentem.



Obr. 6. Přiřazení řešení k incidentu

4.2 Zkušenosti z nasazení aplikace

Pro optimální funkčnost aplikace je stěžejním úkolem vytvoření sady atributů, které budou používány při výběru charakteristik incidentů. Proto je třeba jejich počátečnímu nastavení věnovat dostatek pozornosti. Další charakteristiky, které jsou doplňovány operátory v průběhu používání systému jsou zpřesněním a rozšířením charakteristik vytvořených při zavádění systému. Charakteristiky jsou dostatečně popisné a je možné problémy zařazovat od určitých souvisejících kategorií. Umožňují tedy práci se systémem i zastupujícím operátorům, kteří jsou schopni s jejich pomocí nalézt řešení k většině incidentů nebo alespoň dostatečně popsat situaci spojenou s hlášeným incidentem a předat jej pak kompetentním osobám k řešení. Při práci se systémem vyvstala potřeba setřídění navrhovaných charakteristik podle četnosti výskytu a také podle přímých následníků v konceptuálním svazu z důvodu nárůstu množství charakteristik.

Systém je nasazen do reálného provozu od konce července 2005 a plní svou úlohu podle očekávání a požadavků na něj kladených. Se systémem pracuje 8 hodin denně současně 6 uživatelů, z čehož jsou 2 pracovníci linky technické podpory, kteří jsou v přímém kontaktu se zákazníky, 3 pracovníci vývojového týmu a jeden manažer produktů. Systémem jsou podporovány dva softwarové produkty. Tyto produkty jsou licencovány celkem pro 7000 uživatelů. Jelikož se jedná o ekonomické a daňové produkty, je hlavní nápor na linku technické podpory očekáván v období od března do června. V současné době je v systému kolem 900 incidentů, přičemž denně je do systému zařazeno 10 až 20 nových incidentů. Počet atributů zavedených do systému je 160. Počet nově zaváděných atributů je průměrně 2 - 3 atributy týdně. S vyšší frekvencí dotazů koncem účetního období se dá očekávat nárůst incidentů 3 až 5 krát

větší. Do systému samozřejmě nejsou zaváděny úplně všechny incidenty z linky technické podpory, protože se na linku telefonické podpory obracejí uživatelé i s problémy, které nejsou takového charakteru, aby je bylo třeba zaznamenávat.
Zavedením systému byla podstatně zefektivněna práce na lince technické podpory a bylo umožněno vyhledání a vyhodnocování jednotlivých skupin incidentů, čímž je možné soustředit se při následujících verzích na konkrétní problémové části podporovaných aplikací. Pro porovnání s předcházejícím systémem využívající fulltextového vyhledávání, je použitá metoda nejen přesnější v nalezení relevantních informací, ale zároveň je omezen dopad odlišného vyjadřování a popisu problému různými operátory.
Na základě těchto zkušeností můžeme tedy konstatovat, že formální konceptuální analýza se ukázala jako funkční metoda pro použití v tomto specifickém informačním systému.

5 Další rozvoj

Aplikace se samozřejmě neustále vyvíjí tyto změny probíhají jak na úrovni aplikační, tak i v možnostech výkonného jádra FCA. V současné době se připravují následující rozšíření funkčnosti.

- Přímá vazba na firemní informační systém, pro sdílení informací o klientech.
- Vytvoření webového prezentačního klienta tak, aby si zákazníci mohli požadované řešení nacházet sami.
- Automatizované zpracování došlých emailů na linku technické podpory.
- Doplnění přehledových sestav pro manažerské vyhodnocení, ty pak rozšiřovat podle aktuálních požadavků.
- Provázání na možnost vyhledání zákazníka podle telefonního čísla získaného přes ISDN linku.
- Větší optimalizace klientské části aplikace pro co nejefektivnější zadávání údajů.

Poděkování. Tato práce byla částečně podporována z grantů 1ET101370417 GA AV ČR a 201/05/0079 GAČR

Reference

1. Aamondt, A., Plaza, E. : CASE-Based Reasoning: Foundational Issues, Methodological Variations, and Systém Approaches In: AICom, Vol. 7: 1, pp. 35-59, 1994
2. Carpineto, C.. Romano, G..*Concept Data Analysis. Theory and Application*. Wiley, 2004

3. Ganter, B. Wille, R..*Formal Concept Analysis. Mathematical Foundations*. Springer-Verlag, Berlin, 1999

Annotation:

Knowledge base and information support for hotline

People that answer to customers on hotline often need information whether given problem was resolved in the past. Fast finding analogical problems is very important particularly for less experience operators. In this paper is described information system that helps operators find solution of problem. Theoretical base of system is Formal Concept Analysis

Refinement of case retrieval for Grid service performance prediction through semantic description of input data

Zoltán Balogh, Emil Gatial, Michal Laclavík, Ladislav Hluchý

Institute of Informatics, Slovak Academy of Sciences,
Dubravska cesta 9, 84507 Bratislava,

{balogh,gatial,laclavik,hluchy}.ui@savba.sk

Abstract. The article deals with performance prediction of Grid services. The performance of Grid services is estimated based on the past cases. Authors have identified that in many applications the description of Grid service input data should be considered as important part of the feature vector. Authors propose the refinement of case retrieval process through semantic description of discrete features and service input data. Instance based learning is used to estimate the performance of Grid services.

1 Introduction

This article deals with prediction of runtimes of web services (WS). Estimating WS runtime is particularly critical during construction of composite WS workflows. Each workflow job must be scheduled in a way that the overall workflow run time will satisfy the overall workflow constraints. Such workflows are commonly used in Grids for connecting individual Grid WS to large, complicated and distributed applications. Prediction of WS run times optimizes scheduling and supports efficient use of grid resources. We estimate expected WS run time based on invocation parameters of WS operations and properties of data resources used as processing inputs for a WS. We adopt an instance-based learning (IBL) technique, where the history of WS operations is examined for each class and instance of WS. Such run time prediction is used by workflow schedulers to compute expected run times of a range of WS for the purpose of identifying the most appropriate WS for a given job within given constraints.

Several works have already dealt with problem of selecting where to run an application on the many resources available in the Grid. A great overview of the problem domain is given in [1] by Warren Smith. Two problems are identified in the area of Grid scheduling which can be addressed using predictions: selecting where to run the application on the many resources available in the Grid and gaining simultaneous access to multiple computer systems so that distributed applications can be executed. Overview is also given about the approach of prediction of application run times and queue wait time predictions. Application run time is suggested to be solved by categorization prediction technique or instance-based learning approach (also called Lo-

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 151–158, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

cally Weighted Learning techniques). In [1] they call an experience base the set of all observations. They use Euclidean Overlap Metric to compute the distances between the query and all the relevant observations, whereas it is mentioned that there are many distance functions which might be used. They use distance-weighted average to compute the output feature, i.e. the expected result.

In [2] Kapadia et al. exploit and evaluate three local learning algorithms: nearest-neighbrhor, weighted-average and locally weighted polynomial regression for the purpose of predictive application-performance modeling in Grid. In [2] authors also deal with normalization of features and selection of parameters. The performance of learning algorithms is compared using prediction error and prediction time. The important and for us relevant conclusions of work [2] is that the nearest neighbor algorithm outperforms the more sophisticated locally weighted regression algorithm.

All IBL approaches mentioned in [1] and [2] in fact use a well known methodology called Case Based Reasoning (CBR). All phases of CBR [3] such as case representation, case indexing, case retrieval, case adaptation and case-base maintenance are required to be present in any IBL approach, whereas instance and case are synonyms for a single invocation of an application.

An insufficiency of the above mentioned approaches is that they do not deal particularly with web service performance prediction while WS lifecycle [4] definitely has its particularities, which must be considered for monitoring, processing and estimation of WS results. Another insufficiency is handling of discrete features. No measure of discrete parameter similarity is provided in the current IBL approaches. The last major drawback of existing approaches is unsuitable representation of cases, which neither allow any major extensions nor even simple inference above information stored in a case base. By case extension we mean for example extending the result (i.e. output feature) of a case to additional concept – for example besides run time we could assign to a case whether the WS was available or not. We try to address all mentioned deficiencies in our approach. It is important to note that in this article we consider dedicated Grid environment, i.e. all computational resources are exclusively reserved for the use of considered Grid services.

2 Case Representation for WS Operation Performance Prediction

Information about WS performance usually comes in different formats. Therefore assimilation and integration of information from many different sources is required. A general concept is demanded, which would contain all information required for successful performance prediction. For the purposes of this article we deal only with WS operation run-time prediction.

The run-time of WS operations (WSO) can not be predicted just using run times of previous WS invocations without considering input parameters, because the length of execution is explicitly dependent on the input parameters. Herein we consider two types of input parameters: WS operation invocation parameters (`WSOInvocationParameter`) and WS operation input data properties (`WSOInputDataProperty`) as depicted on Fig. 1. WS operation invocation parameter is any parameter which is used during invocation of a WS Operation. For a WS making some kind of forecast a

length of forecast in minutes is an example of an WSO invocation parameter. WS operation input data property is a descriptor of any input data a WS might consume. For example an hydrological simulation WS uses an area map file as an input data file, where the grid density of this map is a property describing the input data.

Case representation is crucial for later processing of captured WS operation invocations. We model the case structure using ontology. The case which represents a single WS operation invocation is depicted on the following figure:

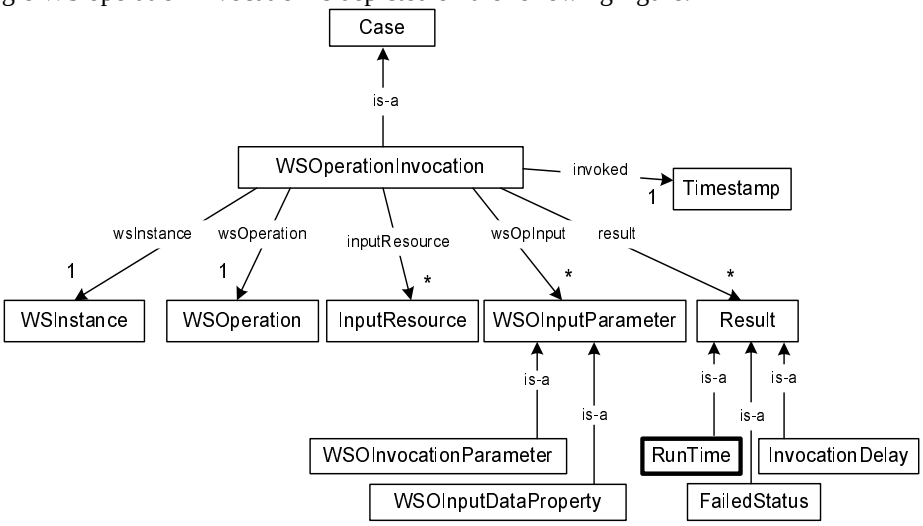


Fig 1. A generic case representation for WS operation performance prediction.

The central part of the ontology is the class `WSOperationInvocation` which is inherited from a `Case` class, obviously implying that "`WSOperationInvocation` is a `Case`". Class `WSOperationInvocation` has the following properties:

- `wsInstance` (an instance of `WSInstance` class) – a concrete WS deployment on a concrete location and on a concrete computer system,
- `wsOperation` (an instance of `WSOperation` class) – a name of operation invoked in scope of the current case,
- `wsOpInput` (an instance of `WSOInputParameter` class) – any number of input parameters we can gather about the current case,
- `invoked` – time of initial invocation of the case and
- `result` (an instance of `Result` class) – describes different resulting aspects generated by the case.

A case can have several types of results. One result can be the length of execution; another is whether the WS operation failed to finish successfully or how much time elapsed between the request and real start of invocation of an operation. The class called `Result` is used to represent such resulting effects (see Fig. 1). Since we are currently dealing only with WSO run-time prediction we use class `RunTime` as our primary result class. Instantiated example of herein presented ontology can be found in the following text.

2.1 Concrete Example of a Web Service Operation Invocation Case

Run times of WSO need to be monitored in order to predict the length of their executions in the future. There are several types of events, each containing different type of information about WS invocations. These various events must be aggregated (assimilated) into a single semantic representation - in our case we use the ontology with the super-class `Case` described in the previous chapter. In addition to information extracted from monitored events, we need to assimilate information from an input resource ontology, web services ontologies and WS property ontologies.

Unification of cases gives us uniform representation of data which we will use for further reasoning. Aggregation of events into one format is performed continuously either in predefined time periods or after occurrence of given number of events. The result of aggregation is therefore a history of WSO invocations in an aggregated format, prepared for further processing. A concrete example of a case with data assimilated from several sources is on the following figure:

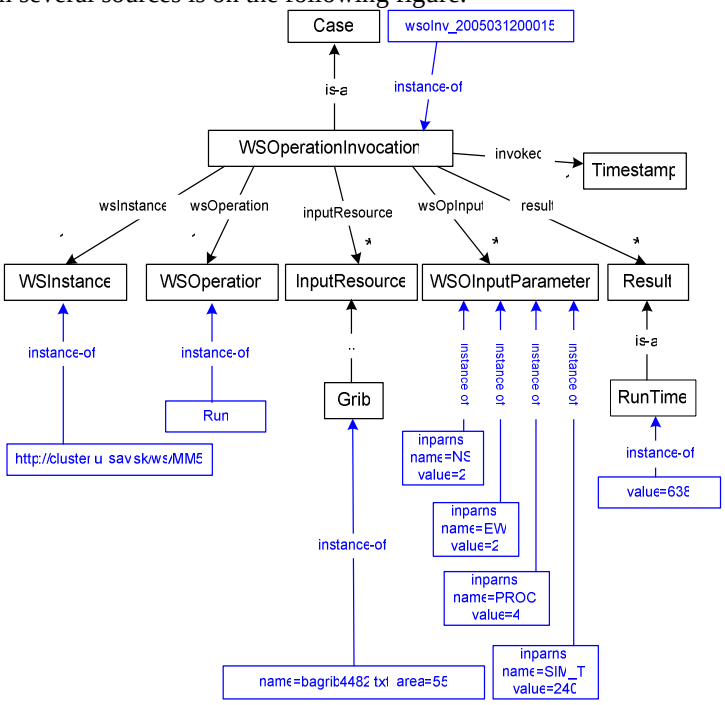


Fig 2. A Sample Case generated from a web service operation.

Fig. 2 depicts a concrete example instance of a WS operation invocation from the K-Wf Grid project [6]. A “Run” operation of a WS with a given URI was captured. Concept called “Grib” represents an input resource file wich is described by name and area size. This concrete WS operation invocation was launched using depicted input parameters. The result measured for this invocation was a runtime with its respective value.

2.2 Estimating future runtime of a single WS instance using reasoning

The estimation of a future runtime for a single WS instance is performed in the following steps:

1. Detection and formulation of "base context" (invocation parameters and input data properties) – a set of variables whose impact on the studied result is examined (in literature authors often use the term "feature vector");
2. Retrieval of cases we want to use for reasoning (might be the whole case base for the chosen WS instance or the number of cases might be limited according to additional limitation settings – an example time limitation might be that we want only cases from the last week);
3. Compute the similarity between the base context and context of each case retrieved from the case base;
4. Identify the most similar case/cases (the similarity measures of individual context properties can be customized);
5. Compute an estimate expected runtime using past runtimes of identified similar cases.

The assumption for a good result, i.e. good future runtime estimation is a sufficiently populated case base.

3 Refinement of case retrieval

Retrieval of similar cases in case-based reasoning (CBR) is in general solved by comparing feature vector of query case with the feature vectors of all cases from the case base. For discrete features usually exact match is done for feature values of query case with another case. Such exact matching has its limitations (returns a very limited set of cases), therefore we propose to model discrete features using ontology. The ontology can be used to find similar concepts thus broaden the set of most similar cases using simple inference.

Let's consider an example: a WS, which processes a single data file. Let's suppose we have description of a web service and all its input data files. Further let's consider that the feature vector is constructed out of the following:

- ws - URI of a WS,
- file_format - file format of a input data resource file and
- file_size - file size of the file.

The value of file format feature in such case is a file extension such as "xml", "rdf", "owl", "doc", "txt", "java" or other standard file extension, which determines the format of the file. In such situation case retrieval is realized by matching the feature vector of current case (URL, file_format and file_size) with feature vectors of all previous cases. The overall case is composed of a feature vector plus the result of a web service invocation.

Table 1. A Sample Case Base

Id	p1	p2	p3	Result
No	**ws**	**file_format**	**file_size**	**Run_time**
1	http://host1/axis/process	owl	104 556	890
2	http://host1/axis/process	owl	50 645	570
3	http://host1/axis/process	rdf	1 599 044	9 566
4	http://host1/axis/process	xml	356 944	1 680
5	http://host1/axis/process	rdf	1 246 003	8 956
6	http://host1/axis/process	rdf	1 233 554	8 877
7	http://host1/axis/process	owl	903 934	6 987

Having a goal of predicting a runtime of a WS deployed at 'http://host1/axis/process' for an input file with file_format "owl" with a known size (let's say 1 200 000B), we would need to retrieve all previous cases where an "owl" file was processed by the same web service in the past. Using our sample case base from the table above, we would retrieve all cases where:

```
(p1 == "http://host1/axis/process") and (p2 == "owl")
```

We would therefore retrieve cases 1, 2 and 7. Due to the fact that our case base is not populated enough with "owl"-based cases, the cases we retrieve would be not very useful. Because the size of the p2 feature significantly differ from our case a prediction using for instance nearest neighborhood method would give none or a very bad prediction (hence cases 1, 2 and 7 would be identified as most similar and subsequently would be used to predict the runtime very inaccurately, since the features of these cases significantly differ from our query case).

Knowing that "rdf", "owl" and other file formats are valid XMLs, we can model these concepts using ontology (Fig 3). In our sample case base we also see that processing of any "xml", "rdf" or "owl" file with a similar file size results to very similar runtimes. We could therefore use cases with "xml" or "rdf" file formats to predict a runtime of a WS processing an OWL file. To do that we propose to use the ontology which models input data file formats:

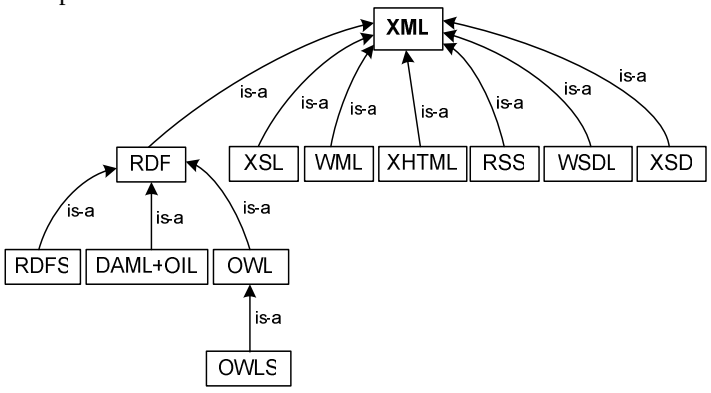


Fig 3. Input data description Ontology example.

Ontology from Fig 3 can be used to refine the retrieval of cases from the case base. We start with retrieval of all the OWL cases. If we detect that retrieved cases are not sufficient to deliver a satisfactory prediction, the retrieval is repeated with super class concept of the current concept (OWL), which is RDF. We continue such cycle, until we get satisfactory set of cases or we reach a concept which has no super class. It is assumed that the WS accepts any input data, which is of class XML or its subclasses.

```
C = classOf ({current_case}.file_format)

repeat
  for each {case} where {case}.file_format inst-of C {
    retrieve {case}
    sim_cases.add ({case})
  }

  if sufficient(sim_cases) {
    predict_result ({file})
    exit
  }
  C = superClass (C)
until (C == NULL)
```

The algorithm above would retrieve all cases, where p2 (i.e. `file_format`) is XML instance or instance of any subclass of XML (RDF, RDF-S, etc.). Thus we would retrieve all cases from the case base (from 1 to 7). In the prediction phase we use only the runtime to predict next runtime of our current case (`file_format` is not used, or can be used as a weighted measure). Depending on the setting of similarity engine we would identify as most similar cases 3, 5 and 6. This would give a more suitable case base for prediction of precise runtime.

4 Conclusion

In this article we are proposing generic ontology for web service runtime prediction and an approach to refinement of case retrieval for instance-based Grid service performance prediction through semantic description of input data. If the used case base has not enough cases, the estimation might be very inaccurate. The trade-off here is that the more specific the base context is the lower number of previous cases will be similar to our current case. Ontology allows us to build semantic structures where concepts are derived from general to more specific. This feature of ontology is used to deduce more general concepts of ontology, thus enlarging the number of cases retrieved for reasoning, thus providing more suitable base for prediction of WS runtimes.

References

[1] Warren Smith, Improving Resource Selection and Scheduling Using Predictions. Pp. 237-253, In: GRID RESOURCE MANAGEMENT, State of the Art and Future Trends. Kluwer Academic Publishers, 2004.

[2] Nirav H. Kapadia, José A. B. Fortes, Carla E. Brodley, Predictive Application-Performance Modeling in a Computational Grid Environment, Eight IEEE International Symposium on High Performance Distributed Computing, pp. 47-54, August 1999.

[3] SANKAR K. PAL, SIMON C. K. SHIU, FOUNDATIONS OF SOFT CASE-BASED REASONING. ISBN 0-471-08635-5, Published by John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2004.

[4] Web Service Management: Service Life Cycle: http://www.w3.org/TR/2004/NOTE-wslc-20040211/

[5] BALOGH Z., GATIAL E., LACLAVIK M., MALISKA M., HLUCHY L.: Knowledge-based Runtime Prediction of Stateful Web Services for Optimal Workflow Construction. Proc. of 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005, R.Wyrzykowski et.al. eds., 2005, Springer-Verlag, ISSN 0302-9743. Poznan, Poland. --in print--

[6] EU IST K-Wf Grid Project IST-2002-511385 http://www.kwfgrid.net/

[7] Faerman M., Su A., Wolski R., Berman F., Adaptive Performance Prediction for Distributed Data-Intensive Applications, August 9, 1999, Proceedings of the ACM/IEEE SC99 Conference on High Performance Networking and Computing, Portland

Appendix:

This work is supported by the EU IST K-Wf Grid Project IST-2002-511385, the Slovak Scientific Grant Agency within Research Project No. 2/3132/23 and APVT-51-024604 Project.

Aktívne učenie pre kategorizáciu textov

Peter Bednár

Katedra kybernetiky a umelej inteligencie, Technická Univerzita v Košiciach,
Letná 9, 042 00 Košice

Peter.Bednar@tuke.sk

Abstrakt. Viaceré aplikácie vyžadujú pre použitie metód kontrolovaného učenia rozsiahlu množinu klasifikovaných trénovacích dát. V tomto článku popíšeme metódu aktívneho učenia, ktorá umožňuje znížiť počet trénovacích príkladov vyžadovaných pre učenie. Uvedený algoritmus je založený na jednoduchej heuristike, ktorá nie je závislá na validačnej dátovej množine, čo umožňuje efektívne učenie na malej množine vybraných trénovacích príkladov.

1 Úvod

Reálne domény vyžadujú pre efektívne aplikovanie metód kontrolovaného učenia rozsiahle trénovacie dáta. Ak je však napr. potrebné pre získanie trénovacích dát vykonať náročné experimenty, alebo ak sú trénovacie príklady vytvorené na základe reálnych prípadových štúdií, trénovacie dáta môžu byť nedostupné resp. ich získanie môže byť príliš nákladné.

Aj v prípade ak je možné jednoducho získať nové dáta, trénovacie dáta je potrebné klasifikovať expertom. Tento proces je zvyčajne časovo obmedzený, takže je často možné pre učenie využiť iba obmedzenú množinu dát. Príkladom takejto domény je kategorizácia textov. Ak je expertom samotný užívateľ, ako je to napr. pri filtrovaní elektronickej pošty, stane sa použitie automatických metód založených na kontrolovanom učení obmedzujúce.

Tento článok sa zaoberá jednou z metód tzv. aktívneho učenia [1][2][3], pri ktorom učiaci algoritmus neprijíma nové trénovacie príklady pasívne, ale zvolí si nasledujúce trénovacie dáta aktívne na základe ich spoľahlivosti predikcie. V nasledujúcej kapitole je popísaný základný algoritmus výberu trénovacích dát. V ďalších podkapitolách je uvedený popis dvoch algoritmov (SVM a rozhodovacie stromy), ktoré sme testovali v kombinácií s aktívnym učením. Na záver sú uvedené výsledky experimentov, ktoré sme získali pri testovaní kategorizácie textov.

2 Aktívne učenie

V tomto článku sa budeme zaoberať metódou učenia, ktorá je založená na aktívnom výbere trénovacích príkladov. Na vstupe algoritmu je množina trénovacích dát, pričom iba časť z trénovacích príkladov je klasifikovaná. Algoritmus potom

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 159–166, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

vyberie z neklasifikovaných dát nasledujúce trénovacie príklady, ktoré predloží expertovi pre klasifikáciu. Cieľom aktívneho učenia je dosiahnuť čo najlepšiu efektívnosť klasifikácie pri čo najmenšom počte klasifikovaných trénovacích príkladov.

V [1] je uvedený algoritmus pre pravdepodobnostné modely, ktorý zaručuje "optimálny" výber trénovacích príkladov (v zmysle najväčšej redukcie pravdepodobnosti chyby znížením variability učenia za predpokladu, že bias učiaceho algoritmu je zanedbateľný). Táto metóda však vyžaduje aby bol na výstupe algoritmu priamo odhad podmienenej pravdepodobnosti triedy, čo si vyžaduje transformáciu predikcie pre diskriminačné klasifikátory. Navyše je v každom kroku potrebné odhadnúť chybu klasifikácie, ktorý však nemôže byť založený na validačnej množine, ktorá by vyžadovalo dodatočné klasifikované dáta. V nasledujúcich experimentoch sme použili jednoduchú heuristiku, ktorá je založená na spoľahlivosti predikcie [1][2][3]. Táto metóda nie je štatisticky optimálna, je však časovo efektívnejšia a nevyžaduje dodatočnú validačnú množinu.

Výber neklasifikovaných príkladov nie je jediná heuristika, ktorú je možné využiť pri aktívnom učení. Ďalšou možnosťou je konštrukcia úplne nového príkladu, ktorý potom učiaci algoritmus prezentuje expertovi. Tento spôsob však nemožno použiť pri kategorizácii textov, pretože vygenerovaný profil dokumentu nie je pre experta zrozumiteľný.

2.1 Support Vector Machine

Support Vector Machine [4] patrí medzi klasifikátory, ktoré sa pokúšajú explicitne vytvoriť nadrovinu oddeľujúcu príklady jednotlivých tried. Pre SVM je deliaca nadrovina určená jednoznačne, tak aby sa maximalizovala vzdialenosť medzi deliacou nadrovinou a najbližšími príkladmi oddeľovaných tried.

Trénovacie dokumenty sú reprezentované ako vektory d_i, $i = 1, ..., |D|$, $d_i \in \Re^{|V|}$ ku ktorým je priradené označenie triedy $y_i = \{-1, 1\}$. Deliaca nadrovina je určená vzťahom $d_i^T \beta + \beta_0 = 0$, kde β je normála deliacej nadroviny, $|\beta_0| / \|\beta\|$ je vzdialenosť nadroviny od počiatku súradníc a $\|\beta\|$ je euklidovská norma vektora β. Kritérium optimálnej deliacej nadroviny je možné definovať ako optimalizačnú úlohu:

$$\min_{\beta,\beta 0} \frac{1}{2} \| \beta \|^2 \qquad (1)$$

za podmienok:

$$y_i (d_i^T \beta + \beta_0) \geq 1 - \xi_i, \ i = 1,...,|D|$$

$$\xi_i \geq 0, \sum_{i=1}^{|D|} \xi_i \leq C$$

kde premenné $\xi_1, ..., \xi_{|D|}$ definujú ako ďaleko leží vektor dokumentu d_i na nesprávnej strane deliacej nadroviny pri lineárne neseparovateľných dátach. Parameter C nepriamo ohraničuje celkový počet nesprávne zaradených trénovacích dokumentov. Učenie SVM klasifikátora teda zodpovedá riešeniu konvexnej optimalizačnej úlohy

(kvadratická účelová funkcia s lineárnymi ohraničeniami podmienok). Riešenie zavedením *Lagrangeových* násobiteľov je dané optimalizovaním duálneho problému:

$$L_D = \sum_{i=1}^{|D|} \alpha_i - \frac{1}{2} \sum_{i=1}^{|D|} \sum_{k=1}^{|D|} \alpha_i \alpha_k y_i y_k d_i^T d_k \qquad (2)$$

za podmienok:

$$\sum_{i=1}^{|D|} \alpha_i y_i = 0 .$$

Násobitele α_i je možné určiť vyriešením duálneho problému (2) niektorou štandardnou metódou pre riešenie konvexných úloh. Metódy riešenia optimalizované pre veľký počet trénovacích vektorov sú uvedené v [5][6]. Výsledná rozhodovacia funkcia pre klasifikáciu do dvoch tried je:

$$y_i = sign(d_i^T \beta + \beta_0) \qquad (3)$$

SVM je možné upraviť pre nelineárnu klasifikáciu transformovaním priestoru príkladov do viacrozmerného euklidovského priestoru pomocou vhodnej kernelovej funkcie. Pre kategorizáciu textov však neprináša nelineárna transformácia výrazné zlepšenie efektívnosti a preto sme použili lineárne SVM.

Pre SVM je možné definovať mieru neurčitosti predikcie na základe šírky oblasti medzi nadrovinami H_1 a H_2, ktorá oddeľuje príklady jednotlivých tried. Heuristika výberu príkladov pre aktívne učenie SVM by mohla byť založená na výbere "najlepšej z najhorších" možností podľa kritéria $C_i = \max(C_{i,1}, C_{i,-1})$ kde $C_{i,1}$ resp. $C_{i,-1}$ je definované ako šírka hraničnej oblasti 2 / ||□|| po pridaní pozitívneho resp. negatívneho príkladu d_i. Podľa definície kritérium C_i predstavuje spodné ohraničenie zníženia neurčitosti ku ktorému môže dôjsť po klasifikovaní a pridaní dokumentu d_i do trénovacej množiny. Táto heuristika však nie je efektívne použiteľná, pretože vyžaduje riešenie dvoch kvadratických optimalizačných úloh pre každý neklasifikovaný príklad. Efektívnejšia metóda by mala umožniť výber nasledujúceho trénovacieho príkladu bez testovania SVM vytvoreného na rozšírenej trénovacej množine.

Alternatívna heuristika použitá v experimentoch je založená na výbere príkladu, ktorý čo najviac zmení výslednú hypotézu klasifikátora, tzn. ktorý čo najviac zmenší existujúcu hraničnú oblasť (pre lineárne separovateľné dáta). Za predpokladu "najlepšieho z najhorších" riešení by takáto heuristika mala vybrať príklad, ktorého kolmá vzdialenosť od deliacej nadroviny je čo najmenšia. Táto metóda je aj časovo efektívnejšia, pretože výpočet kritéria vyžaduje len skalárny súčin medzi vektorom d_i a □ (tzn. určenie predikcie klasifikátora).

2.2 Rozhodovacie stromy

Klasifikácia textov pomocou rozhodovacích stromov [7] je založená na rekurzívnom delení priestoru dokumentov na disjunktné oblasti. Delenie priestoru na podpriestory

je reprezentované nelistovými uzlami stromu, ktorým sú priradené testy deliace pokryté príklady do *n* disjunktných podmnožín, kde *n* je počet vetiev vychádzajúcich z uzla. Vytvorené disjunktné oblasti sú reprezentované listovými uzlami, ktorým je priradené označenie triedy.

Testy priradené nelistovým uzlom rozdeľujú dokumenty do dvoch množín $T1 = \{d_i: w_{i,t} \leq \theta_t\}$ a $T2 = \{d_i: w_{i,t} > \theta_t\}$ podľa výskytu jedného termu $w_t \in V$ (tzn., že výsledný strom je binárny). Rozhodovacie stromy sú generované metódou "zhora nadol" od všeobecnejšej hypotézy smerom k špecifickejšej. Počiatočný strom je tvorený iba jedným uzlom, ktorý pokrýva všetky trénovacie dokumenty. Ak nastane prípad, že všetky príklady pokryté daným uzlom majú rovnakú triedu, delenie (pod)priestoru dokumentov nie je potrebné a uzol sa označí ako listový. Inak algoritmus priradí uzlu logický test, ktorý rozdelí príklady na disjunktné podmnožiny, pre ktoré sa vytvoria nové uzly stromu. Celý proces sa potom rekurzívne opakuje na nových potomkoch, až pokiaľ sa nedosiahne úplné oddelenie príkladov jednotlivých tried.

Výber testu pre nelistové uzly je založený na hodnotiacej funkcii, ktorá zodpovedá "variabilite" označení tried v podmnožinách vytvorených po rozdelení uzla. Pri delení sa otestujú všetky možné testy (pre všetky termy $w_t \in V$ a všetky hodnoty θ_t) a vyberie sa test, ktorý vedie k čo najväčšej redukcii hodnotiacej funkcie (tzn. k čo najväčšiemu počtu príkladov z jednej triedy v každej vytvorenej podmnožine). Ako hodnotiacu funkciu pre výber testov je možné použiť napr. odhad chyby na trénovacích dátach, *Gini* index alebo entropiu.

Uvedená metóda indukcie rozhodovacích stromov vygeneruje hypotézu, ktorá je úplne konzistentná z trénovacími dátami (tzn. že všetky trénovacie príklady sú klasifikované správne). Aby sa zabránilo preučeniu, vygenerovaný strom sa zvyčajne zjednoduší tzv. *orezávaním*, avšak podľa predchádzajúcich experimentov pri kategorizácii textov môže orezávanie výrazne znížiť návratnosť výsledného klasifikátora. V experimentoch s aktívnym učením sme preto použili neorezané stromy. Ako hodnotiaca funkcia pre výber testov bola použitá entropia podobne ako pri algoritme C4.5.

Pravdepodobnosť chyby pre klasifikovaný príklad môžeme pri rozhodovacích stromoch odhadnúť podľa chyby uzla, do ktorého bol príklad zaradený. Ako spoľahlivosť predikcie C_i sme preto použili priamo *Laplaceov* odhad presnosti určený ako:

$$C_i = \frac{p + 0.5}{p + n + 1} \tag{4}$$

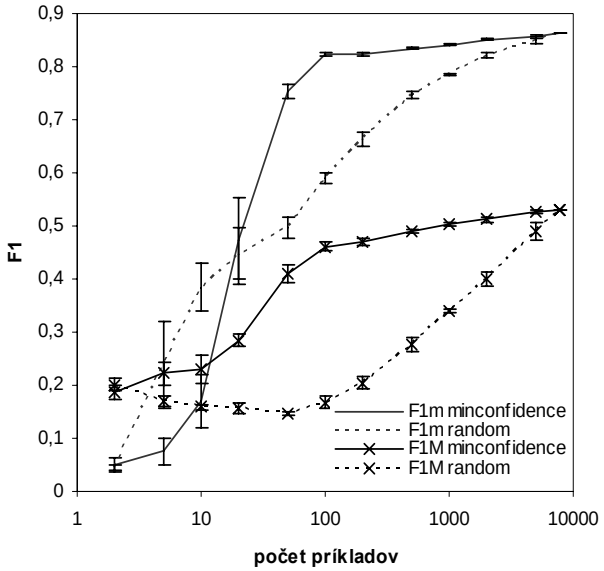
kde p (n) je počet pozitívnych resp. negatívnych príkladov pokrytých uzlom. Na základe takto definovanej (nespojitej) funkcie spoľahlivosti vyberie algoritmus aktívneho učenia v každom kroku náhodne jeden z príkladov, ktorý bol zaradený do uzla s najmenším počtom pokrytých trénovacích príkladov (s najmenšou podporou).

3 Experimenty

Pre experimenty bola použitá dátová množina Reuters-21578, ktorá obsahuje 21578 novinových článkov agentúry Reuters z roku 1987. Pri učení boli použité iba dokumenty, ktoré mali priradenú aspoň jednu z 90 kategórií. Dáta boli rozdelené na trénovaciu (7769 dokumentov) a testovaciu množinu (3019 dokumentov) podľa rozdelenia *ModApte*.

Pri predspracovaní dokumentov boli jednotlivé slová oddelené podľa výskytu prázdnych znakov, text bol prevedený na malé písmená a zo začiatku a konca slova boli odstránené diakritické znaky. Z množiny termov neboli odstránené žiadne stop slová (neplnovýznamové slová, ktoré sa často vyskytujú v dokumentoch nezávisle od ich obsahu, napr. spojky, častice, pomocné slovesá, atď.), avšak boli odstránené všetky slová, ktoré neobsahovali aspoň jedno písmeno. Taktiež pre normalizáciu termov nebola použitá žiadna metóda stemmingu[1].

Cieľom experimentov bolo zistiť, ako efektívne dokáže aktívne učenie zvoliť trénovacie príklady v porovnaní s náhodným výberom. Trénovacia množina bola na začiatku experimentov náhodne inicializovaná jedným pozitívnym a jedným negatívnym príkladom. Postupne bol v každom kroku pridaný do trénovacej množiny 1 (2-100), 10 (100-200), 50 (200-1000) a 200 (1000-7769) príkladov zvolených z pôvodnej trénovacej množiny. Pre testovanie bola použitá celá testovacia množina. Experimenty boli zopakované 5 krát s rôznou náhodnou inicializáciou.



Obr. 1 Závislosť efektívnosti klasifikácie od počtu trénovacích príkladov pre SVM klasifikátor. F_{1m} - mikro F_1, F_{1M} - makro F_1

[1] prevedenie tvaru na koreň slova odstránením sufixov a prefixov

Prvý graf (Obr. 1) zobrazuje závislosť efektívnosti klasifikácie od počtu trénovacích príkladov pre SVM klasifikátor. Chybové úsečky znázorňujú štandardnú odchýlku efektívnosti. Ako mieru efektívnosti sme použili F_1 mieru definovanú ako $F_1 = 2a/(2a + b + c)$, kde a,b a c sú hodnoty kontingenčnej tabuľky pre danú kategóriu[2]. F_1 mieru je možné spriemerovať pre všetky triedy dvoma spôsobmi: pri mikro-spriemerňovaní sa najprv vypočítajú hodnoty pre jednotlivé triedy a tieto sa spriemerujú, pri makro-spriemerňovaní sa najprv vypočítajú globálne hodnoty a, b a c pre všetky triedy z ktorých sa vypočíta miera F_1.

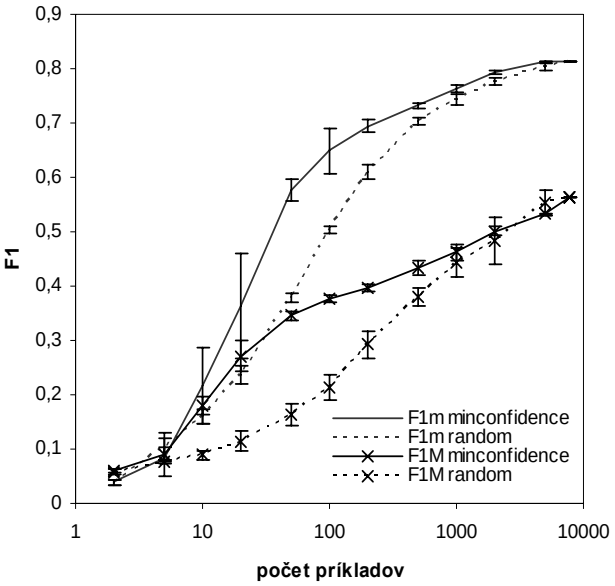
Ako je možné pozorovať, aktívne učenie dosahuje vyššiu efektívnosť než náhodný výber pri menšom počte trénovacích príkladov. Pre makro-spriemerňovanie, ktoré je citlivejšie na efektívnosť tried s malým počtom pozitívnych príkladov, je tento rozdiel výrazný počas celého priebehu. Napr. pri učení na trénovacej množine o veľkosti 100 príkladov (čo je len 1,2% celkového počtu) bola dosiahnutá efektívnosť 46%. Pre dosiahnutie rovnakej efektívnosti pri náhodnom výbere dokumentov je potrebné klasifikovať až 3400 príkladov. Maximálna efektívnosť 53% bola dosiahnutá pri učení na celej trénovacej množine 7769 príkladov.

Pre mikro-spriemerňovanie je priebeh podobný s výnimkou trénovacej množiny o veľkosti 2 až 20 príkladov, kedy je náhodný výber efektívnejší aj napriek vyššej variabilite učenia. Tento jav je spôsobený najmä výberom pozitívnych príkladov, čo spôsobuje zvýšenie počtu chybne klasifikovaných negatívnych príkladov (presnosť klasifikácie dosahuje iba 2 až 10%). Maximálna presnosť 86% bola dosiahnutá na úplnej trénovacej množine, čo znamená že sa nepotvrdili závery niektorých predchádzajúcich experimentov, pri ktorých sa dosiahla najlepšia efektívnosť na podmnožine trénovacích dát. Predchádzajúce experimenty však neuvádzali efektívnosť pre všetky kategórie, ale iba pre podmnožinu kategórií s najväčším počtom pozitívnych príkladov.

Graf na obrázku (Obr. 2) zobrazuje závislosť efektívnosti klasifikácie od počtu trénovacích príkladov pre rozhodovacie stromy.
Pre makro-spriemerňovanie je priebeh podobný ako pri SVM, rozdiel v efektívnosti však nie je až taký výrazný. Pri vyššom počte príkladov (4000 – 7000) dosahuje náhodný výber vyššiu efektívnosť asi o 2%, čo je vzhľadom na vyššiu variabilitu učenia zanedbateľný rozdiel. V porovnaní s SVM pri rozhodovacích stromoch nedochádza k takému rýchlemu nárastu efektívnosti (napr. pri 100 príkladoch je možné dosiahnuť efektívnosť 37%, pričom maximálna efektívnosť 56% bola znova dosiahnutá na celej trénovacej množine).

Na rozdiel od SVM sa pri mikro-spriemerňovaní neprejavil pokles presnosti pri malom počte príkladov, čo je spôsobené zvolenou funkciou spoľahlivosti predikcie. Pri počiatočnom rozhodovacom strome, ktorý má iba jednu pozitívnu a jednu negatívnu vetvu s rovnakou podporou budú mať všetky príklady podľa (4) rovnaký odhad spoľahlivosti predikcie. Aktívne učenie sa teda nebude odlišovať od náhodného výberu. Pri väčšom počte trénovacích príkladov sa však nasledujúce učenie sústredí na zvýšenie spoľahlivosti predikcie uzlov s malou podporou a presnosťou. Aj keď rozdiely nie sú až také výrazné, aktívne učenie prináša zlepšenie efektívnosti aj v tomto prípade.

[2] a je počet správne klasifikovaných pozitívnych príkladov, b je počet nesprávne klasifikovaných pozitívnych príkladov, atď.



Obr. 2 Závislosť efektívnosti klasifikácie od počtu trénovacích príkladov pre rozhodovacie stromy. F_{1m} - mikro F_1, F_{1M} - makro F_1

Keďže zvolená funkcia spoľahlivosti nie je spojitá, rýchlejší nárast efektívnosti by bolo možné dosiahnuť zvolením odlišnej heuristiky aktívneho učenia. Rozhodovacie stromy boli napr. úspešné použité v zložených klasifikátoroch generovaných metódou baggingu [8], ktorá vytvára viacero odlišných hypotéz na rôznych trénovacích množinách vytvorených z pôvodných dát náhodným výberom s opakovaním. Spoľahlivosť predikcie pre neoznačený príklad je možné potom odhadnúť na základe zhody medzi viacerými hypotézami. Kombinácia viacerých hypotéz by navyše znížila variabilitu výsledného klasifikátora, čo by zlepšilo celkovú efektívnosť. Nevýhodou tejto metódy je však časová neefektívnosť učenia, pretože je v každom kroku potrebné učiť viacero modelov.

4 Záver

Vo všetkých uvedených experimentoch dosiahli metódy aktívneho učenia vyššiu efektívnosť pri menšom počte trénovacích príkladov oproti náhodnému výberu. Tento rozdiel bol najviac výrazný pre SVM klasifikátor, kde pri trénovacej množine s 100 príkladmi, čo je iba 1,2% z pôvodnej množiny, poklesne efektívnosť oproti maximálnej hodnote 86% iba o 4% (pre porovnanie, rozhodovacie stromy dosiahli najlepšiu efektívnosť 81%).

Môže sa zdať že nevýhodou aktívneho učenia je časová náročnosť, keďže je treba znova preučiť klasifikátor po pridaní nových trénovacích príkladov. Celkovo je však

na dosiahnutie požadovanej presnosti potrebných oveľa menej príkladov, takže sa dodatočné preučenie a ohodnotenie neoznačených príkladov nemusí časovo prejaviť.

Najväčší prínos aktívneho učenia je však v ušetrení nákladov na získavanie a klasifikáciu nových trénovacích dát, čo umožňuje nasadenie metód kontrolovaného učenia aj v doménach, pri ktorých nie sú dostupné rozsiahle trénovacie množiny klasifikovaných príkladov.

Práca prezentovaná v tomto príspevku vznikla za podpory Vedeckej grantovej agentúry Ministerstva školstva SR a Slovenskej akadémie vied (VEGA) v rámci projektu č.1/1060/04 s názvom "Klasifikácia a anotácia dokumentov pre sémantický web" a za podpory nemecko-slovenského vedeckého projektu DAAD č.8/2004 "Využitie objavovania znalostí v textoch pre extrakciu metadát a sémantické vyhľadávanie".

Referencie

1. Cohn, D., Ghahramani, Z., Jordan, M.: Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4, 129-145. (1996)
2. Tong, S., Koller, D.: Support Vector Machine active learning with Application to Text Categorization, *Machine Learning Research*, vol. 2, 45-66. (2001)
3. Schohn, G. Cohn, D.: Less Is More: Active Learning with Support Vector Machines, *Proc. 17th International Conference Machine Learning*, 839-846, 2000.
4. Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag.
5. Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines. technical report *MST-TR-98-14*, Microsoft Research. (1998)
6. Joachims, T.: Making large-scale SVM practical. Schölkopf, B., Burges, C., Smola, A., edit., *Advances in Kernel Methods - Support Vector Learning*. MIT-Press. (1999)
7. Apté, C., Damerau, F., Weiss, S.: Automated learning of decition rules for text categorization. ACM Transactions on Information Systems, 12(3):233-251. (1994)
8. Machová, K., Puszta, M., Bednár, P.:.A boosting method in combination with decision trees, In: SAMI 2005 : 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, Herľany, Slovakia January 21-22, 2005, 151-162, Budapest : Budapest Polytechnic. (2005)

Annotation:

Active learning for text categorization

Many real applications require a large training dataset for supervised learning. In this paper we will present one method of active learning, which allows to reduce the number of training examples required for effective learning. Presented algorithm is based on simple heuristic that selects examples according to the confidence of the classifier prediction for the given example. This heuristic doesn't require a validation set and can be used effectively to select a small set of labeled examples.

Efektivní marginalizace mnohadimenzionálních kompoziciálních modelů

Vladislav Bína

Fakulta managementu, Jindřichův Hradec,
Vysoká škola ekonomická, Praha
Jarošovská 1117/II, 37701, Jindřichův Hradec

bina@fm.vse.cz

Abstrakt Grafické markovské modely se staly velmi úspěšnými a populárními zejména díky tomu, že byly vyvinuty efektivní algoritmy pro práci s mnohadimenzionálními distribucemi. Odpovídající algoritmy fungují i v rámci kompoziciálních modelů, které jsou alternativou ke grafickým markovským modelům, a to alternativou, která přináší z výpočetního hlediska určité výhody. Právě na jejich využití je založen tento příspěvek, který přináší teoretické základy a zároveň náčrt algoritmu sloužícího k efektivnímu výpočtu marginálních distribucí v kompoziciálních modelech, tedy problému, který dosud není uspokojivě vyřešen ani v softwareových balících pracujících s bayesovskými sítěmi.

Klíčová slova: kompozicionální modely, marginalizace

1 Úvod

Reprezentace a zpracování mnohadimenzionálních pravděpodobnostních distribucí bylo umožněno pokroky dosaženými na poli grafických markovských modelů (viz např. [7]) během posledních asi dvaceti let. Máme zde na mysli nejen bohaté teoretické pozadí, ale i důkladně propracovaný algoritmický aparát, který umožnil vývoj efektivních softwareových balíků (jako např. HUGIN [2]). Jako alternativa ke grafickým modelům byl (od roku 1997) vyvíjen přístup bez použití grafů – *kompozicionální modely* založené na skládání (kompozici) mnohadimenzionální distribuce ze systému několika málodimenzionálních.

Na tomto místě hodláme prezentovat řešení obtížného problému, který dosud nebyl uspokojivě vyřešen dokonce ani v tak propracovaném softwareovém systému jako je např. HUGIN: problému marginalizace mnohadimenzionální distribuce. Pro bayesovské sítě bylo řešení tohoto problému navrženo Rossem Shachterem v [8]. Jeho známý postup je založen na dvou pravidlech: *odstraňování uzlů* a *změna orientace hran*. Zhruba řečeno, efektivita tohoto postupu odpovídá efektivitě námi prezentovaného procesu, ovšem zjednodušeného, bez použití urychlení teoreticky umožněného větou 2 uvednou v kapitole 3 tohoto článku. Konkrétně tato věta využívá rozdílu mezi bayesovskými sítěmi [3] a kompozicionálními modely publikovaného v [6]. Tato výhoda je založena na skutečnosti, že kompozicionální modely mají některé marginály, jejichž výpočet je v bayesovských sítích náročný, vyjádřené přímo.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 167–174, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2 Základní pojmy a značení

V textu budeme uvažovat systém náhodných proměnných s konečným počtem hodnot a s indexy z neprázdné konečné množiny N. Pravděpodobnostní distribuce budou značeny řeckými písmeny. Mějme $K \subset N$, potom $\pi(x_K)$ označuje distribuci proměnných $\{X_i\}_{i\in K}$ definovanou na všech podmnožinách kartézkého součinu $\mathbf{X}_K \stackrel{\text{def}}{=} \times_{i\in K}\mathbf{X}_i$.

Máme-li distribuci $\pi(x_K)$ a $L \subset K$, odpovídající marginál budeme značit $\pi(x_L)$ nebo (analogicky značení Glenna Shafera [9]) $\pi^{\downarrow L}$. Takto zdůrazníme proměnné, pro které je marginál definován. Naopak, chceme-li specifikovat proměnné, které vypouštíme (M), použijeme symbolu π^{-M}. V naší situaci platí $K \setminus M = L$. Abychom zjednodušili značení, budeme množinu všech přirozených čísel menších nebo rovných celému číslu n označovat pomocí $\hat{n} \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$.

K popisu skládání málodimenzionálních distribucí a získání distribuce s vyšším počtem dimenzí použijeme následující operátor kompozice.

Definice 1. *Pro libovolné dvě distribuce $\pi(x_K)$ a $\kappa(x_L)$ je jejich* kompozice *dána vztahem*

$$\pi(x_K) \triangleright \kappa(x_L) = \begin{cases} \frac{\pi(x_K)\kappa(x_L)}{\kappa(x_{K\cap L})} & \text{když} \quad \pi(x_{K\cap L}) \ll \kappa(x_{K\cap L}), \\ \text{nedefinováno jinak,} \end{cases}$$

kde relace $\pi(x_M) \ll \kappa(x_M)$ označuje fakt, že $\pi(x_M)$ je dominována $\kappa(x_M)$*, což (v konečném případě) znamená $(\forall x_M \in \mathbf{X}_M) : (\kappa(x_M) = 0 \implies \pi(x_M) = 0)$.*

Použití operátoru kompozice lze iterativně opakovat a vytvořit *mnohadimenzionální kompozicionální model*. K jejich popisu postačuje seřazený systém málodimenzionálních distribucí $\pi_1, \pi_2, \ldots, \pi_n$, což budeme nazývat *generující sekvencí*. Generující sekvence definuje (reprezentuje) mnohadimenzionální kompoziciální model. Operátor kompozice je aplikován zleva doprava (není komutativní ani asociativní):

$$\pi_1 \triangleright \pi_2 \triangleright \pi_3 \triangleright \ldots \triangleright \pi_{n-1} \triangleright \pi_n \stackrel{\text{def}}{=} (\ldots((\pi_1 \triangleright \pi_2) \triangleright \pi_3) \triangleright \ldots \triangleright \pi_{n-1}) \triangleright \pi_n.$$

V procesu marginalizace budeme potřebovat ještě další důležitý operátor.

Definice 2. *Mějme libovolné dvě distribuce $\pi(x_K)$, $\kappa(x_L)$ a množinu indexů proměnných $M \subset N$, použitím* anticipativního operátoru *parametrizovaného množinou indexů M budeme rozumět následující distribuci*

$$\pi \circledcirc_M \kappa = \left(\kappa^{\downarrow(M\setminus K)\cap L}\ \pi\ \right) \triangleright \kappa.$$

3 Marginalizace v kompoziciálním modelu

V následujícím textu budeme potřebovat dvě jednoduchá lemmata, která vyplývají přímo z definice operátoru kompozice[1].

[1] První tvrzení je triviálním důsledkem definice operátoru kompozice, naznačme myšlenku důkazu lemmatu 2: $\pi \triangleright (\pi \triangleright \kappa)^{\downarrow M} = \frac{\pi(\pi\triangleright\kappa)^{\downarrow M}}{(\pi\triangleright\kappa)^{\downarrow K\cap M}} = \frac{\pi\,\frac{\pi^{\downarrow K\cap M}}{\kappa^{\downarrow K\cap L}}}{\frac{\pi^{\downarrow K\cap M}}{\kappa^{\downarrow K\cap L}}\kappa^{\downarrow K\cap L}} = \pi \triangleright \kappa$

Lemma 1. *Uvažujme dvě distribuce $\pi(x_K)$ a $\kappa(x_L)$. Pokud je kompozice $\pi \triangleright \kappa$ definována, platí $(\pi \triangleright \kappa)^{\downarrow K} = \pi$.*

Lemma 2. *Mějme dvě distribuce $\pi(x_K)$ a $\kappa(x_L)$, je-li jejich kompozice $\pi \triangleright \kappa$ definována a platí-li navíc $L \subseteq M \subseteq K \cup L$, potom $\pi \triangleright \kappa = \pi \triangleright (\pi \triangleright \kappa)^{\downarrow M}$.*

V dalším textu uvažujeme generující sekvence $\pi_1(x_{K_1}) \triangleright \pi_2(x_{K_2}) \triangleright \ldots \pi_n(x_{K_n})$. Proto kdykoliv použijeme symbol π_j, budeme předpokládat, že je definován pro proměnné $\{X_i\}_{i \in K_j}$.

Nejdříve zformulujeme pravidla, která nám umožní snížit dimenzi kompoziciálních modelů o jednu, jejich iterativním použitím lze vypočítat libovolný požadovaný marginál. Důkaz následujícího tvrzení (věty 1) lze nalézt v [4].

Věta 1 (Marginalizace přes jednu proměnnou). *Nechť $\pi_1, \pi_2, \ldots, \pi_n$ je generující sekvence a $\ell \in K_{i_1} \cap K_{i_2} \cap \ldots \cap K_{i_m}$ pro podposloupnost $(i_1, i_2, \ldots, i_m)$ z $\hat{n}$ takovou, že $\ell \notin K_j$ pro všechna $j \in \hat{n} \setminus \{i_1, i_2, \ldots, i_m\}$. Potom*

$$(\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n)^{-\{\ell\}} = \kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_n,$$

kde

$$\begin{aligned}
\kappa_j &= \pi_j, \qquad \forall j \in \hat{n} \setminus \{i_1, i_2, \ldots, i_m\},\\
\kappa_{i_1} &= \pi_{i_1}^{-\{\ell\}},\\
\kappa_{i_2} &= (\pi_{i_1} \circledcirc_{L_{i_2-1}} \pi_{i_2})^{-\{\ell\}},\\
\kappa_{i_3} &= (\pi_{i_1} \circledcirc_{L_{i_2-1}} \pi_{i_2} \circledcirc_{L_{i_3-1}} \pi_{i_3})^{-\{\ell\}},\\
&\vdots\\
\kappa_{i_m} &= (\pi_{i_1} \circledcirc_{L_{i_2-1}} \pi_{i_2} \circledcirc_{L_{i_3-1}} \ldots \circledcirc_{L_{i_m-1}} \pi_{i_m})^{-\{\ell\}},
\end{aligned}$$

a $L_{i_k-1} = (K_1 \cup K_2 \cup \ldots \cup K_{i_k-1}) \setminus \{\ell\}$.

Iterativní použití této věty vždy vede k požadovanému marginálu a plně koresponduje Shachterově proceduře. Použití anticipativního operátoru odpovídá *dědění rodičů* v jeho *pravidle změny orientace hran*. Není jistě překvapením, že výpočetní složitost tohoto procesu silně závisí na počtu výskytů proměnné ℓ mezi argumenty distribucí v generující sekvenci (lze tedy do jisté míry regulovat výpočetní složitost vhodným seřazením vynechávaných proměnných).

Počínaje druhým výskytem uvažované proměnné je zapotřebí nahradit distribuci π_{i_k} výrazem obsahujícím jeden nebo (v dalších výskytech) více anticipativních operátorů. Navíc tento výraz je pořád ještě třeba marginalizovat. Může se snadno stát, že iterativní aplikace této věty se stane výpočetně prakticky neproveditelnou vzhledem k enormní potřebě výpočetního času a paměti.

Pokud je právě marginalizovaná proměnná argumentem jediné distribuce, větu 1 lze zjednodušit do tvaru níže uvedeného důsledku, kdy postačuje marginalizovat právě tuto jedinou distribuci, zatímco ostatní zůstavají nezměněny. To v kontextu bayesovských sítí odpovídá situaci, kdy Shachterovo *pravidlo odstranění uzlu* lze aplikovat buď to přímo (uzel nemá potomky) nebo použití pravidla *změny orientace hrany* nepřinese do uvažované sítě nové hrany.

Důsledek 1. *Nechť $\pi_1, \pi_2, \ldots, \pi_n$ je generující sekvence. Existuje-li $\ell \in K_i$ pro nějaké $i \in \hat{n}$ a $\ell \notin K_j$ pro všechna $j \in (\hat{n} \setminus \{i\})$, potom marginál distribuce reprezentované generující sekvencí lze vyjádřit vztahem*

$$(\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n)^{-\{\ell\}} = \pi_1 \triangleright \ldots \triangleright \pi_{i-1} \triangleright \pi_i^{-\{\ell\}} \triangleright \pi_{i+1} \triangleright \ldots \triangleright \pi_n.$$

Nejefektivnější způsob marginalizace je založen na následujícím tvrzení, které je zobecněním věty 11 z [5]. Popisuje podmínky, za nichž lze skupinu proměnných marginalizovat v jednom výpočetně nepříliš složitém kroku. Důkaz lze nalézt v [1]. Nejdříve však zavedeme pomocný pojem redukce generující sekvence.

Definice 3. *Je-li $\pi_1, \pi_2, \ldots, \pi_n$ generující sekvence a $(j_1, j_2, \ldots, j_m)$ je podposloupností $\hat{n}$ takovou, že existuje $s \in Z = \{j_1, \ldots, j_m\}$, pro které*[2]

$$(\bigcup_{j \in Z} K_j) \cap (\bigcup_{j \notin Z} K_j) \subseteq K_s.$$

Pak říkáme, že s a Z určuje redukci *generující sekvence $\pi_1, \ldots, \pi_n$ (nebo stručněji (s, Z) je redukce).*

Věta 2. *Nechť $s \in Z$ a $Z = \{j_1, \ldots, j_m\}$ určuje redukci generující sekvence $\pi_1, \ldots, \pi_n$. Potom, označíme-li $L = \bigcup_{j \in Z} K_j$, $\mu = (\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n)^{\downarrow K_s}$, a dále $\bar{L}_j = \bigcup_{i \in \hat{j} \setminus Z} K_i$ pro všechna $j \notin Z$, marginální distribuce $(\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n)^{\downarrow L}$ může být vyjádřena jako kompoziciální model*

$$(\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n)^{\downarrow L} = \kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_n,$$

kde

$$\begin{array}{lll} \kappa_j = \pi_j & \text{pro} & j \in Z, \\ \kappa_j = \mu^{\downarrow L \cap \bar{L}_j} & \text{pro} & j \notin Z. \end{array}$$

Tato věta nám bohužel nedává žádný návod jak nalézt množinu indexů Z (ale ani index s), které by určovaly redukci uvažované generující sekvence. K tomu lze využít následující jednoduchá lemmata. Abychom je formulovali dostatečně přehledně a jednoduše zavedeme pomocný symbol. Mějme množinu $Z \subset \hat{n}$ a $j \notin Z$, symbol $W(Z, j)$ označuje podmnožinu indexů

$$W(Z, j) = \left\{ s \in \hat{n} : \left(\bigcup_{i \in Z} K_i \right) \cap K_j \subseteq K_s \right\}.$$

Lemma 3. *Pokud pro $Z \subset \hat{n}$ ($\emptyset \neq Z \neq \hat{n}$) existuje $s \in Z$, pro které $s \in \bigcap_{j \notin Z} W(Z, j)$, pak s a Z určují redukci (uvažované generující sekvence).*

Důkaz. Pro s splňující předpoklady tohoto lemmatu platí $\left(\bigcup_{i \in Z} K_i\right) \cap K_j \subseteq K_s$ pro všechna $j \notin Z$, a tedy $(\bigcup_{j \in Z} K_j) \cap (\bigcup_{j \notin Z} K_j) \subseteq K_s$. □

[2] Symbol $j \notin Z$ je použit pro zjednodušení zápisu namísto korektnějšího $j \in \hat{n} \setminus Z$.

Lemma 4. *Nechť neprázdné $Z \subset \hat{n}$ je různé od $\hat{n}$ a $s \in Z$. Pokud pro nějaké $j \notin Z$, $W(Z,j) \cap Z = \emptyset$, pak neexistuje redukce (uvažované generující sekvence) určená s a Z.*

Důkaz. Dokážeme sporem, nechť existuje $s \in Z$ určující redukci generující sekvence. Pak musí platit inkluze $\left(\bigcup_{i\in Z} K_i\right) \cap \left(\bigcup_{i\notin Z} K_i\right) \subseteq K_s$, neboli $\left(\bigcup_{i\in Z} K_i\right) \cap K_j \subseteq K_s$ pro všechna $j \notin Z$, a tedy $s \in W(Z,j)$ $(\forall j \notin Z)$. Ale nyní Z i $W(Z,j)$ $(\forall j \notin Z)$ obsahuje s, neboli $W(Z,j) \cap Z \ni s$, jinými slovy, průnik $W(Z,j) \cap Z$ je neprázdný, což je spor s předpoklady lemmatu. □

4 Marginalizační algoritmus

Nyní zformulujeme hlavní myšlenky, na kterých je efektivní algoritmus marginalizace v kompoziciálních modelech založen. Budeme minimalizovat použití věty 1.

1. Nejjednodušším způsobem marginalizace je střídavé používání lemmatu 1 a důsledku 1. Proto je využijeme již v prvním kroku a jejich využití zopakujeme kdykoliv budou splněny předpoklady obou tvrzení. Některé distribuce mohou být úplně vynechány (díky lemmatu 1) po použití důsledku 1. Proto má smysl střídavé opakování obou tvrzení.
2. Pokud nelze využít krok 1, pokusíme se aplikovat větu 2 (tato možnost bude v dalším textu rozebrána poněkud podrobněji). V případě uspěchu budeme pokračovat opět krokem 1.
3. Nelze-li použít žádný z předchozích kroků, budeme pokračovat v marginalizaci opakovanou aplikací věty 1, po každém jejím použití zopakujeme krok 1.

S ohledem k efektivitě celé marginalizační procedury má největší vliv sofistikovaná realizace druhého kroku. Lemmata 3 a 4 mohou být východiskem pro efektivní postup hledání s a Z určujících redukci, jejich využití nyní stručně popíšeme.

Uvažujme situaci, kdy chceme vypočítat marginál

$$(\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n)^{\downarrow M},$$

a nelze využít krok 1 algoritmu. Začneme s minimálním Z takovým, že $M \subseteq \bigcup_{i\in Z} K_i$ a vypočítáme $W(Z,j)$ pro všechna $j \notin Z$. Je-li pro nějaké $j \notin Z$ množina $W(Z,j) \cap Z = \emptyset$, je zapotřebí podle lemmatu 4 toto j k množině Z přidat.

Nyní začneme hledat $s \in Z$, které splňuje podmínky věty 2. Bude nás zajímat zda existuje $s \in Z \cap \left(\bigcap_{j\notin Z} W(Z,j)\right)$. V kladném případě jsme nalezli redukci a můžeme použít větu 2, v opačném případě množinu Z rozšíříme. Vezmeme $s \notin Z$ (budeme upřednostňovat takové[3], aby $K_s \cap L$ bylo co největší), přidáme jej k množině Z a nalezneme nové množiny $W(Z,j)$. Poté přidáme k množině Z všechna $j \notin Z$, pro která $s \notin W(Z,j)$. Opakované zvětšování množiny Z (beze

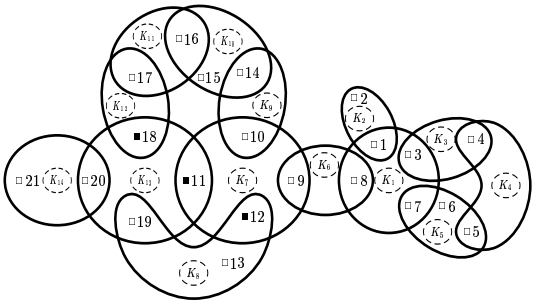
[3] Množina L je zavedena ve formulaci věty 2.

změny s) ukončíme, buďto pokud nalezneme pár s a Z splňující předpoklady věty 2 nebo dospějeme k $Z = \hat{n}$, tedy zjistíme, že uvažované s musí být přidáno k původní Z a naše volba s byla nevhodná. Postup je třeba opakovat s jiným s a rozšířeným Z. Výsledkem je nalezená redukce nebo fakt, že redukce neexistuje.

5 Příklad

Uvažujme distribuce $\pi_1, \pi_2, \ldots, \pi_{14}$ s množinami proměnných (viz též obr. 1):

$$\begin{array}{lll} K_1 = \{1,3,7,8\}, & K_2 = \{1,2\}, & K_3 = \{3,4\}, \\ K_4 = \{4,5\}, & K_5 = \{5,6,7\}, & K_6 = \{8,9\}, \\ K_7 = \{9,10,11,12\}, & K_8 = \{12,13,19\}, & K_9 = \{10,14\}, \\ K_{10} = \{14,15,16\}, & K_{11} = \{16,17\}, & K_{12} = \{17,18\}, \\ K_{13} = \{11,18,19,20\}, & K_{14} = \{20,21\}. & \end{array}$$



Obrázek 1. Množiny proměnných, pro které jsou definovány distribuce $\pi_1, \pi_2, \ldots, \pi_{14}$

Tyto distribuce, jako generující sekvence, definují mnohadimenzionální kompoziciální model $\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_{14}$. Naším cílem bude vypočítat

$$(\pi_1 \triangleright \ldots \triangleright \pi_{14})^{\downarrow\{11,12,18\}}.$$

Prvním krokem bude odstranění distribuce π_{14} z našeho modelu, to je umožněno lemmatem 1. Dále můžeme marginalizovat všechny proměnné, které se vyskytují jen v jedné distribuci. Vypustíme je díky důsledku 1. Máme tedy

$$\begin{aligned} (\pi_1 \triangleright \ldots \triangleright \pi_{13})^{-\{2,6,13,15,20\}} &= \\ &= \pi_1 \triangleright \pi_2^{-\{2\}} \triangleright \pi_3 \triangleright \pi_4 \triangleright \pi_5^{-\{6\}} \triangleright \pi_6 \triangleright \pi_7 \triangleright \pi_8^{-\{13\}} \triangleright \\ &\qquad \triangleright \pi_9 \triangleright \pi_{10}^{-\{15\}} \triangleright \pi_{11} \triangleright \pi_{12} \triangleright \pi_{13}^{-\{20\}}. \end{aligned}$$

Použijeme-li znovu lemma 1, můžeme vypustit distribuci $\pi_2^{\downarrow\{1\}}$. Ve skutečnosti stačilo distribuci π_2 vynechat (bez výpočtu marginálu). Nyní se X_1 vyskytuje pouze mezi argumenty π_1, můžeme ji marginalizovat pomocí důsledku 1

$$(\pi_1 \triangleright \ldots \triangleright \pi_{13})^{-\{1,2,6,13,15,20\}} = \\ = \pi_1^{-\{1\}} \triangleright \pi_3 \triangleright \pi_4 \triangleright \pi_5^{-\{6\}} \triangleright \pi_6 \triangleright \pi_7 \triangleright \pi_8^{-\{13\}} \triangleright \\ \triangleright \pi_9 \triangleright \pi_{10}^{-\{15\}} \triangleright \pi_{11} \triangleright \pi_{12} \triangleright \pi_{13}^{-\{20\}}.$$

Opakovaným užitím lemmatu 1 a důsledku 1 již dalšího zjednodušení nedocílíme.

Aplikujeme tedy druhý krok algoritmu a zjistíme, zda existuje redukce uvedené generující sekvence. V modelu mají zbýt proměnné X_{11}, X_{12} a X_{18}, tedy distribuce π_7 a π_{13}, proto bude na počátku $Z = \{7, 13\}$. Protože je $W(Z, 8) = \{8\}$ ($W(Z, 8) \cap \{7, 13\} = \emptyset$), lze aplikovat lemma 4 a je nutné rozšířit na $Z = \{7, 8, 13\}$. Protože $\bigcap_{j \notin Z} W(Z, j) = \emptyset$, neexistuje redukce určená množinou Z (a libovolným $s \in Z$). Budeme proto množinu Z dále rozšiřovat. Můžeme zvolit mezi $6, 9, 12$, k množině Z zkusmo přidáme (za s zvolíme) např. 9.

Budeme uvažovat $Z = \{7, 8, 9, 13\}$, musíme ale k Z ještě přidat všechna $j \notin Z$ taková, že $s \notin W(Z, j)$. V prvním kroku tedy přidáme $\{6, 12\}$ k Z, ve druhém kroku $\{1, 11\}$ atd. Po 4 krocích Z obsahuje indexy všech distribucí, to znamená, že $s = 9$ nepřipadá pro použití věty 2 v úvahu (neexistuje redukce). Tudíž je nutné přidat 9 k Z a zvolit jiného kandidáta na s. Opět lze uvažovat o 3 indexech, a to $6, 10, 12$. Vybereme-li tentokrát $s = 6$ dostaneme po dvou krocích $Z = \{6, 7, 8, 9, 10, 11, 12, 13\}$, což spolu s $s = 6$ určuje redukci a umožňuje použít větu 2. Její aplikací dostaneme po zjednodušení

$$(\pi_1 \triangleright \ldots \triangleright \pi_{13})^{\downarrow\{8,9,10,11,12,14,16,17,18,19\}} = \\ = \mu^{\downarrow\{8\}} \triangleright \pi_6 \triangleright \pi_7 \triangleright \pi_8^{\downarrow\{12,19\}} \triangleright \pi_9 \triangleright \pi_{10}^{\downarrow\{14,16\}} \triangleright \pi_{11} \triangleright \pi_{12} \triangleright \pi_{13}^{\downarrow\{11,18,19\}},$$

kde $\mu^{\downarrow\{8\}} = (\pi_1 \triangleright \pi_3 \triangleright \pi_4 \triangleright \pi_5)^{\downarrow\{8\}}$.

Nyní již všechny další pokusy o nalezení Z a s určujících redukci selžou, neboli větu 2 nelze aplikovat. Ale bude ještě možné použít větu 1.

Máme distribuci o 10 dimenzích a chceme se dopracovat ke 3-dimenzionální distribuci s proměnnými X_{11}, X_{12}, X_{18}. Zbývá nám ještě marginalizovat 7 proměnných s indexy $8, 9, 10, 14, 16, 17, 19$.

Použijme nyní větu 1 k odstranění proměnné X_8 a dostaneme

$$(\pi_1 \triangleright \ldots \triangleright \pi_{13})^{\downarrow\{9,10,11,12,14,16,17,18,19\}} = \\ = \kappa_1 \triangleright \pi_7 \triangleright \pi_8^{\downarrow\{12,19\}} \triangleright \pi_9 \triangleright \pi_{10}^{\downarrow\{14,16\}} \triangleright \pi_{11} \triangleright \pi_{12} \triangleright \pi_{13}^{\downarrow\{11,18,19\}},$$

kde pro zjednodušení zápisu $\kappa_1(x_9) = \left(\mu^{\downarrow\{8\}} \circledcirc_{\emptyset} \pi_6\right)^{-\{8\}}$

Zbývající marginalizace probíhá analogicky s využitím věty 1. Zdůrazněme, že po každém použití této věty je třeba znovu aplikovat krok 1 algoritmu.

Závěr

Vytvořili jsme teoretické pozadí pro efektivní marginalizaci v rámci kompozicionálních modelů, hlavní přínos spočívá v inovativní možnosti marginalizace množiny proměnných v jednom výpočetně relativně jednoduchém kroku.

Další vylepšení procesu hledání redukce, analýza výpočetní složitosti a praktická implementace zůstávají výzvou pro budoucí práci v rámci problematiky.

Poděkování: Tento výzkum byl podpořen VŠE grantem IGA 611025.

Reference

1. V. Bína, R. Jiroušek, About an Effective Algorithm for Marginalization in Multidimensional Compositional Models. In: T. Kroupa, J. Vejnarová (eds.) Proceedings of the 8th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty. Třešť, Czech Republic, 2005, pp. 1-13.
2. Hugin Explorer, ver 6.0, Computer software, 2005. `http://www.hugin.com`
3. F. V. Jensen, *Bayesian Networks and Decision Graphs*, Springer Verlag, NY, 2001.
4. R. Jiroušek, Marginalization in composed probabilistic models. In: *Proc. of the 16th Conf. Uncertainty in Artificial Intelligence UAI'00* (C. Boutilier and M. Goldszmidt eds.), Morgan Kaufmann Publ., San Francisco, California, 2000, pp. 301-308.
5. R. Jiroušek, Decomposition of multidimensional distributions represented by perfect sequences. *Annals of Mathematics and Artificial Intelligence*, **35**(2002) pp. 215-226.
6. R. Jiroušek, What is the Difference between Bayesian Networks and Compositional Models? In: H. Noguchi, H. Ishii, M. Inuiguchi (eds.) Proceedings of the 7th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty. Awaji Yumebutai ICC, Japan, 2004, pp. 191-196.
7. S.L. Lauritzen, *Graphical Models*. Clarendon Press, Oxford, 1996.
8. R.D. Shachter, Probabilistic inference and influence diagrams. *Oper. Res.*, **36** (1988), pp. 589-604.
9. G. Shafer, *Probabilistic Expert Systems*. SIAM, Philadelphia, 1996.

Annotation:

Effective marginalization of compositional models

Graphical Markov Models became very popular and widely used because of the fact, that effective algorithms for computation with these multidimensional distributions were designed. Similar algorithms can be developed also for computation with Compositional Models, which form an alternative to Graphical Markov Models. This alternative gives us from the point of view of computational effectivity some advantages. Our paper uses those benefits and presents a theoretical basis as well as scheme of algorithms for computing of marginal distributions in compositional models, the task not solved in software packages for Bayesian networks processing.

Visualization of cosymmetric association rules using Hasse diagrams and concept lattices

Michal Burda

Dept. of Computer Science, FEI, VŠB Technical university of Ostrava,
17. listopadu 15, 708 33 OstravaPoruba, Czech Republic

Michal.Burda@vsb.cz

Abstract. Association rules are essential data mining tool and so-called cosymmetric rules form a wide subclass of them. The usual problem of association rules is wide amount of resultant rules – it is often hard to understand all of them. Various visualization methods provide great help in that area. In this paper, I discuss two new cosymmetric rules visualization methods based on Hasse diagrams and concept lattices.

Key words: cosymmetric rules, association rules, visualization, Hasse diagram with negative edges, concept lattice

1 Preface

In this paper, I focus on the possibility of visualizing cosymmetric rules [2]. Cosymmetric rule is a special case of the general notion "association rule", the technique used often in knowledge discovery from databases. In section 2, I briefly introduce the reader to cosymmetric rules. As an example of two visualization methods described in section 4, I use GLASS data taken from [5]. A short description of this dataset is given in section 3.

2 Introduction to cosymmetric association rules

Cosymmetric (or more precisely δ-cosymmetric) rules are motivated in [2]. We provide here a brief introduction only. Cosymetric rules are formulae of the form[1]

$$R(condition_1)[attribute] <^{\star}_{\delta;some-characteristic} R(condition_2)[attribute]. \quad (1)$$

Their meaning is that objects satisfying $condition_1$ have accordingly to $some-characteristic$ the values of attribute $attribute$ significantly lower than objects satisfying $condition_2$. Such rules are based on various statistical tests and model various knowledge of the form that "something is significantly higher than something else". Syntax of the rules complies the Probabilistic logic of typed relations, a language for expressing various association rules – see [3], [1] for more details.

[1] I use often the symbol $p<$ for cosymmetric rules. Note that $A < B$ equals to $B > A$.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 175–182, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

As an example, consider a data table R of patients suffering certain disease. One may be interested in the differences of blood pressure, for instance. So he or she may find the subsequent rule interesting:

$$R(\mathsf{sex} = \text{“male”})[\mathsf{pressure}] >^{\star}_{AW} R(\mathsf{sex} \neq \text{“male”})[\mathsf{pressure}]. \tag{2}$$

The rule simply says that ill men have significantly higher blood pressure than women. The acronym AW in the predicate $>^{\star}_{AW}$ stands for Aspin–Welch statistical test. It is a two sample test for rejecting the hypotheses about zero difference in position (mean value, average). Such rules can be generated from arbitrary data table automatically. However, even a relatively small data set results often into a wide collection of rules which is hard to get acquainted with it.

We apply some constraints on rules to filter out the uninteresting ones. Consider the general pattern of δ-cosymmetric rules (1). When mining such rules, we can generate and test virtually every combination of $condition_1$, $condition_2$ and *attribute*, but doing so makes not much sense. Look at the following rule – although it may be true, the analyst has probably no usage for it:

$$R(\mathsf{eyes} = \text{“blue”} \wedge \mathsf{sex} = \text{“male”})[\mathsf{fat}] >^{\star} R(\mathsf{age} > 30 \wedge \mathsf{wage} < \$200)[\mathsf{fat}]. \tag{3}$$

(It says that blue-eyed men are fatter than people older than 30 with wage below $200.) Therefore, the paper [2] introduces several schemes of interesting cosymmetric rules. In this paper, we focus on the scheme called “one-on-one”. This scheme simply dictates to mine the rules of the following pattern only:

$$R(B = b_i)[A] >^{\star} R(B = b_j)[A] \tag{4}$$

where attribute B called *class attribute* is of categorial domain $\{b_1, b_2, \ldots, b_n\}$, A is referred as *quantified attribute*, and $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$. For example, suppose the class attribute is age and the quantified attribute is heartbeat. If age is categorial attribute with five categories (“0–10”, “11–20”, “21–40”, “41–60” and “61 and more”), we test accordingly to scheme “one-on-one” the differences in heartbeat for every combination of ages. Generally, we should generate and test $\binom{n}{2}$ different formulae for class attribute with domain of size n.

3 The GLASS data

For testing purposes, we have mined the “one-on-one” cosymmetric rules from *Glass Identification Database* taken from [5]. It is a data table with 214 observations of the chemical analysis of 6 different types of glass. The study of classification of types of glass was motivated by criminological investigation. The data table contains 10 attributes: *RI* (refractive index), *Na* (Sodium), *Mg* (Magnesium), *Al* (Aluminum), *Si* (Silicon), *K* (Potassium), *Ca* (Calcium), *Ba* (Barium), *Fe* (Iron) and *type* (type of glass – the class attribute). Database contains six glass types: (1) Float processed building windows, (2) Float processed vehicle windows, (3) Non-float processed building windows, (4) Non-window glass of containers, (5) Non-window glass of tableware, (6) Non-window glass of headlamps. The search of cosymmetric rules involved to perform $9 \times \binom{6}{2} = 135$ tests. 54 of them passed, so we obtained 54 cosymmetric rules (see table 1).

Table 1. The cosymmetric rules from GLASS data. (I use a compressed representation of cosymmetric rules here. E.g. $1 > 6$ in a row of RI stands for a rule $R(\mathsf{type} = 1)[\mathsf{RI}] >^{\star}_{AW} R(\mathsf{type} = 6)[\mathsf{RI}]$ and so on.)

Attribute	Rules
RI	$1 > 6,\ 2 > 6$
Na	$1 < 5,\ 1 < 6,\ 2 < 3,\ 2 < 5,\ 2 < 6,\ 3 > 4,\ 3 < 5,\ 3 < 6,\ 4 < 5,\ 4 < 6$
Mg	$1 > 2,\ 1 > 4,\ 1 > 5,\ 1 > 6,\ 2 < 3,\ 2 > 4,\ 2 > 5,\ 2 > 6,\ 3 > 4,\ 3 > 5,\ 3 > 6$
Al	$1 < 2,\ 1 < 4,\ 1 < 6,\ 2 > 3,\ 2 < 4,\ 2 < 6,\ 3 < 4,\ 3 < 6,\ 4 > 5,\ 5 < 6$
Si	$3 < 6$
K	$1 < 2,\ 1 > 5,\ 2 > 5,\ 3 > 5,\ 4 > 5,\ 5 < 6$
Ca	$2 > 6,\ 3 < 4,\ 4 > 6$
Ba	$1 < 6,\ 2 < 6,\ 3 < 6,\ 4 < 6,\ 5 < 6$
Fe	$1 > 5,\ 1 > 6,\ 2 > 5,\ 2 > 6,\ 3 > 5,\ 5 < 6$

4 Visualization of GLASS data rules

Although the Aspin–Welch one-on-one cosymmetric rules are easy to interpret, the relatively large amount of mined rules is rather hard to understand. In this section we introduce two alternative approaches of visualizing the mined rules. Both of them are based on Hasse diagrams – they differ in semantics only. First approach considers the mined rules as a quasi–ordered set and visualizes it using Hasse diagram with negative edges. The second approach creates concept lattices from the set of rules and uses traditional methods to render them as images.

4.1 Visualization using Hasse diagram with negative edges

A *partially ordered set* (or *poset* for short) is a set P equipped with a partial order relation Q. A *partial order* is a binary relation $Q \subseteq P \times P$ which is reflexive, antisymmetric, and transitive. *Hasse diagram* is a simple picture of a finite partially ordered set. Concretely, Hasse diagram is a graph where each member of a set P is represented as an vertex. A line is drawn upwards from x to y if $\langle x, y\rangle \in Q$, $x \neq y$ and there is no z such that $\langle x, z\rangle \in Q \wedge \langle z, y\rangle \in Q$. Furthermore it is required that the vertices are positioned in such a way that each line meets exactly two vertices: its two endpoints. In the other words, $\langle x, y\rangle \in Q$ iff y is positioned above x and there exists path from y to x. Consider a set of cosymmetric rules created from one quantified attribute. We can define:

$$H_{class,attr} = \left\{ \langle x, y\rangle : x = y \vee R(class = x)[attr] <^{\star} R(class = y)[attr] \right\}. \quad (5)$$

We write simply H_{attr} if *class* is obvious from the context. For example, let *class* be type and *attr* be Fe from GLASS then: (cf. with table 1)

$$\begin{aligned} H_{\mathsf{type,Fe}} = \{\ & \langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle, \langle 4,4\rangle, \langle 5,1\rangle, \\ & \langle 5,2\rangle, \langle 5,3\rangle, \langle 5,5\rangle, \langle 5,6\rangle, \langle 6,1\rangle, \langle 6,2\rangle, \langle 6,6\rangle \ \}. \end{aligned} \quad (6)$$

Obviously, the relation H is reflexive and the properties of cosymmetric rules discussed in [2] guarantee that it is also anti-symmetric. However, the relation H is not transitive. Let $A = R(class = x)[attr]$, $B = R(class = y)[attr]$ and $C = R(class = z)[attr]$. There are some (rare) cases where $A <^{\star} B$ and $B <^{\star} C$

(a) Hasse diagram with negative edge $\langle 1,3\rangle$ (dotted line).

(b) Concept lattice of cosymmetric rules breaking transitivity.

Fig. 1. Cosymmetric rules breaking transitivity.

are true, but $A <^\star C$ is false. Fortunately, the properties of cosymmetric rules justify the opposite rule $A >^\star C$ not to be possible in that case, too. That behaviour is caused with the test behind the rule – e.g. two-sample Student's test behaves that way. It is due the effect of dissimilar dispersion that must be taken into considerations. (Also please note that if the rule is not true it does not imply the opposite rule is valid! It is due the many-valued logic that works behind the rules. See [1], [2], [3] for details.)

Let now $Tr(H)$ be a transitive closure of relation H then $Tr(H)$ is evidently partial order. Let us denote $N = Tr(H) - H$. N is a relation of pairs where the transitivity is broken in H. We can now draw a Hasse diagram of $Tr(H)$ and highlight vertex pairs where the transitivity is broken – we can add some extra edges to Hasse diagram accordingly to items of a relation N (call this edges *the negative edges*). If two nodes of Hasse diagram are connected with negative edge it means that although there is a solid-line path between them, the appropriate order entry does not exist in relation H. As an example consider relation

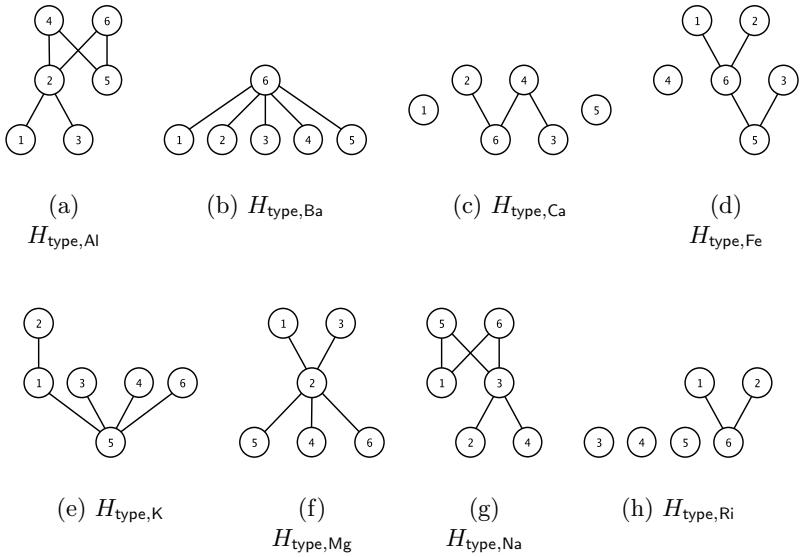
$$G = \left\{ \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,3\rangle \right\}. \tag{7}$$

Its transitive closure equals

$$Tr(G) = \left\{ \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,3\rangle \right\}, \tag{8}$$

so $N = Tr(G) - G = \left\{ \langle 1,3\rangle \right\}$. The corresponding Hasse diagram with negative edges is depicted on figure 1(a). We can see that the classes are ordered 1, 2, 3 from highest to lowest, but there is no evidence of class 1 being significantly higher than class 3.

Hasse diagram of $H_{\mathsf{type,Fe}}$ is depicted on figure 2(d) (it contains no negative edges). It greatly illustrates the structure of Fe values. We can see immediately that there is no statistical difference between class 4 and any other class (with respect to the amount of iron present in the sample, of course). We can see also that glass of type 5 contains the least quantity of iron from types 1, 2, 3, 5 and 6. On the other hand, most iron is presented in classes 1 and 2, and so on... Figure 2 depicts Hasse diagrams of most results from GLASS analysis.



(a) $H_{\mathsf{type,Al}}$ (b) $H_{\mathsf{type,Ba}}$ (c) $H_{\mathsf{type,Ca}}$ (d) $H_{\mathsf{type,Fe}}$

(e) $H_{\mathsf{type,K}}$ (f) $H_{\mathsf{type,Mg}}$ (g) $H_{\mathsf{type,Na}}$ (h) $H_{\mathsf{type,Ri}}$

Fig. 2. Hasse diagrams of all one-on-one cosymmetric rules mined from GLASS data (cf. with table 1). No negative edges were needed.

4.2 Visualization using concept lattices

The second approach of cosymmetric rule visualization presented in this paper is based on concept lattices. We start with the very brief remind of basic mathematical notions. For detailed information see e.g. [4].

Formal context is a triplet $\langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{O}$ represents a finite set of objects, $\mathcal{A}$ is a finite set of attributes and $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$ is an *incidence relation*. Each member $\langle o, a \rangle \in \mathcal{R}$ expresses that the object $o \in \mathcal{O}$ contains the attribute $a \in \mathcal{A}$. For a set $O \subseteq \mathcal{O}$, we define $\phi(O) = \{a : (\forall o \in O)(\langle o, a \rangle \in \mathcal{R})\}$, and for $A \subseteq \mathcal{A}$, $\psi(A) = \{o : (\forall a \in A)(\langle o, a \rangle \in \mathcal{R})\}$. *Formal concept* in context $\langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$ is a pair $\langle O, A \rangle$, where $O \subseteq \mathcal{O}$ is *extent*, $A \subseteq \mathcal{A}$ is *intent*, and $\phi(O) = A \wedge \psi(A) = O$. A set of all concepts in $\langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$ is denoted with

$$\mathcal{B}(\mathcal{O}, \mathcal{A}, \mathcal{R}) = \{\langle O, A \rangle : O \in \mathcal{O} \wedge A \in \mathcal{A} \wedge \phi(O) = A \wedge \psi(A) = O\}. \quad (9)$$

Concept lattice is a set $\mathcal{B}(\mathcal{O}, \mathcal{A}, \mathcal{R})$ together with a relation $\leq_c$ defined as $\langle O_1, A_1 \rangle \leq_c \langle O_2, A_2 \rangle$ iff $O_1 \subseteq O_2$ (or equivalently $A_2 \subseteq A_1$). Since concept lattice is specialization of partially ordered set, we use standard Hasse diagrams to visualize concept lattices, too (without the need to use negative edges).

The idea of visualizing cosymmetric rules using concept lattices is based on the fact that we can create formal context from a set of cosymmetric rules very easily. Let *class* be the class attribute with finite domain and *quants* be a set of quantified attributes. The formal context $K_{class,quants} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$ is defined

Table 2. Context derived from cosymmetric rules based on quantified attribute Fe in GLASS data

$K_{\mathsf{type},\{\mathsf{Fe}\}}$	$\leq$ 1 (Fe)	$\leq$ 2 (Fe)	$\leq$ 3 (Fe)	$\leq$ 4 (Fe)	$\leq$ 5 (Fe)	$\leq$ 6 (Fe)
1	×					
2		×				
3			×			
4				×		
5	×	×	×		×	×
6	×	×				×

as $\mathcal{O}$ to be equal to the domain of the attribute *class* and

$$\mathcal{A} = \{\langle q, o' \rangle : q \in quants \wedge o' \in \mathcal{O}\}. \tag{10}$$

The single context attribute $a = \langle q, o' \rangle$ means intuitively a property of "being lower or equal to o' in value of q". Thus the relation $\mathcal{R}$ should be defined as

$$R = \Big\{ \langle o, a \rangle : o \in \mathcal{O} \wedge a = \langle q, o' \rangle \in \mathcal{A} \wedge \big(o = o' \vee R(class = o)[q] <^{\star} R(class = o')[q]\big) \Big\}. \tag{11}$$

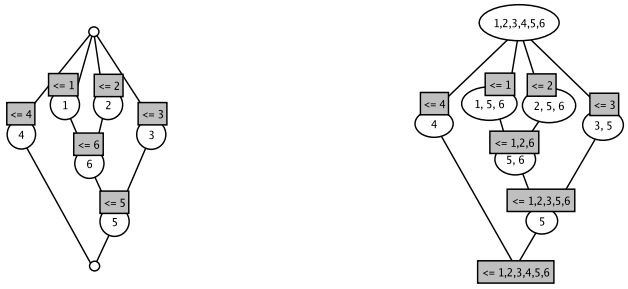
Take again a set of cosymmetric rules for class attribute type and quantified attribute Fe. For context $K_{\mathsf{type},\{\mathsf{Fe}\}} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$ we have the following:

$$\mathcal{O} = \{1, 2, 3, 4, 5, 6\}, \quad \mathcal{A} = \{\langle Fe, 1 \rangle, \langle Fe, 2 \rangle, \ldots, \langle Fe, 6 \rangle\}. \tag{12}$$

Table 2 summarizes values of relation $\mathcal{R}$. Figure 3(a) depicts the concept lattice. Nodes represent concepts and numbers inside the circles denote glass classes that first appear in concepts when traversing the diagram from bottom to top. Labels in rectangles stand for a property of "being lower or equal to given class in amount of Fe". Only top-most concepts are labeled with such attributes. Lower concepts share the attributes with its ascendants (see also figure 3(b)).

Practically, we can interpret such diagrams as follows. We should keep track of a node and look upwards on other nodes and its attributes (denoted with rectangular labels). For instance, look in figure 3(a) again. The fact that node 6 is situated below (and interconnected with) nodes labelled "≤ 1" and "≤ 2" says that the amount of Fe is in class 6 significantly lower than in classes 1 and 2. Moreover, we see immediately the class 5 to be one of the most different because there are labels "≤ 1", "≤ 2", "≤ 3", "≤ 6" above the node 5. One can see some other concept lattices in figure 4. Concept lattice diagram corresponding to Hasse diagram with negative edge (fig. 1(a)) is in figure 1(b) – we see immediately that the only classes lower or equal to 1 are classes 1 and 2 because there is descending path from node containing rectangle with "≤ 1" to nodes labelled with 1 or 2, respectivelly. Similarly, the only classes lower or equal to 2 are classes 2 and 3.

Visualization of cosymmetric rules using concept lattices has one major advantage. Unlikely Hasse diagram, two or more concept lattices can be combined into one big diagram very naturally. We can do so simply by joining several contexts to one. The result of join of two contexts $\langle \mathcal{O}, \mathcal{A}_1, \mathcal{R}_1 \rangle$ and $\langle \mathcal{O}, \mathcal{A}_2, \mathcal{R}_2 \rangle$



(a) Common style of concept lattice visualization

(b) Expanded illustrative version of the diagram

Fig. 3. Concept lattice created from context $K_{\mathsf{type},\{\mathsf{Fe}\}}$

(a) $K_{\mathsf{type},\{\mathsf{Al}\}}$ (cf. with fig. 2(a))

(b) $K_{\mathsf{type},\{\mathsf{Ba}\}}$ (cf. with fig. 2(b))

Fig. 4. Some concept lattices created from cosymmetric rules of the GLASS analysis

is the context $\langle \mathcal{O}, \mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2 \rangle$. Example in table 3 makes it clear. Such context can be naturally transformed to concept lattice (see figure 5).

Interpretation of such concept lattice is alike. We see e.g. immediately that amount of Al of each class is lower or equal to class 2 because rectangle "$\leq$ 2 (Al)" is at the very top of the lattice. We also see that there is no difference in Si, Ri, Na... between class 1 and any other class because there is no descending path from node labelled with rectangle "$\leq$ 1 (Si, Ri, Na...)" to other nodes with the class label, and so on.

5 Conclusion

I have presented here two new methods of cosymmetric rules visualization. Both methods are based on Hasse diagrams – the first method uses the new notion

Table 3. Contexts of first three classes created from GLASS data cosymmetric rules

(a) For attribute Al

$K_{\text{type},\{\text{Al}\}}$	≤ 1 (Al)	≤ 2 (Al)	≤ 3 (Al)
1	×	×	
2		×	
3		×	×

(b) For attribute Ba

$K_{\text{type},\{\text{Ba}\}}$	≤ 1 (Ba)	≤ 2 (Ba)	≤ 3 (Ba)
1	×		
2		×	
3			×

(c) Joined Al and Ba

$K_{\text{type},\{\text{Al,Ba}\}}$	≤ 1 (Al)	≤ 2 (Al)	≤ 3 (Al)	≤ 1 (Ba)	≤ 2 (Ba)	≤ 3 (Ba)
1	×	×		×		
2		×			×	
3		×	×			×

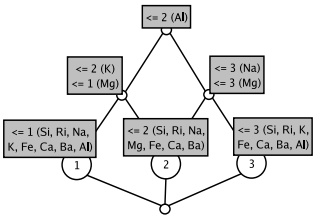


Fig. 5. Concept lattice of all cosymmetric rules related to first three glass classes

of *Hasse diagram with negative edges*. The second method is based on concept lattices viewed with classical Hasse diagrams. My future work in this area will address the possibility and benefits of trilattices [6] usage. Generally, the study of formal cosymmetric rules properties is my main scope.

References

1. Burda, M., Hynar, M., and Šarmanová, J. Probabilistic logic of typed relations (in Czech). In *Znalosti, poster proceedings* (Vysoké Tatry, 2005), pp. 13–16.
2. Burda, M., Mindek, M., and Šarmanová, J. Characteristics of cosymmetric association rules. In *DATESO* (Desná, 2005), K. Richta, V. Snášel, and J. Pokorný, Eds., pp. 20–31.
3. Burda, M., Mindek, M., and Šarmanová, J. Using relational operations to express association rules. In *SYRCoDIS* (St.-Petersburg, Russia, 2005).
4. Ganter, B., and Wille, R. *Formal Concept Analysis*. Springer-Verlag Berlin Heidelberg, 1999.
5. S. Hettich, C. B., and Merz, C. UCI repository of machine learning db's, 1998. Internet: `http://www.ics.uci.edu/~mlearn/MLRepository.html` (May 2005).
6. Wille, R. Triadic concept graphs. In *ICCS'98* (1998), M.-L. Mugnier and M. Chein, Eds., Springer-Verlag Berlin Heidelberg, pp. 194–208.

Případové usuzování v expertním systému NEST

Vladimír Laš, Petr Berka

Katedra informačního a znalostního inženýrství, FIS, VŠE - Praha
Praha 3, Nám. W. Churchilla 4

{lasv,berka}@vse.cz

Abstrakt. Článek popisuje základní způsob případového usuzování, který byl implementován do expertního systému NEST, původně pouze pravidlového. Odvozování probíhá nad skladem předchozích případů, který (na rozdíl od pravidlové báze znalostí) nemusí vytvářet expert. V systému jsou implementovány dva způsoby případového odvozování – kompozicionální a logický – lišící se způsobem využívání případů ve skladu. Tyto způsoby jsou inspirovány kompozicionálním a logickým způsobem práce s neurčitostí v síti pravidel původního systému NEST. V závěru jsou nastíněna další možná rozšíření systému – především hlubší integrace pravidlového a případového usuzování.

Klíčová slova: případové usuzování, expertní systém

1 Úvod

Tento text popisuje, jakým způsobem funguje případové usuzovaní, které bylo přidáno do expertního systému NEST.

Odvozování (kompozicionální respektive logické) v tomto expertním systému bylo původně založeno pouze na pravidlech (klasická IF THEN pravidla). Důvodů pro přidání případového usuzování do systému je několik. Mezi hlavní patří problémy s tvorbou bází znalostí. Pro pravidlové usuzování je nutné mít předem vytvořenou kvalitní bázi znalostí, podle níž systém odvozuje závěry. Pro případové usuzování nám stačí mít sklad případů s dostatečným počtem dříve vyřešených případů (o kterých máme alespoň určitou jistotu, že byly řešeny správně). Tento sklad případů lze získat jednodušším způsobem než kvalitní bázi znalostí. Dalším důvodem je např. možnost odvodit výsledky různými způsoby a tím ověřit jejich správnost.

Navrhovaný způsob usuzování vychází ze stávajících možností systému NEST – ať už jde o reprezentaci příkladů pomocí atributů a výroků, nebo o práci s neurčitostí.

2 Systém NEST

V tomto textu nebudeme popisovat NEST, o tomto systému se lze více dozvědět v [1] či [2]. Zde bychom zmínili pouze dvě základní věci, které přímo souvisí s případovým usuzováním – reprezentace znalostí pomocí atributů, výroků a pravidel a odvozování a způsob práce s neurčitostí .

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 183–190, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2.1 Atributy, výroky a pravidla

Systém NEST nabízí pro reprezentování znalostí atributy, výroky pravidla, kontexty a integritní omezení.

Atributy se používají k vytváření výroků; každý atribut má nějakou hodnotu, podle které se přiřazují váhy příslušným výrokům. Systém NEST umožňuje pracovat s atributy binárními, nominálními a numerickými. Od binárních atributů je odvozen jediný výrok (a případně jeho negace). Od nominálních atributů jsou odvozeny výroky, které odpovídají přípustným hodnotám atributu (systém rozlišuje jednoduché nominální atributy – tyto atributy mohou při konzultaci získat pouze jednu hodnotu a množinové nominální atributy – tyto atributy mohou při konzultaci získat více hodnot). Od numerických atributů jsou odvozeny výroky, které odpovídají rozdělení oboru přípustných hodnot na fuzzy intervaly.

IF-THEN pravidla představují původní základní prostředek pro reprezentování znalostí v systému NEST. Předpoklad pravidla je tvořen disjunktivní normální formou (disjunkce konjunkcí) literálů (výroků nebo jejich negací), závěr pravidla je tvořen seznamem (konjunkcí) literálů. Systém NEST nabízí tři typy pravidel:

- *kompozicionální pravidla* jsou pravidla, kde každý závěr pravidla je doplněn váhou pravidla, která zhruba řečeno vyjadřuje do jaké míry tento závěr platí, pokud předpoklad platí s jistotou. V průběhu konzultace se skládají dílčí příspěvky pravidel, která vedou ke stejnému závěru – proto označení kompozicionální,
- *apriorní pravidla* jsou kompozicionální pravidla bez předpokladu; tato pravidla umožňují přiřadit výrokům implicitní váhu,
- *logická pravidla* jsou nekompozicionální pravidla bez vah. Je-li předpoklad takovéhoto pravidla s jistotou splněn, váha závěru je známa rovněž s jistotou a závěr se považuje za plně vyhodnocený – v ostatních případech se pravidlo neaktivuje. Pouze logická pravidla mohou dát absolutní jistotu +-1.

2.2 Odvozování a práce s neurčitostí

Systém NEST umožňuje pracovat s tzv. váhami, které vyjadřují neurčitost v bázi znalostí (v podobě vah pravidel) i neurčité odpovědi během konzultace (jako váhy přiřazené příslušným výrokům na základě hodnot atributů zadaných při konzultaci). Práce s neurčitostí vychází z algebraické teorie P. Hájka [3]. Váha je tedy z intervalu [-1,1] a v systému jsou definovány kombinační funkce NEG (pro výpočet váhy negace výroku), CONJ (pro výpočet váhy konjunkce výroků), DISJ (pro výpočet váhy disjunkce výroků), CTR (pro výpočet příspěvku pravidla k váze závěru) a GLOB (pro složení příspěvků více pravidel se stejným závěrem). Na rozdíl od původní algebraické teorie není váha vyjádřena jedním číslem, ale reprezentuje ji interval vah. Interval chápeme tak, že výrok může získat libovolnou váhu z daného intervalu. Interval tedy simuluje opakované konzultace pro všechny možné hodnoty z intervalu. Vzhledem k monotonii kombinačních funkcí (funkcí používaných při odvozování) stačí pracovat pouze s mezemi intervalů.

Použité typy pravidel nabízejí dva možné způsoby odvozování:

- *kompozicionální odvozování* v rámci kompozicionálních a apriorních pravidel. V průběhu konzultace se skládají dílčí příspěvky všech pravidel vedoucích ke stejnému závěru. Nějaké pravidlo je pak aktivováno a dá nenulový příspěvek k váze závěru, pokud (1) kontext pravidla (je-li zadán) je odvozen s kladnou vahou a rovněž (2) předpoklad pravidla je odvozen s kladnou vahou – kontext (je-li zadán) přitom modifikuje váhu pravidla. Jde tedy o způsob dovozování známý ze systému SAK (a analogický systémům Equant, FEL-EXPERT, MYCIN, PROSPECTOR a pod.) ,
- *nekompozicionální odvozování* v rámci logických pravidel. Jak již bylo uvedeno v předcházejícím odstavci, logická pravidla odvozují závěr s jistotou – s využitím klasického dedukčního pravidla modus ponens. V průběhu konzultace pak jedno aktivované pravidlo plně vyhodnotí daný závěr. "Skládání" příspěvků pravidel tedy má charakter disjunkce, tak jak je tomu například v komerčních expertních systémech Nexpert Object nebo Kappa PC.

Z tohoto způsobu rozdělení odvozování na kompozicionální a logické vychází i způsob práce případového usuzování, jak je uvedeno dále.

3 Jak funguje případové usuzování

Případové usuzování (Case-Based Reasoning, CBR) je založeno na principu analogie: v neznámé situaci použij to řešení, které se osvědčilo v situaci podobné. Znalosti jsou tedy reprezentovány v podobě databáze již vyřešených problémů (případů) a usuzování má podobu hledání případu, který bude nejpodobnější nastalé rozhodovací situaci [4].

Případy vyjadřující znalosti v systémech CBR lze reprezentovat v zásadě dvěma způsoby:

1. jako nestrukturované vektory hodnot atributů,
2. jako strukturované objekty, nesoucí informace o situacích, kdy je použití případu relevantní i o akcích, které se mají provést.

Použitý způsob reprezentace pak předurčuje způsob ukládání (indexování) případů i měření podobnosti. V prvním případě vystačíme s některou s běžných metrik. Pro měření vzdálenosti mezi numerickými hodnotami se obvykle používá eukleidovská metrika

$$d_E(x_1, x_2) = \sqrt{\sum_{j=1}^{m} \delta_E(x_{1j}, x_{2j})}, \quad kde\ \delta_E(x_{1j}, x_{2j}) = (x_{1j} - x_{2j})^2 \qquad (\mathbf{1})$$

pro měření vzdálenosti mezi kategoriálními hodnotami se používá tzv. překrytí (overlap)

$$d_O(x_1, x_2) = \sum_{j=1}^{m} \delta_O(x_{1j}, x_{2j}),$$
$$kde: \begin{array}{l} \delta_O(x_{1j}, x_{2j}) = 0 \; pro \; x_{1j} = x_{2j} \\ \delta_O(x_{1j}, x_{2j}) = 1 \; pro \; x_{1j} \neq x_{2j} \end{array} \quad (2)$$

kde $\mathbf{x}_1$, $\mathbf{x}_2$ jsou dva vektory o *m* složkách.

V systému NEST jsme zvolili první způsob reprezentace (a tedy i odvozování). Případy jsou chápany jako údaje z provedených konzultací: obsahují tedy informace o odpovědích na dotazy i o odvozených cílech – obojí vyjádřeno jako váhy příslušných výroků. Odvozování vah cílů nové konzultace pak probíhá na základě dřívějších případů (konzultací). K tomu je zapotřebí, abychom měli sklad případů s dostatečným počtem dříve vyřešených případů. Nová konzultace (nový případ) je pak porovnávána s těmito staršími případy, je určena podobnost k jednotlivým případům a na základě této podobnosti je spočtena váha cílů nové konzultace.

Podobně jako v případě použití pravidel, i v případě použití případů nabízí NEST dva přístupy: kompozicionální odvozování a logické odvozování. Oba způsoby definují 3 základní funkce:

- Podobnost vah dvou výroků – výsledkem je číslo z intervalu [-1;1], které určuje, jak „podobné“ jsou váhy dvou stejných výroků z různých konzultací (při práci s intervaly musíme počítat s intervalem $[w_1, w_2]$, kde w_i je prvkem [-1;1]).
- Podobnost dvou konzultací – výsledkem je opět číslo z intervalu [-1;1], které určuje, jak „podobné“ jsou dvě konzultace. Funkce pro výpočet využívá podobnost vah jednotlivých výroků.
- Výpočet váhy cílů - je funkce, která na základě podobnosti 2 konzultací (podobnosti nové konzultace s jednotlivými starými případy) vypočítá váhy závěrů nové konzultace.

Základní rozdíl mezi kompozicionálním a logickým způsobem odvozování spočívá v tom, že zatímco v kompozicionálním způsobu jsou brány v úvahy všechny případy ve skladu a váha závěru je odvozována skládáním vah těchto případů, v logickém způsobu odvozování je vybrán nejpodobnější případ a váha závěru je určena na základě tohoto případu.

Cíle (cílové výroky) nové konzultace mohou být určeny dvěma způsoby. Pokud v bázi znalostí existují pravidla, mohou být cíle určeny na základě těchto pravidel stejným způsobem jako v případě pravidlového usuzování. Druhou možností je přímo ve skladu případů označit atributy, které mají být brány jako cílové.

Jednotlivé atributy mohou mít ve skladu případů určeny svoji váhu. Tato váha je pak zohledňována při výpočtu podobnosti 2 konzultací (atributy s vyšší váhou se na odvozování podílejí více než atributy s nižší váhou). Pokud atribut nemá váhu zadanou, bere se automaticky maximální váha (tedy 1).

4 Kompozicionální odvozování

4.1 Výpočet podobnosti vah dvou výroků

Mějme výrok *v*, který má v konzultaci k_a váhu w_a a v konzultaci k_b váhu w_b.
Podobnost vah výroku pak definujeme jako:

$$pv_k(k_a, k_b) = 1 - |w_a - w_b| \qquad pv_k \in [-1;1] \tag{3}$$

pv_k = -1 znamená, že váhy jsou zcela odlišné, pv_k = 1 znamená, že váhy jsou stejné.
Pro práci s intervaly (uvažujme intervaly $[w_{a1}, w_{a2}]$ a $[w_{b1}, w_{b2}]$):
Jestliže $aw_1 > bw_2$ nebo $bw_1 > aw_2$ (intervaly se nepřekrývají) pak

$$pv_k(k_a, k_b) = [\min(1 - |w_{a1} - w_{b2}|, 1 - |w_{a2} - w_{b1}|), \max(1 - |w_{a1} - w_{b2}|, 1 - |w_{a2} - w_{b1}|)] \tag{4}$$

Jinak (intervaly se překrývají):

$$x_1 = |w_{a1} - w_{b1}|,\ x_2 = |w_{a1} - w_{b2}|,\ x_3 = |w_{a2} - w_{b1}|,\ x_4 = |w_{a2} - w_{b2}|$$
$$pv_k(k_a, k_b) = [1 - \max(x_1, x_2, x_3, x_4), 1] \tag{5}$$

4.2 Výpočet podobnosti dvou konzultací

Mějme konzultaci k_1 a konzultaci k_2. Podobnost konzultací pak definujeme jako:

$$pk(k_1, k_2) = \frac{\sum_{a\ je\ dotaz} \frac{wa}{nva} \sum_{i=1}^{pva} pv(v_{k1i}, v_{k2i})}{\sum_{a\ je\ dotaz} wa} \tag{6}$$

kde

- v_{kji} značí i-tý výrok (odvozený od atributu *a*, který je dotazem) v případu k_j
- *wa* je váha atributu *a* nastavená v bázi znalostí,
- *nva* je počet výroků, které jsou odvozeny od atributu *a* (koeficient 1/*nva* zavádíme proto,že každý atribut má být při výpočtu podobnosti zohledňován jen jednou; je-li tedy od atributu odvozeno vice výroků, je tímto způsobem celkově.započítán jen jednou)
- Σ probíhá přes všechny atributy-dotazy

pk = -1 znamená, že případy (váhy odpovídajících si výroků) jsou zcela odlišné, *pk* = 1 znamená, že případy (váhy odpovídajících si výroků) jsou stejné.
Pro práci s intervaly se podle vzorce zvlášť spočte hodnota pro horní mez a zvlášť pro dolní mez.

4.3 Výpočet váhy cíle nové konzultace

Mějme bázi znalostí s případy k_i a novou konzultaci (nový případ) k_n. Mějme cílový výrok c s vahou w_c

$$w_{cn} = \frac{\sum_i \max(0, pk(k_n, k_i)).w_{ci}}{pkon_+} \quad (7)$$

kde

- $pkon_+$ je počet případů v bázi znalostí kde $pk(k_n,k_i) > 0$
- Σ jde přes všechny případy v bázi znalostí.

Jedná se vlastně o vážený součet vah cílů jednotlivých případů, kde jako váhy slouží podobnost jednotlivých případů s novou konzultace.
Pro práci s intervaly se podle vzorce zvlášť spočte hodnota pro horní mez a zvlášť pro dolní mez.

5 Logické odvozování

5.1 Výpočet podobnosti vah dvou výroků

Mějme výrok *v*, který má v konzultaci k_1 váhu w_1 a v konzultaci k_1 váhu w_2. Dále mějme zadaný práh podobnosti p. Podobnost vah výroků pak definujeme jako:

$$pv_l(k_1,k_2) = 1 \; jestliže \; pv_k(k_1,k_2) > p$$
$$pv_l(k_1,k_2) = -1 \; jinak \quad (8)$$

To znamená, že pokud podobnost vah výroků spočtena kompozicionálním přístupem přesáhne zadaný práh, pak je logická podobnost rovna 1, jinak se rovná -1. Pro práci s intervaly se spočte stejným způsobem pro minimální i maximální váhu.

5.2 Výpočet podobnosti dvou konzultací

Počítá se stejným způsobem jako u kompozicionálního přístupu.

5.3 Výpočet váhy cíle nové konzultace

Mějme bázi znalostí s případy k_i a novou konzultaci k_n. Mějme cílový výrok c s vahou w_c

$$w_{cn} = \frac{\sum_{i \in M} w_{ci}}{|M|} \quad (9)$$
$$kde \quad M = \{i \mid i = \arg\max_j [pk(k_n, k_j)]\}$$

Nebo-li váha nové konzultace se rovná váze konzultace, která je nejpodobnější a v případě, že je nejpodobnějších konzultací více, vezme se průměr z jejich vah.

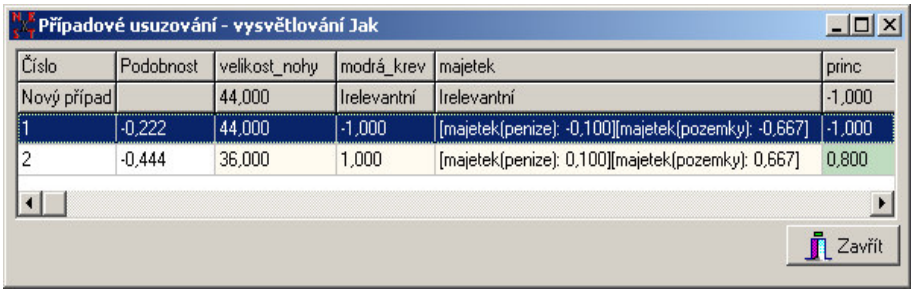
Pro práci s intervaly je případ s podobností [$a1$, $a2$] podobnější nové konzultaci než případ s podobností [$b1$, $b2$] jestliže:

$$(a_1 + a_2) - (b_1 + b_2) > 0 \qquad (10)$$

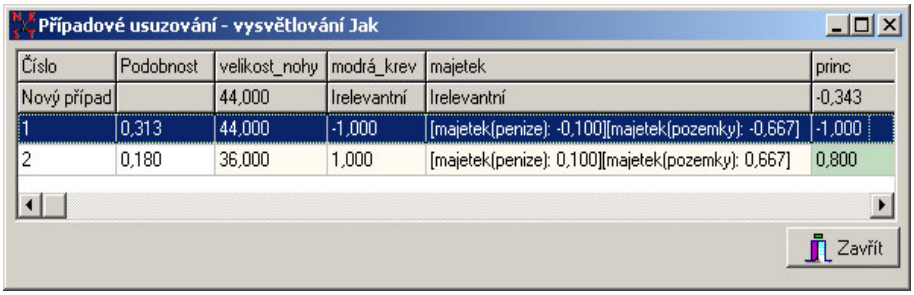
6 Implementace systému

Implementace případového usuzování v systému NEST plynule navazuje na původní implementaci systému, jak je popsaná v [1]. Zatím je vytvářena pouze stand-alone verze (klient-server verze bude dodělána až pod odladění všech problémů).

Konzultace je možno provádět dvěma způsoby: pomocí dotazníku nebo nahráním odpovědí ze souboru. Oproti konzultaci s bází pravidel tedy odpadá dialogový režim práce. Podobně jako u pravidlového odvozování ani zde nemusí být zadány hodnoty všech atributů. Nezodpovězené dotazy získají implicitní váhu „neznámá“ (w=[-1,1]) případně „irelevantní“ (w=0). Obr. 1 a Obr. 2 ukazují příklad konzultace. Při logickém odvozování je váha cíle (výroku princ) stejná jako váha cíle bližšího z případů 1 a 2 (jde tedy vlastně o analogický postup, jaký nabízí metoda 1-nn), při kompozicionálním odvozování je váha cíle odvozena z vah cíle u všech případů v bázi.



Obr. 1 Příklad logického odvozování



Obr. 2 Příklad kompozicionálního odvozování

Sklad případů je nyní reprezentován pomocí souboru xml (obdobně jako báze znalostí s pravidly), přičemž ukáže-li se v budoucnu, že případové odvozování nad tímto souborem probíhá příliš pomalu, bude vytvořen vlastní formát pro skladování dat (případně bude použita relační databáze).

7 Další rozvoj systému

V prvním kroku rozvoje systému je potřeba především sehnat dostatečně obsáhlý sklad případů a vytvořit pravidlovou bázi znalostí z příslušné oblasti. Na těchto základech by mělo proběhnout důkladné testování výše popsaných algoritmů.

V dalším rozvoji by mělo dojít k užšímu propojení případového a pravidlového usuzování. Jednou uvažovanou možností je integrování obou přístupů, kdy výsledky konzultace získané zvlášť na základě pravidel a zvlášť na základě případů budou kombinovány do definitivního závěru konzultace. Druhou uvažovanou možností využití jen jednoho způsobu založené na automatickém (či poloautomatickém) rozhodování, který způsob je pro danou konzultaci vhodnější.

V poslední části rozšíření bude snaha naučit systém upravovat pravidla v bázi znalostí na základě skladu případů.

Reference

1. Berka P., Laš V., Svátek V., Nový expertní systém NEST. Brno 25.02.2004 – 27.02.2004. In: Snášel V. (ed.). *Znalosti 2004*. Ostrava : VŠB TU Ostrava, 2004, s. 256–267. ISBN 80-248-0456-5
2. Laš V. http://lisp.vse.cz/NEST, Expertní systém NEST
3. Hájek P. Combining Functions for Certainty Factors in Consulting Systems. *Int. J. Man-Machine Studies 22*, 1985
4. Watson I., Marir F., Case-based reasoning: An review. *The Knowledge Engineering Review*, Vol. 9:4, 1994, 327-354.

Annotation:
Case-Based Reasoning in the Expert System NEST
This paper describes a basic type of case-based reasoning as it was implemented in the expert system NEST (originally a rule-based system). The reasoning proceeds over the cases storage, which doesn´t need to be created by expert contrary to the rule knowledge base. There are two types of case-based reasoning in the system – compositional and logical. These types differ in the way of using cases storage. Future development of the system is outlined at the end of the paper.

Vývoj systému NEST je částečně podporován granty GACR 201/05/0325 a IGA 25/05

Information Retrieval from Web Pages Employing Classification and Clustering Methods

Kristína Machová, Valentín Maták, and Peter Bednár

Department of Cybernetics and Artificial Intelligence,
Technical University, Letná 9, 042 00 Košice

Kristina.Machova@tuke.sk, valentin.matak@centrum.sk,
Peter.Bednar@tuke.sk

Abstract. The paper describes possible representation models and ways of weighting text documents, which can be found on the Internet. The focus is on automatic extraction of information from texts including pre-processing of text documents. The paper presents also results of experiments, which were carried out using the 20 News Groups and Reuters-21578 collections of documents. These experiments enabled to uncover how the cardinality of training set and a suitable weighting of text documents can influence the precision of document classification. The results of experiments with k-means clustering and k-means clustering with controlled initialisation are presented as well.

1 Introduction

A lot of information is stored on various places of the world in an electronic form. This paper presents some aspects of information retrieval [10] from web pages with the aid of machine learning. Since information located within web pages contains some level of noise, the application of pre-processing methods and selecting a suitable representation are necessary. As far as the representation is concerned, a suitable weighting of text documents is important. The weighted and pre-processed text documents form a suitable input for classification or clustering methods of machine learning.

2 Used Methods

We used classification and clustering machine learning methods [3], [5]. In the frame of classification, some evaluation of employed classifiers is necessary. The quality of the used classifiers can be measured with the aid of various coefficients calculated from a contingency table [4]. In our experiments, we used the precision coefficient defined according to the following formula:

$$\pi_j = \frac{TP_j}{TP_j + FP_j},$$

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 191–198, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

where TP_j is the number of correctly predicted positive examples of the class c_j and FP_j is the number of incorrectly predicted positive examples of the class c_j.

In the frame of this work, we focused on classification of text documents from web pages. We performed tests using the kNN classifier (k Nearest Neighbours) [5], which is based on examples. This classifier stores all training examples (documents) in its memory.

In our experiments, we also focused on text document clustering [6]. We employed the k-means algorithm, which is defined in the following way. Let us assume n objects and k clusters. Each object represents a vector in a d-dimensional space. In this case, each cluster can be represented as a centre of gravity of those objects, which belong to the cluster. We used cosine similarity metrics. One of disadvantages of this method is the risk of falling into a local minimum. This falling depends on the initial random selection of initial examples – documents. Better results can be achieved using a modification of the algorithm by employing the incremental actualisation of centres of clusters.

3 Text Document Processing

The purpose of the presented work was to classify retrieved text information from web pages to a set of classes, which represent a domain of user interests. From the point of text document processing, we were interested in the dependency of the classification precision on the type of used weightings of text documents. Before we discuss the used type of weighting of text documents, we want to mention, that we used the vector representation model to represent documents.

The process of automatic extraction consists of several steps: lexical analysis – token formation, elimination of words without meaning, lemmatisation and weighting. The lexical analysis was performed in our tests by "Lower case filter". The elimination of words without meaning was made with the aid of "Stop words filter", lemmatisation (stemming) was carried out by "Stem filter" and finally weighting was accomplished by "index filter". All filters were from the library "Jbowl" [1]. This "Jbowl" library is an original piece of software system developed in Java to support information retrieval and text mining tasks. It is being developed as an open source with modular framework for pre-processing, indexing and further exploration of text collections. The system is described in more detail in [2].

3.1 Text Document Weighting

The words can be of various importance for document representation. That is why some relative values - weights must be defined. These weights can be used while reducing the number of used terms. In this way the weights represent a selective force of the terms. The selective force expresses how good the term represents the content of a document. Those terms have higher selective force, that are not so frequent throughout the collection of documents, but are more frequent within a particular document (or a group of documents). The term, which finds all documents from the

corpus, has the minimum selective force. The process of weight definition is called weighting. Various types of weighting can be found in [8].

In our work the following weighting schemes have been tested: ***Binary weighting.*** Weight function is $F: TxC \rightarrow \{0, 1\}$, where C is the document corpus and T is the set of terms, for which $F(d_i, t_j) = 1$ in the case when at least one occurrence of the term t_j can be found in the document d_i, Otherwise $F(d_i, t_j)=0$. ***TF weighting (TF - term frequency).*** Only the term importance with regard to particular documents is taken into account and term importance with regard to the whole corpus of documents is not considered. The weight function is defined: $F: TxC \rightarrow IN$(set of natural numbers). $F(d_i, t_j) = k$ represents the frequency of the term t_j in the document d_i. ***TF-IDF weighting*** is a combination of TF and IDF weightings. IDF – inverse document frequency is used for a global weighting $G(t_j) = idf_j = log(N/df_j)$, where N is the number of used documents in the corpus and df_i is the number of documents with the occurrence of the term t_i. ***Inquery weighting (information retrieval).*** This weighting is more complicated, but its advantage is the absence of any parameters, which have to be experimentally set. Weights are defined as:

$$w_{ij} = \frac{tf_{ij} \log \frac{N+0.5}{n}}{(tf_{ij} + 0.5 + 1.5 ndl_i) \log(N+1)},$$

where n is the number of documents in which the term t_i can be found, N is the number of documents in the corpus and ndl_j is the normalised length of a document defined as the relation of the document's length to the average length of all documents located in the corpus. ***Sparck, Jones and Robertson weighting*** [7]. The weight function is represented by the following definition, where parameter $b \in <0,1>$ represents the effect of the document frequency and parameter $K1$ controls the influence of the term frequency.

$$w_{ij} = \frac{tf_{ij} idf_j (K1+1)}{K1(1-b+ndl_i b) + tf_{ij}}$$

4 Experiments

In our experiments, two data sets (collections of documents) where used: ***20 News Groups*** is a simple data set, which is composed from Internet discussion documents. It contains 19953 documents assigned (classified) into only one of twenty categories. The dimension of the lexical profile is 111474. Its advantage is an implicit classification to only one category. A division of this data set into training and test sets was realised by a random selection using the proportion 1:1. ***Reuters-21578*** contains articles of the press agency Reuters. Each document from 21578 documents carries information obtained in the process of intellectual indexing – assignment to some of 406 categories. Classification to more categories is possible. In the presented work, the ApteMod version in the XML format was used. This version consists of a training part (7770 documents) and a test part (3019 documents). Many researchers reported the results with this data set split. Documents are represented by lexical profile of the

dimension 24242. The ApteMod version was created from the original Reuters collection by removing uncategorised documents and categories with very small number of documents. The ApteMod version was modified to the ApteModMdf collection by removing documents with more than one category and keeping only documents classified into one of 13 the most populated categories. The ApteModMdf collection contains 5953 documents in training and 2307 in test subsets. All experiments were performed in the programming language Java™ 1.4.2_04.

4.1 Influence of Weighting on Classification Precision

For subsequent processing of documents by classification or clustering methods, the type of used weighting is very important. Thus, we performed experiments in order to compare precision of classification achieved by the kNN method on both above mentioned document corpuses while experimenting with the type of used weighting. We used the number of seeds $k=45$. This number was selected experimentally by the method "leave one out cross validation" from the range of 1 to 50. Experiments with the 20 News Groups corpus were carried out in the following order. First, the number of terms (the dimensionality of lexical profile) was reduced using the information gain criterion. Next, the corpus was divided into training and test sets in proportion 1:1 by a random selection. Five experiments were realised for each type of weighting. **Table 1** contains achieved results for these weightings: Sparck, Jones & Robertson, Inquery, TFIDF, binary and TF. The TFIDF weighting was used in two versions: a classic TFIDF weighting denoted as TFIDF(ntc) and a modified schema TFIDF(ltc) where weight calculations are made according to the following formula:

$$w_{ij} = [\log(tf_{ij}) + 1] idf_{ij} = [\log(tf_{ij}) + 1] \log(\tfrac{N}{df_j}) \cdot$$

The weighting according to Sprack, Jones & Robertson (SJR) seems to be the best choice in the sense of the highest average precision of classification. This type of weighting together with Inquery weighting required adding information about the average length of documents. The SJR weighting seems to be the most robust weighting scheme from those we experimented with. The Inquery weighting shows results, which can be compared with the best SJR weighting, but is simpler because of the absence of tuning parameters. TFIDF(ltc) weighting seems to be better than TFIDF(ntc) weighting, because of using the modified TF. The used logarithm decreases differences between the weight representing a frequently occurring term and the weight of a term with only one occurrence. The logarithm function is only slightly increasing while the original TFIDF(ntc) weighting increases linearly.

Table 1. Precision of classification on 20 News Groups according to various types of weighting

Fold.	Spark&	Inquery	TFIDF (ltc)	Binary	TFIDF (ntc)	TF
1	0.834236	0.830225	0.822002	0.794826	0.790614	0.735058
2	0.827818	0.827016	0.818492	0.790112	0.791617	0.738969

3	0.837345	0.836141	0.828620	0.795929	0.794725	0.739571
4	0.835540	0.832130	0.824208	0.788608	0.791115	0.738468
5	0.841757	0.838448	0.830325	0.797333	0.792920	0.745187
Average precision	0.835339	0.832792	0.824729	0.793361	0.792198	0.739450
Standard Deviation	0.005075	0.004572	0.004824	0.003797	0.001653	0.003654
Max. Precision	0.841757	0.838448	0.830325	0.797333	0.794725	0.745187
Min. Precision	0.827818	0.827016	0.818492	0.788608	0.790614	0.735058
Ordering	**1**	**2**	**3**	**4**	**5**	**6**
%	100.0	99.7	98.7	95.0	94.8	88.5

An identical experiment was realised using the Reuters-ApteMod and Reuters-ApteModMdf document sets. Vectors in Reuters sets were shorter (9848) than vectors in 20 News Groups (24242). Investigated weightings showed similar tendency of differences in quality as when using the 20 News Groups collection. The results can be found in **Table 2**.

Table 2. Precision of classification using Reuters-ApteMod and Reuters-ApteModMdf according to various types of weighting

	Spark&	**Inquery**	**TFIDF (ltc)**	**Binary**	**TFIDF (ntc)**	**TF**
ApteMod	0.917642	0.912874	0.907672	0.919809	0.882531	0.876896
Apte-ModMdf	0.887723	0.890443	0.892385	0.902486	0.878399	0.869852

In the same way as before, the advantage of using the scheme TFIDF(ltc) to using TFIDF(ntc) and the preference of binary weighting to TF-based weightings were confirmed. The weighting SJR has proven to be suitable for fine distinguishing documents of similar categories. Therefore, we represented documents by weights calculated exclusively according to this weighting in all subsequent experiments.

4.2 Influence of unlabeled data with predicted categories on precision

In this experiment the 20 News Groups set was divided into training and test parts using a ratio 1:1. The experiment itself consists of ten separate experiments carried out in two modes. The first mode (column 3 in **Table 3**) was based on measuring the overall classification precision using the complete test set. The second mode (column 4 in **Table 3**) represents the case when the category for remaining **(100% - i*10%)** documents from the training set was predicted using the kNN classifier which was trained on only **i*10%** documents from the training set where **i** denotes the i-th experiment. Very often we have information about the class of training examples only for a part (**i*10%** documents) of our training set. To obtain information about the class for the unlabeled part (**100% - i*10%**) of the training set is often too expansive

or impossible. Thus, we need to estimate or predicate this information. We estimated the label of the unlabeled examples with the initial classifier build using only the labeled data. The final classifier was learned using this complete extended training set.

Table 3. Influence of category prediction on precision

Training [%]	Prediction [%]	Precision of kNN	Precision of kNN with prediction
10	90	0.0991280	0.3059036
20	80	0.1795129	0.5051619
30	70	0.2602987	0.6137115
40	60	0.3444923	0.6706425
50	50	0.4364037	0.6879824
60	40	0.5294177	0.7281748
70	30	0.6262404	0.7499248
80	20	0.7164478	0.8104641
90	10	0.7942267	0.8159767
100	0	0.8353212	0.8353212

The achieved results clearly indicate that using prediction increases the precision of classification. In the last tenth experiment, the training set was the same in both modes.

4.3 Clustering by k-means on 20 News Groups

In the case of information retrieval from various web pages, no categories are specified to which retrieved documents belong. These categories can be defined with the aid of clustering. We have realised a set of experiments with the *k*-means clustering method using documents from 20 News Groups. The number of clusters to be formed was twenty (20 categories exist in the used corpus of documents). Documents from the corpus were weighted using the SJR weight function. **Table 4** presents the achieved values of average precision related to individual categories (represented by the second column) and positive standard deviation of precision according to individual categories (in the third column). Since the random initialisation was made, each cluster was initialised by a randomly selected document from the set of twenty categories. The table represents achieved precision with great dispersion – standard deviation oscillates within the interval <10,20> %, therefore the hypothesis about strong dependence on the initialisation procedure seems to be strongly supported.

4.4 Clustering by k-means with controlled initialisation on 20 News Groups

This set of experiments was parametrically identical to the previous set of experiments. The only difference was that the initialisation was not random but controlled. Our system initialised the **i**-th cluster by an example which represents an average of ten randomly selected examples belonging to the **i**-th category. **Table 4** presents the achieved values of average precision related to individual categories (represented by the fourth column) and positive standard deviation of precision according to individ-

ual categories (represented by the fifth column). The results have proven the importance of controlled initialisation for decreasing standard deviation.

Table 4. Average precision and standard deviation of the clustering by the k-means algorithm without and with controlled initialisation

Clustering	**without controlled initialisation**		**with controlled initialisation**	
Category	**Precision**	**Standard deviation**	**Precision**	**Standard deviation**
1	44,96	10,96	64,09	7,86
2	38,87	13,94	66,91	3,37
3	46,80	11,11	61,40	2,14
4	37,49	04,69	50,40	1,89
5	41,89	11,78	75,44	3,47
6	53,68	16,09	83,81	2,92
7	69,61	17,94	80,69	1,88
8	56,48	21,39	86,62	1,00
9	71,44	20,13	94,41	0,85
10	77,04	20,96	94,20	0,59
11	74,43	15,72	87,62	0,93
12	59,01	16,96	80,23	1,81
13	46,65	18,27	76,72	3,07
14	87,78	10,11	94,94	1,41
15	72,51	14,82	84,35	1,60
16	49,43	06,54	59,76	3,84
17	47,87	10,42	58,98	3,50
18	53,69	16,55	79,22	2,51
19	32,87	13,49	64,35	10,41
20	27,66	03,83	37,89	5,79

5 Conclusions

The paper presents experiments with different weighting schemes carried out for the purpose of using them for knowledge retrieval from web-pages. A comparison of particular types of text document weighting was performed and supported experimentally. Experiments, which were focused on the *k*-means clustering method are described as well.

Sparck, Jones and Robertson weighting usually isn't used very often. So results of experiments with this method are valuable. The paper brings a new possibility of using unlabeled data for better classification accuracy. Using 20NewGroups on testing brings more precise quantitative evaluation of the new created clusters.

Clustering and classification methods are suitable for application in template based composition [9], for library applications, applications for design and realisation of Internet crawling and so on.

The work presented in the paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the 1/1060/04 project "Document classification and annotation for the Semantic web".

References

1. Bednár, P. (2005): *API Java knižnice HTML Parser.* http://sourceforge.net/projects/jbowl
2. Bednár, P., Butka, P., Paralic, J.: Java Library for Support of Text Mining and Retrieval. ZNALOSTI 2005, Stará Lesná, Vyd. Univerzity Palackého Olomouc, 2005, 162-169, ISBN 80-248-0755-6.
3. Berka, P.: Dobývání znalostí z databází. Academia – nakladatelství Akademie věd České republiky, Praha, 2003, 366 stran, ISBN 80-200-1062-9.
4. Machová, K., Puszta, M., Bednár, P. (2005): *Improving of the results of classification algorithms by Boosting method.* ZNALOSTI 2005, Stará Lesná, Vyd. Univerzity Palackého Olomouc, 2005, 81-84.
5. Mitchell, T.M.(1997): *Machine Learning*. McGraw-Hill Companies, Inc., Singapore, 1997, 414 ps., ISBN 0-07-042807-7.
6. Muresan, G., Harper, D.J. (2001): *Document Clustering and Language Models for System-Mediated Information Access*. Proc. of the 5th European Conference on Research and Advanced Technology for Digital Libraries ECDL'01, Darmstad, September 2001, ISBN 3-540-42537-3.
7. Robertson, S. E. and Sparck Jones, K. (1997). *Simple proven approaches to text retrieval*. Technical report, TR-356, Cambridge University Computer Laboratory, Cambridge, UK.
8. Salton G., & Buckley C. (1988). *Term weighting approaches in automatic text retrieval*. Information Processing and Management, 24(5): 513-523.
9. Svátek, V., Vacura, M.: Automatic Composition of Web Analysis Tools: Simulation on Classification Templates. RAWS 2005, Proc. of the 1st International Workshop on representation and Analysis of Web Space, VŠB-Technical University of Ostrava, TiskServis, Ostrava, 2005, 78-84, ISBN 80-248-0864-1.
10. Van Rijsbergen C.J. (1979): *Information Retrieval.* Department of Computing Science, University of Glasgow.

Využití suffixových stromů pro vylepšení vektorového dotazu

Jan Martinovič[1], Tomáš Novosád[2], Václav Snášel[1]

[1]Katedra informatiky, FEI, VŠB - Technická Univerzita Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba

[2]Katedra matematické informatiky,
Přírodovědecká Fakulta, Univerzita Palackého v Olomouci

{jan.martinovic,vaclav.snasel@vsb.cz}@vsb.cz, novosadt@inf.upol.cz

Abstrakt. V dnešním světě existují různé kolekce dokumentů. Proto vznikají různé systémy využívajících booleovských, vektorových a dalších modelů k reprezentaci dokumentů. Většina z těchto modelů pro svá omezení nedovoluje uživateli nalézt všechny očekávané dokumenty. Tato práce popisuje některé přístupy k vylepšení vektorového dotazu. Věnuje se rozšíření vektorového dotazu založeného na shlukování dokumentů pomocí sufixových stromů a následného hledání nejpodobnějších dokumentů k danému dokumentu. Následně jsou navrženy metody pro oddálení nerelevantních dokumentů od dotazu za pomocí dotazování na vývoj tématu. Dále v této práci provádíme srovnání různých metod shlukování jak pomocí sufixových stromů tak například aglomerativního hierarchického shlukování. Srovnání provádíme vždy oproti klasickému vektorovému dotazu.

Klíčová slova: suffixové stromy, shlukování dokumentů, vývoj tématu, vektorový model

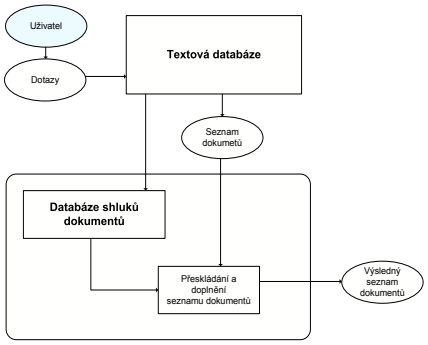
1 Úvod

Pro vyhledávání v kolekcích dokumentů existují různé systémy využívajících booleovských, vektorových, pravděpodobnostních a dalších modelů k reprezentaci dokumentů, dotazů, pravidel a procedur umožňujících určit shodu mezi požadavkem uživatele (*dotaz*) a dokumenty. Většina z těchto modelů obsahuje řadu omezení. Tato omezení neumožňují uživateli nalézt všechny dokumenty, které považuje za relevantní. Mezi očekávanými dokumenty (*relevantní dokumenty*) nalezneme i dokumenty špatné (*nerelevantní*) a některé relevantní dokumenty nejsou v seznamu nalezených dokumentů vůbec obsaženy.

V této práci navrhujeme metodu TOPIC-ST pro hledání nejpodobnějších dokumentů k určenému dokumentu, založenou na suffixových stromech a její použití na vylepšení odpovědi vektorového dotazu. Podobnost je v tomto případě dána právě sdílenými frázemi mezi jednotlivými dokumenty. V práci navazujeme na výsledky prezentované na konferenci DATESO 2004 [2], Znalosti 2005 [6]. Systém pro vylepšení odpovědi vektorového dotazu je zobrazen na obrázku 1.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 199–206, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

TOPIC-ST je odlišný v tom, že se na dokument nedíváme jako na pouhou množinu slov, ale dokument bereme jako množinu frází (v našem případě věty). Tento přístup byl již úspěšně dříve aplikován [12]. Fráze se již dlouho používaly, aby nahradily IR systémy založené na indexování slov [13]. TOPIC-ST algoritmus je závislý na struktuře nazývané Suffixový strom (suffix tree) [14, 4]. Tato struktura se používá k tomu, aby nám našla množiny dokumentů, které sdílejí společné fráze (věty) respektive podfráze, jak ukážeme dále. Oproti klasickým shlukovacím algoritmům [9], má tento postup výhodu v tom, že věta - fráze má určitý hlubší význam v tom, že slova jdoucí po sobě nám tvoří určitou uspořádanou posloupnost, oproti tomu, když vezmeme dokument jenom jako množinu slov a neuvažujeme jejich uspořádání.



Obr. 1. Vyhledávací systém pro vylepšení odpovědi vektorového dotazu. K vylepšení dotazu je použita metoda SORT-EACH a databáze shluků dokumentů.

2 Vektorový model

V polovine 20. století Lunh navrhl systém pro vyhledávání informací v textových kolekcích. Systém byl založen na porovnávání informací mezi uloženými dokumenty a informacemi v *dotazu* od uživatele. Ve vektorovém modelu jsou dokument a dotaz reprezentovány *vektorem* [11]. Jednou z výhod tohoto modelu je možnost usporád podle klesající podobnosti k dotazu dokumenty ruzných formátu (jakými jsou napríklad textové dokumenty, obrázky, video) [1, 11].

Pokud bylo pro indexaci n dokumentů použito celkem m různých termů $t_1 \dots t_m$, potom je každý dokument d_i reprezentován vektorem:

$$d_i = (w_{i1}, w_{i2}, \dots, w_{im}) ,$$

kde w_{ij} je váha termu t_j v dokumentu d_i. Váha s největší hodnotou odpovídá termu s největší důležitostí. Indexový soubor vektorového modelu reprezentuje matice, kde i–tý řádek odpovídá i–tému dokumentu, a j–tý sloupec j-tému termu. Dotaz ve vektorovém modelu můžeme reprezentovat pomocí m–místného

vektoru vah. Na základě dotazu q můžeme pro každý dokument d_i spočítat *koeficient podobnosti*. Tento koeficient si můžeme představit jako "vzdálenost" vektoru dokumentu a vektoru dotazu. Pro výpočet podobnosti jsme použili *kosinovou míru*:

$$sim(q, d_i) = \frac{\sum_{k=1}^{m} (q_k w_{ik})}{\sqrt{\sum_{k=1}^{m} (q_k)^2 \sum_{k=1}^{m} (w_{ik})^2}}$$

3 Suffixový strom

Datová struktura suffixový strom se používá na porovnávání řetězců, popřípadě na vyhledávání v řetězcích. Například na vyhledání nejdelšího opakujícího se podřetězce [14], porovnávání řetězců [3], nebo komprese textu [10]. V následujíc kapitole popíšeme datovou struktur suffixového stromu - jeho definice, jak ho postavit a další hlavní vlastnosti.

Následující popis této struktury je převzat z knihy od Dana Gusfielda, která odpovídá na otázky týkajících se řetězců, stromů a posloupností [4]. Co je důležité si uvědomit, že se na dokument díváme jako na posloupnost slov, ne znaků.

Definice 1. *Suffixový strom T pro m-slovný řetězec S je orientovaný strom s přesně m listy očíslovanými od 1 do m. Každý interní uzel různý od kořene má minimálně dva potomky a každá hrana je označena neprázdným podřetězcem slov z S. Žádné dvě hrany vycházející z jednoho uzlu nezačínají stejným slovem. Klíčová vlastnost tohoto stromu je v tom, že pro každý list i, když budeme procházet strom od kořene k tomuto listu, dostaneme zřetězením hran (jejich označení) právě suffix daného řetězce S, začínajícího na i-té pozici. Takže dostaneme $S[i \ldots m]$.*

V podobném smyslu, suffixový strom množiny řetězců, nazývaný zobecněný suffixový strom [5], což je kompaktní trie všech suffixů všech řetěců v dané množině.

Definice 2. *Zobecněný suffixový strom T pro množinu S obsahující n řetězců S_n, každý o délce m_n, orientovaný strom s kořenem, který obsahuje přesně $\sum m_n$ listů označených pomocí dvojice čísel (k, l), kde k je z intervalu 1 až n a l je z intervalu 1 to m_k. Každý interní uzel různý od kořene, má nejméně dva potomky a každá hrana vycházející z tohoto uzlu je popsána neprázdným podřetězcem slov z řetězce v množině S. Žádné dvě hrany vycházející z tohoto uzlu nemůžou začínat stejným symbolem (v našem případě slovem). Pro libovolný list (i, j), zřetězením popisů hran na cestě od kořene k listu (i, j), dostaneme suffix řetězce S_i, který začíná na pozici j, takže dostaneme suffix $S_i[j \ldots m_i]$.*

4 Vylepšení odpovědi ve vektorovém modelu

4.1 Algoritmus získání vývoje tématu

Pro získání vývoje tématu z hierarchie shluků jsme definovali algoritmus TOPIC [6, 7], který používá počet dokumentů ve vývoji, jako omezující kritérium.

V tomto článku popisujeme jehou novou modifikaci, která je založena na suffixových stromech.

Suffix tree topic algoritmus je založen na nalezení frází, které jsou společné pro určitou skupinu dokumentů. V našem kontextu bereme frázi jako sekvenci po sobě jdoucích slov délky jedna a víc. Dále zde definujeme pojem base cluster (základní frázový shluk) [15], což je množina dokumentů, které sdílejí společnou frázi. Algoritmus TOPIC-ST vychází z dříve publikovaných prací [15, 17].

Suffix tree topic algoritmus má tři kroky. (1) *"Čištění"* dokumentu, (2) nalezení base clusterů pomocí zobecněného suffixového stromu, (3) ke každému dokumentu určit n nejpodobnějších dokumentů.

Co je velmi potěšující je, že suffix tree topic je inkrementální algoritmus [17]. Nevyžaduje jakékoliv předzpracování kolekce dokumentů. Tak jak dokumenty přicházejí, tak jsou postupně zpracovávány jeden za druhým. Nepotřebujeme tedy například dopředu vědět kolik je dokumentů, kolik je slov, jaká jsou slova apod.

"Čištění dokumentu": V tomto kroku převedeme dokument z textové podoby do formy, se kterou budeme dále pracovat. Z textu nejdříve vybereme větné celky. Toto můžeme dělat různými způsoby, například rozdělíme dokument nějakým terminálním znakem (tečka, čárka, otazník, vykřičník, atd). Dále provedeme jednoduchou lematizaci slov pomocí Porterova lematizačního algoritmu (Porter's stemming algorithm) [8].

Dále z textu odstraníme tzv. stop slova, což jsou slova, která se v textu objevují příliš často a proto pro nás nemají význam. Stop slova odstraňujeme dvěma způsoby. První je takový, že stop slova odstraňujeme pouze ze začátku a konce věty, kvůli zachování sémantiky věty, druhý způsob odstraní stop slova z celé věty. Z provedených experimentů (sekce Testování) se ukázalo, že první metoda je přesnější z hlediska nalezení relevantních dokumentů, leč náročnější na konstrukční čas a místo potřebné v paměti. Druhou metodou se potřebný čas i místo v paměti redukoval na 50% první metody a přesnost ve výsledku (relevantní nalezené dokumenty) se snížila minimálně, jak vidět na obrázku 3(a).

Nalezení base clusterů (základních shluků): Na nalezení base clusterů se můžeme dívat jako na vytvoření invertovaného indexu frází pro naši kolekci dokumentů. K tomuto účelu nám právě velmi dobře vyhovuje suffixový strom. Suffixový strom postavený z kolekce dokumentů je kompaktní trie obsahující všechny suffixy všech řetězců respektive vět, získaných výše zmíněným způsobem z celé kolekce dokumentů. Nyní každý interní uzel v tomto stromě reprezentuje skupinu dokumentů, které sdílejí frázi jednoznačně určenou právě tímto uzlem [15, 17]. Takovýto interní uzel nazýváme base cluster (základní shluk). Dále, všechny base clustery (obsahující dva a více dokumentů) se ve stromě objevují jako interní uzly.

Pro to, abychom ušetřili strojový čas, použijeme penalizační funkci, která penalizuje clustery, jejichž efektivní délka fráze je nulová [15]. Takové base clustery již dále neuvažujeme. Efektivní délka fráze je nulová právě tehdy když obsahuje

pouze stop slova (děláme pouze v případě prvního způsobu odstraňování stop slov, jelikož v druhém způsobu odstraňování stop slov je zaručeno, že jsou již všechny odstraněny po fázi čištění dokumentu).

Dále každému base clusteru B přiřadíme skóre $s(B)$. Toto skóre použijeme v třetím kroku TOPIC-ST. V odstavci Testování jsme popsali čtyři různé způsoby jak počítat skóre base clusteru. Postupnými experimenty se nejlépe osvědčilo skóre, které odpovídá následující formuli:

$$s(B) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} tfidf(w_j, d_i)}{n},$$

kde n je počet dokumentů v base clusteru B, m je délka sdílené fráze v base clusteru B, w_j je j-tý term sdílené fráze a d_i je i-tý dokument v B.

Term frequency - inverse dokument frequency (tfidf) je obecně používaná technika v IR, která individuálním termům přiřadí váhy vůči danému dokumentu. Počítáme $tfidf(w_j, d_i)$ hodnotu termu w_j v dokumentu d_i použitím následující formule:

$$tfidf(w_j, d_i) = (1 + \log(tf(w_j, d_i))) * log(1 + \frac{N}{df(w_j)})$$

kde $tf(w_j, d_i)$ značí počet výskytů termu w_j v dokumentu d_i, N je celkový počet dokumentů v naší kolekci a $df(w_j)$ je počet dokumentů, ve kterých se term w_j objevuje.

Hledání nejpodobnějšího dokumentu k danému dokumentu: V tomto třetím a posledním kroku TOPIC-ST hledáme ke všem dokumentům v naší kolekci nejpodobnější dokumenty, seřazené od nejvíce podobného po nejméně podobný. Pro to jsme vytvořili následující pravidlo:

Definice 3. *Nejpodobnější dokument k danému dokumentu je ten, který je s ním v nejvíce base clusterech (základních frázových clusterech). Skóre podobnosti je dáno jako součet skóre jednotlivých base clusterů, ve kterých jsou dané dva dokumenty společně přítomny.*

4.2 Algoritmus SORT-EACH

Tento algoritmus přesune dokumenty v kolekci získané vektorovým dotazem tak, aby dokumenty ze stejného vývoje tématu byly za sebou:

1. Provedeme vektorový dotaz a získaný setřízený seznam dokumentů označíme C_V. Seznam je setřízen podle podobnosti dokumentů k dotazu.
2. Označíme výsledný setřídený seznam dokumentu C_S a určíme počet dokumentů, který má obsahovat - *count*.
3. Určíme $level_{C_T}$ – počet dokumentů ve vývoj tématu.
4. Provedeme následující třídění (třídění bude ukončeno buď po vyčerpání dokumentů v C_V, nebo pokud $|C_T| = count$):

// seznam dokumentů procházíme od nejpodobnějších dokumentu k dotazu
for each D_V **in** C_V **do**
  if $C_S = \emptyset$ **then**
    D_V *pridej na konec* C_S
    continue
  endif
  Nalezneme seznam $C_T \backslash \{D_V\}$ pro D_V pomocí algoritmu TOPIC-CA,
    tak aby $|C_T| = level_{C_T}$.
  for each D_T **v** C_T **do**
    if $D_T \in C_S$ **then**
      pridej D_V *do* C_S *za* D_T
      break
    endif
  endfor
  if $D_V \notin D_S$ **then** D_V *pridej na konec* C_S
endfor

5. Výsledný seznam je C_S.

5 Testování

Udělali jsme sérii několika různých variant TOPIC-ST. Jednak jsme zkoušeli různým způsobem odstraňovat stop slova, což je uvedeno v odstavci "*čistění*" dokumentu. Dále jsme experimentovali s jinými formulemi pro výpočet skóre základních clusterů. Nejlépe se osvědčila výše zmíněná formule jak je vidět na obrázku 3(a). Nyní popíšeme jednotlivé experimenty. Variace se vždy týkají výpočtu skóre base clusteru a "*čistění*" dokumentu. Jsou zde čtyři různé způsoby jak počítat skóre:

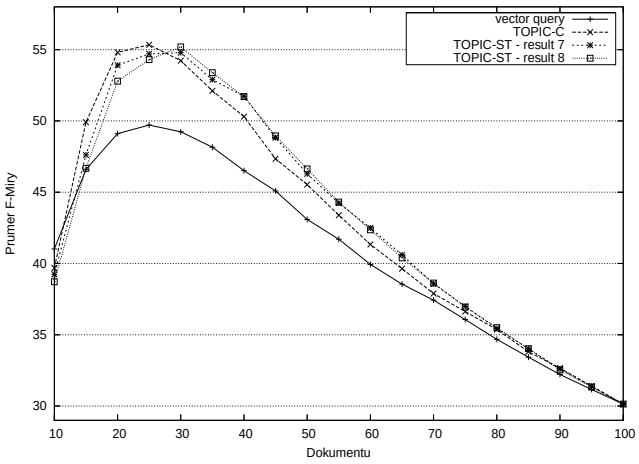
1. skóre = efektivní délka fráze × počet dokumentů v base clusteru
2. skóre = počet base clusterů ve kterých se dokumenty vyskytují společně
3. skóre = naše výše zmíněná formule s tfidf vážením
4. skóre = efektivní délka fráze × počet dokumentů v base clusteru × suma $tfidf$ hodnot jednotlivých termů v base clusteru

a další čtyři způsoby jak odstraňovat stop slova:

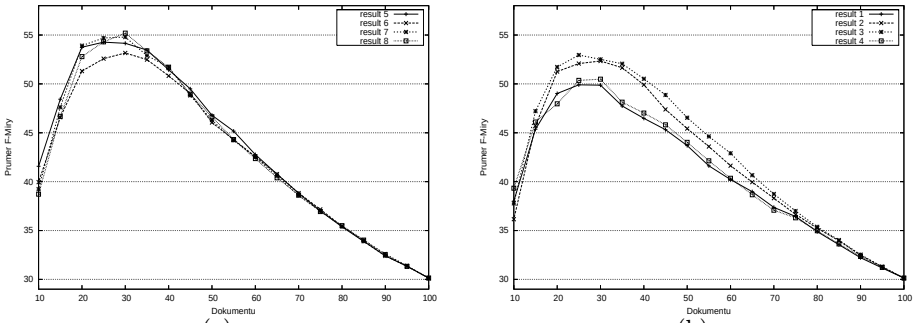
1. stop slova odstraněna ze začátku a konce věty
2. stop slova odstraněna z celé věty
3. stop slova odstraněna ze začátku a konce věty + odstraněna zipf's slova
4. stop slova odstraněna z celé věty + odstraněna zipf's slova

Následně jsme provedli následující kombinace testů:

result 1 – skóre (1), čištění (1)
result 2 – skóre (2), čištění (1)
result 3 – skóre (2), čištění (2)
result 4 – skóre (1), čištění (2)
result 5 – skóre (3), čištění (2)
result 6 – skóre (3), čištění (4)
result 7 – skóre (3), čištění (1)
result 8 – skóre (3), čištění (3)



Obr. 2. Srovnání vylepšení vektorového dotazu pomocí aglomerativního shlukování, dvou metod TOPIC-ST result 7, result 8 a klasického vektorového dotazu



(a) (b)

Obr. 3. Srovnání jednotlivých metod algoritmu TOPIC-ST

6 Závěr

V článku byla popsána metoda vylepšení odpovědi vektorového dotazu pomocí sufixových stromů. Nejlepší výsledky jsou dosahovány, když je pro výpočet skóre použito tfidf vážení. Dále je patrné, že není vhodné odstraňovat stop slova z celého dokumentu. Nejlepších výsledku je dosahováno, pokud jsou stop slova odstraněna pouze z počátku a konce věty. Dalším porovnáním, které jsme provedli, bylo srovnání metody vylepšení výsledku vektorového dotazu pomoci sufixového stromu a pomoci shlukování. Výsledky obou metod se jeví podobné a bude potřeba provést srovnání těchto metod při testech prováděných na dalších textových kolekcích. V další práci se chceme věnovat testování výše uvedených metod na větších kolekcích dat a jejich vzájemnému srovnání. Dále je potřeba otestovat vliv délky seznamu získávaného metodou TOPIC-ST.

Reference

1. Berry M. W." *Survey of Text Mining.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
2. Dvorsky J., Martinovic J., and Snasel V.: Query expansion and evolution of topic in information retrieval systems. In *DATESO*, pages 117–127, 2004.
3. Ehrenfeucht A. and Haussler D.: A new distance metric on strings computable in linear time. Discrete Applied Math, 40, 1988.
4. Gusfield D.: Algorithms on strings, trees and sequences: computer science and computational biology, chapter 6. Cambridge University Press, 1997.
5. Gusfield D.: Generalized suffix tree for a set of syringe, Cambridge University Press.
6. Martinovič J., Snášel V.: Using The Topic Evolution for Vector Query Improvement, ZNALOSTI 2005, Stará Lesná, ISBN: 80-248-0755-6.
7. Martinovič J., Gajdoš P.: Vector model improvement by FCA and Topic Evolution, DATESO 2005, Desná, ISBN: 80-01-03204-3.
8. Porter M. F.: An algorithm for suffix stripping, Program, 14 (3):130-137, 1980. A description of the most widely used English stemmer.
9. Rasmussen E.: Clustering Algorithms. In W. B. Frakes and R. Baeza-Yates (eds.), Information Retrieval, pages 419-42. Prentice Hall, Eaglewood Cliffs, N. J., 1992.
10. Rodeh M., Pratt V. R. and Even S.: Linear algorithm for data compression via string matching. Journal of the ACM, 28(1):16-24, 1981.
11. Salton G. and Buckley C.: Term weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513–523, 1988.
12. Salton G., Yang C. S. and Yu C. T.: A theory of term importance in automatic text analysis. JASIS, 26(1):33-44,1975.
13. Zhai C., Tong X., Milic-Frayling N. and Evans D. A.: Evaluation of syntactic phrase indexing - CLARIT NLP track report, TREC-5, 1997.
14. Weiner P.: Linear pattern matching algorithms. In Proceedings of the 14th Annual Symposium on Foundations of Computer Science (FOCS), pages 1-11, 1973.
15. Zamir O., Etzioni O.: Web Document Clustering: A Feasibility Demonstration, SIGIR 98, 46-54, 1998.
16. Zamir O., Etzioni O., Madani O., Karp R. M.: Fast and Intuitive Clustering of Web Documents, 1997, (2001).
17. Zamir O.: Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results, Doctoral dissertation, University of Washington, 1999.

Annotation

Improving Vector model using Suffix trees

There are many ways how to search for documents in document collections. These methods take advantage of boolean, vector, probabilistic and other models for representation of documents, queries, rules and procedures which can determine correspondence between user requests and documents. Most of these models have several restrictions. These restrictions not allow user to find all expected documents. This work describing method for vector model improvement by using suffix trees and searching for the most similar documents to the given one. Next there are suggested methods how to push away irrelevant documents with help of quering on topic evolution. We also comparing different this new method with for example agglomerative hierarchical clustering. Every method is compared with classical vector query.

Práce s tacitními znalostmi v České republice

Ludmila Mládková

Katedra managementu, FPH, VŠE v Praze, W. Churchilla 4,
Praha 3, 130 00 Praha

mladkova@vse.cz

Abstrakt. Tacitní znalost je neviditelná, tichá. Organizace a jejich manažeři mají tendenci ji přehlížet a podceňovat. Bohužel, následky bývají nepříjemné. S tacitními znalostmi můžeme pracovat dvojím způsobem. Můžeme je převést na znalosti explicitní nebo je nechat v tacitní formě a tzv. je sdílet. Tacitní znalost je vždy vázána na osobnost či charakter svého nostitele, jedince nebo skupiny. Při externalizaci dochází k jejímu poškození. Proto je v mnoha případech vhodnější ponechat a sdílet ji v tacitní formě. Sdílení tacitní znalosti napomáhají tři nástroje – příběhy, komunity a učňovství. Článek je věnován stručné charakteristice těchto nástrojů a výsledkům výzkumu, který sleduje práci organizací v České republice s tacitními znalostmi.

Klíčová slova: znalost, tacitní znalost, příběh, komunita, učňovství

1 Význam tacitních znalostí pro organizaci

Význam tacitní znalosti pro organizaci je ohromný. Jak praxe ukazuje, největší znalostní bohatství mají organizace v hlavách pracovníků, v tacitní formě, vázané na individuální rysy jedince či skupin. Získat a využít tento potenciál není nic jednoduchého.

Tacitní znalost je neviditelná, tichá. Organizace a jejich manažeři mají tendenci ji přehlížet a podceňovat. Bohužel, následky bývají nepříjemné. Například americké zbrojařské organizace propustily počátkem 90. let 20. století velké množství expertů. Vlastníci a manažeři měli dojem, že konec studené války přinesl i konec zbrojení. Experti ponechali v databázích organizací velké množství explicitních znalostí – plány raket, složení chemikálií, procesy řešení speciálních situací. Své tacitní znalosti si však odnesli sebou. Očekávání ohledně vývoje ve vojenství se však nenaplnilo a zbrojařský průmysl v polovině 90. let zjistil, že explicitní znalosti jsou bez tacitních znalostí propuštěných odborníků téměř k ničemu. Organizace musely přijmout pracovníky zpět, mnohdy za pro ně nevýhodných podmínek.

Tacitní znalost je individuální záležitostí. Například řízení auta probíhá z velké části v tacitní dimenzi. Posadíme-li dva zkušené řidiče jednoho po druhém do stejného auta a necháme je projet stejnou trasu, při bedlivém pozorování zjistíme, že každý z nich řídil jinak. Oba mají podobnou tacitní znalost, ale tato znalost není shodná. Důvodů, proč je tomu tak, je více:

- lidé mají své osobní mentální filtry pomocí kterých selektivně vybírají z různých událostí aspekty, kterým dávají přednost. Například, budou-li se dva lidé dívat na sklenici s vodou, jeden může tvrdit, že je napůl plná a druhý, že je napůl prázdná. Oba mají sice pravdu, ale výsledné sdělení se

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 207–218, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

výrazně odlišuje. Na základě selektivně vybraných aspektů pak lidé hodnotí události,

- lidé mají různou absorpční kapacitu. Absorpční kapacita je schopnost absorbovat nové informace a vytvářet z nich znalosti. Závisí na předchozích znalostech jedince,
- velmi důležitý je i kontext. Dva jedinci mohou být vystaveni stejnému podnětu, ale pouze jeden z nich vytvoří novou znalost. Obvykle proto, že podnět pro něho přišel ve vhodnější okamžik, například nebyl rozptýlen výrazným zájmem o něco jiného. Nebo prostě tento jedinec lépe dokázal zhodnotit realitu a podnět pro něho byl více relevantní než pro jeho kolegu,
- důvodem může být i znalost sama, protože typ znalosti může ovlivnit proces sdílení. Někteří lidé lépe přebírají znalost ve formě příběhu, jiní mají rádi diskusi v komunitě, popřípadě dávají přednost koučingu. To je zcela individuální.

Výše uvedené body platí jak pro jedince, tak i pro skupiny, neboť i skupina může být nositelem tacitní znalosti [5].

2 Nástroje sdílení tacitních znalostí

V praxi jsou po staletí používány tři nástroje sdílení tacitních znalostí. Jde o *příběhy, komunity a učňovství*.

Příběhy pomáhají přenášet tacitní znalost mezi dvěma lidmi i ve skupině lidí. Dokáží překonat bariéry v myslích jedinců a vytvořit sdílenou základnu pro přenos tacitní znalosti.

Komunity jsou malé nebo větší skupiny či sítě lidí, kteří se scházejí za účelem sdílení znalostí a učení se jeden od druhého. Komunity mohou fungovat na základě osobního styku nebo jsou virtuální. Členy sdružuje do komunity nějaký cíl. Tímto cílem může být práce na řešení problému, zájem, touha naučit se něco nového, atd.

V praxi se lze setkat s různými typy komunit. Typ obvykle závisí na cílech a účelu, které komunita sleduje. Komunity lze členit různým způsobem také podle toho, k čemu má členění sloužit.

Učňovství je tradiční nástroj sdílení tacitních znalostí. Pedagogický proces a proces učení probíhají přímo na pracovišti nebo v místech, která reálná pracoviště připomínají. Organizace většinou tento základní princip chápou a snaží se vytvářet prostředí podnětné pro učení (například dílny nebo laboratoře).

Tradiční styl učňovství je založen na dělbě práce a jasně definovaných dovednostech nutných pro vykonávání povolání. Učňovství může také mít formu koučingu či mentoringu. V rámci učňovství nepřebírá učeň pouze znalosti profesní, ale také sociální dovednosti a archetypy [5].

3 Příběhy jako nástroj práce s tacitními znalostmi

Příběhy lze identifikovat ve většině textů a komunikace [7]. Příběhy jsou komplexní systémy symbolů a schémat používaných k tvorbě identity, změně společenských

praktik a předávání a tvorbě hodnot jak jedince, tak i skupiny. Dávají věcem a událostem smysl, představují normy, zkušenosti a vysvětlení reality, obsahují základní hodnoty, čímž ovlivňují, co pracovníci organizace považují za akceptovatelné a co ne. Jsou součástí sociálních procesů probíhajících v organizaci, spoluúčastní se tvorby mocenských vazeb a vztahů. Slouží jako nástroj ukládání a sdílení zkušeností a znalostí. Určují hodnoty organizace a výrazně ovlivňují její kulturu.

Příběh je nástrojem sdílení tacitní znalosti. Dokáže zachytit smysl a v hlubokém podvědomí existující významy a modely i tacitní vazby a vztahy mezi událostmi a zapůsobit na city. Vhodně zvolený příběh pomáhá překonat mentální bariéry, které si většina lidí vůči novým znalostem buduje. Zlepšuje tak komunikaci mezi lidmi, zvyšuje jejich kreativitu, pomáhá hledat společné hodnoty a řešit konflikty. Příběhy dokáží uvolnit představivost pracovníků organizace a zvýšit její inovační potenciál. Příběh může být katalyzátorem změn. Příběhy vznikají pomocí organického myšlení, ne mechanického a proto jejich význam v současné době vzrůstá.

Příběh lze také chápat jako virtuální zkušenost.

4 Komunity jako nástroj práce s tacitními znalostmi

Druhým účinným nástrojem, který mají manažeři k dispozici pro práci s tacitními znalostmi jsou komunity. V organizaci existují, ať již manažeři o nich vědí či ne, ať je podporují nebo ne.

Také komunity existovaly v lidské společnosti vždy. Komunitu lze definovat jako skupinu lidí, kteří mají společný zájem o nějakou oblast lidské činnosti (doménu) a jsou ochotni a schopni sdílet znalosti a učit se jeden od druhého. Tento společný zájem, nebo chcete-li zaujetí, vytváří velmi silné vazby mezi členy komunity a napomáhá sdílení tacitních znalostí v rámci aktivit spojených s doménou. Lidi vede do komunit zájem, touha sdílet problémy, zkušenosti, názory a společně řešit zajímavé úkoly.

Příkladem komunity je fungující rodina, skupina přátel, kteří se pravidelně schází za účelem sportovních nebo společenských aktivit. Komunitami s velmi silnými vztahy bývají komunity náboženské. Životaschopnou komunitu mohou vytvářet již dva lidé, ale často má komunita členů více.V organizacích vznikají komunity obvykle tam, kde se řeší nějaký problém nebo lidé musí intenzivně dlouhodobě spolupracovat, popřípadě učit se jeden od druhého. Komunita není tým, ale tým může být komunitou. Komunity zvyšují spolupráci pracovníků organizace a to i ve strukturách shora dolů, zlepšují inovativnost organizace, zlepšují rychlost a kvalitu rozhodování, napomáhají investicím do lidského kapitálu.

5 Učňovství jako nástroj práce s tacitními znalostmi

I *učňovství* je tradiční nástroj sdílení tacitních znalostí. Je to metoda „řízené reflexe“. Mistr dohlíží na proces učení a pomocí rad a komentářů usměrňuje učně žádoucím směrem.

Učňovství lze v organizacích používat k různým účelům. Kromě klasických učňovských programů se můžeme setkat s různými metodami, založenými na shodných principech, nejčastěji s koučováním a mentorováním. Na rozdíl od učňovství jde obvykle o postupy nespecifikované smlouvou. Vztah mezi koučem či mentorem a žákem nebývá ukončen předáním určité sady znalostí, ale je zaměřen na dlouhodobý rozvoj osobnosti učně či na pomoc zkušeného kouče při řešení problémů. Koučování může být použito i přímo k řízení podřízených.

Pedagogický proces a proces učení v rámci učňovství probíhají přímo v terénu, ať jim je les, pole nebo pracoviště či místa, která reálná pracoviště připomínají, například trenažér či laboratoř. Při učňovství dochází k předávání znalostí jak v teoretické, tak v praktické podobě. V rámci učňovství nepřebírá učeň pouze znalosti profesní, ale také sociální dovednosti a archetypy. V době před průmyslovou revolucí trvalo velmi dlouho a učeň si osvojoval veškeré znalosti nezbytné pro vykonávání řemesla. Později učňovství vycházelo z dělby práce a jasně definovaných dovedností nutných pro vykonávání povolání.

Má-li učňovství fungovat, je třeba dodržet některé zásady:

- učeň je ochoten a schopen převzít od mistra tacitní i explicitní znalosti,
- mistr, kouč, mentor, učitel má znalosti a je ochoten a schopen je učni předat,
- prvky řemesla nebo práce jsou převzatelné a je možné se je naučit,
- lze vhodně zvolit pracoviště, ve kterém budou probíhat interakce mezi učněm a mistrem.
- a v případě klasického učňovství
- učeň a mistr jsou spojeni a motivováni závaznou společenskou smlouvou.

Předávání znalostí mezi mistrem a učněm je aktivní. Učeň si osvojuje nové znalosti pomocí přirozeného procesu sledování, přijímání a vtiskávání.

6 Výzkum zaměřený na způsob práce organizací v České republice s tacitními znalostmi

V roce 2002 začal na katedře managementu Grantový projekt vedený prof. Ing. Janem Trunečkem nazvaný Projekt synergie, který zkoumal zda organizace v České republice dokáží vhodným způsobem zkombinovat nástroje řízení, které používají a dosáhnout tak synergie. Na tento grant v roce 2004 navázal grant GAČR 402/04/0867 Tvorba komplexního přístupu k práci s tichými a explicitními znalostmi organizací v České republice, rok 2004 – 2005, opět pod vedením profesora Trunečka. Výzkumným cílem tohoto grantu bylo zjistit, zda lidé v českých organizacích chápou pojem tacitní znalost, zda dokáží s tacitními znalostmi pracovat a zmapovat nástroje sdílení tacitních znalostí používané v organizacích v České republice.

Výzkum probíhal v organizacích patřících do různých odvětví a šlo o organizace jak ryze české, tak i vlastněné zahraničními subjekty, jejich velikost byla rozdílná. Výzkum byl prováděn pomocí interview, založeného na dotazníku. Otázky byly nepřímé, aby v respondentovi neevokovaly „správnou“ odpověď.

Rozhovory ve vybraných organizacích prováděli studenti, dopředu vyškolení v problematice managementu znalostí a práce s tacitními znalostmi. Samotný dotazník jim byl velmi podrobně vysvětlen. Po ukončení rozhovoru studenti dotazník

analyzovali a navrhli opatření, která by organizace měla přijmout v případě, že se rozhodne svou práci s tacitními znalostmi zlepšit. Vyplněný dotazník a doporučení byly součástí závěrečné zkoušky z předmětu. Při této zkoušce zkoušející přezkoumal kvalitu dotazníku a analýzy s doporučeními a v případě potřeby požádal studenta či studenty, aby se vrátili do organizace a provedli korekci. Dotazníky, které nesplňovaly požadavky, byly odmítnuty a do výzkumu nebyly zařazeny.

Dotazník se skládá z několika částí:

Tabulka 1. Dotazník

Část	Cíl části
Úvod	Základní údaje o organizaci.
Řízení lidských zdrojů	Základní otázky týkající se struktury zaměstnanců (věk, pohlaví, manažerské pozice), typu organizační struktury, stylu práce, znalostní strategie a důvěry.
Informační systémy	Analýza informačních technologií a jejich vztahu k informačním systémům nám poskytuje údaje o používaných nástrojích práce s explicitními znalostmi.
Schůzky, týmy a jiné metody	Jde o popis prostředí organizace a jeho vhodnosti pro sdílení tacitních znalostí.
Učňovské programy	Základní otázky týkající se klasických učňovských programů.
Školení nových zaměstnanců	Ukazuje, jak se noví zaměstnanci po znalostní stránce „hodí“ do organizace a jaké jsou nároky na jejich doškolení.
Koučing	Otázky týkající se koučingu.
Komunity	Otázky týkající se komunit.
Příběhy	Otázky týkající se příběhů.

V tuto chvíli je ještě brzy zveřejňovat seriózní výsledek výzkumu, protože praktická část výzkumu začala v únoru 2005, proto má výzkumný tým k dispozici pouze prvních 25 dotazníků. Bohužel, to, co z nich vyplývá, není příliš optimistické. Organizace v České republice dávají přílišný důraz na explicitní znalosti a podceňují význam znalostí tacitních. Pouze jedna ze sledovaných organizací řídí znalosti komplexně. V 13 organizacích se nám povedlo identifikovat komunity, bohužel pouze 4 organizace je vědomě podporují. Jedna organizace má funkční virtuální komunitu.

Tabulka 2. Komunity

Komunity	ano	ne
V organizaci pracují komunity	13	11
Organizace podporuje vznik formálních komunit	4	19
Organizace podporuje vznik neformálních komunit	4	21
Organizace podporuje vznik virtuálních komunit	1	22

Výzkumný tým byl velmi zklamán výsledky části věnované koučingu. Domnívali jsme se, že pro organizace na rozdíl od ostatních nástrojů práce s tacitními znalostmi je koučing obvyklým nástrojem. Tato domněnka se nám nepotvrdila, protože jen 15 organizací používá koučing ve své práci. Velmi nás naopak překvapilo, že pouze 5 organizací koučuje své vrcholové manažery.

Tabulka 3. Koučing

Praktikuje vaše organizace koučing?	ne	10
	koučujeme důležité zaměstnance na nemanažerských postech.	7
	koučujeme liniové manažery	9
	koučujeme střední manažery	9
	koučujeme vrcholové manažery	5

Manažeři 5 organizací sledovaného vzorku používají vědomě příběhy k řízení. Pracovníci 19 organizací, tedy většiny, si vypráví negativní příběhy, v 7 organizacích se vypráví především příběhy pozitivní.

Tabulka 4. Příběhy

Pracovníci si vypráví příběhy většinou o	překonaných problémech, katastrofách a nebezpečích	19
	úspěších a vítězstvích	7
Jsou příběhy ve vaší organizaci úmyslně používány k řízení?	ano	5

Z dosavadních výsledků výzkumu vyplývá, že organizace se ve svém přístupu k tacitním znalostem a nástrojům jejich řízení výrazně odlišují. Z celkového počtu 25 organizací pouze jedna řídila tacitní znalosti uvědoměle a uvědoměle používala nástroje, které umožňují jejich sdílení. Bohužel však představitelé této organizace nesouhlasili se zveřejněním jejího jména. U dalších dvou organizací byla práce s tacitními znalostmi velmi intenzivní, ačkoliv pracovníci a manažeři původně pojem tacitní znalost neznali. V obou případech byl v čele organizace charismatický manažer s lídrovskými dovednostmi. Velmi zajímavý byl především způsob jakým tyto organizace využívaly potenciál příběhů. Oba vedoucí pracovníci vyprávěli příběhy s cílem ovlivnit a zlepšit chování svých pracovníků. Jeden z manažerů používal příběhy dokonce k zlepšování motivace pracovníků, kteří se museli vyrovnat se změnovým programem a to velmi úspěšně. Bohužel, tři zmíněné organizace představovaly z hlediska celého výzkumu pouze výjimky [5].

7 Závěr

Tacitní znalosti jsou pro moderní organizaci důležité. Nástroje, které pomáhají řídit práci s tacitními znalostmi, tedy příběhy, komunity a učňovství, se rychle rozvíjejí. V současnosti probíhá ve světě velké množství výzkumných i jiných aktivit, které jsou zaměřeny na komplexní práci s tacitními znalostmi nebo přímo na některý z nástrojů zmíněných v tomto článku.

Výzkum provedený v organizacích v České republice ukázal, že naše organizace mají v práci s tacitními znalostmi velké mezery a používají-li je, pak (kromě jediné ze sledovaných organizací) pouze intuitivně. Velkým zklamáním pro výzkumný tým byla také skutečnost, že pouze 2 organizace z 25 sledovaných ve výzkumu, souhlasili se zveřejněním svých zkušeností.

Reference

1. DAVENPORT, T. H., PRUSAK, L. *Working Knowledge*. Harvard Business School Press. UK. 1998. ISBN 0-87584-655-6
2. DENNÍNG, S. *The Springboard. Butterworth-Heinemann*. USA. 2001. ISBN 0-7506-7355-9.
3. DENNING, S. *Telling Tales.* HVB Review, May 2004, Vol 82, Issue 5
4. MAGUIRE, J. *The Power of Personal Storrytelling: spinning Tales to Connect with Others*. NY. Jeremy P. Tarcher/Putnam. 1998
5. MLÁDKOVÁ, L. *Moderní přístupy k managementu, tacitní znalost a jak ji řídit.* Beck. 2005
6. MLÁDKOVÁ, L. *Management znalostí*. VŠE. Praha. 2005. ISBN 80-245-0474-X.
7. RICOER, P. *Myths as a Bearer of Possible Worlds*. 1991
8. SCOTT, J.E. *Facilitating Interorganisational Learning with Information Technology.* Journal of Management Information Systems. Podzim Vol 17 Issue 2. 2000.
9. VEBER, Jaromír a kol. *Management, základy, prosperita, globalizace*. Management Press. Praha. 2000. ISBN 80-7261-029-5.
10. WENGER, E. Communities of Practice: Learning. Meaning and Identity.1998.
11. WHITMORE, J. *Koučování: Rozvoj osobnosti a zvyšování výkonnosti.* Management Press Praha 2004. ISBN 80-7261-101-1

Annotation:

Work with tacit knowledge in the CR

Tacit knowledge is invisible. Organizations and their managers tend to underestimate it. Unfortunately results of their neglect are often very unpleasant. People can deal with tacit knowledge in two ways. They can transform it to explicit knowledge or leave it in tacit form and share it. Tacit knowledge is always linked to the individual or group of humans, his/her/their character. When externalized, tacit knowledge is often damaged. It is more

convenient to leave it in tacit form and share it. There are three tools that enable us to share tacit knowledge – storytelling, communities and apprenticeship. The article briefly characterize these tools and provides the reader with results of the research on tacit knowledge made in Czech companies.

A comparison of two approaches to temporal granularity modelling

Daniela Ponce

Dept. of information technologies, FIM, University Hradec Králové,
Rokitanského 62, 500 03, Hradec Králové

daniela.ponce@uhk.cz

Abstract. This paper deals with two approaches to temporal granularity modelling: temporal granularity and theory of granular partitions. A possible conversion and compatibility between the temporal granularity and temporal granular partition is also discussed. Finally, some further questions related to the theory of granular partitions are formulated.

Keywords: temporal granularity, theory of granular partitions, ontological modelling

1 Introduction

Our ability to conceptualize the reality at different levels of granularity and switch between them as needed is the basic condition of our intelligence and flexibility ([4]). In this paper we compare two approaches to granularity modelling: temporal granularity and granular partition. The first approach is time specific, the second approach is more general.

We take first a closer look at the granularity of time as defined by the theory of temporal granularity.

2 Theory of temporal granularity

The problem of temporal granularity modelling is usually tackled together with the problem of time representation and temporal reasoning ([7]). We introduce a definition based on the classical definition Bettini-Wang-Jajodia ([1]). It will be referred as the definition of the theory of temporal granularity (TTG).

Definition. *Let I be the set of indeces such that I is a subset of the set of all integers, and let* $\leq_I$ *denote the usual order on I. Let T be the absolute time, and let* 2^T *denote the set of all its subsets. Let* $\leq_T$ *denote the temporal order on T. The temporal granularity on (I, T) is a mapping* μ *from I to* 2^T *such that for each i, j from I such that* $i <_I j$, $\mu(i) \neq \emptyset$ *and* $\mu(j) \neq \emptyset$ *hold:*

(1) for each t_i , t_j *from T such that* $t_i \in \mu(i)$ *and* $t_j \in \mu(j)$ *holds:* $t_i <_T t_j$,

(2) for each k from I such that $i <_I k$ *and* $k <_I j$ *holds:* $\mu(k) \neq \emptyset$.

The first condition means a requirement on the monotony of the mapping. The second condition means a requirement that a mapping of an index be non-empty, if the

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 215–222, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

mapping of smaller and greater indeces is non-empty. The μ(i) set is called the i^{th} tick of the temporal granularity or more generally a granule of the temporal granularity. Commonly used temporal granularities like day, month or year satisfy the definition of the temporal granularity.

2.1 Properties of granularity

The definition of the temporal granularity allows such granularities like working day (skipping some subsets of time between granules), working month (defining granules of different size and skipping some subsets of time within granules) or ruling years of a ruler (finite number of granules, leaving out some subsets of time). We can define several properties of the temporal granularity ([3]).

A granularity is **externally continuous**, iff non-empty granules follow one other, so no gaps[2] exist between non-empty granules; it is **internally continuous**, iff granules do not contain gaps; it is **continuous**, iff it is externally and internally continuous. A granularity is **complete**, iff the granules cover the complete absolute time, and **uniform**, iff the size of all non-empty granules is equal.

The temporal granularity day is externally and internally continuous (and therefore continuous), complete and uniform. The temporal granularity working day is not externally continuous, it is internally continuous, it is not complete and it is uniform. The temporal granularity month is externally and internally continuous (and therefore continuous), complete, and it is not uniform. The temporal granularity working month is externally continuous but it is not internally continuous, nor complete, nor uniform.

2.2 Systems of granularities and their properties

In many problem domains we need to work with such a model which enables to formulate temporal references within various temporal granularities and which guarantees the correct interpretation of assertions with temporal references including conversions from one temporal granularity to other. In systems with multiple temporal granularities we can examine relationships existing between the granularities ([5]).

The granularity G1 is **finer** than the granularity G2, iff every granule of G1 is a subset of a granule of G2. Example: the temporal granularity day is finer than the temporal granularity month; the temporal granularity week is not finer than the temporal granularity year.

The granularity G1 is a **subgranularity** of the granularity G2, iff every granule of G1 is equal to a granule of G2. Example: the temporal granularity leap year is a subgranularity of the temporal granularity year.

The granularity G1 is **smaller** than the granularity G2, iff every granule of G1 is a subset of the granule of G2 with the same index. Example: the temporal granularity working week is smaller than the temporal granularity week.

[2] A gap is a non-empty interval of absolute time which is not a subset of any granule.

The granularity G2 is a **union** of the granularity G1, iff every granule of G2 is a union of any number of succeeding granules of G1. Example: the temporal granularity month is a union of the temporal granularity day.

The granularity G2 is **shifted** with respect to the granularity G1, iff there exists a constant (greater than or equal to zero) such that adding the constant to indeces of all granules of G1 we get G2. Example: the temporal granularity year_of_christian_calendar is shifted with respect to the temporal granularity year_of_jewish_calendar.

3 Theory of granular partitions

The theory of granular partitions (TGP) is motivated by the fact that human cognition of the reality is based on pattern recognition. In the cognition of the reality we use „a certain grid of labelled cells and we recognize objects of reality as being located in these cells“ ([2]). The granular partition is such a grid of cells. An object of reality is understood as any portion of reality which can be distinguished from the rest of reality.

The basic terms of the granular partition theory are cell, partition, subcell, projection and location. The cell roughly corresponds to the cognitive construct „concept“, „class“ or „type“ in the sense that it refers to the totality of qualities possessed by objects that are instances of the same concept, individuals belonging to the class, objects of the same type. The partition means a system of cells.

The subcell is a relationship holding between two cells. As it follows from the context of the theory, in some cases the subcell corresponds to the taxonomic relationship is-a, while in other cases to the mereological relationship is-a-part-of[3]. The is-a-part-of relationship is not further specified. It can be interpreted as any of the six types ([6]) of the part-whole relationships: homeomeric/heteromeric variant/invariant configuration of parts within a whole, and variant/invariant collection of parts within a whole.

The projection is a relation holding between a cell of a partition and such an object of the reality which possesses all qualities referred by the cell[4]. The location is a relation holding between an object and such a cell of the partition that all qualities possessed by the object are referred by the cell.

The granular partition theory specifies relations between cells and partitions and also relations between partitions and objects in reality. Several restrictions are imposed on the partition (for details see [2]) yielding a structure isomorphic to a one-root oriented tree of finite-length branches. Tree nodes correspond to partition cells, and two nodes are connected by oriented edge iff the corresponding cells are in the relation subcell. The root of the tree corresponds to such a cell of the partition, which is not a subcell of any other cells of the partition. This cell is called the root cell of the partition.

[3] When two cells are in a mereological relation, four cases can occur: identity, proper parthood, overlapping and disjointness.

[4] It is supposed that the projection of the partition on a reality is unique.

Restrictions are also put on the projection and location relationships. They are required to be inverse one of the other and functional. Therefore at most one object[5] can be located into any cell, and any object can be located into at most one cell. For example, cells labelled Mark Twain and Samuel Langhorne Clemens collapse into one cell, as the same person would be located in both cells. Objects located into cells of the partition are said to be recognized by the partition.

It is also required that the partition preserve the mereological structure of objects. Preserving the mereological structure means that the following implication holds for all cells of the partition: object located in a subcell of some cell is a part of the object located in that cell. However, if two cells are not subcell one of the other, the theory is silent about the mereological relation of the objects located in such two cells. For example, in the partition thought of by a child the cell labelled tadpole may not necessarily be a subcell of the cell labelled the life cycle of frog.

The partition may be defined in such a way that in some cells of the partition no objects will be located, and vice versa, some objects will not be located in any cell of the partition. For example, no object will be located into the cell labelled Atlantis. On the other hand, when a new animal is discovered, there may be no cell into which it can be located.

The granular partition theory requires that there exist an object which is located into the root cell of the partition. This object is called the domain of the partition.

Generally, a partition does not recognize the proper parts of whole objects which it recognizes. If a partition recognizes a whole and also its parts, it is because there is a cell in the partition which recognizes the whole and there are also cells which recognize the parts of the whole. Recognition of parts therefore stops at leaves of the partition. Objects located in leaves of a partition may have parts, but the parts are not recognized by the partition.

3.1 Granularity in the granular partition

In the granular partition, granules in general are disjoint, or one is a proper part of the other, because we are allowed to construct the partition in such a way only that cells are in the subcell relation one to the other. The granular partition is such a system of granules, which encaptures the is-part-of relation holding between granules of different granularities, but the granularities themselves are not explicitly defined.

We introduce the following recursive definition of the granularity in the granular partition: (1) The root cell of the partition is a granularity.; (2) If a part of the partition is a granularity, then replacing any cell of the granularity by its all immediate subcells defined in the partition yields a granularity, too.

[5] It should be noted here that an object of the reality may in fact mean a multi-object. A cell labelled Mimi is projected to an individual cat named Mimi, but a cell labelled cat is projected to a multiobject all-cats. We can say that a cell labeled cat is projected to that portion of reality that possesses qualities of „being of the biological species cat“.

3.2 Temporal granular partition

The granular partition theory is general, because it does not specify what type of reality portions the cells refer to. Therefore we can apply the granular partition to time domain, too. We will denote it as temporal partition. In the following we give the interpretation of those basic terms of the granular partition theory only, which are interpreted specifically to the temporal context.

In the temporal context, absolute time is considered as the reality and an interval of absolute time is considered as a portion of reality. An interval of absolute time can be of zero-length (instant of time), and it can also be non-contagious. There may exist a temporal concept which refers to a kind of temporal interval. Example of such temporal concept is a concept of standard time unit (second, minute, ...), calendar units (week, month), calendar timing-references (concrete days of week, concrete months of year). The relationship being subcell can be interpreted in two ways: as is-a relation and as is-a-part-of relation. Is-a relation is understood in the same way as for the object partitions. Is-a-part-of relation is understood as is-a-subinterval-of relation between time intervals. When performing projection and location operations, we search for such portions of absolute time which correspond to the temporal qualities referred by the cell.

The relation holding between two cells may not be overlapping. The overlapping is not allowed as it would imply that a cell should be a subcell of two different cells. Examples of overlapping temporal granules are pairs “day, weekend” (the overlap is Sunday, because Sunday is-a day and Sunday is-a-subinterval-of weekend), “week, month” (the overlap is day, because day is-a-subinterval-of week and day is-a-subinterval-of month), or “day, time-to-rest” (the overlap is Sunday, because Sunday is-a day and Sunday is-a time-to-rest). Whether the overlap is detected, depends on our knowledge of the two temporal concepts referred by the two cells and/or on our knowledge of the mereological relation of the objects the cells project onto.

In the partition theory for objects we required that the projection and location relations were functions. We satisfied this requirement by specifying the context in such a way, that we restricted the temporal dimension of the reality to a fixed time-interval within which neither the reality nor our knowledge about it does change.

When we construct a temporal partition, we also need to specify the context (called time-reference specification), in which the projection and location relations can become functions. Such a time-reference specification includes e.g. specification of absolute-zero-time-point in the absolute time, time authority (geo-political system), time zone, but eventually other specification, too.

3.3 Properties of temporal granular partition

Projection and location relations can be further restricted. We can thus define several properties related to the projection and relation of the granular partition ([2]).

A granular partition is **mereologically monotone** iff it reflects the mereological structure of the reality. That is, if any two objects of the granular partition, object-the-part and object-the-whole, that are recognized by the partition are located in such

cells, that object-the-part is located in the subcell of the cell in which object-the-whole is located.

A granular partition is **complete** iff every cell of the partition recognizes some object, so there are no empty cells. For example, a temporal granular partition with cells corresponding to ruling eras of Egyptian pharaohs is complete, if the ruling interval of every pharaoh is known, and it is not complete, if we know about some particular pharaoh, but we do not know when he/she ruled.

A granular partition is **exhaustive** iff it recognize all of those objects in its domain which satisfy $some\,special\,condition \leftarrow e \triangleright g \triangleright object\,size \rightarrow \triangleright For\,example$ a temporal granular partition with cells corresponding to ruling eras of Egyptian pharaohs is not exhaustive, if we know about a particular pharaoh era, but we know nothing about the ruling pharaoh.

A granular partition is **full** iff every cell of the partition is identical to the mereological sum of its immediate subcells. For example, a temporal granular partition composed of cells ruling dynasty, ruling eras of dynasty members is not full, if a mythical and therefore never-ruled emperor cell is included as well.

A granular partition is **cumulative** iff every cell of the partition projects onto the mereological sum of the objects projected by the immediate subcells of the cell. For example, a temporal granular partition composed of cells week, Monday, Tuesday, …, Sunday is cumulative, because there is no temporal part of week which is not located in any of the cells Monday, Tuesday, …, Sunday.

4 Temporal granularity vs. temporal granular partition

Trying to match (temporal) granular partition to (temporal) granularity, we find that the notion of cell roughly corresponds to the notion of granule. When we define a granular partition, we define a system of granules. However, the resulting granule system is different from the granule system defined by the temporal granularity.

First of all, the temporal granularity defines granules in such a way that any two granules of the temporal granularity are disjoint[6]. It can be easily seen that for any temporal granularity such a temporal granular partition can be constructed, which consists of an (ordered) finite or infinite list of cells and which projects onto the temporal granularity. In this partition no subcell relations hold and it lacks the unique root cell. We can define a cell referring to "any time in the absolute time" and make it the root of the temporal partition. All other so-far-defined cells of the temporal partition will be made its subcells.[7] We will call such a partition the flat partition. In the theory of temporal granularity, such an extension means that we add a trivial

[6] The definition of the temporal granularity has a stronger implication: not only are the temporal granules pairwise disjoint, but their convex hulls (as defined for intervals) are pairwise disjoint, too.

[7] Note, however, that the granular partition theory allows to define such a temporal granular partition, which includes a cell corresponding to a temporal granule with a hole in it (a temporal granule with a temporal subinterval left out from it). Such a temporal partition will project to a system of temporal intervals which cannot be ordered as required by the theory of temporal granularity.

temporal granularity (of a single temporal granule covering the whole absolute time) to the already existing granularity and we thus get a system of two granularities.

The temporal granularity specifies also restrictions concerning the ordering of granules but these restrictions can not be specified by the means of the theory of granular partition.

A temporal granularity is defined in the domain of reality, where the reality is the absolute time. When we want to define a temporal granularity corresponding to a temporal granular partition, we need more knowledge about the partition projection onto the reality.

The granular partition theory restricts the relation between the partition and its projection onto reality. However, the partition theory does not restrict the mereological structure of the reality onto which the partition projects, because it only ensures that the partition does not misrepresent the mereological relation between the (temporal) objects of the reality. If we have no knowledge about the relation of two concepts, they will be modelled as two minimal cells. When we project the two cells onto reality, they project onto two (temporal) objects which may stand one to other in any of the possible mereological (interval) relations.

The temporal granularity restricts the relation of temporal objects; therefore, a more restricted definition of the partition is needed in order to use it for temporal domain and receive similar effects as those following from the theory of temporal granularity. Or, a restriction can also be specified on the concepts which are modelled by cells of a partition. We can restrict the concepts used to those about which we have knowledge on their relation.

4.1 Special case of flat temporal granular partition

All cells except the root cell of a flat partition are pairwise disjoint and thus they can be compared to the (disjoint) granules of the temporal granularity. All cells are also subcells of the root cell and the root cell projects onto the whole absolute time. In the case of a flat temporal granular partition, we consider the completeness and exhaustiveness properties only, because other properties are formulated as restrictions on subcell relations.

Whether the partition is complete or not, its possible correspondence to a temporal granularity is not affected. The fact that a cell is not empty and there is an object located in it has no correspondence in the definition of the temporal granularity. On the other hand, continuity (external and/or internal) of the temporal granularity has no correspondence in the granular partition.

If the granular partition consists of a single cell (and the root cell) such that it is projected on a multi-object, then the partition corresponds to the uniform temporal granularity. Example of such temporal granular partition is a single cell labelled day or month. If a cell labelled month has no subcells defined in the partition, then there is no way within the partition how to compare the extent of two special months, e.g. February and August.

If the granular partition is exhaustive, then it corresponds to the complete temporal granularity. Each (temporal) object satisfying the special condition will be located in a cell, so no (temporal) object will remain left out by the partition. The contrary

implication may not hold: a complete temporal granularity does not necessarily correspond to the exhaustive partition, because nothing ensures that a (temporal) object will pass the special condition required by the exhaustive granular partition.

5 Conclusion and further questions

The theory of granular partition defines (temporal) granularity in such a way, that we capture not only the partition of the reality into granules, but also the structuring knowledge on the cells of the partition as well. In the case of mereologically monotone partition, reasoning about the partition's mereological structure we can derive conclusions about the granular structure of the reality.

However, the theory of granular partition needs more restrictions on its possible interpretations. For example, the subcell relation is interpreted in some cases as taxonomical relation and in other cases as mereological relation. Such vague specification may cause problems especially when several cells and their relationships are analyzed. We suggest that the subcell relationship should be interpreted as mereological relation only, and the taxonomical relation should be captured by ontology. Granular partition could then include complete the ontological structure of a domain. It can be interesting to interprete specifically the six different types of the mereological relationships for cells of the temporal domain.

In the comparison of the temporal granularity ([1]) and temporal granular partition based on ([2]) we paid no attention to the comparison of systems of temporal granularities and particular granularities (derived or not) of the temporal granular partition.

References

1. Bettini, C., Wang, X.S., Jajodia, S.: *A general framework for time granularity and its application to temporal reasoning*. Technical Report ISSE-TR-96-10 (1996).
2. Bittner, T., Smith, B.: A Theory of Granular Partitions. In *Duckham, M., Goodchild, M.F., Worboys, M.F. (Eds.): Foundations in Geographic Information Science* (2002) 113-145.
3. Combi, C., Franceschet, M., Peron, A.: Representing and reasoning about temporal granularities. *Journal of Logic Computation* **14** (1) (2004) 51-77.
4. Hobbs, J.R.: *Granularity*. (2002)
5. Ning, P., Jajodia, S., Wang, X.S.: An algebraic representation of calendars. *Annals of Mathematics and Artificial Intelligence* **36** (2002) 5-38.
6. Odell, J.J.: Six Different Kinds of Composition. *Journal of Object-Oriented Programming* **5** (8) (1994) 10-15.
7. Ponce, D.: Některé otázky reprezentace času v kontextu znalostních systémů. In *Proc. Kognice a umělý život '05, Smolenice* (2005). (*in Czech*)

A GUHA Style Association Mining Using Prefix Tree Data Structure

Lukáš Vlček

Department of Computer Sciences, FEI, VŠB - Technical University of Ostrava
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
lukas.vlcek@gmail.com

Abstract. In this article a new implementation of GUHA method (an original method for association mining) using data storage mechanism built on top of a prefix tree is proposed. Usage of prefix tree for itemset and association mining has already been introduced (FP-growth); however, it is not directly applicable to the GUHA method without prior generalization. This article describes such generalization where only one pass through data set is needed. Unlike other data structures used for association mining, prefix tree is also suitable for direct visualization which can bring additional information about data in question.

Key words: association mining, GUHA method, generalized FP-growth algorithm

1 Introduction and Motivation

The term *association mining* was first widely popularized during the 90's in connection to market basked analysis [2]. In its original form, association mining was used to infer co-occurences from binary data[1].

The core of every association mining algorithm is the need to identify frequent patterns called *frequent itemsets*. This task itself became popular to the point that a specialized workshop [10] was started in 2003. When all frequent itemsets are identified and their frequencies in the data are known then one can derive association rules in the form $\varphi \rightarrow \psi$ where both φ and ψ represent conjunctions of positive binary attributes. An association rule is evaluated according to its *confidence* and *support*. They are calculated as conditional probability $P(\psi|\varphi)$ and probability $P(\varphi \wedge \psi)$, respectively. If both the values exceed user specified thresholds, the association rule is accepted and it is called an *implication*.

Interestingly, it is very natural for association mining methods to be applied to tasks with both positive and negative attributes. Soon, after all, it was found that simple co-occurence is far from real implication and generalized correlations and associations based on statistical tests were introduced as well [3, 4]; however, the task of itemset mining stays focused on identification of positive

[1] Form of data is binary matrix where rows correspond to individual market transactions and columns to purchased items (only positive attributes are considered).

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 223–230, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

frequent patterns without further investigating the type of associations between individual data attributes. This fact becomes hurdle when one wants to apply new itemset mining algorithm in association mining task if both positive and negative attributes are taken into account.

This article introduces generalized form of itemset mining algoritm FP-growth within scope of GUHA – a well developed association mining method with rich system of various association functions.

In this article, first the GUHA method is briefly reminded, then a generalized itemset mining algorithm is presented and it is shown that not only GUHA can benefit from it but also it makes sense to visualize results for the user in order to bring additional information about the data.

2 A GUHA Method

GUHA method (General Unary Hypothesis Automaton) was born in the 60's and it has been continuously developing since then [1, 9] and periodically being re-implemented. GUHA assumes data in the form of a rectangular matrix $\mathcal{M}$ where rows correspond to objects and columns to properties. Properties are boolean values where the presence of a specific property is coded by *true* or *false*. We are looking for association rules in the form $\varphi \sim \psi$ where both φ and ψ are disjunctive non-empty sets of properties of $\mathcal{M}$ such that the truth[2] value of φ and ψ is equal to *true*. Then, the pair $\{\varphi, \psi\}$ and the given data matrix $\mathcal{M}$ determine four frequencies:

a = the number of objects satisfying both φ and ψ
b = the number of objects satisfying φ but not ψ
c = the number of objects satisfying ψ but not φ
d = the number of objects satisfying neither φ nor ψ

This can be represented as a *four-fold table (4f-table)*:

$\mathcal{M}$	ψ	$\neg\psi$	
φ	a	b	r
$\neg\varphi$	c	d	s
	k	l	m

where $r = a + b$, $s = c + d$, $k = a + c$, $l = b + d$ are marginal frequencies and $m = a + b + c + d = r + s = k + l$ is the total number of objects in matrix $\mathcal{M}$.

A *quantifier* $\sim$ is given by a function $Tr_{\sim}(a, b, c, d)$ associating each 4f-table either 1 (*true*) or 0 (*false*). The formula $\varphi \sim \psi$ is true in the data, if and only if $Tr_{\sim}(a, b, c, d) = 1$ for a given 4f-table [1, 5, 8].

Examples of GUHA quantifiers are listed in tab. 1. For evaluation of FIMPL it is sufficient to know frequencies a, b only. Note that FIMPL is similar to *implication* described above. But if we want to evaluate Fischer [1] or AA quantifier

[2] Both positive and negative properties are taken into account such that φ and ψ are atomic or composed boolean expressions, e.g.: $\varphi = (A \wedge \neg B \wedge C)$, $\psi = (\neg D \wedge E)$.

Table 1. Example of GUHA quantifiers

Quantifier	**Evaluation of** $Tr_{\sim}(a,b,c,d)$	**Parameters**
Founded implication (FIMPL)	$a \geq Base, \frac{a}{a+b} \geq p$	$Base$, p
Fischer's quantifier (Fischer)	$a \geq Base$, $ad > bc$, $\sum_{i=a}^{min(r,k)} \sigma(i,r,k,m) \leq \alpha$ where $\sigma(a,r,k,m) = \frac{r!s!k!l!}{m!a!b!c!d!}$	$Base$, α
Above average (AA)	$a+b>0$, $a+c>0$ $\frac{a(a+b+c+d)}{(a+b)(a+c)} \geq (1+p)$	p

[7] (see also [4]) then we have to know all frequencies a, b, c, d from 4f-table and this is the point where itemset mining algorithms lack enough generality, as they are focused on positive literals only. It means that they don't cover all frequencies in the 4f-table. This fact makes them hardly applicable to the GUHA method without prior generalization. The following text presents generalization of itemset mining algorithm FP-growth in connection to the GUHA method.

3 Generalized FP-growth Algorithm

Various itemset mining algorithms have been proposed during recent years; however, recent results [10] show that among all known algorithms none is superior to the others. A comparative study [6] gives a glance at approaches that have been succesfull so far. Probably the youngest algorithm (called FP-growth) is based on the idea of preprocessing the data to the form of a prefix tree and then using directly this new data structure for efficient itemset mining [11, 12]. A parallelized version of this algorithm has been shown as well [13]. We are building on this idea by extending it to a more general form which allows evaluation of GUHA quantifiers based on frequencies from 4f-table.

We consider each object in data $\mathcal{M}$ a code word (more formaly we can say that code word is vector of object properties) and because both positive and negative properties are treated equaly then all code words are of the same length. Given data from tab. 2. the first row is represented by code word $\{A, B, C\}$, the second one by code word $\{A, \neg B, C\}$, etc.

Table 2. Data $\mathcal{M}$; the *rowNo* value defines order in witch rows will be processed.

rowNo	A	B	C
1	1	1	1
2	1	0	1
3	1	0	0
4	0	0	0
5	0	0	1

Before we can construct prefix tree from code words we need to define an order function for properties. Let P be set of all properties in data $\mathcal{M}$, then $ord(x, y)$ is **order function** such that

$$ord(x,y) = \begin{cases} +1 \text{ if } frq(x) > frq(y) \\ +1 \text{ if } frq(x) = frq(y) \text{ and } chr(x) > chr(y) \\ -1 \text{ otherwise} \end{cases}$$

where $x, y \in P$ and $x \neq y$ and $frq(x)$ is function $frq : P \to \mathcal{N}$ which is determined by evidence of given property x in $\mathcal{M}$. Function $chr(x)$ defines lexicographical order. Now we say that $x > y$ iff $ord(x, y) = 1$ and $x < y$ iff $ord(x, y) = -1$.

The order on properties can be defined in other ways but the order function above will suffice our needs.

3.1 Prefix Tree

A prefix tree is composed of code words where all properties in each code word are ordered according to $ord(x, y)$ and satisfies the following propeties:

1. Tree nodes correspond to individual properties in code words.
2. All nodes in every branch are ordered according to order function such that $ord(x, y) = 1$ where x is parent and y is its child.
3. Every node (except top most parents and leaves) has only one parent and set of children $\{c_1, \ldots, c_m\}$ where $1 \leq m < i - 1$ and i is the number of properties in $\mathcal{P}$.
4. For each node n, there is a function $f(n)$; $f : n \to \mathcal{N}$.
5. If node n is not a leaf and if it has m children then $f(n) = \sum_{k=1}^{m} f(c_k)$.
6. $\sum^{\mathcal{M}} f(n)$ = total evidence of property n in data $\mathcal{M}$.
7. Evidence of given sequence of nodes $\mathcal{S} = \{n_1, \ldots, n_m\}$ in $\mathcal{M}$ is determined by $\sum^{\mathcal{M}} f(n_i)$ where $\forall x \in \mathcal{S} : ord(x, n_i) = 1$ over all tree branches containing all nodes from $\mathcal{S}$ (note that this property indirectly defines value $f(n)$ for any leaf).

Let $\mathbf{V}$ be vector of all properties from $\mathcal{P}$. Accoring to data in tab. 2. vector $\mathbf{V} = \{\neg B, A, C, \neg A, \neg C, B\}$. Order of properties in $\mathbf{V}$ is given by order function $ord(x, y)$ and data $\mathcal{M}$.

3.2 Prefix Tree Composition

Now, let's demonstrate the construction of a prefix tree based on data in tab. 2. We will be processing one row at a time for the sake of clarity. Particular steps are illustrated in fig. 1.

We start with empty $\mathbf{V}$ and empty $\mathcal{M}$. When the first row is added into $\mathcal{M}$ then $\mathbf{V}$ turns into $\{A, B, C\}$ and the prefix tree is initiated by brach with three nodes (a). With the second row the $\mathbf{V}$ is updated to $\{A, C, B, \neg B\}$ and nodes B and C in the tree are interchanged. Then, it is sufficient to add only one node $\neg B$ into tree (b). With the third and fourth rows the tree nodes need to be reordered a little bit more (c) and one completely new branch is added, respectively (d).

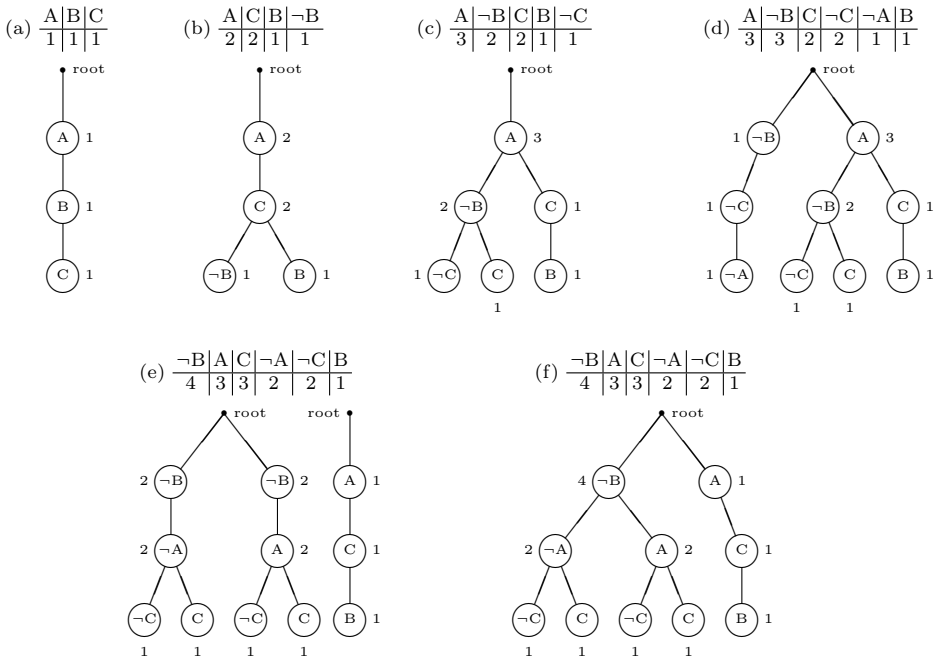


Fig. 1. Step by step construction of prefix tree based on data in tab. 2. There is an actual vector $\mathbf{V}$ above every composition along with $\sum^{\mathcal{M}} f(n)$ for every node in it.

Tree node reordering. In fact reordering of tree nodes happens when any node is propagated towards the tree root. For node n and its parent p it is enough to simply interchange them if $f(n) = f(p)$. This is the situation when parent has only one child. But if $f(n) < f(p)$ then the situation gets more complicated. Such situation is depicted in fig. 2. If we need to propagate node n_1 towards root by one position then we have to split its parent n_3 into two nodes n_3' and n_3'' where $f(n_3') = f(n_1)$ and $f(n_3'') = f(n_3) - f(n_1)$. This action requires only one additional node to be created and some manipulation with pointers. Then, it is enough to simply interchange nodes n_1 and n_3'.

However, not every newly created node is really useful. In some cases there exists other composition of the tree where new node is not needed. Back to fig. 1. this situation is illustrated in composition (e) which is result of adding fifth row from tab. 2. to composition (d). We can see that there exists other composition (f) where only one node with $\neg B$ is needed. Creation of redundant nodes is the price we pay for node reordering, on the other hand mining association rules from both (e) and (f) yields the same results. In fact we don't always need compact form of the tree (tree without redundant nodes) but sometimes it makes sense to get rid of unnecessary nodes by rebuilding the whole tree.

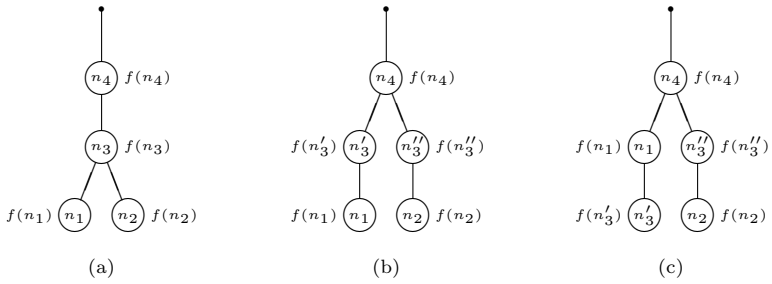


Fig. 2. Reordering of tree nodes. If node n_1 is propagated towards root by one position then new node n'_3 has to be created.

4 Algorithm Pseudo Code

The following is pseudo code of algorithm which produces prefix tree data structure from given data matrix.

```
01: model = new Model();
02: vector V = model.getV();
03: while ( Data_Source.data_available() == true ) {
04:   new_data_row = Data_Source.get_data();
05:   V.update_from( new_data_row );
06:   if ( V.order_is_changed() == true ) {
07:     model.reorder_nodes();
08:   }
09:   new_data_row.reorder_according_to( V );
10:   model.add( new_data_row );
11:   if ( model.rebuild_required() == true ) {
12:     model.rebuild();
13:   }
14: }
```

We use `model (01)` as a container for prefix tree data structure and vector **V** `(02)`. We don't expect incoming data rows to have any particular order of properties. Every time `new_data_row` is received we check if adding it into `model (05)` causes change in order of properties in **V** `(06)`. If order is changed then we need to reorder whole `model` according to new order `(07)`. Then we reorder `new_data_row` according to new order too `(09)` which allows us to add it into `model (10)`.

Frequencies of properties in **V** are changed with every `new_data_row` and so can change their order. It may be necessary to reorder tree nodes in `model` if order of properties is changed. Often such node reordering leads to increased number of nodes and `model` then requires more memory then would be needed by a compact form. To keep the `model` compact we have to rebuild it `(12)`. It can be done by extracting individual branches from current `model` and storing them into new `model`.

It is a question how often the `model` should be rebuilt as this can be a time consuming process. On the other hand the more compact `model` the more maintainable it is and the association mining process becomes faster.

For a small data set it is sufficient to rebuild `model` after a fixed number of `new_data_rows` or just once after the last data row. The more complicated solution would be to check for an increase in memory usage and act according to given system constraints. Other approach for large data sets could be to check for significant changes in properties frequency using goodness of fit statistics.

5 Mining Association Rules from Prefix Tree

In order to evaluate a GUHA quantifier $Tr_{\sim}(a, b, c, d)$ we need to know frequencies in the 4f-table. It is easy to derivate frequency of any sequence of properties from prefix tree. To make it more efficient we can interconnect all nodes representing the same property by pointers including property in $\mathbf{V}$. Then, properties listed in 3.1 make prefix tree traversal more efficient. For details see [11–13].

6 Visualization of Prefix Tree

When mining association rules only specific information from $\mathcal{M}$ is needed. More specifically, we can discard information about order of columns and rows in original data set $\mathcal{M}$. Prefix tree fits perfectly into this situation because we are able to reconstruct original $\mathcal{M}$ with the exception of order of columns and rows.

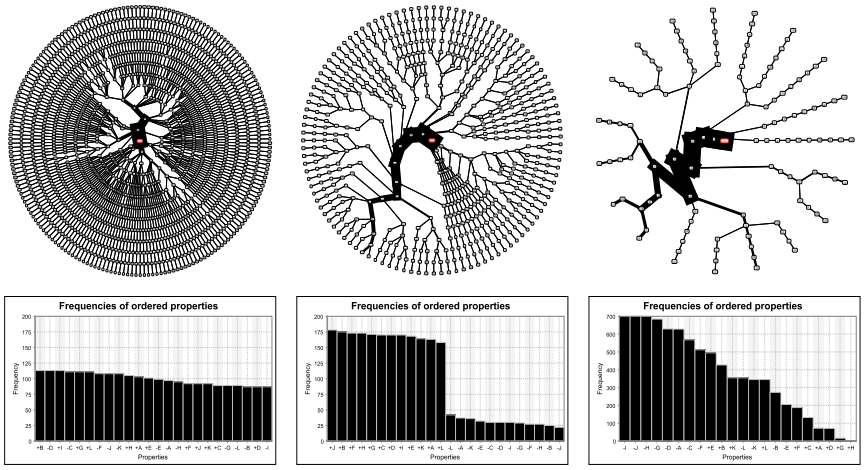
Visualization of prefix tree to the user seems to be an interesting option because shape of prefix tree can say something about density and distribution of patterns through data set. This can be a useful technique especially when one is facing random data whithout previous knowledge. Special attention should be given to data which appears to contain typical features of randomly distributed data (see fig. 3).

7 Conclusion

A generalized form of itemset mining algorithm FP-growth was described. It was shown that it is possible to construct general prefix tree data structure where both positive and negative properties are stored. It was also shown that prefix tree can be build dynamically during one pass through data. Moreover it was shown that prefix tree can be used in connection to the GUHA method because it allows evaluation of all quantifiers based on 4f-table. Visualization of prefix tree can also bring additional information about explored data.

References

1. Petr Hájek, Tomáš Havránek. *Mechanizing Hypothesis Formation - Mathematical Foundation for a General Theory.* Springer, Berlin 1978. ISBN 3-540-08738-9.



(a) (b) (c)

Fig. 3. Synthetic data with properties distributed according to uniform distibution (a) and Gamma distribution (b). In both cases $||\mathcal{M}|| = 200$. Data extraceted from web application log file where user activities are recorded (c). Here, each row of $\mathcal{M}$ corresponds to one type of user activity. Note that cardinality $||\mathcal{M}|| = 700$ but only fraction of nodes was needed for the construction of this prefix tree. Good compress ratio promises significantly frequent patterns to be uncovered.

2. Rakesh Agrawal, Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. *VLDB 1994*.
3. Sergey Brin, Rajeev Motwani, Craig Silverstein. Beyond Market Baskets: Generalization Association Rules to Correlations. *ACM-SIGMOD 1997*.
4. Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, Shalom Tsur. Dynamic Itemset Counting and Implication Rules for Market Basket Data. *ACM-SIGMOD 1997*.
5. Petr Hájek. Logics for data mining (GUHA rediviva). *Neural Network World, 2000, 10*.
6. Jochen Hipp, Ulrich Güntzer, Gholamreza Nakhaeizadeh. Algoritms for Association Rule Mining - A General Survey and Comparison. *ACM SIGKDD*. SIGKDD Explorations 2000.
7. Jan Burian. Datamining a AA (Above Average) kvantifikátor. *ZNALOSTI 2003*. Ostrava 2003. ISBN 80-248-0229-5. (in Czech)
8. Petr Hájek. Briefly on the GUHA method of Data Mining. *Journal of Telecommunications and Information Technology, 2003, 3*.
9. Petr Hájek. Metoda GUHA v minulém století a dnes. *ZNALOSTI 2004*. ISBN 80-248-0456-5. (in Czech)
10. FIMI'03 - Frequent Itemset Mining Implementations, *http://fimi.cs.helsinki.fi/*
11. G. Grahne, J. Zhu. Efficiently Using Prefix-trees in Mining Frequent Itemstes. *FIMI 2003*. Helsinki, Finland 2003.
12. M. El-Hajj, O. R. Zaïane. COFI-tree Mining: A New Approach to Pattern Growth with Reduced Candidacy Generation. *FIMI 2003*. Helsinki, Finland 2003.
13. O. R. Zaïane, M. El-Hajj, P. Lu. Fast Parallel Association Rule Mining Without Candidacy Generation. *FIMI 2003*. Helsinki, Finland 2003.

Část II (Part II)

Enabling Grids for E-sciencE – The EU EGEE Project

Luděk Matyska[1,2]

[1]CESNET, Zikova 4, 162 00 Prague
[2]Institute of Computer Science, Masaryk University Brno, Botanická 68a
602 00 Brno, Czech Republic

Ludek.Matyska@muni.cz

Abstract. The EGEE project builds a pan European Grid that will serve as an information technology infrastructure for the European Research and Educational Area (EREA). Its own middleware—gLite—provides all the basic services needed to run a Grid, including workload management and scheduling, information and monitoring services, and data management. The 70 project partners are organized in 10 federations, with a Regional Operations Center taking care of all EGEE Grid operation related activities in the respective region. The resource owners are encouraged to join their resources to the EGEE Grid, as while keeping full control over the use of the resources, they can offer their users new services and easy access to remote resources of their peers. Virtual organizations put together users with similar, usually application oriented needs. A specific Virtual Organization for Central Europe, VOCE, is run by the CE federation to provide an application neutral environment for users without explicit affiliation to any other VO. Immediately after initial training, also provided by EGEE, users can join VOCE and freely use its resources to get more deep understanding of EGEE Grid potential. The EGEE II project is currently under review to provide the pan European Grid services till the end of the 6FP.

1 Introduction

Last year, the European Commission launched the EGEE project—Enabling Grid for E-science in Europe—to build a large scale production quality Grid infrastructure that will span the whole Europe and will provide service for all the major research communities. The project has been conceived and is lead by CERN, as the high energy physics community is in deep need for a large scale infrastructure to store and process huge amounts of data that will be produced when the experiments at the Large Hadron Collider (LHC) will start in 2007.

The EGEE was launched in April 2004 as a two year first phase of a four year project, with 70 partners—including Russia and few oversea participants—and a budget of more than 35 million Euros. To cope with a complexity of such a large project, all partners were organized in ten federations, which covered individual European regions: Central Europe, France, Germany and Switzerland, Ireland

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 233–236, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

ad UK, Italy, Northern, Russia, South-East Europe, South-West Europe, and CERN as a federation of its own. Czech Republic is represented by CESNET, the national Internet and in recent years also Grid providers for the academic and research community.

Early in 2005, while retaining its acronym, the project has been renamed to *Enabling Grids for E-SciencE*, dropping Europe from the name. This change reflected interest from non European countries and regions (most specifically Asia) to join the project's effort, but demanding that he project will not limit itself to Europe only. The project thus got a truly global influence, working closely with all the major Grid related activities.

The work is performed within *Networking*, *Service*, and *Joint Research Activities*, which are assigned 28 %, 48 %, and 24 %, resp., of the total budget.

Human networking is the key focus of the Networking activities. This naturally includes project management, but the most important work is done as part of the training and induction and application support activities. Other aspects of human networking are covered by the dissemination and outreach—including partnership with industry and press—and international collaboration, that works with partners outside Europe.

The setup and daily operation of the Grid is a responsibility of Service activities. Each federation is responsible for running a Regional Operations Center (ROC), that coordinates the resources, monitor them and provides the first line of support for the regional users. The computing and data storage resources are not directly paid by the project. They are provided by partners and collaborators—the so called Resource Centers (RC)—in an expectation that bringing them together will provide a qualitatively new services. The ROC coordinates deployment of the EGEE middleware, called *gLite*, helps individual Resource centers to run and monitor it and also takes care of the resource allocation in the region.

The middleware itself is developed and tailored as part of the Joint Research Activities of the EGEE project. The Workload Management System (WMS) the Data Management and the Monitoring are the three most important gLite components developed as part of the JRA1. The performance of the Grid is evaluated from statistical data collected and processed by the JRA2, which is also responsible for the overall quality assurance of the project's software. The security aspect is taken for by the JRA3 and network related development is done within the auspices of the JRA4.

2 The gLite middleware

To make a collection of computing resources, data resources and the network a Grid, a middleware must be deployed to coordinate these individual components into a unified distributed system. The EGEE project aimed to use as much of the existing middleware components, developed as part of other project, modify them if necessary, develop the missing pieces and put all these parts together under a brand new name of the *gLite* middleware. However, the necessity to build a

production Grid from the first month of the project lead to the adoption of LCG middleware, that has been put together as part of the CERN Grid project (the Large Collider Grid). This software was based on a preceeding CERN lead DataGrid project and has been specifically tailored to suit the high energy physics community. Hence, for the first year of the EGEE project, its Grid has been based on the LCG middleware. This lead to some misunderstanding of what constitutes the EGEE product and what is the proper relationship between EGEE and LCG—the use of LCG middleware does not mean that EGEE is the HEP Grid. On the contrary, currently the LCG is using the EGEE middleware, adding HEP specific functions, services and properties.

The current gLite structure is described in the recent EGEE deliverable [2]. The gLite provides an open source implementation of the foundation services that are application independent as well as a set of higher-level application specific services that are built on top of it. The gLite is also able to accommodate higher-level services from other sources.

3 Grid operation and Virtual Organizations

Each site that is willing to provide resources for EGEE Grid must deploy the basic foundation gLite services. After successful certification, provided by a corresponding ROC, the services are continuously monitored (see [3]).

The EGEE users has access to the Global Grid User Support (GGUS) system, that allows them to book a request in one place, regardless of where a problem on the Grid occurred (in fact, a user usually knows no reason nor a place of his/her problem). Central European region, as some of the others, provides also regional and even country level entry point to the GGUS.

While the infrastructure operation is organized by regions, users are grouped around the Virtual Organization (VO) concept. Each group of people having similar requirements on the infrastructure use—mostly based on a common set of applications—is expected to create its own VO, that takes care of all the higher-level middleware services. The VO establishes its own resource broker(s), data management and catalogue services, and logging and bookkeeping services that keep track of running jobs. Also, VO experts join the user support team to provide application specific expertise to problems related to that VO. Also, the VO takes care of the resources, negotiating with resource owners for the appropriate share, and including also resources from the user communities themselves. The EGEE project established more than 40 VOs, mostly in areas where users are already well organized. Apart from the HEP, which formed several large VOs, bioinformatics and astrophysics are the most prominent communities, followed by computational chemistry, geo-sciences, Earth observation community and others. However, most of these VOs are are thinly populated, as the barriers to join the VO are rather high.

The Central Europe has followed a different approach. A VOCE, Virtual Organization of the Central Europe, is an application neutral VO that has been build and is maintained by CESNET. All CE partners provided some resources

and the VOCE is available for all users that are interested to use EGEE Grid but are not (yet) well organized to create and support their own VO. The existence of VOCE allows users to access and use the EGEE production Grid immediately after the initial training, that is also regularly provided in all CE countries. Users get access to all the EGEE services, including a set of applications ranging from general purpose packages like Matlab to highly sophisticated applications already ported to the EGEE Grid. Any person with a personal certificate issued by an approved CA can register to VOCE (see [4]) to get access to its international resources.

4 EGEE II

In September 2005, CERN together with more than 80 partners submitted a proposal for the second phase of the project, the EGEE II. The proposal passed successfully the first phase of the review and will step into the negotiation phase, where th exact budget for this second phase will be determined—proposal asked for a budget nearing 40 millions of Euro. The EGEE II project aims to continue the activities of EGEE, extending the research communities directly involved and providing a specific support for smaller research groups that do not have resources nor available man power to directly participate in operating and managing the infrastructure.

The EGEE II structure is very similar to the one EGEE uses, but more focused on the production level quality of the gLite software developed and better integration of gLite development and Grid operation. Therefore, the development related to the security and computer networks has been "dissolved" into the JRA1 and SA1 activities. Another Service activity, SA3, will be established to take over the integration, testing and certification of the gLite middleware.

5 Conclusion

The EGEE project has successfully set up a large scale Grid infrastructure that is operating even outside the Europe. Ten of thousands of processors are connected via an EGEE provided middleware—currently the LCG version being replaced by the genuine EGEE gLite middleware. The EGEE Grid, while mostly used by large organized groups like high energy physics or bioinfomratics communities, is open to European researchers to put their resources on it in exchange to become part of a large infrastructure, that may both provide a link between separate communities and also to offer—as a best effort service—free resources. Smaller groups will be supported by application neutral VOs, like the VOCE.

References

1. The EU EGEE project web pages: `http://www.eu-egee.org`
2. gLite architecture: `https://edms.cern.ch/document/594698/1.0/`
3. The EGEE GOC: `http://goc.grid-support.ac.uk/gridsite/gocmain/`
4. The VOCE entry page: `http://egee.cesnet.cz/en/voce/index.html`

Nástroje pre získavanie, organizovanie a udržovanie znalostí v prostredí heterogénnych informačných zdrojov*

Pavol Návrat, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
Ilkovičova 3, 842 16 Bratislava, Slovensko
{navrat,bielik}@fiit.stuba.sk

Abstrakt V príspevku opisujeme výskumný projekt, ktorý sa zameriava na prácu s informáciami a znalosťami v prostredí heterogénnych informačných zdrojov Internetu. Úlohou je skúmanie modelov pre prácu s informáciami a znalosťami: model informačného obsahu, model metadát, model používateľa a kontextu a skúmanie metód odvodzovania s dátami reprezentovanými v modeloch aj s využitím ontológií. Výsledky riešenia možno vidieť v dvoch rovinách: podstatná časť predstavuje výsledky základného výskumu v oblasti podpory životného cyklu znalostí v prostredí heterogénnych zdrojov, druhou je aplikácia výsledkov v návrhu nástrojov v pilotnej realizácii v doméne pracovných ponúk.

Kľúčové slová: informácie, znalosti, web so sémantikou, ontológie.

1 Zameranie projektu

Množstvo dostupných informácií a znalostí neustále rastie. Ich rozsah, rozdiel v kvalite a dostupnosť sa stále zvyšuje aj vďaka vzniku a rozšíreniu siete Internet a jeho služieb (najmä. WWW - angl. World Wide Web). Tieto možno použiť ako vhodné prostredie pre skúmanie nových spôsobov získavania znalostí z heterogénnych zdrojov, organizovania, overovania, ohodnocovania a udržovania aktuálnosti informácií a znalostí.

Problém získavania a spracovania informácií a znalostí, ktoré sú skryté na webe v súčasnosti zaujíma viaceré výskumné tímy z celého sveta. Ako príklady môžeme uviesť napríklad projekt AKTORS podporovaný anglickou vládou (www.aktors.org), projekty podporované Európskou úniou ako On-To-Knowledge (www.ontoknowledge.org), REWERSE (rewerse.net), Knowledge Web (knowledgeweb.semanticweb.org), či projekt SIMILE (simile.mit.edu),

* Táto práca vznikla v súvislosti s riešením rovnomenného projektu štátneho programu výskumu a vývoja "Budovanie informačnej spoločnosti" a tiež za podpory Agentúry na podporu vedy a techniky na základe Zmluvy č. APVT-20-007104.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 237–242, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

ktorý predstavuje spoluprácu konzorcia W3C, MIT Libraries a MIT Computer Science and Artificial Intelligence Laboratory. Mohli by sme sem zaradiť aj viaceré projekty skúmajúce jednotlivé aspekty práce s informáciami a znalosťami na webe v rámci Čiech a Slovenska.

Na Slovensku doteraz boli aktivity v uvedenej oblasti značne roztrieštené. Získaním zákazky na dodanie riešenia úlohy výskumu a vývoja "Nástroje pre získavanie, organizovanie a udržovanie znalostí v prostredí heterogénnych informačných zdrojov" štátneho programu výskumu a vývoja "Budovanie informačnej spoločnosti" Slovenskou technickou univerzitou v Bratislave (pracovisko Ústav informatiky a softvérového inžinierstva, FIIT STU) so spoluriešiteľmi z Univerzity Pavla Jozefa Šafárika v Košiciach (pracovisko Ústav informatiky PF UPJŠ), Ústavu informatiky Slovenskej akadémie vied a fy Softec sa utvoril viac ako 30-členný tím informatikov pozostávajúci z prevažne mladých výskumníkov. Význam tohto projektu možno vidieť v dvoch rovinách: (1) v rovine výskumnej, keď témou sú dnes veľmi aktuálne otázky práce s informáciami a znalosťami v rozsiahlom heterogénnom priestore Internetu, (2) v rovine integrácie slovenského výskumu, keď projekt otvoril možnosť spolupráce pomerne veľkého počtu (najmä mladých) výskumníkov.

Vychádzajúc z hlavného cieľa projektu, ktorým je zlepšiť poskytovanie aktuálnych a relevantných informácií z webu, hlavnú náplň projektu predstavuje skúmanie nových spôsobov práce s informáciami a znalosťami v prostredí heterogénnych zdrojov, špeciálne s nedokonalou a vágnou informáciou. V súlade s uvedeným cieľom sa výskum sa orientuje najmä na tieto oblasti:

- modely heterogénneho prostredia (neurčitosť, systémy pre modelovanie nedokonalej informácie; modely aplikačnej domény; modely používateľa; modely kontextu; modely navigácie; metadáta a ontológie; viacjazyčnosť; multiagentové systémy),
- získavanie znalostí (odporúčanie informácií; získanie modelu používateľa/prostredia; špeciálne jazyky pre flexibilné dopytovanie; tvorba ontológií),
- organizovanie znalostí (ontológie; rôzne induktívne metódy; zhlukovanie; indexovanie; siete malého sveta),
- prezentovanie znalostí (adaptívna navigácia; adaptívna prezentácia obsahu).

Aplikačnou doménou pre projekt je poskytovanie informácií a znalostí o ponuke voľných miest na trhu práce. O spoločenskej dôležitosti takejto aplikácie a najmä jej skvalitnenia oproti súčasnému stavu snáď netreba rozsiahlo argumentovať. Ide o typický problém, ktorý má heterogénny charakter hneď vo viacerých dimenziách. Samotné zdroje, obsahujúce ponuky, môžu byť veľmi rozdielne - od svetových koncernov po lokálne firmy. Spôsob ich prezentácie bude rôzny, i keď určité spoločné znaky v štruktúre aj v obsahu sa dajú odhaliť. Jazyk prezentácie bude tiež rôzny – v Európskej únii je viac než tucet oficiálnych jazykov, o ďalších ani nehovoriac. Aj v nich sa však môžu ponuky práce vyskytovať a nemusia byť o nič horšie než tie ostatné. Ďalšími dimenziami rôznorodosti sú napr. profesijná skupina, regionálne špecifiká, ktoré môžu spoluurčovať tvar a obsah informácie o pracovnej príležitosti.

V celej tejto húštine viacnásobne rôznorodých informácií treba vyhľadať, sprístupniť a prezentovať používateľovi tie, ktorú sú výsledkom čo najpresnejších odporúčaní, zohľadňujúcich jeho potreby alebo preferencie. Poskytované informácie musia byť aktuálne – o ich aktuálnosť alebo jej preverenie sa bude starať agent v pozadí. Nakoniec bude samozrejme rozhodujúce, nakoľko sa takýmto nástrojom zlepší prístup uchádzačov k takým informáciám a znalostiam o ponuke pracovných príležitostí, ktorá mu pomôže nájsť si prácu.

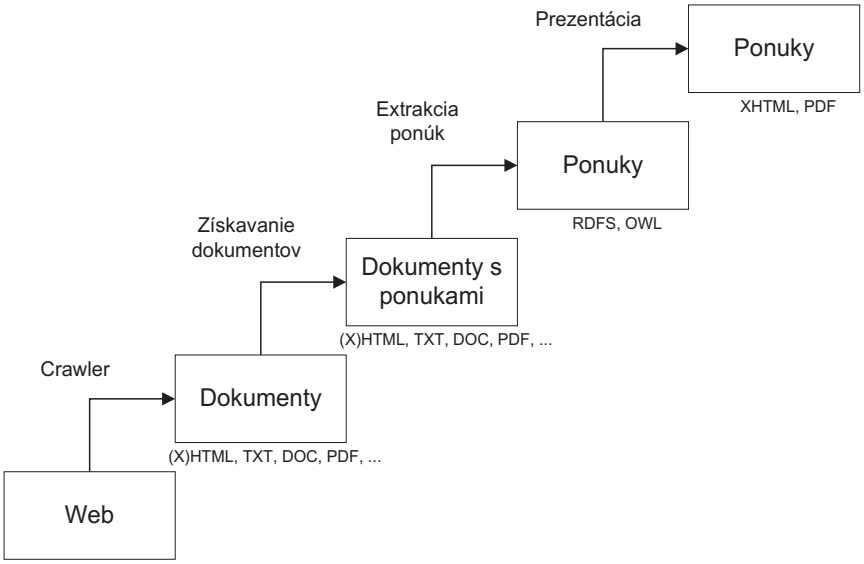
Úlohou pilotnej aplikácie je efektívne vyhľadať informácie a znalosti o voľných pracovných miestach a sprostredkovať ich potenciálnym uchádzačom o tieto pozície a tým:

- zlepšiť proces výberu vhodného pracovného miesta uchádzačom zefektívnením práce so zdrojmi ponúk,
- zvýšiť šance uchádzača nájsť si prácu vyhovujúcu jeho možnostiam a požiadavkám sledovaním a prispôsobovaním získaných informácií a znalostí jednotlivému uchádzačovi spolu s poskytnutím pracovných príležitostí zo zdrojov, ktoré doteraz boli pre uchádzača skryté,
- umožniť zamestnávateľom nájsť vhodných uchádzačov, ktorí spĺňajú požiadavky.

2 Od dát na webe k informáciám a znalostiam pre používateľa

Vzhľadom na obmedzený rozsah tohto príspevku nie je možné postihnúť všetky aspekty projektu. Určitý obraz o projekte môže poskytnúť pohľad na navrhované nástroje a ich miesto v procese transformácie dát z webu na informácie a znalosti prezentované používateľovi. Nástroje realizujú reťaz získavania a spracovania dát a ponúk tak, že pracujú na rôznych úrovniach poznania sémantiky jednotlivých zdrojov. Postupne od získavania dát z Internetu, o ktorých môžeme predpokladať, že obsahujú ponuky, pokračuje reťaz cez identifikáciu dokumentov s ponukami, extrakciu ponúk, ich organizovanie a prezentáciu používateľovi (pozri obr. 1). Môžeme hovoriť o transformácii časti webu na web so sémantikou, kedy sa postupne transformuje reprezentácia existujúcich dokumentov do takej reprezentácie, ktorá na svojej najvyššej úrovni obohatí prezentované informácie o sémantické koncepty využiteľné pri ich automatizovanom spracovaní [3].

Návrh komponentov pre nástroje je ovplyvnený aj faktom, že softvérové systémy vo všeobecnosti a zvlášť tie týkajúce sa oblasti hypermédií, sa stále vyvíjajú väčšinou bez organizovaného znovupoužitia, ktoré je jednou z podmienok automatizácie vývoja softvéru. Ad hoc znovupoužitie sa v praxi neosvedčilo. Nevyhnutný je vývoj pre znovupoužitie, ktorý je základom doménovo-orientovaných prístupov vývoju softvéru [11,12]. Tieto prístupy sa zameriavajú na celú doménu aplikácie (problému), ktorá však musí byť vhodne vymedzená. Výsledkom je architektúra a komponenty pre celú rodinu softvérových systémov v danej doméne, z čoho sa potom vytvárajú špecifické softvérové systémy. Táto oblasť je tiež predmetom výskumu v rámci projektu.



Obrázok 1. Transformácia dát z webu na prezentované informácie.

Nástroje na získavanie dát a ponúk

- identifikovanie potenciálnych zdrojov ponúk na Internete – RIDAR (Relevant Internet Data Resource Identification)
- vyhodnotenie relevantnosti analýzou obsahu stránky a vyhodnotenie relevantnosti vzhľadom na požadovanú oblasť – ERID (Estimate Relevance for Internet Documents)
- prehľadávanie a sťahovanie stránok na základe vyhodnotenia relevantnosti obsahu stránky, vkladanie metadát o stiahnutej stránke – WebCrawler [5]
- extrakcia individuálnych ponúk zo zdrojov, v ktorých sú ponuky akumulované v jednom zdroji, napr. v jednom súbore – OSID (Offer Separation for Internet Documents)
- konverzia rôznych formátov dokumentov (HTML, doc, pdf, ...) na text – DocConverter
- editor pracovných ponúk – JOE (Job Offer Editor)

Nástroje na vyhľadávanie

- fulltextové vyhľadávanie spolu s indexáciou textových dokumentov – DaiRTFS (Data Access and Integration Rich FullText Search)
- dopytovanie nad textovými informáciami s podporou relačnej databázy vrátane indexovania – JDBSearch
- vyhľadávanie ponúk s kritériami – Criteria Search
- prístup k rôznym dátovým zdrojom pomocou webových služieb; integrácia dát z jednotlivých zdrojov

Nástroje na analýzu a organizovanie dát a ponúk

- anotácia textových dokumentov s cieľom získavania informácií o ponukách – OnTea [8]
- klasifikácia s použitím induktívneho logického programovania – IGAP [7]
- hľadanie duplicitných informácií a znalostí
- vyhľadávanie komunít v sieťach vytvorením grafu siete, ktorý používateľovi uľahčuje navigáciu – JobCluster Navigator [4]
- pravdepodobnostné zhlukovanie dokumentov – aspektový model [9,10]
- fuzzy zhlukovač – OSFCL (One-Sided Fuzzy Concept Lattice Clusterer)
- odhaľovanie a spravovanie súvisiacich slov – RWM (Related Words Manager)
- nájdenie najlepších objektov z viacerých usporiadaných zoznamov ohodnotených objektov na základe agregačnej funkcie alebo monotónnej klasifikácie, top-k agregátor – Top-k [6]

Rámec pre prezentáciu informácií reprezentovaných ontológiou

- prezentácia ponúk pomocou pohľadov [2]
- prispôsobovanie používateľovi, jeho záujmom, cieľom a ďalším charakteristikám vyjadreným modelom používateľa [1]

3 Záver

Hlavným výstupom projektu je vytvorenie internetového portálu, ktorého účelom je overovanie výsledkov výskumu tým, že bude využívať nové spôsoby práce so znalosťami v prostredí heterogénnych informačných zdrojov pre aplikačnú oblasť pracovných ponúk. Internetový portál sa vytvorí nástrojmi pre získavanie, organizovanie a udržovanie znalostí vytvorených ako overenie metód a techník navrhnutých v rámci výskumu. Výhody a základné charakteristiky navrhnutého riešenia projektu možno zhrnúť do týchto bodov: využitie ontológií (usudzovanie, podobnosť dokumentov), aktívne vyhľadávanie informácií o pracovných príležitostiach (sledovanie firemných sídiel, využitie sprostredkovateľských sídiel), sústredenie ponúk z heterogénnych zdrojov, robustnosť z hľadiska formulácie dopytu.

Projekt je v súčasnosti v druhom roku riešenia. V rámci projektu vznikol pôvodný návrh softvérovej architektúry pre inteligentný manažment znalostí a tiež pôvodný návrh komponentov pre nástroje na získavanie, spracovanie a organizovanie informácií a znalostí. Jednotlivé nástroje sú v rôznych etapách vývoja, niektoré vo fáze špecifikovania, iné vo fáze navrhovania a viaceré nástroje už vo fáze overovania. Ich zaraďovanie do navrhnutej softvérovej architektúry je postupné. Na to, aby bolo možné pracovať na nástrojoch súbežne, vytvárame umelú ontologickú bázu pracovných ponúk, ktorá bude slúžiť na overovanie nástrojov. Je predpoklad, že by mohla byť využiteľná aj v iných projektoch s podobným zameraním.

Referencie

1. Bieliková, M., Kuruc, J. Sharing User Models for Adaptive Hypermedia Applications. In H. Kwasnicka, M. Paprzycki (Eds.), *Proc. of ISDA 2005*, Sept. 2005, Wroclav, Poland, IEEE Computer Society Press, Loa Alamitos, pp. 506-511.
2. Bieliková, M., Grlický, V., Kuruc, J. Rámec pre prezentáciu informácií reprezentovaných ontológiou. In P. Vojtáš (Ed.), *Proc. of ITAT 2005 - Workshop on Theory and Practice of IT*, Sept. 2005, Račkova dolina, pp.325-334.
3. Filkorn, R., Návrat, P. Feature-based Filtering in Semantic Web. In B. Thalheim and G. Fiedler(Eds.), *Emerging Database Research in East Europe, Proc. of the Pre-Conference Workshop of VLDB 2003*, pp. 46-50.
4. Frivolt, Gy., Bieliková, M. An Approach for Community Cutting. In V. Svátek, V. Snášel (Eds.), *RAWS 2005 - Proc. of the 1st Int. Workshop on Representation and Analysis of Web Space*, Sept. 2005, Prague, Czech Republic, pp. 49-54.
5. Gatial, E., Balogh, Z., Laclavík, M., Ciglan, M., Hluchý, L. Focused web crawling mechanism based on page relevance. In P. Vojtáš (Ed.), *Proc. of ITAT 2005 - Workshop on Theory and Practice of IT*, Sept. 2005, Račkova dolina, pp.41-46.
6. Gurský, P., Horváth, T. Dynamic search of relevant information. In L. Popelinsky (Ed.), *Proc. Znalosti 2005*, Vysoke Tatry, Slovakia, pp. 194-201.
7. Horváth, T., Krajči, S., Lencses, R., Vojtáš, P. An ILP model for a graded classification problem. *J. Kybernetika*, 40 (2004) pp. 317-332.
8. Laclavík, M., Gatial, E., Balogh, Z., Habala, O., Nguyen, G., Hluchý, L. Semantic annotation based on regular expressions. In P. Vojtáš (Ed.), *Proc. of ITAT 2005 - Workshop on Theory and Practice of IT*, Sept. 2005, Račkova dolina, pp.305-306.
9. Polčicová, G., and Návrat, P. Semantic Similarity in Content-based Filtering. In Manolopoulos, Y. and Návrat, P. (Eds.), *Proc. of ADBIS 2002 - Advances in Databases and Information Systems*, Springer LNCS 2435, pp. 80-85, 2002.
10. Polčicová, G., Tiňo, P. Making sense of sparse rating data in collaborative filtering via topographic organization of user preference patterns. In *Neural Networks*, Elsevier, Vol. 17, (2004), pp. 1183-1199
11. Smolárová, M., Návrat, P. Software Reuse: Principles, Patterns, Prospects. *Journal of Computing and Information Technology*, 5(1997), 1, 33-48.
12. Vranić, V. Multi-Paradigm Design with Feature Modeling. *ComSIS (Computer Science and Information Systems)*, Vol. 2, No. 1, (2005), pp. 79-102.

Annotation:

Tools for knowledge acquisition, organization and maintenance in the environment of heterogeneous information sources

This paper describes research project that is aimed at design of new ways of processing of information and knowledge in heterogeneous environment, in particular acquisition, representation, organization and maintenance of actual knowledge and developing tools for supporting new model of heterogeneous environment. Results of research will be verified on pilot applications with job market as an application area. This allows processing information about job market much more effectively – when looking for new job positions, when mediating services connected with employment or when selecting an applicant for a job for various job positions.

Workflow in the Digital Mathematics Library Project

How Mathematics is Stored and Retrieved

Petr Sojka

Masaryk University in Brno, Czech Republic
sojka@fi.muni.cz

Abstract. This paper is a progress report of the retrodigitization project of the Czech Digital Mathematics Library, DML CZ. We are aiming to digitize valuable mathematical journals and books (250,000 pages) published in the Czech and Slovak Republics, and make them publicly available in digital form. We describe here the project workflow: the key concept is a gradual enhancement of the digital material by 'knowledge enhancing' filters applied to the markup-rich XML data.

1 Introduction

The proposed WDML, World Digital Mathematics Library, [2] in a digital and easily accessible form is a dream of mathematicians over the world. In addition to the projects Cornell funded by NSF, EMANI: electronic mathematical archiving network, NUMDAM: Numérisation de documents anciens mathématiques, German digital research library funded by German Research Foundation and realized at Göttinger Digitalisierungs Zentrum, a DML-CZ: Czech Digital Mathematical Library project for the retrospective digitization of library materials of mathematical journals and books published in in the Czech and Slovak Republics (200–300,000 pages in total) is being realized [8]. We identify the following main steps in retrodigital document processing:

acquisition document acquisition, preparation, copyright issues handling;
scanning document scanning, main metadata entering, scanning checks;
image processing main OCR, image enhancements;
semantic processing document markup enhancement, semantic processing, document classification, citation linking, document clustering, indexing;
presentation visualization techniques of the document repository, digital library web portal, interfaces to other services and search engines.

In this overall architecture of processing, the *raw data* is transformed into *information* and ultimately, *knowledge*. We workflow is built on *extensible*, *open* formats (XML), while the key data processing on *extensible*, *open source* tools that gradually enhance and enrich the scanned data into a mathematical knowledge library of a new type. The best current practices of previous projects

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 243–247, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

(NUMDAM, JSTOR and the Digitization of the Otto Encyclopædia [7]) are being followed so as not to reinvent the wheel.

The structure of this paper is as follows: Section 2 describes the scanning phase and digital storage issues. We describe the technique of gradual markup enrichment in Section 3. The closing remarks in Section 4 deal with the organization, presentation and delivery of the digitized material.

2 Scanning and Image Processing

Scanning at 600 DPI and 4-bit depth is taking place in Jenštejn near Prague, in the Digitization Center of the Library of Academy of Sciences using two A2 book scanners Zeutschel OS 7000. The original TIFF files are archived before entering the production line. The scanning process itself is inexpensive—pure scanning costs are reported at about ten percent of the whole page processing price [2] thanks to the high degree of automation.

With the Book Restorer software, dozens of image corrections are being made in automated overnight jobs. These include: adjustments to the histogram, binarization, deskew, despeckle, geometrical correction and lighting correction (1D & 2D). Then the data are fed into Sirius, which was developed for the National Library in Prague by Elsyst Engineering. The main metadata (page numbers) are entered and page numbers scanned. The system allows export in XML format with links to the normalized images.

From Image to Text with Visual Markup Documents consisting of a host of images are not much useful unless a full text layer has been created by optical character recognition (OCR) techniques. The OCR program, ABBYY Finereader/PDF Transformer, can provide the text of a document with *visual markup* as a text-under-image searchable PDF or HTML. The text is encoded in Unicode. The links between the textual and visual layers of a document are stored and preserved for further document processing, as XML encoded data.

3 Document Markup Enhancement

Scanned text with *visual markup* has to be converted into a *logical markup* to enable high precision search techniques. This is a difficult step, often ambiguous, reverting the process of typesetting. To face that, as most mathematics papers are typeset with the TEX engine, TEX typesetting rules and fonts have to be taken into account so as to enrich document objects with a *structural markup* in MathML. As MathML (XML namespace), allows for the storage of both *presentation/visual* (e.g. TEX) and *logical/content markup*, it is an ideal format for storing both layers of information. We are currently testing the Infty system [9] for its structural recognition of mathematics, and if successful, it will be merged with FineReader API.

Structure Markup Enrichment The structure markup of document paper can be flat (marking only the key parts of a paper such as title, author, abstract), or it can be more detailed. The level of detail can be specified or enforced by a *Document Type Definition*, DTD, or similar formalisms such as XML Schema, which allow even more detailed type checking.

For many markup tasks, regular expressions can be developed and used [7]. Even more demanding tasks such as the identification and markup of bibliographic entries in a document are performed by smart regular expressions matching various citation styles as shown in CiteSeer [3] or ACM Digital Library projects. Similarly, eXstyles by Inera allows the definition of thousands of rules for the semi-automated editing of poorly marked-up (Word) files. In our workflow, we have sets of regular expressions to locate the main metadata (title, authors, etc.) in the scanned OCR.

The main mathematical databases of reviews and abstracts as Zentrallblatt MATH and MathSciNet already contain most needed metadata (at least for journals)—they are collected as the first iteration of metadata.

An application for metadata editing is being developed at the Institute of Computer Science, Masaryk University, which pairs scanned images with OCR texts, trying to locate available metadata against them. A metadata copyeditor sees all these data in WYSIWYG form, and makes the necessary editing of mandatory metadata fields. After this step, both the metadata and the OCR text with partial XML markup (e.g.bibliographic entries) are saved.

There are numerous ways that lead to markup enhancement, so for every document object, several versions of markup richness should be stored in a repository, allowing the building of pipes of programs to further enhance the markup. Today's tools and technologies of *corpora management* allow for the effective handling of a corpus, the size of the entire published mathematical literature to date.

The problem of *language identification* is an example of a markup enrichment filter. A relatively easy task is identifying the language of document chunks (paragraphs). It is easily extensible, because only bigram statistics of a new language have to be computed. This way, the source language tag for every paragraph or sentence can be added automatically.

Data Storage, Indexing, Semantic Processing All digital documents should be efficiently and effectively stored in a *digital repository system* to ensure

- a platform for digital data enhancement processing,
- the long-term unambiguous identification and preservation of digital material,
- open access-friendly *digital rights management* (DRM).

Currently, the data are loaded into the open source system Kramerius. Various modules for it are being developed (indexing, PDF generation on the fly).

The software developed at the NLPlab at the Faculty of Informatics, Masaryk University has superior scaling capabilities for indexing, querying, browsing and statistical computations of textual corpora: the corpus manager Manatee and it's graphical user interface Bonito. This system is capable of efficient storage, indexing and processing of billions of words. It will be used for a full-text search across the entire repository.

Today, *natural language processing* technologies make it possible to process digital documents not only on the level of syntax, but semantic processing is also needed to achieve the ideal of the *Semantic Web* for which many problems have yet to be solved.

XML languages, such a RDFS in RDF or DAML+OIL, provide a means of machine interpretable document semantics, compatible with our model of linked document layers based on XML.

Mathematics Subject Classification Scheme (MSC) was compiled by the Editorial Offices of Mathematical Reviews (MR) and Zentralblatt MATH (Zbl) and is widely accepted by the publishers of mathematical journals. These are the main sources leading towards the creation of a taxonomy of mathematics (referred to as *ontology* in semantic web jargon) to which documents will be linked.

4 Identification, Visualization and Dissemination

All the document object parts accumulated by intelligent document processing should be stored and linked together. It is essential to have an *open access interface* at least to the archive metadata available for citation indexes and search engines. It should also be accompanied by a *document object identifier* (DOI) or persistent URL (PURL). The expanding use of DOI for scientific data is evident, and the DOI system is on track to becoming an ISO standard.

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) will be used to allow search engines and people on the Web to harvest stored metadata.

DjVU For document delivery, tagged text-under-image format PDF (or PDF/X for printing) such as a rich media container can be *generated* from primary sources, containing layers of scholar's interests. In addition, DjVU format in it's searchable form will serve for image low-bandwidth dissemination as well.

The success of tools based on the TouchGraph engine [6] such as Amazon or Google browsers in the style of WebODAV [1] inspired us to use a similar approach for handling digital library visualization and presentation [4]. Metadata, classification links and relations are stored in the KAON database [5] as an RDFS. It has been verified that the amount of metadata of the whole mathematics literature worth archiving (estimated at about only 50 million pages) is achievable and could be visualized on a modest workstation, even with a rich set of RDF data and document descriptions.

Closing Remarks We have designed an architecture of, and a methodology for building a fully fledged mathematics archive. Semantic enhancements filtering, metadata linking and visualization play a major rôle in the architecture. The research has been supported by the Czech National Programme *Information Society*, Grants No. 1ET208050401, 1ET200190513 and 1ET100300419.

References

1. Mao Lin Huang, Peter Eades, and Robert F. Cohen. WebOFDAV: Navigating and Visualizing the Web On-line with Animated Context Swapping. In *Proceedings of the 7th International WWW Conference*, pages 638–642, 1998.
2. Allyn Jackson. The Digital Mathematics Library. *Notices of the AMS*, 50(4):918–923, 2003.
3. Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6):67–71, 1999.
4. Zuzana Nevěřilová and Petr Sojka. XML-Based Flexible Visualisation of Networks: Visual Browser, 2005. Submitted.
5. Daniel Oberle, Raphael Volz, Boris Motik, and Steffen Staab. An extensible ontology software environment. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 311–333. Springer-Verlag, 2004.
6. Alexander Shapiro. TouchGraph LLC at SourceForge, 2005.
7. Petr Sojka. Publishing Encyclopædia with Acrobat using TEX. In *Towards the Information-Rich Society. Proceedings of the ICCC/IFIP conference Electronic publishing '98*, pages 217–222, Budapest, Hungary, April 1998. ICCC Press.
8. Petr Sojka. From Scanned Image to Knowledge Sharing. pages 664–672, Graz, Austria, June 2005. Know-Center in coop. with Graz Uni, Joanneum Research and Springer Pub. Co.
9. Masakazu Suzuki, Fumikazu Tamari, Ryoji Fukuda, Seiichi Uchida, and Toshihiro Kanahori. INFTY — An integrated OCR system for mathematical documents. pages 95–104, Grenoble, France, 2003. ACM.

Projekt M-CAST: vývoj vícejazyčného systému agregace informací

Vilém Sklenák, Petr Strossa

Katedra informačního a znalostního inženýrství, FIS,
Vysoká škola ekonomická v Praze, nám. W. Churchilla 4, 130 67, Praha 3
{sklenak, kizips}@vse.cz

Abstrakt: Cílem projektu M-CAST je vyvinout vícejazyčný systém umožňující integrovat a prohledávat rozsáhlé kolekce textů (a multimédií). Je popsána struktura navrhovaného systému, dosavadní práce na jeho implementaci a zejména na jeho rozšíření o český jazykový modul.

Klíčová slova: vyhledávání informací, vícejazyčné vyhledávání, lingvistika, ontologie

1 Úvod

Cílem projektu M-CAST[1] (Multilingual Content Aggregation System based on TRUST Search Engine, projekt programu eContent #22 249) je vyvinout vícejazyčný systém, který bude umožňovat tvůrcům obsahu integrovat a prohledávat rozsáhlé kolekce textů (a multimédií), jako jsou internetové knihovny, informační zdroje nakladatelství a tiskových agentur nebo databáze vědeckých informací.

2 O systému M-CAST obecně

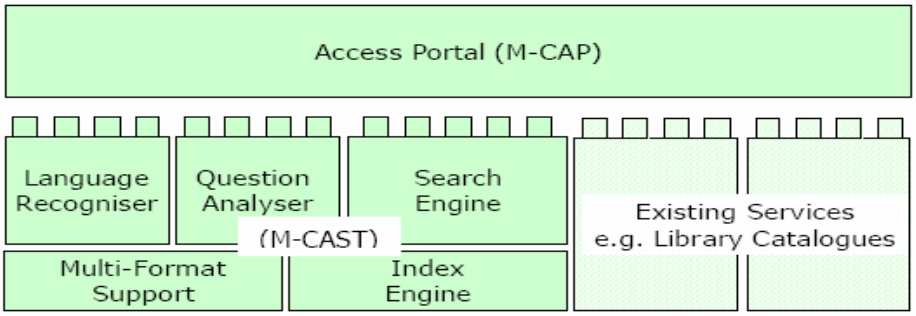
Systém M-CAST umožní tvorbu digitálních knihoven prostřednictvím agregace dat dostupných v různých formátech a z různých zdrojů. Systém bude testován dvěma knihovnami, pro které budou vytvořeny portály agregující multimediální informace (M-CAP, Multimedia Content Aggregation Portals) na základě existujících portálů a systémů. Tyto portály budou schopny hledat odpovědi na dotazy v přirozeném jazyce v rozsáhlých kolekcích vícejazyčných dat. Prezentační vrstva portálu bude multimediální, díky čemuž bude možné zpřístupnit digitalizované kopie starých tisků, právních dokumentů, notových záznamů, obrázků a videa, i když indexovány budou pouze jejich textové popisy.

Systém M-CAST bude založen na výsledcích projektu TRUST – Multilingual Semantic and Cognitive Search Engine for Text Retrieval Using Semantic Technologies[2] (Vícejazyčný sémantický a kognitivní mechanismus vyhledávání textů využívající sémantické technologie, IST-1999-56416), který byl financován z Pátého

[1] http://www.m-cast.infovide.pl
[2] http://www.trustsemantics.com

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 248–251, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

rámcového programu EU pro rozvoj vědy a výzkumu. Vyhledávací stroj TRUST umí v současnosti vyhledávat ve čtyřech jazycích (francouzštině, italštině, polštině a portugalštině [2]). V rámci projektu bude transformován z jednouživatelské aplikace pro PC na serverovou aplikaci pro operační systém UNIX nebo Windows. Jazykové zdroje [1] systému TRUST budou aktualizovány a v systému nadále využity. K dosud využívané jazykové ontologii (taxonomii) bude vytvořena alternativa založená na standardním mezinárodním desetinném třídění (MDT), celosvětově používaném v knihovních systémech. Systém bude obohacen o dva další jazyky: angličtinu (bude řešena italským partnerem projektu TRUST) a češtinu (bude doplněna v rámci projektu M-CAST).



Agregační systém M-CAST bude ústředním prvkem agregačního portálu M-CAP a bude vyvinut podle zásad metodologie návrhu systémů pro správu obsahu založených na znalostech (Knowledge-based Content Management Application Design Methodology). Tuto metodologii vytvořila firma Infovide, S.A. v rámci jiného projektu financovaného z Pátého rámcového programu EU – ICONS[3] – inteligentní systém pro správu obsahu (Inteligent Content Management System, IST-2001-32429).

Portály budou nasazeny a testovány ve dvou veřejných knihovnách: v Polské internetové knihovně[4], jejímž provozovatelem je Kopernikova knihovna v Toruni, a v Národní knihovně České Republiky v Praze[5]. Cílem je zpřístupnit vícejazyčné digitální informační zdroje pro zodpovídání dotazů v přirozeném jazyce. Výsledkem hledání budou fragmenty textu, které obsahují odpovědi na dotaz .

Výsledkem projektu bude komerční produkt pro management vícejazyčných znalostí. Očekává se komerční využití výsledků v knihovnách (vyhledávání informací, předmětové katalogy), správě sbírek (akvizice a agregace dat, statistiky výpůjček, vyřazování z fondu), bibliografických databázích, informačních službách (selektivní šíření informací, personalizace, objevování znalostí) a v sémantických datových sítích.

Pro potřeby projektu bylo vytvořeno konsorcium těchto subjektů:

- Infovide S.A., Varšava, Polsko (koordinátor projektu)
- Polská internetová knihovna, Toruń, Polsko
- TiP sp. z o.o., Katowice, Polsko
- Synapse Développement SARL, Toulouse, Francie

[3] http://www.icons.rodan.pl
[4] http://www.pbi.edu.pl
[5] http://www.nkp.cz

- Priberam Informática Lda., Lisabon, Portugalsko
- Expert System S.p.A., Modena, Itálie
- Národní knihovna České republiky, Praha, Česká republika
- Vysoká škola ekonomická v Praze, Česká republika

3 Čeština v systému M-CAST

Lingvistické moduly dosavadního systému TRUST pro jednotlivé jazyky byly řešeny do značné míry nezávisle – obecně sdílely jen základní funkční schéma, které by bylo možné vyjádřit v těchto bodech:

- Texty mají být indexovány jednotlivými slovy, resp. (v případě homonymních / polysémních slov) jejich jednotlivými významy, dále idiomy a jmennými frázemi obsaženými v příslušných slovnících, vlastními jmény obsaženými v příslušných slovnících, pojmenovanými entitami rozpoznanými podle určitých obecných pravidel, koncepty jazykové ontologie (taxonomie) a jmény domén podle speciálního seznamu.
- Indexovaný text tedy musí být podroben zejména morfologické analýze a lematizaci (což je úkol různě náročný podle konkrétního jazyka); vhodná (avšak nikoli absolutně nezbytná) je i určitá omezená syntaktická analýza – alespoň v takové míře, do jaké se o syntaktické kategorie opírají pravidla rozpoznávání významů homonym a polysémních slov: např. některé významy sloves může být vhodné rozlišovat podle toho, zda je či není přítomen předmět nebo příslovečné určení jistého typu.
- Doporučuje se používání derivačního slovníku (tj. slovníku popisujícího vzájemné odvozování slov různých slovních druhů – např. slova *testovat, testování, testovací, testovaný, testující* všechna odkazují na *test*) a tezauru nebo alespoň slovníku synonym.
- Specifickým prvkem každého jazykového modulu je analyzátor typů dotazů (v současnosti je popsáno asi 80 základních typů faktografických dotazů, které se mohou poněkud lišit způsobem vyhledávání potenciálních odpovědí v textech). Rovněž s tímto bodem souvisí potřeba určité omezené syntaktické analýzy. Úlohy dotazování a indexování textů v daném jazyce ovšem mohou být řešeny odděleně a nezávisle – pro daný jazyk není principiálně nutné řešit obě zároveň.

Při realizaci českého lingvistického modulu bylo rozhodnuto v maximální míře využít existující programové kódy pro polštinu, které vyvinula firma TiP, a nově sestavovat pokud možno jen slovníky a datové tabulky ve formátech požadovaných těmito programovými kódy. Dosavadní výsledky potvrzují schůdnost takového postupu. Hypotetická alternativa spočívající v získání a zakomponování jednotlivých modulů, o kterých je známo, že již někde existují (např. pro morfologickou analýzu češtiny), byla zamítnuta jako pravděpodobně celkově komplikovanější.

V současnosti probíhající práce se soustřeďuje zejména na morfologický analyzátor češtiny. Základem v tomto směru je prozatím formalizovaný popis přibližně 300 vzorů skloňování podstatných jmen, 50 vzorů skloňování a stupňování přídavných jmen (zahrnujících i odvozování příslovcí) a 150 vzorů časování sloves. (Každý vzor

je definován určitou – v závislosti na slovním druhu – posloupností ohýbacích koncovek; pro jeden tvar může být specifikováno více alternativních koncovek; v rámci různých vzorů mohou navíc různé tvary vyžadovat různé typy kmenových změn. Principem je, že každé paradigma ohýbání slova odlišující se od jiných je považováno za vzor, i kdyby se jím řídilo jen jediné slovo, nerozlišujeme tedy "pravidelné" a "nepravidelné" způsoby ohýbání slov. Přitom může být shledáno zajímavým, že podle pokusného přiřazení zmíněných vzorů v pracovní databázi celkem 120 000 podstatných jmen, zkompilované z různých volně dostupných zdrojů, existují dva vzory – konkrétně "kost" a "stavení" – pokrývající dohromady 70 % všech podstatných jmen; 11 nejfrekventovanějších vzorů pokrývá skloňování 90 % všech podstatných jmen; pomocí 56 vzorů bychom dokázali popsat skloňování 99 % všech českých podstatných jmen; zhruba polovina z uvedených přibližně 300 vzorů pro podstatná jména popisuje skloňování pouze 1–3 slov, 79 vzorů bylo přiřazeno jen jedinému podstatnému jménu. Podobné výsledky jsme získali i pro přídavná jména a slovesa.)

Zároveň aktuálně probíhají přípravné práce na derivačním slovníku a na slovníku synonym a antonym.

4 Závěr

Projekt M-CAST je plánován jako dvouletý, vlastní práce začaly na jaře 2005. Kromě zdokonalení /doplnění zdrojů pro jednotlivé jazyky byly zahájeny práce na vytvoření ontologie na bázi MDT, podle plánu by měl být na počátku roku 2006 k dispozici funkční prototyp systému. Projekt bude vrcholit koncem roku 2006.

Tento příspěvek vznikl díky podpoře projektu eContent EDC no. 22249 M-CAST. Autoři dále děkují pracovníkům firmy TiP, a zejména Piotru Fuglewiczovi, za pomoc při vývoji české lingvistické podpory pro systém M-CAST.

Reference

1. Amaral, C.; Figueira, H.; Mendes, A.; Mendes, P.; Pinto, C. A Workbench for Developing Natural Language Processing Tools. URL: http://www.priberam.pt/docs/WorkbenchNLP.pdf
2. Amaral, C.; Laurent, D.; Martins, A.; Mendes, A.; Pinto, C. Design and Implementation of a Semantic Search Engine for Portuguese. URL: http://www.priberam.pt/docs/LREC2004.pdf

Annotation:

Project M-CAST: Development of Multilingual Content Aggregation System

The goal of M-CAST project is to develop a multilingual system for integration and searching in large collections of texts (and multimedia). The structure of the proposed system, the work done on its implementation and especially on its Czech language module is described.

Automatické generování vědeckých portálů

Pavel Smrž

Fakulta informačních technologií, Vysoké učení technické v Brně,
Božetěchova 2, 61266 Brno

smrz@fit.vutbr.cz

Abstrakt. Tento příspěvek se zabývá projektem PortaGe, jehož cílem plně automatické, případně počítačem významně podporované vytváření vědeckých webových portálů na základě zadaných počátečních dat a omezení. Uvedeme motivaci naší práce a základní předpoklady, z nichž projekt vychází. Popíšeme hlavní funkce jednotlivých modulů a upozorníme na úskalí vytváření vědeckých portálů. Zvláštní pozornost je věnována použití ontologií, jak pro formování vlastního portálu, tak pro klasifikaci a personalizaci jeho obsahu. V závěrečné části nastíníme hlavní směry budoucího vývoje systému.

Klíčová slova: vědecké webové portály, extrakce informací

1 Úvod

Hlavním způsobem vyhledávání informací na současném webu je zadání klíčových slov či frází do některého ze standardních vyhledávačů. Digitální knihovny, např. ACM DL (http://portal.acm.org/dl.cfm), Springer Link (http://www.springerlink.com) nebo arXiv e-Print Archive (http://arxiv.org), přidávají možnost vyhledávání publikací na základě metainformací – jména autora, názvu práce, časopisu, konference atd. Problémem však zůstává vyhodnocení relevance výsledku hledání vzhledem k potřebám a zájmům uživatele. Současné systémy nejsou schopny pracovat z kontextem hledání.

Projekt PortaGe si klade za cíl rozšířit možnosti specializovaných vyhledávačů vědeckých a výukových materiálů typu Google Scholar (http://scholar.google.com) nebo CiteSeer (http://citeseer.ist.psu.edu/). Typickým uživatelem je doktorand či mladý vědecký pracovník, který potřebuje rozšířit svoje znalosti v určité podoblasti výzkumu. Předpokládá se, že zájem o danou problematiku je dlouhodobé povahy, je tedy vhodné uživatele upozornit na nové publikace, projekty, události, výzvy atd. v příslušné oblasti výzkumu.

PortaGe vytváří vědecký webový portál v dané doméně na základě inicializačních dat. Kromě standardních klíčových slov, seznamu autorů, relevantních časopisů, konferencí a projektů, které oblast zájmu charakterizují, může uživatel zadat počáteční dokumenty, např. webovou stránku obsahující seznam účastníků relevantního semináře, projetku atd., případně vybrat vhodnou sadu uzlů aktuální ontologie (která může být daná předem či automaticky extrahovaná z vyhledaných dokumentů.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 252–255, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2 Základní funkce systému PortaGe

PortaGe zpracovává výstupy několika informačních služeb:

- výsledky dotazů ze systému Google Scholar;
- nalezené články a konferenční příspěvky z placených digitálních knihoven (v současnosti jsou dostupné ACM DL a Springer Link);
- informace z volně dostupných webových zdrojů (arXiv e-Print Archive a CiteSeer)
- metainformace o dokumentech dostupných pouze v papírové podobě (fakultní knihovna).

Systém se skládá z několika propojených modulů, které vykonávají následující činnosti:

1. Metavyhledávání v digitálních knihovnách – vzhledem ke skutečnosti, že mnohé placené i nekomerční archivy vědeckých článků neumožňují automatické stahování a lokální ukládání nalezených dokumentů, musí PortaGe často pracovat s neúplnou informací (např. pouze abstrakt a začátek příslušného článku). Plnou podobu dokumentu se však často podaří doplnit během dalšího zpracování, například z autorovy domovské stránky se seznamem publikací.
2. Uspořádání výstupů z různých vyhledávačů podle předpokládané relevance k dotazu uživatele. Výsledky z Google Scholar, CiteSeer a ArXiv e-Print jsou kombinovány z výstupem metavyhledávače nad digitálními knihovnami ACM DL a Springer Link.
3. Převod do jednotného formátu – soubory v formátech PDF, Postscript, HTML a DOC (experimentálně) jsou staženy a převedeny pomocí standardních nástrojů do prostého textu.
4. Lokální indexování dokumentů – dokumenty v textové podobě jsou zařazeny do fronty pro zpracování lokálním vyhledávačem, založeným na vektorovém modelu. Modul plánovač může změnit priority indexování určité skupiny dokumentů.
5. Extrakce metainformací z textové podoby článků – systém se pokusí na základě jednoduchých heuristik rozpoznat autora a název článku, případně další relevantní informace, a porovnat je s případnými údaji získanými moduly zpracování metadat z webových zdrojů a analýzy stránek obsahujících seznamy publikací.
6. Extrakce sémantických vztahů pro vytváření ontologií – textová podoba článků je dále předložena systému OLE [1], který na základě daného souboru sémantických extrakčních vzorů vytváří ontologickou reprezentaci daného dokumentu a rozšiřuje doménovou ontologii pro danou oblast (vědecký portál).
7. Klasifikace dokumentů – extrahované sémantické vztahy, stejně jako převedená textová podoba slouží pro klasifikaci dokumentů podle kritériích daných dosavadní definicí portálu. V současné době jsou podporovány jednoduché metody klasifikace založené na bayesovském přístupu a kernelových metodách [2], především SVM.
8. Analýza citací – jedním z podkladů pro vytváření sítí dokumentů a vyhodnocování míry relevance vzhledem k uživatelově preferencím (např. upřednostňování přehledových článků) je citační analýza částí dokumentů věnovaných odkazům

na relevantní literaturu. Modul využívá implementaci poskytnutou autory služby CiteSeer.

9. Analýza stránek obsahujících seznamy publikací – pro ověření shody nalezených autorských stránek s kritérii zadanými uživatelem, pro získání seznamu publikací daného autora a pro zjištění nových prací se používá modul (v současné době v experimentální podobě), který se na základě jednoduchých pravidel a heuristik pokouší najít anglickou verzi autorových stránek a odkaz na seznam jeho publikací. Pokud jsou údaje vyhodnoceny jako relevantní a stránka obsahuje plné texty článků v podporovaných formátech, příslušné dokumenty jsou staženy a předány k lokálnímu indexování a dalšímu zpracování.
10. Zpracování metadat z webových zdrojů – některé digitální knihovny a někteří autoři uvádějí na webových stránkách metainformace, které jsou cenné pro další zpracování (doplňkové informace o autorovi, publikaci, řešených projektech atd.). Tato data jsou porovnána s údaji v lokální databázi, případné konflikty jsou zaznamenány pro další zpracování (v současnosti manuální).
11. Analýza změn webových stránek – jednou z nejdůležitějších funkcí systému PortaGe je sledování změn zdrojů, které byly v předchozím zpracování (automaticky nebo v interaktivním módu) vyhodnoceny jako relevantní vzhledem k uživatelově potřebám, zájmům a omezujícím podmínkám. Příslušné webové stránky jsou ve stanovených intervalech stahovány a porovnávány s předchozím stavem. Pokud jsou zjištěny potenciálně významné rozdíly (snaha o filtraci změn omezených na prezentační vrstvu), stránka je předána k dalšímu zpracování extrakčním modulem a porovnávány jsou dále údaje o konkrétních publikacích.
12. Plánovač aktualizace informací – vzhledem k náročnosti pravidelného sledování stránek, jejich analýzy a extrakce a zpracování citací je začleněn modul, který na základě zadaných pravidel a omezení, odhadu náročnosti vycházející z předchozích zpracování a předdefinovaných i zadaných priorit určuje pořadí, ve kterém budou jednotlivé úlohy zpracovávány. Problémem plánovače je nastavení granularity jednotlivých podúkolů. Bohužel se dosud poměrně často stává, že výsledky některých analýz nejsou v požadované době k dispozici, a systém tedy musí pracovat s neúplnými daty.
13. Zpracování metadat z webových zdrojů – některé digitální knihovny a někteří autoři uvádějí na webových stránkách metainformace, které jsou cenné pro další zpracování (doplňkové informace o autorovi, publikaci, řešených projektech atd.). Tato data jsou porovnána s údaji v lokální databázi, případné konflikty jsou zaznamenány pro další zpracování (v současnosti manuální).

3 Role ontologií při výstavbě vědeckých portálů

Poznamenejme přímo v úvodu této části, že s pojmem „ontologie“ zde pracujeme poměrně volně, označujeme jím celou škálu různě strukturovaných zdrojů – od jednoduchých pojmových struktur a tezaurů až po propracované systémy detailních sémantických vztahů, které je často obtížné reprezentovat pomocí běžných standardů.
Základním použitím ontologií v systému PortaGe je definování vlastní struktury portálů. Stěžejní část ontologie obsahuje koncepty pro vydavatele, knihy a knižní

řady, časopisy a jejich zvláštní vydání věnovaná určitým tématům, konference, jejich specializované semináře, případně další vědecká setkání, projekty, výzkumné týmy, autory, články, webové stránky atd. PortaGe předpokládá, že většinu těchto informací je možné sdílet napříč vědeckými oblastmi (některé vědní disciplíny se však mírně odlišují v konceptualizaci příslušných oblastí výzkumu). Primární obecná ontologie (ve formátu OWL) musí být samozřejmě rozšířena o konkrétní instance jednotlivých pojmů (např. názvy důležitých časopisů a konferencí v určitém podoboru). To je možné provést manuálně, nebo „vypočítat" na základě vyhledaných zdrojů.

Ontologie jsou dále používány pro klasifikaci obsahu dokumentů. Jak bylo dříve ukázáno, lze při obecném hledání nahradit tuto metodu tzv. nepravou zpětnou vazbou. Pro velmi úzce zaměřené portály se silně omezenou množinou relevantních dokumentů použitelných pro trénování automatických metod se však použití sémantických vztahů obsažených v ontologiích jeví jako optimální. Zde opět přichází ke slovu automatická extrakce informací z dříve stažených dokumentů – informace (znalost) získaná v předchozích krocích je použita pro klasifikaci akutálních dokumentů.

Ontologie umožňují rovněž detailní určení kontextu příslušného hledání. Uživatel může omezit dotaz na konkrétní dokumenty vyjadřující jistý sémantický vztah mezi pojmy uvedenými v ontologii (např. požadovat pouze články, kde „skryté Markovovy modely" slouží jako „metoda_pro" potřeby „analýzy struktury RNA". Je zřejmé, že on-line zpracování takových dotazů je velmi náročné, proto je vhodné lokální předzpracování dříve vyhledaných dokumentů.

V případě víceuživatelských portálů jsou informace z ontologií využívány pro personalizaci obsahu. Uživatelské profily definují, o jaký typ informací má daný člověk či skupina především zájem. Začínající postgraduální student může například požadovat úvodní či přehledové statě (určováno na základě analýzy citací), zkušení odborníci se mohou spíše zajímat o aktuální novinky, případně se zaměřit na novátorské použití (např. první použití metody známé z jiných oblastí výzkumu).

4 Závěry a směry budoucího vývoje

Je zřejmé, že vytvoření plně automatického generátoru vědeckých portálů je velmi ambiciózním cílem. Některé moduly představeného systému PortaGe jsou dnes již plně funkční a běžně používané, na jiných zbývá ještě mnoho práce. Náš další výzkum v této oblasti se zaměří především na lepší integraci modulu extrakce sémantických vztahů (dnes jako samostatná aplikace) a rovněž na pokročilé algoritmy klasifikace dokumentů založené na kernelových metodách.

Reference

1. Smrž, P., Nováček, V. Text Mining for Semantic Relations that Support a Scientific Portal Generator. Přijatý příspěvek na konfereci *SOFSEM 2006.*
2. Shawe-Taylor, J., Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. ISBN 0521813972.

The *Rainbow* Project: Multiway Analysis of Website Content and Structure

Vojtěch Svátek

Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
svatek@vse.cz

Abstract. The *Rainbow* project aims at automated analysis of websites, especially at extraction, classification and retrieval of useful information. Within the project, various knowledge-based and inductive-learning-based methods are developed. In addition to individual tools, a 'semantic' infrastructure for their integration arises.

1 Project Motivation

The World-Wide Web is a unique repository of information from various domains. However, its content is generally intended for human consumption, thus lacking the structure necessary for processing by conventional software systems. Although the number of web resources endowed with explicit semantic descriptions [3] steadily increases, their relative frequency is still rather small, and unbalanced; for example, websites of SMEs are almost untouched by this technology. Automated analysis of the web space, possibly followed with conversion of acquired information to semantic web format, thus remains a hot topic of research.

The way information is presented on the web combines multiple types and representations of data. Different methods of web data analysis may provide complementary and/or supplementary information. Reducing the analysis to a single method may thus lead to significant information loss. On the other hand, a monolithic application encompassing many methods would be impossible to maintain and reuse. The approach taken in the *Rainbow* project was to develop/adapt independent tools for different types of data and analysis techniques, and to provide them with a web service interface including a lightweight description of 'semantics'.

2 Project History

The *Rainbow* project started in 2001 as an informal joint venture of a group of researchers and students at the University of Economics, Prague. Their joint interest was the rich but heterogeneous and unreliable content of the WWW, in particular, that of websites of smaller companies.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 256–259, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

In the first phase of the project (2001–2003), the methods and tools were mostly designed for non-specific business websites. There were two main implemented outcomes:

- A method for extraction of 'company profile' sentences based on *lexical indicators* discovered via statistical analysis of web directory[1] headings and associated pages [4].
- A method for discovery of navigational structure of company websites, based on graph-theoretical analysis of *local link topologies* [16].

Besides, less sophisticated tools for e.g. page classification based on *URL strings* or for collection of the content of selected *META tags* were developed. The overall *web service infrastructure* was also built in this phase, as well as a simple *result browser*, which eased manual experiments combining multiple tools [8]. The integration of different tools is, to date, manual, based on WSDL descriptions and textual documentation structured according to a pre-defined XML template (this is only feasible given the low number of tools involved). The state of the project in the end of the first phase was summarised in [9].

In the second phase (2003–2005), the project was undertaken with support of the CSF grant no. 201/03/1318, in collaboration with the Technical University of Ostrava and the Free University of Amsterdam (as foreign partner). In the course of this phase, a more restricted area was chosen for experiments, namely, that of websites offering *bicycle products*. The main (implemented) achievements were:

- An *information extraction* tool employing Hidden Markov Models and a 'wrapper ontology' for extraction of bicycle product offers from online catalogues [6]
- A collection of tools for *image analysis*, based on singular value decomposition (SVD), colour histograms and image dimensions [5]
- A *procedural application* that calls individual analysis tools, collects results and converts them to the RDF format (as standard language of the semantic web) [14]
- A *result repository*[2] with simple HTML interface allowing for end-user *search and navigation* [13]
- By the collaborating group at VSB-TU Ostrava, the text-and-XML indexing and query tool *AmphorA* has been adapted for provision of *source data* using the web service technology [2].

Another stream of research focused on abstract descriptions of analysis tools. Different types of knowledge models were subsequently developed:

- A four-dimensional *descriptive framework* called TODD (for task-object-data-domain as its four dimensions)
- A collection of *ontologies* capturing various aspects of the web space

[1] DMOZ, `http://www.dmoz.org`

[2] Based on *Sesame*, see `http://openrdf.org`.

– A collection of *problem-solving methods* (PSM) defining different ways of solving the tasks of classification, extraction and retrieval.

The models are similar in spirit to those developed in projects such as CommonKADS [7] or IBrow [1]; however, they are finer-tuned to the WWW as subject matter for 'inferencing'. The coverage of the models was tested through re-describing real (own as well as others') applications in a semi-formal language [10]. Furthermore, simulations of automated composition of more complex applications from simple components were carried out for the specific task of pornography classification [11, 12]; component simulations were derived from real tools developed within a PhD thesis [15].

Additional bibliography for the project as well links to implemented tools can be found at `http://rainbow.vse.cz`.

3 Future Work

As the exploitation of web content and structure is by far not a resolved problem, the motivation for extending the outcomes (and possibly, duration) of *Rainbow* project is high. The most interesting research goals can be summarised as follows:

– To *automatically build* web analysis applications on the fly, using PSM-based architectural templates and ontological descriptions of individual tools
– To effectively combine *inductively learnt models* with *wrapper ontologies*, for newly addressed domains.
– To make full benefit of the available *XML indexing and querying* technology (as pre-processor to knowledge-based analysis), capable of providing an appropriate XML environment to the knowledge-based web analysis tools.

We envisage to elaborate on these issues in the framework of several newly commenced EU-funded projects as well as in a follow-up CSF project.

Acknowledgements

The *Rainbow* project has been partially supported by the grant no.201/03/1318 of the Czech Science Foundation "Intelligent analysis of the WWW content and structure". The author is thankful to numerous local collaborators on the project, especially (in alphabetic order) to Martin Kavalec, Jirka Kosek, Martin Labský, Ondřej Šváb and Miroslav Vacura, as well as to colleagues from the collaborating institutes: Marek Andrt, Michal Krátký, Pavel Praks and Václav Snášel from the VSB-Technical University of Ostrava, and Frank van Harmelen and Annette ten Teije from the Free University of Amsterdam.

References

1. IBROW homepage, `http://www.swi.psy.uva.nl/projects/ibrow`.

2. Andrt M., Krátký, M., Svátek, V., Snášel, V.: AmphoraWS – webova sluzba pro vyhledavani ve strukturovanych dokumentech. [AmphoraWS – Web service for querying semi-structured data.] In: Datakon'04, Brno 2004.
3. Antoniou, G., van Harmelen, F.: A Semantic Web Primer. MIT Press, 2004.
4. Kavalec, M., Svátek, V.: Information Extraction and Ontology Learning Guided by Web Directory. In: ECAI Workshop on NLP and ML for ontology engineering. Lyon 2002.
5. Labský, M., Vacura M., Praks P.: Web Image Classification for Information Extraction. In: First International Workshop on Representation and Analysis of Web Space (RAWS-05), online `http://CEUR-WS.org/Vol-164`.
6. Labský, Svátek, V., Šváb, O., Praks P., Krátký, M., Snášel, V.: Information Extraction from HTML Product Catalogues: from Source Code and Images to RDF. In: 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), IEEE Computer Society, 2005.
7. Schreiber, G., et al.: Knowledge Engineering and Management. The CommonKADS Methodology. MIT Press, 1999.
8. Svátek, V., Berka, P., Kavalec, M., Kosek, J., Vávra, V.: Discovering company descriptions on the web by multiway analysis. In: New Trends in Intelligent Information Processing and Web Mining (IIPWM'03), Zakopane 2003. Springer-Verlag, 'Advances in Soft Computing' series, 2003.
9. Svátek, V., Kosek, J., Labský, M., Bráza, J., Kavalec, M., Vacura, M., Vávra, V., Snášel, V.: Rainbow - Multiway Semantic Analysis of Websites. In: 2nd International DEXA Workshop on Web Semantics (WebS03), IEEE Computer Society 2003.
10. Svátek, V., Labský, M., Vacura, M.: Knowledge Modelling for Deductive Web Mining. In: Proc. International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004), Whittlebury Hall, Northamptonshire, UK, Springer Verlag, LNCS, 2004.
11. Svátek, V., ten Teije, A., Vacura, M.: Web Service Composition for Deductive Web Mining: A Knowledge Modelling Approach. In: Znalosti 2005, High Tatras, 2005.
12. Svátek, V., Vacura, M.: Automatic Composition of Web Analysis Tools: Simulation on Classification Templates. In: First International Workshop on Representation and Analysis of Web Space (RAWS-05), online `http://CEUR-WS.org/Vol-164`.
13. Šváb, O., Labský, M., Svátek, V.: RDF-Based Retrieval of Information Extracted from Web Product Catalogues. In: SIGIR'04 Semantic Web Workshop, Sheffield, 2004.
14. Šváb, O., Svátek, V.: Procedurální propojen nástroju pro extrakci informací z webových sídel. [Procedural combination of tools for information extraction from web sites.] In: Poster session of Znalosti 2005, Stará Lesná.
15. Vacura, M.: Recognition of pornographic WWW documents on the Internet (in Czech), PhD Thesis, University of Economics, Prague, 2003.
16. Volavka F., Svátek V.: Identifikace navigační struktury webové prezentace na základě topologie odkazů. [Identification of navigation structure of website based on link topology.] In: Znalosti 2004, Brno 2004.

Využitie atribútovo orientovanej indukcie na zlepšenie výsledkov zhlukovej analýzy

František Babič

Katedra kybernetiky a umelej inteligencie, FEI, Technická univerzita Košice,
Letná 9, 040 01, Košice

Frantisek.Babic@tuke.sk

Abstrakt. CRISP-DM je najpoužívanejšou metodikou v procese objavovania znalostí v databázach (KDD). Pozostáva zo šiestich fáz, kde fáza modelovania (data mining) je kľúčovým krokom celého procesu. Medzi hlavné úlohy data mining-u patrí zhluková analýza, ktorej výsledkom je popis zhlukov pomocou hodnôt jednotlivých atribútov. Tento popis je možné zlepšiť použitím atribútovo orientovanej indukcie, čo bolo aj mojím cieľom. Vstupom do algoritmu je dátová množina, súbor s hierarchiami a ukončovacia podmienka: počet rôznych hodnôt atribútov alebo počet záznamov v relácii. Dosiahnuté výsledky sú jednoduchšie, ľahšie interpretovateľné doménovým expertom.

Klíčová slova: AOI, zhlukovanie, hierarchie

1 Úvod

Atribútovo orientovaná indukcia (AOI) je predstaviteľom metód automatickej generalizácie pri popisnom dolovaní v dátach. Používa sa pri stručnom a výstižnom popise danej množiny dát – charakterizácii. Zhluková analýza je proces zoskupovania fyzických alebo abstraktných objektov do tried na základe podobnosti objektov. AOI je jednou z možností ako zlepšiť výsledky zhlukovania, dosiahnuť jasnejší a jednoduchší popis jednotlivých zhlukov.

2 AOI – atribútovo orientovaná indukcia

Algoritmus AOI je predstaviteľom automatickej generalizácie. Používateľ nastaví len jeho vstupné parametre: stupeň generalizácie a ukončovaciu podmienku.

Neformálny postup AOI[3]:

1) Celý postup začína výberom relevantnej množiny dát, ktorá vytvorí základnú reláciu.
2) Preskúma sa stupeň generalizácie dát, čiže počet navzájom rôznych hodnôt pre jednotlivé atribúty a potom ich existujúce hierarchie konceptov.
3) Generalizácia prebieha v dvoch krokoch:
   a) odstránenie atribútu, ak má veľký počet navzájom rôznych hodnôt atribútu a neexistuje hierarchia konceptov pre daný atribút, alebo koncepty vyššej úrovne sú vyjadrené pomocou iných atribútov.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 260–263, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

   b) ak má atribút veľký počet navzájom rôznych hodnôt a existuje pre neho hierarchia konceptov, hodnoty na nižšej úrovni sú nahradené hodnotami na vyššej – všeobecnejšími.
4) Po aplikácii generalizácie sa vypočíta podpora jednotlivých záznamov
5) Krok generalizácie sa opakuje dovtedy, kým výsledná generalizovaná relácia nespĺňa jednu z dvoch možných ukončovacích podmienok:
   a) počet záznamov v databáze klesne pod určenú hranicu
   b) počet navzájom rôznych hodnôt všetkých atribútov klesne pod určenú hranicu
6) Posledným krokom je zobrazenie výslednej generalizovanej relácie.

3 Experimenty

3.1 Predspracovanie

Doménovou oblasťou je databáza nezamestnaných v okrese Humenné na východe Slovenska. K dispozícii som dostal 8 čiastkových databáz, ktoré popisujú rôzne sledované charakteristiky. Na základe konzultácií s expertom, jednoduchých štatistických analýz a logických súvislosti som vyselektoval kľúčové atribúty ako napríklad: Stupeň vzdelania, Prax, Dodatočné vzdelanie, Zdravotný stav atď.

Výsledná dátová množina obsahovala 7966 záznamov charakterizovaných 19 atribútmi. Niektoré atribúty ako napr. Dodatočné vzdelanie, vznikli kombináciou dvoch alebo viacerých pôvodných atribútov. U niektorých pôvodných atribútov som upravil použité hodnoty, napr. reálny atribút Vek som generalizoval podľa štatistických noriem. Zvolil som si rozdelenie do siedmich tried: 0-14, 15-24, 25-34, 35-44, 45-54, 55-64, 65-; ktoré vytvára pravidelnejšie usporiadanie. Rozdelenie na pred, po a produktívny vek vytvára dominantnú triedu produktívny vek.

Napriek všetkým problémom, ktoré sprevádzali predspracovanie, existovala tu aj jedna výhoda a to výskyt chýbajúcich hodnôt. V celej databáze chýbalo len niekoľko hodnôt, ktoré neboli štatisticky významné. Zvolil som alternatívu manuálneho doplnenia na základe podobnosti, aby nedošlo k zmene kontextu a zachovali sa logické súvislosti.

3.2 Zhluková analýza

Na identifikáciu zhlukov som použil algoritmus KMeans, ktorý reprezentuje elementárnu aglomeratívnu metódu zhlukovania. Táto vykazuje porovnateľné, ak nie lepšie výsledky v porovnaní so zložitejšími algoritmami. Tento algoritmus je implementovaný vo freeware Weka, ktorý bol vyvinutý na univerzite Waikato na Novom Zélande

Prvým krokom bol výber počtu sledovaných zhlukov. Pri počte dvanásť dochádzalo k situácii, že viacero zhlukov bolo nutné agregovať, pri troch bol popis príliš všeobecný. Najvhodnejším sa stal počet zhlukov päť. Pri experimentoch som zanedbal binárne atribúty, z dôvodu zahltenia populácie dominantnou hodnotou. Tri

numerické atribúty som transformoval na nominálne, a z nominálnych atribútov som odstránil tie s nevyváženým usporiadaním hodnôt. Výsledok viď Tab.1..

Zhluk	Centroid	Počet záznamov
Zhluk č.0	0 7 1 2 9 9 0 0 1 2 1	28%
Zhluk č.1	1 4 2 2 7 7 0 0 0 2 1	30%
Zhluk č.2	0 2 6 1 0 3 0 0 0 5 1	18%
Zhluk č.3	1 2 2 1 8 8 0 0 1 2 1	11%
Zhluk č.4	1 8 2 2 9 9 0 0 0 2 1	14%

Tab. 1. Popis zhlukov KMeans pri 11 atribútoch

Napríklad zhluk č.1 popisuje mužov vo veku 30-34 rokov, ženatých, vyučených v odbore Technické vedy a náuka, s poslednou a požadovanou profesiou: remeselník, v dobrom zdravotnom stave, Humenné- mesto, žiadna zmena v evidencii.

3.3 Aplikácia AOI na výsledky zhlukovania

Vstupom do AOI je základná relácia. Výstupom zhlukovej analýzy je päť súborov, na každý zvlášť sa aplikuje algoritmus AOI. Prvým krokom bolo vytvoriť hierarchiu konceptov pre jednotlivé atribúty. Pri atribútoch, ku ktorým existovala hierarchia, bolo nutné transformovať hodnoty na hodnoty na najnižšej úrovni hierarchie príslušného atribútu. Hierarchie som vytváral po zvážení významových a logických vzťahov medzi atribútmi a najmä ich hodnotami. Z celkového počtu 19 atribútov som vybral 5 atribútov: Vek, Stupeň vzdelania, Zdravotný stav, Ukončené (vzdelanie), Odkiaľ (miesto bydliska). Pre ostatné atribúty, ako napríklad Pohlavie, nie je možné vytvoriť hierarchiu konceptov z viacerých dôvodov ako napríklad neexistujúca štruktúra údajov.

Pre všetky atribúty som vytvoril jeden súbor s hierarchiami, ktorý je dodatočným vstupom do algoritmu. Ukončovaciu podmienku, najmenší možný počet rôznych hodnôt jednotlivých atribútov, som stanovil na dva.
Pri tvorbe algoritmu v prostredí Java som využil existujúce štruktúry systému Weka. Dosiahnuté výsledky dokumentuje Tab.2..

Zhluk	Počet záznamov pred apl. AOI	Počet záznamov po apl. AOI	Min. podpora	Max. podpora
Zhluk č.0	2253	817	1	94
Zhluk č.1	2357	686	1	135
Zhluk č.2	1397	518	1	55
Zhluk č.3	839	294	1	60
Zhluk č.4	1120	362	1	52

Tab. 2. Výsledky algoritmu AIO pre jednotlivé zhluky

4 Záver

Mojím cieľom bolo pripraviť pilotnú aplikáciu, ktorá by ukázala kompetentným osobám na ÚPSVaR v Humennom možnosti KDD. Hodnotenie prebiehalo ako

prezentácia dosiahnutých výsledkov dvom pracovníkom ÚPSVaR, konkrétne osobe zaoberajúcej sa štatistickým spracovaním a osobe zaoberajúcej sa samotnou evidenciou. Ich závery a hodnotenia som použil ako spätnú väzbu pri experimentoch. Ako vidno z Tab.2., z dosiahnutých výsledkov je jednoduchšie identifikovať zástupcov pre daný zhluk. Ak by sa ešte znížil počet relevantných atribútov pre daný zhluk, došlo by k redukcii len na niekoľko charakteristických záznamov s vysokou podporou. Samozrejme redukcia je do veľkej miery závislá od predchádzajúcich rozhodnutí a použitej dátovej množine.
Aplikácia AOI na výsledky zhlukovania priniesla jednoduchší popis zhlukov, s ním spojenú lepšiu interpretáciu dosiahnutých znalostí doménovému expertovi. Ako možné a vhodné oblasti aplikácie sme v spolupráci s expertom stanovili oblasti:

1. Výber najlepších potenciálnych zamestnancov pre danú pracovnú pozíciu
2. Perspektívnosť učebných odborov, rekvalifikačné kurzy
3. Smerovanie investícií v rámci euroregiónov

Poďakovanie

Práca prezentovaná v tomto príspevku vznikla za podpory Vedeckej grantovej agentúry Ministerstva školstva SR a Slovenskej akadémie vied (VEGA) v rámci projektu č. 1/3135/06 s názvom "Metódy a prostriedky pre tvorbu integrovaných distribuovaných aplikácií na báze ambientov – agentov vyššej úrovne".

Referencie

1. Babič F.: Využitie metód objavovania znalostí pre riešenie reálnej aplikácie, plomová práca. KKUI, FEI TU Košice, 2005.
2. Berka P.: Dobývaní znalostí z databáz. 1.vyd. Praha: Academia, 2003. ISBN 80-00-1062-9
3. Han J. - Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001. ISBN 1-55860-489-8
4. Paralič J.: Objavovanie znalostí v databázach. 1.vyd. Košice: Elfa, 2003. ISBN 80-89066-60-7

Annotation

Use of attribute oriented induction on improvement results of cluster analysis

The objective of this task is to improve results of cluster analysis by AOI (attribute oriented induction). KMeans is one of the simplest unsupervised algorithms that solve the clustering problem. Algorithm AOI is one of techniques of automatic generalization. This technique is used in the fourth phase of CRISP-DM, which is main methodology in Knowledge discovery in databases(KDD). KDD is the practice of automatic search of large for patterns. Input of algorithm is basic relation, file with hierarchy of each attribute and finishing condition. Achievement results are simple, domain expert can better estimate real generality.

Medicínská data a jejich analýza (data mining)

Ladislav Beránek

Katedra informatiky, Jihočeská Universita, Jeronýmova 10, České Budějovice

beranek@pf.jcu.cz

Abstrakt. Předkládaný příspěvek se zabývá charakteristikou medicínských dat z hlediska úloh a algoritmu KDD. Tato data jsou heterogenní a variabilní, podléhají určitým etickým a právním hlediskům. Mohou dále pocházet z různých zdrojů v různé struktuře a kvalitě. Znalostní technologie aplikované v této oblasti jsou však nástrojem, bez kterého se v současné době lékařský výzkum neobejde.

Klíčová slova: medicínská data, charakteristika medicínských dat,

1 Úvod

Tento příspěvek si klade za cíl seznámit odbornou veřejnost, která se zabývá získáváním, zpracováváním, zpřístupňováním a managementem znalostí, s některými aspekty analýzy (data mining) medicínských dat vzniklými na základě zkušenosti autora. Medicínská data mají určité specifické vlastnosti, které je třeba brát v úvahu při jejich zpracování. Jedná se o data, která jsou velmi heterogenní, která podléhají určitým etickým a právní aspektům (nehledě na specifický status medicíny samotné) a která jsou ve srovnání s ostatními biologickými daty obtížněji analyzovatelná a obtížněji se z nich získávají jakékoliv závěry.

2 Specifičnost medicínských dat

Medicínská data jsou obecně velice heterogenní. Pocházejí totiž zpravidla z různých zdrojů jako vyšetření lékaře, data z laboratoře, vyšetření na různých medicínských přístrojích atd. a jsou pak uchovávána v různé formě jako poznámky lékaře, jako výsledky vyšetření - výpis z různých lékařských přístrojů, jako obrazová data a podobně. Všechna tato data mohou nést část informace vztahující se k diagnóze a teprve jejich shrnutím lze dospět k relevantním závěrům. Proto žádný datový zdroj nelze zanedbat bez důkladné analýzy.
Hlavní příčiny heterogenity jsou tedy zejména:

- komplexita medicínských dat - v současné době je značná část medicínských dat uchovávána ve formě obrazů. Většina takových obrazových dat se jen obtížně zpracovává automatickými nástroji. Např. pro interpretaci záznamů EEG nebo ECG signálu je třeba další analýzy lékařů specializovaných v tomto oboru. Tendence je provádění stále komplexnějších vyšetření, což vede i k tomu, že se uchovává velké množství dat a je třeba nových nástrojů k jejich analýze.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 264–267, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

- různé postupy lékařů při pořizování a uchovávání těchto dat. Například při prvním vyšetření lékař uvede diagnózu, která se postupně dalšími vyšetřeními zpřesňuje nebo mění. V datech pro žádost o laboratorní vyšetření se uvede prvotní diagnóza. Provede se vyšetření, po kterém se zpřesní případně změní prvotní diagnóza. Data z vyšetření se uloží, ale obsahují původní diagnózu. Konkrétní případ pokusů o provedení průzkumových analýz vyšetření z krve selhal, protože diagnózy uvedené na žádankách k vyšetření byly jen prvotním odhadem. Plánované pokusy použití metod klasifikace dat nevedly k uspokojivým výsledkům.
- forma zápisu dat - v matematice se používá kanonická forma zápisu, která v sobě zahrnuje všechny ekvivalentní formy stejného konceptu. Například kanonická forma jedné poloviny je ½ a existuje algoritmus pro redukci nekonečného množství ekvivalentních výrazů, jako je 2/4, 3/6, 4/8, 5/10 …, na ½. Používání kanonické formy je zavedeno v mnoha disciplínách. V medicíně však základní ani elementární zápisy nemají kanonickou formu, takže například:
  colon adenocarcinoma , metastáze jater
  colonic adenocarcinoma , metastáze jater
  …
  znamená popis stejné diagnózy.
  Podobných zápisů k jednotlivým diagnózám existuje celá řada, které mohou dále obsahovat různé přívlastky podle uvážení specialisty, který diagnózu provádí.
  V současné době proto jedna z oblastí analýz medicínských dat je zavádění standardů pro medicínská data v souvislosti s databázemi a XML standardy [3]. Obdobná situace je i v ostatních biologických oborech.

3 Specifický statut medicínských dat

Medicínská data jsou data obsahující informace o lidských bytostech. Proto jsou na kladeny různé požadavky etické a právní, které by měly zabránit zneužití těchto citlivých dat. Na tuto problematiku pohlížet je možno pohlížet z pohledu vlastnictví dat, zákonných požadavků na tyto data, a dalších.

- vlastnictví dat - v současné době existují medicínské záznamy v různých databázích a registrech. Prakticky o každém člověku v naší republice je veden nějaký medicínský záznam. V souvislosti s využíváním těchto dat pro analýzy, zejména při využívání různých zdrojů budou vyvstávat otázky, které v současné době nejsou úplně zodpovězeny, ale mohou se stát zdrojem případných sporů. Jsou to otázky typu: Je individuální pacient vlastníkem dat o sobě? Má právo vědět a dát souhlas jak se s těmito daty nakládá? Je vlastníkem dat například soukromá nemocnice? Soukromý lékař? Nebo pojišťovna, která platí dané úkony? Jak postupovat, na koho se obrátit, pokud by při analýze bylo potřeba využít různé zdroje?
- zákonný status medicínských dat - medicínská data jsou v naší republice označována jako citlivá data (dle zákona 101/2000 O ochraně osobních dat)

a při jejich zpracování jsou vyžadovány specifické procedury. V tomto zákoně je zmíněn i proces anonymizace dat, tedy proces úpravy dat tak, aby z nich nebylo možné odvodit, k jakému subjektu se vztahují. Taková data lze zpracovávat a používat pro testy a analýzy. V současné době je úroveň zabezpečení dat u medicínských organizací subjektů různá. Neexistuje povinný standard ani nejsou prováděny prověrky zabezpečení dat. I v této situaci se v budoucnu situace změní, podobně jako se mění přístup u státních organizací v souvislosti s vydáním Národní strategii informační bezpečnosti zpracovanou Ministerstvem informatiky [5], a v budoucnu bude muset být této oblasti věnována větší pozornost.

- obava z negativního hodnocení - s případy odmítnutí poskytnutí dat je možné se setkat zejména u menších soukromých laboratoří a praxí. V pozadí takového odmítnutí je zpravidla obava z možného negativního hodnocení práce dané laboratoře, chybného hodnocení nebo zbytečného vyšetření znamenající větší náklady pro zdravotní pojišťovny a podobně.

4 Zpracování medicínských dat

Pro zpracování medicínských dat je třeba volit opatrně postupy. Často se objevují následující problémy:

- medicínská data jsou velmi objemná a jak již bylo uvedeno jsou velmi heterogenní, mohou pocházet z různých zdrojů a mohou mít různou strukturu. Důležitá fáze zpracování je proto předzpracování dat, zejména za účelem reprezentace dat v podobě, která je vhodná již ke zpracování zvoleným algoritmem. V této fázi je možné například možné i vybrat relevantní údaje a snížit tak počet atributů.
- zdravotnické databáze jsou stále aktualizovány. To vyžaduje použití kontrolu používaných metod a jejich neustálou aktualizaci například ve smyslu výběrových kriterií a podobně.
- medicínské informace obsažené v databázích jsou často nekompletní nebo nepřesné, např. v lékařské hodnocení z pravidelných prohlídek může být uvedeno „pacient se cítí slabý“ i tato data však mohou obsahovat důležitou informaci. Je třeba důkladně zvážit jejich úpravu či vyloučení.
- variabilita [4] je u biologických dat tedy i medicínských velkým problémem při analýze. Bez předběžných úprav pak např. aplikace metod shlukové analýzy nedají uspokojivé výsledky.
- chybějící data v databázích je problém, který se u medicínských dat vyskytuje poměrně často. Je to způsobeno tím, že např. některá vyšetření nejsou provedena, např. z důvodů stavu pacienta apod. V takovém případě je obtížné nahradit chybějící hodnotu hodnotou odvozenou z exitujících hodnot. V takovém případě je způsob náhrady pečlivě zvážit.
- přístup k modelování - odborná medicínská veřejnost přijímá spíše deduktivní přístup vycházející z věcných znalostí problematiky než kybernetický přístup, při kterém je modelovaný systém pojímán jako fiktivní skříňka transformující určité vstupy (příčiny) na výstupy (důsledky). Proto se

vyplatí postupovat podle osvědčených metodik, např. podle šesti fázové metodiky CRISP-DM

- V experimentálních vědách se používají některé základní předpoklady, které často nelze splnit u medicínských dat. V humánní medicíně je prvotním úkolem péče o pacienta a pouze druhotně lze využít data pro výzkumné účely. Klasické statistické testy předpokládají, že experiment lze opakovat a že pravidla jsou stanovena před započetím testu. Není možné pak měnit pravidla uprostřed experimentu, protože tím se daný experiment znehodnotí. Ve zdravotnické praxi však může dojít k tomu, že je nutno tyto pravidla a podmínky měnit. V zájmu péče o pacienta. Často také není možné opakovat testy nebo zvětšovat výběr. Těmto podmínkám odpovídají i data v databázích a je třeba pečlivě posoudit okolnosti, za kterých byla tyto data získána.

5 Závěr

Aplikace znalostních technologií na medicínská data má některá specifika, na která se tento příspěvek snažil poukázat. Tato specifika souvisí s tím, že data jsou heterogenní a variabilní, podléhají určitým etickým a právním hlediskům. Mohou dále pocházet z různých zdrojů v různé struktuře a kvalitě. Interpretace lékaře tvoří zpravidla základní komponentu těchto dat. Znalostní technologie aplikované v této oblasti jsou však nástrojem, bez kterého se v současné době lékařský výzkum neobejde.

Reference:

[1] Cios,K.J., *Medical Data Mining and Knowledge Discovery*. Springer-Verlag, 2000, Heidelberg

[2] Kurgan,L.A., Cios, K.J.,Tadeusiewicz,R,Ogiela,M. and Goodenday,L.S., *Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis*, Artificial Intelligence in Medicine, 23(2), 149-169, 2001

[3] U.S.National Library of Medicine. 2002. *Unified Medical Language System. Knowledge Sources*. 2002: 13th edition. http:www.nlm.nih.gov/research/umls

[4] Knížek,J., Stránský,P., Beránek,L., *Statistická analýza síly testu – staronový trend v experimentální medicíně*, Konference Lékařská biofyzika, 25.-27.5.2005, Valtice

[5] http://www.micr.cz

Annotation:

Medical data and their analysis (data mining)

This paper deals with the characteristics of medical data from the point of view of problems and algorithms KDD. Medical data are heterogeneous and variable and they subject to specific ethical and law aspects. The can come from various sources in different structure and quantity. The knowledge technologies applied in this area are the tool without it the medical research can not progress well at the present time.

Znalostní strategie pro znalostní management

Vladimír Bureš

Katedra informačních technologií - Univerzita Hradec Králové,
Rokitanského 62, 500 03, Hradec Králové

vladimir.bures@uhk.cz

Abstrakt. Znalostní management na organizační úrovni, tzn. znalostně orientované řízení organizace, je stále více předmětem zájmu mnoha odborníků. Na této úrovni hraje významnou roli strategie ve smyslu dlouhodobého záměru a směru dosažení vytýčených cílů. Z tohoto důvodu je nutné vedle již zavedených pojmů jako informační, komunikační nebo personální strategie věnovat pozornost také znalostní strategii. Příspěvek je zaměřen na popis pozice znalostní strategie v procesu zavádění znalostního managementu do organizace, jednotlivé aktivity, které je nutné v rámci tvorby znalostní strategie vykonat, použitelné nástroje, metody a techniky, dokumenty, které by měly v procesu tvorby vzniknout, nebo kritické faktory úspěšnosti tohoto procesu.

Klíčová slova: znalostní management, znalostní strategie, zavádění, cíle

1 Úvod

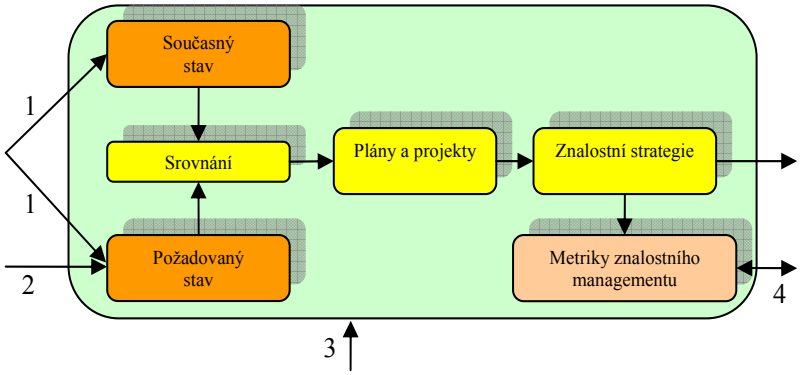
V současné době je se možné setkat s prací se znalostmi na dvou odlišných úrovních. Na základní úrovni se jedná o technologicko-informatickou disciplínu označovanou jako „management znalostí“ (MZ). Na následující úrovni jde o manažersko-ekonomickou disciplínu označovanou jako „znalostní management“ (ZM). Zatímco MZ je vnímán jako snaha o práci s konkrétními objekty, kterými jsou znalosti, ZM představuje spíše znalostně orientované řízení organizace, u kterého nemusí být podle některých autorů [3] znalosti ani hlavním předmětem zájmu. MZ se tedy pohybuje v oblastech jako objevování znalostí v databázích, zpracování přirozeného jazyka, strojové učení, značkovací jazyky nebo zkvalitňování sumarizátorů textů. Oproti tomu ZM čerpá z MZ a využívá jeho výstupů ke svým aktivitám a k dosahování vlastních cílů, které jsou ve většině případů na úrovni vyšších organizačních celků (podnik, tým, apod.). Konkrétní obsah výstupů z MZ je tak většině manažerů vzdálený a neznámý. Ačkoliv to není příliš ideální situace, nejedná se o nic zvláštního. Stejně je tomu také u výstupů jiných oborů (např. málokterý manažer zná přesné postupy generování pseudonáhodných čísel nebo vzorce, podle kterých se vypočítávají některé ukazatele, v běžně používaném produktu MS Excel). Paralelu k originálnímu pojmu “knowledge management” a jeho dvěma úrovním je možné vidět například u anglického pojmu „project management“, který je možné přeložit buď jako „řízení projektu“ nebo „projektové řízení“. Zatímco první pojem naznačuje řízení konkrétního projektu, řízení ve smyslu stanovení, plnění a kontroly jeho časových, finančních a věcných úkolů nebo využívání softwarových aplikací, druhý pojem, projektové řízení, značí projektově orientované řízení podniku, tzn. řízení jednotlivých zakázek v organizaci a chodu celé organizace na základě jednotlivých

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 268–271, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

projektů. Je tedy zřejmé, že se v obou případech jedná o odlišné pojmy. Avšak stejně jako je řízení projektu nutnou součástí a podmnožinou projektového řízení, je MZ jedním z několika základů ZM jako multidisciplinárního oboru lidské činnosti.

2 Znalostní strategie

Znalostní strategie (ZS) je klíčovou problematikou při zavádění a realizaci ZM. Podle metodiky KM-Beat-It [1] je možné k vytvoření ZS přistoupit až v případě provedení analýzy výchozího stavu v organizaci, resp. v případě, jsou-li známy současné podmínky v organizaci (existence a využití znalostních technologií, stav organizační kultury a její podpora ZM, realizované znalostní procesy, atd.). Účelem ZS je tedy zrealizování potřebných činností, které dovedou organizaci ke stanovení požadovaného (cílového) stavu a strategie, jakým tohoto stavu dosáhnout. Celý proces tvorby ZS je nastíněn na obrázku 1.

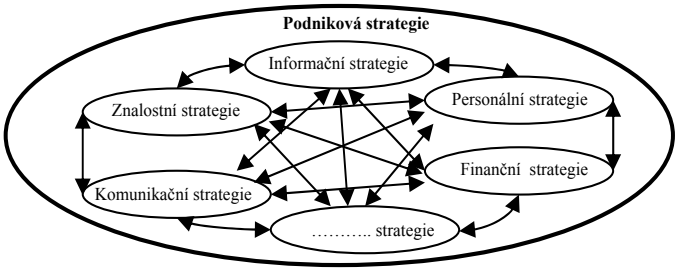


Obr.1. Proces tvorby znalostní strategie

Z obrázku je zřejmé, že proces tvorby ZS má ze systémového pohledu několik vstupů. Prvním vstupem jsou výsledky již zmíněné analýzy výchozího, resp. současného stavu. Tyto výsledky ovlivňují dva základní prvky procesu tvorby ZS. Jednak výsledky slouží svému primárnímu účelu, kterým je popis současného stavu, a jednak jsou výsledky jedním ze zdrojů informací o požadovaném stavu (mezi další je možné zařadit např. benchmarking nebo analýzu konkurence). Druhý vstup představuje informace o podnikové strategii a podnikových cílech, třetí vstup je vyjádřením zpětné vazby, díky které je možné se k procesu tvorbě ZS vrátit, a čtvrtý vstup zajišťuje vztah s vytvořeným systémem podnikových metrik. Množství vstupů dokazuje význam a důležitost tohoto procesu pro ZM. Na základě porovnání současného stavu se stavem požadovaným je možné určit zásadní nedostatky současné situace a navrhnout způsob, jakým je možné existující mezery uzavřít, tzn. vytvořit seznam všech aktivit ZM, které je možné realizovat. Při tomto rozhodování je nutné brát na vědomí a respektovat podnikovou strategii.

2.1 Vztah znalostní strategie a podnikové strategie

Podniková strategie dává smysl a cíl všem podnikovým aktivitám a zavádění ZM nevyjímaje. Vycházet ovšem z předpokladu, že existence podnikové strategie a její prosazování automaticky zajistí úspěšnost podniku, je liché. Je zřejmé, že v situaci chybně definované podnikové strategie bude mít její prosazování pro organizaci negativní, spíše než pozitivní dopad. Pokud je to možné, je vhodné při tvorbě ZS využít vizionáře, kteří se účastnili tvorby podnikové strategie. Těmto vizionářům totiž neschází fantazie a jsou schopni dívat se na realitu novými, neotřelými způsoby [4]. Stejně jako u podnikové strategie je možné při tvorbě ZS využít rozličné techniky, postupy nebo metody. Jako příklad může sloužit SWOT analýza (analýza silných a slabých stránek), KSP (Knowledge Strategy Process) nebo analýza CSF (kritické faktory úspěšnosti). Vztah podnikové strategie a ZS je patrný z obrázku 2.



Obr.2. Vztah podnikové a znalostní strategie

2.2 Charakteristiky procesu tvorby znalostní strategie

Obdobně jako podniková strategie je také právě vytvořená ZS strategickým dokumentem, který je nutné pravidelně upravovat a přizpůsobovat interním a externím změnám. Mezi základní charakteristiky tvorby ZS způsobujících a zároveň zajišťujících různorodost a dynamiku tohoto procesu patří:

- V rámci tohoto procesu se stanovují a vybírají jednotlivé *aktivity ZM*, na základě kterých je posléze možné určit a zrealizovat žádoucí projekty, jednorázové akce, rozhodnutí, atd. Společným cílem všech těchto aktivit je podpora a rozvoj ZM v organizaci. Jelikož má každá organizace jiné výchozí podmínky, bude se u každé z nich lišit jak počet aktivit, tak i jejich zaměření, finanční požadavky, potřeby personálního zajištění nebo doba trvání. Každá aktivita má pak již vlastní životní cyklus, použitelné metody, techniky a nástroje, atd.
- Součástí procesu tvorby ZS je také *stanovení metrik, indikátorů a ukazatelů ZM*, které budou využity ke sledování vývoje a především úspěchu procesu zavádění ZM. Jejich stanovení není až tak zásadní problém. Mnohem složitější může být stanovení vazeb tohoto systému metrik se systémem podnikových metrik.
- Významné jsou také *dokumenty*, které při tvorbě ZS vznikají. Patří mezi ně například vlastní ZS, explicitně definované cíle ZM nebo vytvořený systém metrik a ukazatelů ZM.

- Mezi *kritické faktory* tvorby ZS patří rozlišení časového charakteru cílů ZM, výběr vhodných aktivit, které budou realizovány v rámci naplňování ZS nebo propojení systému metrik a indikátorů ZM s podnikovým systémem metrik.
- V rámci jednotlivých činností je možné využít vedle již zmíněné SWOT analýzy nebo benchmarkingu také například brainstorming, brainwriting, systémy pro podporu rozhodování, vývojové diagramy, Ganttovy diagramy, SW pro řízení projektů nebo různé komunikační nástroje (blíže viz [2]).

3 Závěr

Z výše uvedeného je zřejmé, že ZS hraje při zavádění a realizaci ZM na organizační úrovni klíčovou roli. Mezi jednotlivé činnosti, které je při vytváření ZS nutné vykonat, patří definice požadovaného cílového stavu, srovnání požadovaného stavu a stavu současného a identifikace hlavních rozdílů, vytvoření seznamu aktivit ZM, výběr aktivit, zpracování plánů a projektů, vytvoření ZS a stanovení metrik ZM a jejich vazby k systému podnikových metrik. Veškeré aktivity je však nutné koordinovat s podnikovou strategií a průběžně aktualizovat společně se změnami podnikové strategie a existujícím stavem a podmínkami v organizaci.

Reference

1. Bureš V. Methodology of Knowledge Management Implementation. *3rd Intl. Conf. on Multimedia and Information & Communication Technologies.* Cáceres, Spain, Vol. 3, str. 963-968, 2005. ISBN 609-5997-X.
2. Bureš V. KM-Beat-It: Metodika zavádění znalostního managementu. *E+M Ekonomie a Management (7) 3.* Liberec, 2005. ISSN 1212-3609.
3. Sveiby K. http://www.sveiby.com.au. Knowledge Management - The Viking way.
4. Voříšek J. *Strategické řízení IS a systémová integrace.* Management Press, Praha, 2002. ISBN 80-85943-40-9.

Poděkování: Příspěvek vznikl za podpory grantu GAČR 406/04/2140 KNOMEDIAS

Annotation:

Knowledge strategy for knowledge management

The organisational level of knowledge management, i.e. knowledge-orientated management of the organization, has nowadays been attracting the attention of several experts. A strategy in sense of long-term intentions and the fulfilment of aims has a significant role at this level. That is why, it is necessary to create knowledge strategy and link it with communication strategy, personal strategy or information strategy. This paper is focused on the position of knowledge strategy in a process of knowledge management implementation, individual activities that should be performed, utilisable tools, techniques or methods, created documents, and CSF.

Užití grafických struktur při práci s texty[2]

Jiří Jelínek

Katedra managementu informací, FM, Vysoká škola ekonomická,
Jarošovská 1117/II, 377 01, Jindřichův Hradec

jelinek@fm.vse.cz

Abstrakt. Grafy jako forma reprezentace datových struktur jsou velmi oblíbené a to především pro svou názornost a značnou flexibilitu. Stále rostoucí množství informací a dat dostupných v elektronické formě pro ně otevírá další oblasti nasazení. Jedná se zejména o reprezentaci a zpracování textových dokumentů, tedy blíže nestrukturované textové informace. Záměrem příspěvku je ukázat praktické využití grafů pro detekci jedno i víceslovných termů a zachycení vazeb mezi nimi v množině dokumentů a to v rámci projektu podpory uživatele WWW, který je na Fakultě managementu VŠE řešen již delší dobu.

Klíčová slova: text mining, WWW, grafy, víceslovné termy, textové grafy

1 Úvod

Grafy jako forma reprezentace datových struktur jsou velmi oblíbené. Rozvoj WWW a stále rostoucí množství informací a dat dostupných v elektronické formě otevírá další možnosti použití těchto struktur. Ty se otevírají při reprezentaci textů, tedy blíže nestrukturovaných dat. I moderní jazyky pro sémantický popis obsahu dokumentu či WWW stránky jako jsou RDF či OWL vycházejí z reprezentace sémantického popisu ve formě trojic subjekt – vztah – objekt, což je v podstatě reprezentace ve formě orientovaného grafu. Otázkou je, jakou transformaci dokumentu či množiny dokumentů na grafy zvolit.

Literatura k tématu grafické reprezentace textu je značně obsáhlá. Jedním z prvních, kdo přišli s myšlenkou grafické reprezentace dokumentů byl J. Sowa, který přišel s tzv. konceptuálními grafy [3], které tvoří základ mnoha dnes užívaných metod. Jiný postup převodu text > graf je popsán v [1], kde jsou z textu generovány tzv. textové grafy, v jejichž uzlech nejsou pouze termy, ale je v nich zachyceno i přesné umístění navazujícího termu (především v případě sloves v uzlech). Problematice využití systému WordNet pro zpracování textu je věnován příspěvek [2] s důrazem kladeným na výběr správného synsetu podle větného kontextu.

2 Navržené postupy

Základní myšlenkou popsaného postupu je získat reprezentativní vzorek významných termů, a to jak jedno tak víceslovných, společně s jejich vzájemnými

[2] Příspěvek vznikl za podpory grantu č. 34/04 Interní grantové agentury VŠE Praha.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 272–275, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

vazbami danými jejich vzájemnou blízkostí v textu. Výsledná struktura pak může být užita při podpoře uživatele webu.

Především pro nízké nároky na přípravu dat, byl jako zdroj textových dat zvolen systém RSS (RDF Site Summary). Vybrány byly RSS kanály předních zpravodajských agentur (v angličtině) zabývající se světovými událostmi, ekonomikou a technikou. Databáze obsahuje v současnosti cca 450 tis. dokumentů (zpráv), což vytváří dobré předpoklady pro získání reprezentativních výstupů. Zvolený výběr vstupních dat nic nemění na možném užití stejného postupu pro práci s texty WWW stránek.

Převod vstupních dat do podoby grafu byl proveden v několika krocích. Každý vstupní dokument byl nejprve předzpracován, což kromě běžných drobných úprav označuje především aplikaci množiny tzv. „stop words“, tedy slov bez podstatného přínosu, které pouze zvětšují složitost výsledného grafu. Tato slova byla z textu vypuštěna. Následně byl text analyzován a převeden do podoby orientovaného grafu tak, že slova tvořila uzly grafu a hrany byly označeny hodnotou vyjadřující, kolikrát se v daném textu vyskytovala příslušná slova těsně vedle sebe a v daném pořadí. Orientace hran grafu byla dána tokem textu, tedy od předcházejícího slova (termu) k bezprostředně následujícímu.

Pro každý dokument t_i z množiny vstupních dokumentů T tak získáváme orientovaný graf $G_i = \{U_i, H_i\}$, definovaný množinou uzlů (jednoslovných a později i víceslovných termů) U_i a množinou hran H_i. Je patrné, že v množině U_i se každý term z textu vyskytuje pouze jednou.

Dále je vytvořen jediný souhrnný orientovaný graf $G = \{U, H\}$, který vznikne integrací jednotlivých grafů G_i. Tato integrace přidává do G pouze ty uzly a hrany, které se v něm ještě nevyskytují. U již existujících hran se pouze mění jejich popis, který je dán součtem ohodnocení z jednotlivých grafů G_i. Výsledná množina uzlů U je dána sjednocením všech množin U_i bez opakování uzlů. Výsledný graf G je tedy reprezentací množiny dokumentů T a hodnoty jeho hran h vyjadřují absolutní četnost výskytu dotčených termů (uzlů) v textu vedle sebe.

Hodnoty ohodnocení hran byly dále rozděleny do tří skupin pomocí horní a dolní mezní hodnoty. K vyhledávání víceslovných termů byla užita horní mezní hodnota h_h ohodnocení hrany mezi dvěma uzly $h_{kl} = (u_k, u_l)$. Ta určuje, zda se dva sousední termy vyskytují společně tak často, že je lze považovat za víceslovný term. Následuje iterativní postup, který v každé iteraci identifikuje termy o počtu slov o jedno vyšším, než v iteraci předchozí:

1. V grafu G najdi takové uzly u_k, u_l, pro které je hodnota hrany $h_{kl} > h_h$. Nalezené uzly ulož jako uspořádané dvojice do množiny $F = \{(u_k, u_l)\}$.

2. Generuj znova všechny grafy G_i tak, že pokud narazíš na dvojici slov (u_k, u_l) obsaženou v množině F, nahradíš tuto dvojici jediným uzlem s označením daným kombinací původních označení u_k a u_l.
3. Z dílčích grafů G_i vytvoř nový graf G.
4. V případě, že nebylo dosaženo mezního počtu iterací (maximální zadané délky víceslovného termu) pokračuj bodem 1.

Uvedený algoritmus upřesňuje významovou strukturu a slučování uzlů umožňuje objevit víceslovné termy, které jsou dále reprezentovány jediným uzlem v grafu G.

Místo použití mezních četností h_h a h_d je možné variantně pracovat i s jejich normovanými hodnotami m_h a m_d, které jsou porovnávány s normovanými hodnotami ohodnocení hran

$$m_{kl} = \frac{h_{kl}}{\sum_{h \subset H} h} \tag{1}$$

Graf G může být ještě tzv. prořezán, což znamená, že v něm budou ponechány pouze ty hrany, jejichž ohodnocení bude vyšší než stanovená dolní mezní hodnota h_d. Tímto postupem jsou zanedbávány okrajové a málo se vyskytující vazby.

Výsledkem aplikace uvedeného postupu je získání grafu termů o zadané maximální délce (počtu slov) včetně jejich vzájemných vazeb. Výsledný graf může sloužit jako vstup pro další zpracování směřující k vytvoření sémantické sítě a také je možné ho použít jako podpůrnou informaci pro uživatele při jeho vyhledávání v prostředí WWW. Uživateli mohou být (na základě vazeb z grafu G) nabídnuta slova či slovní spojení, které souvisejí se zadaným termem (vyskytují se často v jeho blízkosti). Uplatněním voleb uživatele následně vzniká zpřesněný dotaz, který pak může být předán některé běžné vyhledávací službě na WWW.

3 Výsledky a další postup

Výše uvedený postup byl ověřován na rozsáhlých vzorcích vstupních dat (100 – 300 tis. zpráv o délce do 250 znaků). Obvyklý počet uzlů ve výsledném grafu byl 100 – 200 tis., po vyčištění a redukci pomocí množiny zakázaných slov klesl tento počet na cca 50 tis. Nad tímto grafem probíhala detekce víceslovných termů, která se ukázala jako smysluplná do max. délky termu 3 - 4 slova. Výpočet výsledného grafu byl značně časově náročný. Celý proces však bude realizovaný off line, on line bude pouze užití vzniklé struktury pro vyhledávání souvislostí, známých termů a orientaci.

Jak vyplynulo z provedených experimentů, zásadní otázkou je definice množiny zakázaných slov, především problém zahrnutí určitých slovních druhů do této množiny, zejména sloves. Jejich vypuštění z dalšího zpracování výrazně zjednodušuje výstupní graf, může však dojít i k významné ztrátě informace. Pro detekci slovních druhů jiných než podstatná jména bylo uvažováno též o užití databáze WordNet [4].

Bohužel, slova vyskytující se v dokumentech jsou velmi často registrována do několika jazykových skupin současně (podst. jméno, sloveso). Užití WordNetu tímto jednoduchým způsobem se proto ukázalo jako značně problematické. Situaci by mohlo vyřešit použití větného kontextu (viz [2]) nebo hlubší jazyková analýza textu nástroji NLP (Natural Language Processing), která pracuje s grafickou strukturou dokumentu na podstatně detailnější úrovni.

4 Závěr

Tento příspěvek se zabýval užitím grafů pro reprezentaci a zpracování textových informací. Byl prezentován konkrétní postup transformace text > graf a jeho možné užití pro extrakci především víceslovných termů a podporu získávání sémantické informace z textu .

V současné době probíhají další práce na efektivnější implementaci uvedených postupů a na metodách umožňujících další zpřesnění získávaných výstupů např. zpracováním textu pomocí jednoduchého NLP.

Reference

1. Nuutila E., Törmä S.: Introduction to Text Graphs, Kolin Kolistelut -- Koli Calling 2003, *Proceedings of the Third Finnish/Baltic Sea Conference on Computer Science Education. Oct. 3-5, 2003 in Koli, Finland.* University of Helsinki Department of Computer Science, Report B-2003-3, Helsinki 2003. ISBN 952-10-1439-3.
2. Ramakrishnanan G., Bhattacharyya P.: *Text Representation with WordNet Synsets using Soft Sense Disambiguation*, In: Ingenierie des Systemes d'Information Journal (ISI-NIS Journal), Special Issue on the Application of Natural Language to Information Systems, 2004.
3. Sowa J. F.: Conceptual Graphs, http://www.jfsowa.com/cg/index.htm
4. WordNet, http://www.cogsci.princeton.edu/~wn/.

Annotation:

The Use of Graphs for Text Processing

Graphs as form of data structures representation are very popular namely above all for their plasticity and considerable flexibility. Growing amount of information and data accessible in electronic form open up new areas of usage for them. These are especially representation and processing of documents, that means unstructured text information. Intention of this contribution is to show practical usage of graphs for single and multi-word term detection and their relations on set of documents in terms of research project of web user support that is already solved on Faculty of Management for a longer time.

Vizualizace textových dat pomocí grafu[2]

Jiří Jelínek, David Kunčar, Jiří Přibil

Katedra managementu informací, FM, Vysoká škola ekonomická,
Jarošovská 1117/II, 377 01, Jindřichův Hradec

jelinek@fm.vse.cz, {david.kuncar,jiri.pribil}@gmail.com

Abstrakt. Příspěvek představuje nástroje pro vizualizaci grafických struktur reprezentujících vztahy mezi termy v analyzovaných textech. Tyto nástroje byly vytvořeny v rámci projektu podpory uživatele WWW, který je na Fakultě managementu VŠE v současné době řešen. Hlavním cílem projektu je poskytnout uživateli WWW takové doplňkové informace, které mu mohou pomoci při vyhledávání relevantních informací ve WWW prostoru.

Klíčová slova: text mining, WWW, textové grafy, vizualizace

1 Úvod

Záměrem příspěvku je ukázat nástroje pro vizualizaci grafů, které byly vytvořeny v rámci projektu podpory uživatele WWW, který je na Fakultě managementu VŠE řešen již delší dobu a jehož hlavním cílem je poskytnout uživateli WWW doplňkové informace, které mu mohou pomoci při vyhledávání ve WWW prostoru [3]. Grafické výstupy budou užity pro prezentaci souvisejících pojmů a lepší orientaci uživatele.

V rámci uvedeného projektu byly řešeny postupy transformace textů do formy orientovaných grafů. Grafické struktury jsou i pro nezkušeného uživatele velmi názorné a pochopitelné a s jejich pomocí lze uložené informace a znalosti snadno upravovat a spravovat stejně jako je přímo využít např. ve formě myšlenkových map. Pro podporu uživatele bylo však nutné řešit též otázku vizualizace grafických struktur pro následnou prezentaci uživateli.

2 Stav problematiky, navrhované postupy a dosažené výsledky

Jedním z prvních, kdo přišli s myšlenkou grafické reprezentace dokumentů byl J. Sowa, který přišel s tzv. konceptuálními grafy [7]. Tato reprezentace slouží jako výchozí model pro mnoho dalších, které z ní vycházejí. Rovněž naše reprezentace z ní vychází.

Konstrukci a vizualizaci konceptuálních grafů se věnuje rovněž [1], případně [5]. 3D přístup k vizualizaci grafů je nastíněn v [6].

V rámci projektu podpory uživatele se grafickými reprezentacemi na našem pracovišti zabýváme již několik let [4]. V tomto případě byl výstupem metod řešících detekci termů orientovaný graf, jehož uzly představují významné termy a hrany

[2] Příspěvek vznikl za podpory grantu č. 34/04 Interní grantové agentury VŠE Praha.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 276–279, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

označují bezprostřední sousedství termů v textu. Hrany jsou kromě toho ohodnoceny číslem vyjadřujícím počet případů, ve kterých spolu dané termy bezprostředně sousedí. Tvar uvedeného grafu je dán kromě popisovaného tématu také osobností autora a jeho vyjadřovacími návyky.

V případě vizualizace grafických struktur je nejprve nutné zvolit jeden z možných přístupů:

1. Graf je zobrazován celý, přičemž rozmístění uzlů je provedeno buď automaticky nebo s pomocí uživatele.
2. V grafu je zvolen centrální uzel, který je následně zobrazen společně se svými předchůdci a následníky a to do předem stanovené úrovně (horizontu).

První přístup je velmi náročný na výpočetní výkon, zejména u grafů, ve kterých jde počet uzlů do stovek a tisíců (v našem případě cca 16 – 50 tis. uzlů). Přesto je možné objevit nástroje, které tuto činnost uspokojivě provedou. Jejich příkladem může být např. systém CMap [2], který byl pro vizualizaci testován i v tomto případě a to především z důvodu kompatibility s používanou databází a jednoduchosti exportu a importu dat. I tento systém však vykazoval pro rozsáhlejší grafy značnou pomalost. Kromě něho byly testovány ještě produkty Tulip Software [8] a uDraw [9].

Při experimentování s různými zobrazeními byl zvolen druhý přístup. Pro práci s grafickými strukturami bylo vytvořeno WWW rozhraní umožňující jak prosté prohlížení, tak editaci grafických struktur uložených v připojené databázi. Jeho omezením v současnosti je, že v grafické formě zobrazuje pouze vybraný uzel a jeho bezprostřední předchůdce a následníky.

Proto byla vytvořena další programová komponenta pro zobrazování grafových struktur, založena na algoritmech použitých v open source vizualizačním nástroji Graphviz, který je vytvářen firmou AT&T Bell Labs. Grafové struktury jsou v ní generovány na základě uživatelsky zadaných parametrů *hloubka* (typu celé číslo) a *root* (typu řetězec). Na jejich základě je pak ze vstupních dat uložených v databázi ve formátu *počáteční slovo vazby – počet dvojic slov – koncové slovo vazby* na základě SQL dotazu (resp. více SQL dotazů) vybrána množina záznamů obsahující pouze taková slova, která jsou spojena vazbou (i přes více uzlů) se slovem předaným v parametru *root*. Tím je tedy vytvořen podgraf původního grafu reprezentovaného všemi vazbami uloženými v databázi.

Vybrané vazby jsou následně přeloženy do formátu DOT [10], obecně přijímaného formátu pro popis planárních grafů, který je dále zpracováván knihovnami vizualizačního software Graphviz. Výstupem tohoto zpracování je pak seznam jednotlivých uzlů a hran „orientovaného“ grafu určujících jejich umístění ve 2D prostoru. Pro toto rozmístění je přitom možno použít hned několik algoritmů:

- *dot*, primárně určený pro tvorbu hierarchických nebo vrstvených struktur orientovaných grafů;
- *neato* používající algoritmus Kamada-Kawai;
- *twopi* a *circo* vytvářející radiální grafové struktury.

Jako nejvýhodnější se v praxi ukázal algoritmus *neato*, byť je původně směřován do oblasti neorientovaných grafů; jeho výstupy maximálně zohledňují nepřekrývání jednotlivých hran obsažených v grafu a činí tak graf maximálně přehledným - výsledný graf je totiž vždy planární (což je umožněno zejména tím, že díky zvolenému způsobu grafické reprezentace do každého uzlu vstupuje pouze jedna jediná hrana).

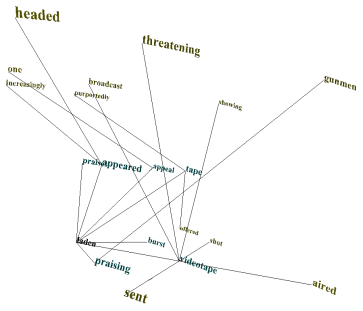
Cílovým výstupním formátem naší komponenty je pak formát SVG, stále více se prosazující formát pro tvorbu vektorové grafiky nejenom pro prostředí internetových stránek. Použití této technologie umožňuje zachovávat maximální možnou kvalitu výstupních grafových struktur a jejich snadné použití v dalších grafických aplikacích umožňujících následné zpracování.

Samotná grafová struktura ve formátu SVG je pak doplněna o odkazy, pomocí nichž je možno procházet zobrazený graf dále; komponentě je předána nová hodnota parametru root a graf je příslušně překreslen.

V rámci výzkumů druhého přístupu k vizualizaci grafů byla řešena rovněž otázka možného 3D zobrazení grafů a to především vzhledem k rostoucímu výkonu výpočetních systémů a pokroku v oblasti 3D zobrazování obecně. V rámci výzkumného projektu byl vytvořen vizualizační nástroj exportující popis grafu do 3D zobrazení v jazyce VRML.

Hlavním problémem bylo zvolit vhodný algoritmus umisťující jednotlivé uzly grafu do 3D prostoru. Po provedených experimentech se jako použitelný postup ukázalo umisťování uzlů na krychle různých velikostí (podle „vzdálenosti“ od hlavního uzlu). Skript obdrží přes webový formulář hledané slovo (root), ke kterému následně vyhledá v databázi slova, která mají k slovu hledanému vazbu. Poté započne samotný výpočet pro následné 3D zobrazení. Skript prochází získaná data a každému jednotlivému slovu přiřadí koordináty podle výše zmíněného algoritmu. Tento postup se rekurzivně opakuje pro všechna získaná slova a slova s nimi svázaná až do definované úrovně (hloubky) a to dopředně i zpětně. Struktura dat je graficky znázorněna pomocí hran vedoucích mezi souvisejícími slovy. Výstupem skriptu je samostatný VRML soubor, ve kterém jsou úrovně rekurze barevně rozlišeny. Každé zobrazené slovo je zároveň i hypertextovým odkazem, který vede k provedení skriptu s tímto slovem jako slovem hledaným. Praktické výstupy ukázaly, že smysluplné je použití této vizualizace do druhé úrovně zobrazení.

Obr. 1. Výstup 2D zobrazení



Obr. 2. Výstup 3D zobrazení

3 Závěr

Tento příspěvek se zabýval možnými postupy vizualizace grafů pro prezentaci textových informací. Byly představeny různé přístupy k tomuto problému a jejich

praktická řešení. Výstupy práce budou použity v budovaném systému podpory uživatele WWW, kde napomohou lepší orientaci uživatele při procesu vyhledávání.

Další práce v této oblasti se soustředí na zlepšování popisovaných metod tak, aby dále vzrostla názornost prezentace informací a grafické zobrazení bylo pro uživatele WWW co nejsnáze pochopitelné.

Reference

1. Coburn, A.: Text Modeling and Visualization with Network Graphs, *ACH/ALLC, June 15-18, University of Victoria, British Columbia, Canada, 2005*
2. IHMC CmapTools - Concept Map: CmapTools - Home Page, http://cmap.ihmc.us/
3. Jelínek, J.: Podpora orientace a vyhledávání v prostředí WWW. *The 3rd annual conference Knowledge 04, Brno, Czech Republic, 2004,* In: Poster proceedings of the 3rd annual conference Knowledge 04, pp. 13-16, VŠB-TUO Ostrava Czech Rep., 2004
4. Jelínek, J.: Využití vazeb mezi termy pro podporu uživatele WWW. *Mezinárodní konference Znalosti 2005, 9. – 11. 2. 2005, Stará Lesná, Slovensko,* In: Sborník příspěvků 4. ročníku konference Znalosti 2005, pp. 218-225, VŠB-TUO FEI Ostrava, ISBN: 80-248-0755-6
5. Mutton, P. - Golbeck J.: Visualization of Semantic Metadata and Ontologies, *Seventh International Conference on Information Visualization (IV03),* In: Proceeedings of the Seventh International Conference on Information Visualization (IV03), pages 300-305. IEEE, July 2003, ISBN 0-76951988-1
6. Slay, H. – Philips, M. – Thomas, B. - Vernik R.: Keg Master: a Graph-Aware Visual Editor for 3D Graphs, *AUIC 2003, Adelaide, Australia, 2003*, In: Proceedings of the Fourth Australian user interface conference on User interfaces 2003 - Volume 18, pp. 99-103, Australian Computer Society, Inc. Darlinghurst, Australia, Australia, 2003, ISBN ~ ISSN:1445-1336 , 0-909925-96-8
7. Sowa, J. F.: Conceptual Graphs, http://www.jfsowa.com/cg/index.htm
8. Tulip Software home page, http://www.tulip-software.org/
9. uDraw(Graph), http://www.informatik.uni-bremen.de/uDrawGraph/en/index.html
10. The DOT Language, http://www.graphviz.org/pub/scm/graphviz2/doc/info/lang.html

Annotation:

Visualization of the Text Data with Graphs

The aim of this contribution is a brief introduction to several tools for graph visualization. These tools were developed within the scope of a WWW user support project, solved at the Faculty of Management. The main purpose is to afford extended information, which could be helpful to a WWW user searching relevant information in WWW space.

Possible Interpretations of Binary Factor Analysis

Aleš Keprt

Department of Computer Science, Faculty of Sciences
Palacký University, Olomouc, Czech Republic
Aley@Keprt.cz, Keprt@inf.upol.cz

Abstract. Binary factor analysis (BFA) is a nonhierarchical binary data analysis, based on reduction of binary space dimension. It allows us to find hidden relationships in binary data, which can be used for efficient data compression, data mining, or intelligent data comparison for information retrieval. This short paper reveals some nontrivial and indirect interpretations of BFA, whose appeared as a side results of the research in the field of BFA algorithms.

Keywords: binary factor analysis, dimensionality reduction, compression, clustering

1 Introduction

Binary factor analysis (BFA) is a nonhierarchical binary data analysis, based on reduction of binary space dimension. It allows us to find hidden relationships in binary data, which can be used for efficient data compression, data mining, or intelligent data comparison for information retrieval.

BFA is usually defined and described by a simple mathematical formula

$$\mathsf{X} \approx \mathsf{F} \odot \mathsf{A} \tag{1}$$

i.e. as a decomposition of binary data matrix X to the binary product of two smaller binary matrices F and A. (Although this equation is correct, it is just a math formula, and doesn't show any real–world interpretation of BFA.)

The quality of the solution is measured by discrepancy function d, which counts the number of different bits between data matrix X and binary matrix product $\hat{\mathsf{X}} = \mathsf{F} \odot \mathsf{A}$ (d is nonnegative whole number, lower is better).

This short paper shows a few possible interpretations of BFA, ranging from pure mathematics to geometry and informatics.

2 Interpretation of BFA

We defined BFA problem in the form of matrix equation 1. This mathematical equation is the base of the research of BFA solving algorithms, but doesn't lead to a simple interpretation of what BFA is and what is it good for. We will try to uncover some possible interpretations which appeared during the research primarily aimed at the BFA solving algorithms.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 280–283, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2.1 Matrix decomposition

Matrix decomposition is used as a primary definition of BFA (see equation 1). It's not a real–world interpretation, but it is listed here for completeness. Note that since BFA is based on binary algebra, we can't use any known matrix decomposition method of linear algebra. It means that equation 1 gives the definition of BFA problem, but doesn't show any algorithm for solving it.

2.2 Reduction of dimensionality

The basic interpretation of equation 1 is the **reduction of dimensionality**. This refers to a vector model, where row vectors x_i are understood as points in a binary vector space of dimension p. We try to find a suitable projection which transforms all x_i vectors to a subspace of dimension m. Since m is considerably smaller than p, that projection is a reduction of dimensionality.

Matrix A defines the relation over X (data in original space) and F (data in reduced space). We can use simple binary matrix product $\mathsf{F} \odot \mathsf{A}$ to project reduced data back to original space. The BFA problem is how to get the best pair of matrices F and A in respect of lowering the discrepancy d.

2.3 Comparing and searching documents

Reduction of dimensionality is often used to compare and search documents. The BFA itself cannot be used for searching, it can only serve as a helper which can compare two documents.

Here, the word **document** must be understood in general. We usually work with text documents or graphical images, but it can also be anything else, which need to be shared among more objects of the same kind. Each document i is encoded to a binary vector x_i. BFA can reduce the vector space in a way that transform similar bit–coded documents x_{i_1} and x_{i_2} to considerably shorter vectors f_{i_1} and f_{i_2}. The level of similarity may be defined as number of 1s which lie on the same position in these vectors. Or we can use some more sophisticated method, but this all lies outside of the scope of this work. BFA only tell us if the two particular documents are similar, or not, or "how much" are they similar to each other.

2.4 Lossy data compression

Reduction of dimensionality leads also to the compression. If vector space is reduced to a considerably lower dimension, we obviously gain also significant compression. It is a lossy compression, as in the general case we can't reconstruct the original data from the compressed one. It became a lossless compression only for problems where we manage to get $\mathsf{X} = \hat{\mathsf{X}}$.

Matrix X stores the original data, matrix F stores compressed data, and matrix A is a decompression matrix. We get the original data from the compressed

one by performing a matrix product $\mathsf{F} \odot \mathsf{A}$. Compressing is much more problematic, as there isn't a simple way of turning original data X into compressed one F, even with the knowledge of A. The compression process actually leads to a nontrivial binary matrix pseudo–division X/A (see [1] for details).

2.5 Geometric interpretation

From geometric point of view, we are back to reduction of dimensionality. This applies both to classical and binary factor analysis: It's a projection from a p–dimensional space to a "smaller" (i.e. less–dimensional) m–dimensional space.

In classical (linear) factor analysis (see [2]), the key to the solution is in projection of variables with a high correlation (i.e. similarly behaving ones) to the same axis. This means that similarly behaving variables are "grouped together". In binary case, there is no correlation in the classic sense, but we still try group similarly behaving variables together. The main difference in binary case is the space itself – the space is reduced to vertices of a hypercube. Data rows x_i are vertices of p–dimensional hypercube, and we try to find a suitable projection of them to a "smaller" m–dimensional hypercube, which is a binary subspace of the original data space. The solution matrix A is a basis (a set of base vectors) defining the binary subspace within original data space. Then each f_i is an image of x_i in reduced space, and product $\hat{x}_i = f_i\mathsf{A}$ puts f_i back to the original data space. The BFA problem is to find such a projection, where $\hat{x}_i$ maps as close to x_i as possible for all i.

Hypercube–interpretation also shows geometric interpretation of neighborhood and distance. Each vertex x_i of p–dimensional hypercube has p neighboring vertices, each of them reflect a single bit inverse of x_i. Distance among two vertices x_{i_1} and x_{i_2} is equal to the shortest edge–path from x_{i_1} to x_{i_2} (or vice versa, as it's commutative). This is also equivalent to the number of different bits in the two vectors, i.e. $\mathrm{nzb}(x_{i_1} \text{ xor } x_{i_2})$. This is also a native interpretation of discrepancy defined in section 1, as vector discrepancy can be interpreted as "distance" from one vertex to another one on a hypercube. Matrix discrepancy is then a sum of vector discrepancies for all row vectors of two matrices.

2.6 Differences to other kinds of clustering

Data clustering (see [3]) is a common technique for statistical data analysis, which is used in many fields, like machine learning, data mining, pattern recognition, image analysis or bioinformatics. Clustering is the process of partitioning a data set into subsets (called "clusters"), so that the data in each subset (ideally) share some common trait – often proximity according to some defined distance measure.

BFA is similar to clustering, but it should be understood as "something different". When doing clustering, we group variables into clusters, and usually expect each single variable to be assigned to exactly one cluster. In BFA, factor is a synonym for cluster, but it has different qualities. A variable can be

assigned to arbitrary number of binary factors, including no one at all. In clustering a variable can't be assigned to more than one cluster without lowering the weight (i.e. relevance level) below 1, which leads to a fuzzy–clustering. In opposite, the "Boolean nonlinearity" of BFA brings the 100% free relationships between variables and factors. It means that when a variable is assigned to a particular factor, it has absolutely no consequences to its relations to the other factors. This leads to the fact that BFA restricted to non–overlapping binary factors may be possibly implemented using some suitable clustering method, but standard BFA with its Boolean arithmetic and overlapping factors can't be implemented this way. The reverse relation applies as well, i.e. we can't use BFA in the place of clustering, as BFA with its Boolean arithmetic can't substitute standard clustering methods.

The so–called "Boolean nonlinearity" of BFA is caused by the $\oplus$ operator. Its $1 \oplus 1 = 1$ causes that adding a variable to other factors yields other "plus 1" with no additional effect on bits where we already have 1. And contrariwise, when there are 1s in some bits, there is no power to put them out of there, as there are no additive inverse elements. The nonexistence of additive inverse elements is also one of the reasons why we sometimes call for a higher penalty on negative discrepancy – too many 1s in solution can't be put away later on, and these unwanted 1s *are* the negative discrepancy.

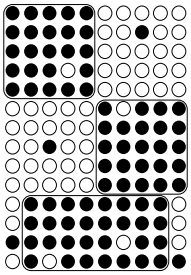


Fig. 1. An example of overlapping binary factors (circled).

You can see an example on figure 1 – we are interested in columns: There are three overlapping binary factors [1–5], [6–10] and [2–9]. In a classical clustering, you would get just two clusters [1–5] and [6–10]. The bottom four rows are weaker and would be outspread in these two strong clusters.

References

1. Keprt, A., Snášel, V. Pseudo-dělení binárních matic a jeho aplikace. In proceedings of *Znalosti 2005.* Vysoké Tatry, Slovakia. Published by VŠB Technická Univerzita, Ostrava, **2005**, pp. 41–50, ISBN 80-248-0755-6.
2. Überla, K. *Faktorenanalyse* (2^{nd} edition). Springer–Verlag, Berlin–Heidelberg–New York, 1971. ISBN 3-540-04368-3, 0-387-04368-3.
(slovenský překlad: Alfa, Bratislava, 1974)
3. *Wikipedia – The Free Encyclopedia.* http://en.wikipedia.org/

Příprava dat pro clickstream analýzu

Tomáš Kliegr

Katedra informačního a znalostního inženýrství, Fakulta informatiky a statistiky,
VŠE Praha, nám. W. Churchilla 4, 130 67, Praha 3

xklit05@vse.cz

Abstrakt. Ex post získávání clickstreamu z log souborů je pracné a vznikají při něm chyby. Alternativou je identifikace návštěvníka pomocí session cookie již během návštěvy a ukládání clickstreamu, tak jak vzniká, rovnou do databáze. Vzniklý proměnlivě dlouhý clickstream je následně transformován do tvaru o konstantním počtu atributů vhodném pro algoritmy hledající asociační pravidla.

Klíčová slova: clickstream, asociační pravidla, identifikace session

1 Úvod

Sledování tras (posloupností zobrazených stránek) návštěvníků na webu a jejich vyhodnocování – tzv. clickstream analýza – může poskytnout cenné informace pro e-commerce aplikace. Pokud je například věrohodně doloženo, že návštěvníci přicházející z jistého webu si častěji něco zakoupí než návštěvníci odjinud, získá tím marketingové oddělení firmy cennou informaci pro lepší zaměření reklamy.

Log soubory tvořené webserverem neobsahují skutečný clickstream, ale pouze izolované záznamy představující jednotlivé stránky. V nich je nejdříve potřeba na základě aplikace řady heuristických pravidel identifikovat jednotlivé návštěvy (session). Použití těchto pravidel je pracné a nespolehlivé[2]. Výsledek tohoto procesu se převádí do relační databáze, jež je zdrojem dat pro data miningové procedury.

V tomto článku ukáži přístup k pořizování clickstreamu založený na bázi cookies, který zaznamenává návštěvy – tak jak vznikají – přímo do relační databáze.

Možná nejobtížnější krok během fáze přípravy dat je transformace proměnlivě dlouhého clickstreamu do jedné tabulky o konstantním počtu atributů[3]. Ta je vhodným vstupem pro nejtypičtější způsob zpracování clickstreamu [1] – pro procedury založené na hledání asociačních pravidel.

Na základě případové studie cestovní kanceláře bude demonstrován přístup k této úloze, který je založený na zobecnění clickstreamu na *profil návštěvníka*. Vlastní modelování dat bylo provedeno pomocí procedury 4ft-Miner systému LISp-Miner [3].

Struktura článku včetně názvů kapitol a jejich obsahu se rámcově drží metodiky pro dolování znalostí z databází CRISP-DM[4].

[2] Časté problémy: více paralelních session z jedné IP, extrémně krátké návštěvy, extrémní počet návštěv z jedné adresy (zřejmě indexovací roboti), extrémně dlouhé návštěvy (roboti) [2].
[3] Většina DM nástrojů akceptuje jako vstup tabulku o kontaktním počtu sloupců.
[4] CRISP-DM 1.0 Step-by-step data mining guide (dostupn□ na wwww.crisp-dm.org).

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 284–287, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2 Porozumění problematice

Cestovní kancelář od clickstream analýzy očekává, že dostane odpověď na kardinální otázku: kde se vyplatí inzerovat? Zajímá ji tedy a) odkud (z kterých webů) přijdou návštěvníci, kteří si něco koupí, a b) jaká je struktura zájmů návštěvníků ve vztahu k odkazujícímu webu. Cílem modelování se tedy stává klasifikace návštěvníků.

3 Porozumění datům

Aby mohl být vytvořen *profil návštěvníka*, musí být k dispozici sekvence stránek, které navštívil. Byla proto vytvořena ASP.NET webová aplikace s pracovním názvem KM propojená s měřeným webem pomocí tzv. měřícího obrázku umístěného na každé stránce měřeného webu. Tato aplikace[5] identifikuje nového návštěvníka již během zobrazení první stránky z měřeného webu, zjistí odkud na web přišel, zaznamená další zajímavé informace z HTTP hlavičky, a zejména mu přidělí jedinečnou session cookie. Při dalším požadavku od téhož uživatele v rámci jedné session je tato cookie vrácena a uživatel identifikován. Takto vytvářený clickstream je v reálném čase ukládán do MS SQL databáze. Tento přístup má následující přednosti:

Pracuje se všemi prohlížeči s povolenými cookies. Někteří uživatelé mají svůj prohlížeč nastaven kvůli ochraně soukromí tak, že přijímá pouze vlastní cookie. Komerční měřící systémy, které neumisťují měřící obrázek či kód na měřený web, ale ponechávají ho na svém serveru, se pokouší identifikovat návštěvníka pomocí cizí cookie. Ta však není vždy přijmuta. Zato aplikace KM je součástí měřeného webu, jí vydávané cookie jsou tak vlastní a tím pádem důvěryhodné a častěji přijímané[6].

Automatická filtrace robotů. Aplikace KM je aktivována pouze tehdy, pokud si prohlížeč uživatele vyžádá její měřící obrázek. Většina crawlerů (s výjimkou tzv. imagebotů) obrázky nestahuje. Jejich návštěvy tak nejsou zaznamenány (srovnej [2]).

Malé nároky na diskovou kapacitu. Log soubory obsahují duplicitní řetězce. Aplikace KM oproti tomu ukládá clickstream do relační databáze. S využitím číselníků pro eliminaci duplicit a toho, že požadavky na obrázky nejsou ukládány, činí úspora při použití databáze až 95% kapacity ve srovnání s log soubory[7].

4 Příprava dat

Sesbírané clickstreamy jednotlivých návštěvníků jsou různě dlouhé. Cílem této fáze bylo vytvořit univerzální *profil návštěvníka* jako zobecnění clickstreamu. P*rofil návštěvníka* by měl být tvořen vhodně zvolenou množinou odvozených atributů tak, aby byla pro libovolně dlouhý clickstream zajištěna dobrá vypovídací schopnost.

---

[5] Kód aplikace byl odvozen z článku: www.15seconds.com/issue/021119.htm

[6] Společnost WebTrends odhaduje, že v oblasti maloochodního prodeje je odmítáno pouze 1-4% vlastních cookie (cookie prvních stran) oproti 16,9% cízích cookie (cookie třetích stran).

[7] Hrubý odhad založený na údajích spol. Log Rover. http://www.logrover.com/

Nejdříve byla za asistence doménových expertů vytvořena taxonomie stránek s ohledem na jejich obsah a služby jež poskytují[8]. Jednotlivé stránky pak byly pomocí databázové uložené procedury *Segment* zařazeny do obecných (Cat) a podrobnějších (rozšiřujících) kategorií (ECat) podle kritéria *služba* a podle *obsahu* do témat (Topic). Dále je stránkám přiřazen čas na stránce (t), pořadí v clickstreamu (P) a skóre (S). Tím je provedena kategorizace dat na několika úrovních granularity.

Tabulka 1. Clickstream jednoho návštěvníka zpracovaný procedurou *Segment*

P	URL	Kategorie	Roz. kat.	Téma	Čas	Skóre
1.	Example.cz/z49-Richterovyboudy.htm	Zájezd	Není	Krko–noše	60	60
2.	Example.cz/turistika-alpy/	Hledání	Katalog	Alpy	120	203
3	Example.cz/z98-lyzovaniRakousko.htm	Zájezd	Není	Alpy	240	503
4.	Example.cz/kdojsme.asp	Obecné	Kdo jsme	Není	30	71

Skóre. Za atributy P a S se skrývá tato úvaha: cílem návštěvníka je nalézt zájezd, který nejvíce vyhovuje jeho preferencím. Své hledání s každou zobrazenou stránkou zpřesňuje. Více o zájezdu, který hledá, tak napoví stránky hlouběji v clickstreamu. Skóre vyjadřuje absolutní váhu dané stránky. Zohledněn je především čas strávený na stránce[9], přičemž větší váhu mají stránky s vyšším pořadovým číslem P.

$$S = (\mathrm{Ln}(P) + 1) * t$$

Uvedený vzorec je experimentální, jeho precizace je záležitostí dalšího výzkumu.

Na výstup procedury *Segment (*viz Tabulka 1) navazuje další uložená procedura *Merge*, která vytváří profil návštěvníka. Jejím výstupem je už požadovaná tabulka o konstantním počtu atributů. Do Tabulky 2 bylo reprezentativně vybráno 15 z celkových 29 atributů vytvořených pro charakterizaci jedné návštěvy:

Základní atributy (6 atributů): Celkový čas na webu, Počet zobrazených stránek, Den v týdnu, Hodina, Odkazující doména – Referer (složená z URL, Typ refereru).

Atributy zobecňující stopu (11 at.): Šířka zájmu, Nejoblíbenější téma (Topic, S), Hledání celkem (S) a analyticky–Fulltext (S), Rozšířené (S), Katalog (S), Informační stránky celkem (S) a analyticky–Slevy (S), Pojištění (S), O CK (S).

Významné body stopy (12 at.): Vstupní stránka, Výstupní stránka, Stránka konverze. Každý bod je charakterizován atributy Název stránky, Cat, ECat a S.

Většinu z atributů lze dobře pochopit intuitivně. Blíže tedy jen k těm méně zřejmým. *Šířka zájmu* odpovídá počtu různých témat, do kterých zobrazené stránky patří. *Nejoblíbenější téma*[10] je téma s maximálním součtem skóre. Tento nejdůležitější atribut je odhadem tématické oblasti, ze které si chce návštěvník zakoupit zájezd.

[8] Tento přístup je podobný [4] v tom, že jsou vytvořeny dvě paralelní taxonomie. Jedna se zabývá obsahem – Téma (Topic) je např. „Alpy“, druhá poskytovanou službou – např. Cat je „Hledání“ a detailnější ECat „Hledání katalogové“ (viz dále).

[9] U poslední stránky v sekvenci tento údaj není měřen. Předpokládá se 30 sekund.

[10] V této kapitole předpokládáme, že návštěvník si chce zakoupit právě jeden zájezd. U jinak zaměřeného webu může mít ale návštěvník více cílů (kupovaných produktů).

Tabulka 2. Výstup procedury *Merge* – výběr z vygenerovaných atributů.

Referer	Typ refereru	# Stránek	Nej téma	Nej téma skóre
zoohoo.cz	*Fulltext*	*4*	*Alpy*	*706*
Výstupní stránka Náz.	Výstupní stránka Skóre	Výstupní stránka Cat	Výstupní stránka ECat	Šířka zájmu
kdojsme.asp	*71*	*Info*	*Kdo jsme*	*2*
Info celkem	Slevy	O CK	Hledání celkem	Hledání katalog
71	*0*	*71*	*263*	*263*

5 Závěr (modelování, vyhodnocení výsledků a využití)

První výsledky získané ze zpracovaných dat pomocí pouhé frekvenční analýzy jsou povzbudivé a přenositelné do praxe. Například podmíněná pravděpodobnost *Konverze|Typ refereru* je výrazně vyšší pro katalogové vyhledávače (12,8 %) vůči průměru (7,7 %), což lze marketingově využít ke změně inzertních kanálů.

Představený postup zpracování clickstreamu je vhodným výchozím bodem pro další výzkum zaměřený na zobecnění tvorby profilu návštěvníka pomocí technologií Sémantického webu a na analýzu již připravených dat pomocí systému LISp-Miner.

Prohlášení. Podporováno granty GAČR (201/05/0325) a IGA 25/05 VŠE

Reference

1. Facca, F., Lanzi P.: Mining interesting knowledge from weblogs: a survey. In: Data & Knowledge Engineering 53. Elsevier, 2005
2. Hofgesang P, Kowalczyk W, From Anomaly Detection to Visitor Profiling, sborník konference Discovery Challenge 2005. Porto 2005
3. Rauch, J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Foundations of Data Mining and Knowledge Discovery. Berlin 2005
4. Vanzin, M., Becker, K.: Exploiting Knowledge Representation for Pattern. In: Proc. of the Workshop on Knowledge Discovery and Ontologies. Pisa 2004

Annotation:

Preparing data for clickstream analysis

Acquiring clickstream data from log files is laborious and prone to errors. An alternative is to identify the visitor with a session cookie already during the visit; emerging clickstream is immediately saved to a database. Next, it is shown how can a varying length clickstream be transformed into a constant number of attributes in a way that makes the transformed data a suitable input for association rules mining procedures.

Agent Architecture based on Semantic Knowledge Model*

Michal Laclavík, Zoltán Balogh, Emil Gatial, Ladislav Hluchý

Institute of Informatics, Slovak Academy of Sciences,
Dubravska cesta 9, 845 07 Bratislava, Slovakia

laclavik.ui@savba.sk

Abstract. In this paper we describe a semantic knowledge model of an agent as well as implementation and use of such model using the Jena semantic library and the JADE agent system. The solution has been used and evaluated in the Pellucid and K-Wf Grid projects.

1 Introduction

Multi-Agent Systems or Agent Oriented Programming is a highly powerful method in distributed heterogeneous information systems requiring representation and reasoning with knowledge. At present, MAS lack interconnection with current commercial technological standards and the results of the semantic web research [6]. In the MAS knowledge is usually represented by states, rules or predicate logic. This is extremely powerful but it is hard to capture knowledge from a person or from current information systems in such a way, furthermore it is difficult to present knowledge information expressed in e.g. predicate logic to the end user. Ontology as understood in the Semantic Web is closer to current information systems because it is based on XML/RDF and enables an easier capture or return from/to a person or an existing information system.

For this reason we have decided to integrate semantic web results into MAS and create architecture, methodology and software for such integration.

Agent architectures are the fundamental engines underlying the autonomous components that support effective behavior in the real-world, dynamic and open environments. There are several basic types of agent architectures: Reactive Architecture; Belief Desire Intention Architecture – BDI; Behavioral Architecture. In literature, the main focus is on the externals of the agents, their communication with environment and other agents. The internal knowledge model is left for an agent creator. The most advanced, but not sufficient approach is in the JADE [1] agent system. JADE support for ontologies or agent knowledge modeling [2] is based on Java classes. The JADE agent model is not sufficient in several ways. Such model cannot support features of semantic ontology representations such as OWL, does not have a query engine for the FIPA-SL language [3] and a model based on the predicate logic

* This work is supported by the project K-Wf Grid, EU RTD IST FP6-511385, VEGA No. 2/3132/23, APVT-51-024604 and SPVV 1025/2004.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 288–291, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

is hard to be communicated with a user and existing commercial systems. Thus we have created an RDF/OWL based model. Agent communication based on RDF/OWL was presented in other works such as [4], but this work lacks a generic internal model, it only offers a theory how RDF/OWL can be used for agent communication. In this work we considered parts of those two models and extended it with the event based memory model. The event based model was previously used in many areas but in the agent field only work of Anderson [5] deals with the event model as a typical reactive architecture. On the other hand in our model we use events to take action and to have a history of an action allowing us to have history of an environment in any moment of its history which can be preprocessed at any time and different results can be achieved with the same knowledge model.

2 Agent Knowledge Model

Our architecture is based on behavioral architecture, where an agent memory model based on OWL-DL is used. First we will describe agent memory formally by description logic (DL) compatible with OWL-DL and graph description and subsequently we will move forward to modeling and implementation using RDF/OWL. A model is based on events. The idea is taken partially from the JADE ontology model – predicates and concept. As we described, the JADE ontology model [2] has limitations.

Our model is based on five main elements: Resources, Actions, Actors, Contexts and Events. Figure 1 shows formal graph representation of a model. The proposed model is described using DL too[7].

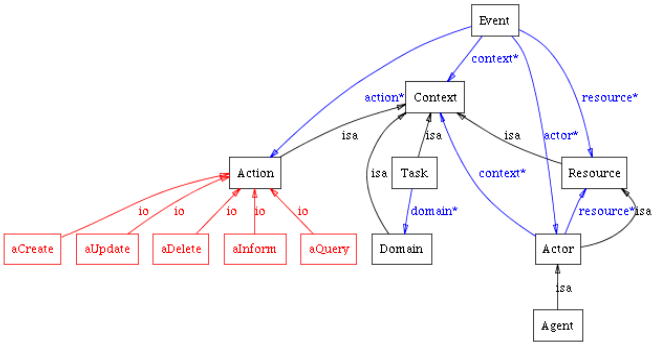


Fig. 1. Basic Ontology for Knowledge Modeling

Here we describe only the Actor class using DL, which consists of important properties: context.Actor, resource.Actor. The context.Actor represents actor current context. The system or application environment can be found based on stored events. Events model environment state. The current state of the environment or the actor related environment/context is thus affected by relevant new events. The resource.Actor property stands for all current resources of the actor. This resources are results of actors intentions or objectives and thus they are dependable on current actors environment state/context (context.Actor). Functions/algorithms for context and resource updating were specified above.

$$\begin{aligned} \text{Actor} \subseteq\ & \\ & \text{resource.Actor}(\{\text{resource}\}) \cap \\ & \text{context.Actor}(\{\text{actor}\}) \cap \\ & \text{Resource} \end{aligned} \tag{1}$$

$$\text{contextActor}(\{\text{actor}\}) = f_C(\forall \square \{\text{event}\}; \text{actor.Event}(\{\text{actor}\}) \in \{\text{event}\}) \quad (2)$$

$$\text{resource.Actor}(\{\text{resource}\}) = f_R(\text{contextActor}(\{\text{actor}\})) \quad (3)$$

This is where the brain of the system is located. An advantage of such model is that it enables to achieve better results when such algorithms are changed in the future, using the same model and data. Due to storing all events we can model the environment in any moment from past and process it later from any starting point with improved algorithms for context and resource updating.

2.1 Modeling Methodology for Agent Design

The developed methodology integrates several parts of different methodologies and follows CommonKADS methodology. However CommonKADS is not tied with any modeling tool, knowledge representation or ontologies. We present knowledge based on the OWL ontology and we model it in the Protégé ontology editor. Thus defined methodology for knowledge model is similar to [8]. Ontology modeled with Protégé reflects CommonKADS models and it has several commonalities with the JADE ontology model. Design of the system is based on UML, AUML and MAScommonKADS. When using this methodology, good results can be archived only after several iterations of the process and remodeling after the first developing, use and evaluation of the first system version.
Modeling a knowledge model for an application we follow first three CommonKADS models: Organizational or in our case the Environment Model; the Task Model; the Agent or the Actor Model.
Modeling the knowledge model we have to extend the generic agent model (Figure 1) with new elements and relations.
Results of models is the knowledge model which consists of :

- Ontology developed in Protégé which can be exported in the OWL format.
- Algorithms for each actor (algorithms often are similar or same), which updates actors' context f_C (2) and resources f_R (3).

The design methodology of a system is based on three UML diagram types in a similar way as known in object oriented programming: Use Case Diagram; Sequence Diagram; Class Diagram.

2.2 Design and Specification of Agent Software Library

The developed library is based on the JADE agent system and the Jena library. It covers functionalities which we identified as missing in current agent architectures such as JADE, the Agent Knowledge Model based on RDF/OWL; Action resource as basics for communication and incorporated in the OWL ontology; Sending ACL based RDF/OWL messages; Receiving ACL based RDF/OWL messages; Incorporation of received information into a model; XMLRPC receiving messages; XMLRPC returning RDF and XML; An inference Engine or RDQL messages handling.

The JADE based agent can be developed using this library to support the Jena OWL model as the Agent memory and furthermore it is possible to include XML-RPC based functionalities for presenting knowledge or receiving events from external systems as RDF messages based on a used ontology. Moreover it allows communication among agents based on RDF/OWL. The developed library is published on JADE website as the 3rd party software [9] as a way to bring together Jena RDF/OWL and JADE Multi-agent system features.

6 Conclusion and Future Work

The paper describes how semantic web results can be used in multi-agent systems. An agent knowledge model has been created; it can model the agent environment, agent context and its results. The agent library to support such a model has been developed extending the JADE agent system. It has been proved that such a model and agent architecture are implementable and can be used in different knowledge management applications. The approach has been successfully used and evaluated in several R&D projects, mainly in the Pellucid IST project. The developed library for using such a semantic model has been published on the JADE agent website and it is available to the worldwide JADE community. Presented work will be further developed and extended. More detailed information on modeling, software library and concrete scenarios and use can be found in [10].

References

1. JADE (Java Agent DEvelopment Framework) Website, 2004, http://jade.cselt.it/
2. Giovani Caire, JADE Tutorial Application-defined Content Languages and Ontology, 2002
3. FIPA, FIPA SL Content Language Specification, 2000
4. Marek Obitko, Vladimir Marík, OWL Ontology Agent based on FIPA proposal, Znalosti 2004, Brno, Czech Republic,2004
5. JAMES J. ANDERSON, An Agent-based Event Driven Foraging Model, Natural Resource Modeling, Volume 15, Number 1, 2002
6. Michael Luck, Peter McBurney, Chris Preist, Agent Technology: Enabling Next Generation Computing, A Roadmap for Agent Based Computing, 2003
7. Franz Baader, Deborah McGuinness, Daniele Nardi, The Description Logic Handbook, ISBN:0521781760, January 9, 2003
8. Guus Schreiber, Monica Crubezy, Mark Musen, A Case Study in Using Protege-2000 as a Tool for CommonKADS, ,2001
9. Michal Laclavik, AgentOWL; http://jade.tilab.com/community-3rdpartysw.htm#AgentOWL
10. Michal Laclavik: Ontology and Agent based Approach for Knowledge Management; PhD Thesis; Institute of Informatics, SAS, field: Applied Informatics, June 2005

O význame slovných druhov pre klasifikáciu dokumentov v slovenskom jazyku

Marián Mach

Katedra kybernetiky a umelej inteligencie, FEI, Technická univerzita v Košiciach,
Letná 9, 041 20 Košice, Slovenská republika

Marian.Mach@tuke.sk

Abstrakt. Článok sa zaoberá klasifikáciou dokumentov do jednej z dostupných tried. Pozornosť je sústredená na vplyv uvažovania jednotlivých slovných druhov na kvalitu klasifikácie meranú ako mikro tak aj makro charakteristikami. Experimenty boli realizované s využitím kolekcie dokumentov v slovenskom jazyku.

Kľúčové slová: Klasifikácia dokumentov, slovné druhy, presnosť a návratnosť, mikro a makro spriemernenie

1 Úvod

Pri automatickej kategorizácii dokumentov sa dominantne používajú klasifikačné prístupy založené na štatistických vlastnostiach dokumentov (prítomnosť slov, frekvencia ich výskytu, ap.). Tento príspevok je pokusom o malé vykročenie smerom k analýze textu. Stavia na hypotéze, že pre účely klasifikácie by bola postačujúca iba limitovaná analýza. Jedným zo spôsobov limitácie je jej obmedzenie iba na niekoľko málo slovných druhov. Aby sme overili prijateľnosť tohto prístupu, realizovali sme niekoľko experimentov s cieľom zistiť, ako sa zmení kvalita klasifikácie dokumentov pri uvažovaní iba niektorých slovných druhov.

2 Kolekcia dokumentov

Pre experimentovanie bola použitá kolekcia "Markíza" – kolekcia dokumentov spravodajského charakteru. Dokumenty z kolekcie boli predspracované v niekoľkých krokoch. Najprv boli z kolekcie odstránené všetky triedy s menej ako 20 dokumentami. Následne bola vykonaná lexikálna analýza dokumentov (vyčlenenie slov podľa "bielych"znakov, spojenie slov rozdelených cez konce riadkov, prevod na malé písmená, odstránenie nealfanumerických znakov).

Nasledujúcim krokom bolo určovanie slovných druhov. V tomto kroku sa uvažovali všetky slovné druhy s výnimkou čísloviek (typická podoba lexikálnej analýzy zanedbá všetky "slová" neobsahujúce ani jedno písmeno). Slovný druh bol priradzovaný iba slovám, ktoré sa vyskytovali aspoň v piatich dokumentoch. Určovanie slovných druhov bolo realizované čiastočne automaticky s použitím

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 292–295, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

morfologických slovníkov slovenského jazyka, vytvorených v minulom období na materskom pracovisku, a čiastočne manuálnym priradzovaním (asi jedna tretina slov). Počas tejto etapy nebola vykonávaná žiadna disambiguácia – a teda jedno slovo mohlo byť priradené k viacerým slovným druhom. Posledným krokom predspracovanie kolekcie dokumentov bolo odstránenie tých dokumentov, ktoré neobsahovali slová všetkých skúmaných slovných druhov.

Výsledkom bola kolekcia 21 878 dokumentov kategorizovaných do 84 kategórií (každý z dokumentov patril práve do jednej z týchto kategórií). Štyri najpočetnejšie kategórie spolu obsahovali viac než polovicu všetkých dokumentov (každá s viac než tisíc dokumentmi); 58 kategórií malo menej ako sto dokumentov, z toho 35 ich malo menej ako 50. Dokumenty boli dlhé 12 až 2872 slov, pričom 12 705 malo dĺžku v rozmedzí 100-299 slov a 19 784 v rozmedzí 50-499 slov. V kolekcii sa vyskytovalo 70 998 rôznych slov, pričom 29 842 z nich sa vyskytovalo v menej ako desiatich dokumentoch, 17 217 v 10 až 19 dokumentoch, 13 369 v 20 až 49 dokumentoch a 833 vo viac ako 500 dokumentoch.

3 Popis experimentov

V úlohe klasifikátora bol použitý naivný Bayesov klasifikátor vo verzii podľa [1] (využívajúci princíp indiferencie a spriemerňovania modelov). Jeden klasifikátor bol použitý pre všetky kategórie, pričom pre klasifikáciu používal informáciu o prítomnosti 200 slov, ktoré boli selektované na základe informačného zisku z množiny uvažovaných slov (danú práve skúmaným slovným druhom).

Pre vyhodnocovanie kvality klasifikácie boli použité presnosť a návratnosť a miera F1 kombinujúca obe uvedené miery s rovnakou váhou. Výsledky klasifikácie dosiahnuté v rámci jednotlivých tried boli spriemernené, pričom bolo použité ako mikro[1] tak aj makro spriemerňovanie [2].

Pri testovaní bola použitá metóda 5x2 krížovej validácie, pričom pre testovanie signifikancie rozdielov medzi dosiahnutými výsledkami bol použitý kombinovaný 5x2 cv F test [3].

4 Výsledky experimentov

Pri prvom experimente boli brané do úvahy vždy iba slová jedného slovného druhu. Pri slovách priradených k viacerým slovným druhom sa uvažovala "optimistická disambiguácia" – slovo patrilo k slovnému druhu ak ten druh bol v zozname slovných druhov daného slova. Neuvažovali sa tie slovné druhy, ktoré mali v celej kolekcii dokumentov menšie zastúpenie ako 200 slov (citoslovce – 57 slov, predložky – 86, spojky – 72). Dosiahnuté výsledky[2] sú v tab. 1.

[1] Keďže dokumenty boli klasifikované iba do jednej triedy, pri mikro spriemernení presnosť, návratnosť a ich kombinácia F1 sú rovnaké a preto je uvádzaná iba F1.

[2] Výsledky sú uvedené v percentách referenčnej hodnoty, pričom ako referenčné boli uvažované výsledky dosiahnuté bez obmedzenia slovných druhov.

Tabulka 1. Optimistické určovanie slovných druhov.

slovný druh	makro spriemernenie			mikro spriem.
	presnosť	návratnosť	F1	F1
vlastné podstatné mená	91.22 ± 3.98	81.90 ± 1.80	85.71 ± 1.96	102.75 ± 0.79
všeobecné podstatné mená	87.03 ± 2.98	70.78 ± 1.32	77.08 ± 1.53	99.90 ± 0.86
prídavné mená	78.04 ± 4.40	50.17 ± 1.24	59.45 ± 1.68	94.14 ± 0.67
zámená	43.20 ± 2.38	47.13 ± 1.12	45.30 ± 1.47	55.63 ± 0.81
slovesá	70.66 ± 2.96	69.16 ± 1.71	69.80 ± 1.95	73.20 ± 1.68
príslovky	46.62 ± 1.10	50.72 ± 1.27	48.83 ± 0.75	66.26 ± 0.80
častice	57.42 ± 4.22	43.98 ± 1.14	48.97 ± 2.02	60.20 ± 1.13

Tabulka 2. Pesimistické určovanie slovných druhov.

slovný druh	makro spriemernenie			mikro spriem.
	presnosť	návratnosť	F1	F1
vlastné podstatné mená	93.35 ± 4.10	73.63 ± 1.36	81.12 ± 1.89	105.75 ± 0.98
všeobecné podstatné mená	88.49 ± 4.68	71.24 ± 1.31	77.86 ± 1.94	100.70 ± 0.68
prídavné mená	72.55 ± 4.99	47.56 ± 1.17	55.96 ± 1.62	94.21 ± 0.96
zámená	33.73 ± 2.30	31.80 ± 1.31	32.56 ± 0.95	53.89 ± 0.60
slovesá	67.30 ± 3.85	65.78 ± 1.87	64.40 ± 2.38	72.19 ± 1.54
príslovky	45.10 ± 2.80	35.96 ± 0.86	39.44 ± 1.34	67.17 ± 0.47

Z hľadiska mikro spriemerňovaných výsledkov je použitie iba podstatných mien veľmi vhodnou alternatívou (pri všeobecných sa dosiahli rovnaké výsledky ako pri uvažovaní všetkých slov, pri vlastných boli výsledky ešte priaznivejšie). O niečo horšie hodnoty sa dosiahli pri použití prídavných mien, ostatné slovné druhy nepredstavujú vhodnú alternatívu. Z hľadiska makro výsledkov opäť podstatné mená dosahujú najlepších hodnôt, avšak zaostávajúcich za výsledkami dosiahnutými pri uvažovaní všetkých slov.

Rovnaký experiment bol vykonaný s uvažovaním "pesimistickej disambiguácie" – slovo patrilo k slovnému druhu iba ak ten druh bol jediným druhom v zozname slovných druhov daného slova (a teda slovo s viacerými priradenými slovnými druhmi nebolo vôbec uvažované). Navyše voči predchádzajúcemu experimentu sa neuvažovali ani častice, ktorých zastúpenie pokleslo pod 200 slov (teraz iba 69 slov). Dosiahnuté výsledky sú v tab. 2.

Všeobecné podstatné mená dosiahli porovnateľné výsledky pri oboch experimentoch. Z hľadiska mikro výsledkov podľa použitého testu [3] rovnaké výsledky sa dosiahli aj pri prídavných menách a príslovkách. K signifikantnému zlepšeniu došlo iba u vlastných podstatných mien (ale toto zlepšenie mikro výsledkov bolo doprevádzané zhoršením makro návratnosti).

Tabulka 3. Počty spoločných termov.

vlastné podst. mená	všeobecné podst. mená	prídavné mená	zámená	slovesá	príslovky
172.1 ± 1.64	192.2 ± 0.6	167.2 ± 1.99	130.2 ± 1.25	185.5 ± 0.67	111.0 ± 1.48

Pre porovnanie vplyvu "optimistického" a "pesimistického" spôsobu disambiguácie, tab. 3 uvádza počty slov, ktoré sa vyskytli v súbore 200 slov použitých pre klasifikáciu pri oboch prístupoch.

Oproti prvému prístupu, pri druhom sa už nevyskytli napr. slová označené súčasne ako prídavné mená a príslovky (digitálne, mimoriadne), všeobecné podstatné a prídavné mená (vylúčení, zamestnaní), všeobecné podstatné mená a príslovky (domov, časom), príslovky a častice (mimochodom, doslova), príslovky a slovesá (nadnes, neraz, určite), všeobecné podstatné mená a slovesá (hostia, zápas, kole, kop), slovesá a častice (veď, raz, nič), slovesá a predložky (popri), atď. Vidno, že obmena bola najmenšia pri všeobecných podstatných menách, ktoré vďaka tomu v oboch prípadoch dosiahli veľmi podobné výsledky.

5 Záver

Experimenty potvrdili prijateľnosť obmedzenia obsahu dokumentov na určité slovné druhy, pričom ako použiteľné alternatívy sa ukázali vlastné a všeobecné podstatné mená resp. ich kombinácia (výsledok ďalších v článku neuvedených experimentov s kombinovaním slovných druhov). Ovšem pre všeobecné prijatie tohto záveru je nevyhnutné reprodukovať experimenty aj s využitím kolekcií dokumentov s inými charakteristikami ako použitá kolekcia.

Tento príspevok vznikol vďaka podpore VEGA grantu č. 1/1060/04 "Klasifikácia a anotácia dokumentov pre sémantický web".

Reference

1. Cerquides, J.: Improving Bayesian Network Classifiers. PhD Thesis, Departament de Llenguatges i Sistemes Informatics, Universitat Politecnica de Catalunya, 2003.
2. Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Computing Surveys, vol. 34, no. 1*, 2002, 1–47.
3. Alpaydin, E.: Combined 5 x 2 cv F Test for Comparing Supervised Classification Learning Algorithms. *Neural Computation, vol. 11*, 1999, 1885–1892.

Annotation:

About importance of word categories for classification of documents in Slovak

The paper is dedicated to classification of documents into one of available classes. The focus is on how considering particular word categories can influence classification quality measured using both micro and macro characteristics. Experiments were carried out using a collection of documents in Slovak.

Management znalostí pro inteligenci prostředí ve znalostní ekonomice

Peter Mikulecký

Katedra informačních technologií, FIM, Univerzita Hradec Králové,
Rokitanského 62, 500 03, Hradec Králové

Peter.Mikulecky@uhk.cz

Abstrakt. Naše společnost postupně přechází z vývojové fáze, které se kvůli významné úloze informací v nich začalo říkat ***informační společnost*** do nové vývojové fáze, zdůrazňující mimořádnou roli znalostí. Tato ***znalostní společnost*** musí být zákonitě založená i na novém typu ekonomiky, tzv. nové či síťové ekonomice, pro kterou se snad již vžil název ***znalostní ekonomika***. Znalostní ekonomiku lze charakterizovat z mnoha aspektů, většina z nich kromě jiného předpokládá i změnu prostředí, ve kterém bude tato ekonomika působit. Ve svém příspěvku upozorňujeme, že výkonnosti znalostní ekonomiky by prospělo, kdyby její fungování bylo podpořeno zvýšenou inteligencí prostředí (ambient intelligence).

Klíčová slova: znalostní ekonomika, znalostní společnost, inteligence prostředí

1 Úvod

Nacházíme se v etapě přechodu společnosti z jedné vývojové fáze do jiné, pravděpodobně vyšší. Ta první etapa zvýrazňovala důležitost informací pro její fungování, bývá zvykem ji proto nazývat informační společností. Ta druhá, uvědomujíce si, že bez kvalitních znalostí nelze sebelepší informace správně použít, si vysloužila název znalostní společnost. S kvalitativní změnou celé společnosti souvisí samozřejmě i změna její ekonomiky. Odkdy v 90. letech minulého století došlo k revolučnímu technologickému pokroku v oblasti vysokorychlostních komunikací a zejména k širokému rozvinutí Internetu a jeho služeb, můžeme být svědky vzniku úplně nového, globálního trhového prostředí, které bývá ve svých různých projevech charakterizované jako ***postindustriální ekonomika***, ***síťová ekonomika***, ***digitální ekonomika***, nebo snad nejčastěji ***znalostní ekonomika***.
Mezi tři základní charakteristické rysy nové ekonomiky lze zahrnout zejména to [1], že nová ekonomika může znamenat možnost vyššího trendu ekonomického růstu v důsledku toho, že se

- uplatňují efektivnější mechanismy ekonomických činností založené na ICT,
- může dojít k ovlivnění cyklických procesů vlivem snížení přirozené míry nezaměstnanosti souhrou toho, že ICT stláčí inflaci a globální konkurence drží inflační trendy ve mzdách pod kontrolou,
- zdrojem růstu se stávají rostoucí výnosy ze síťových efektů a vnějších vlivů, které pozitivně ovlivňují některé části nové ekonomiky.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 296–299, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

V odborné literatuře (viz např. [2]) lze najít několik základních charakteristik znalostní ekonomiky. Stručně si je rozeberme, neboť většina z nich souvisí se změnou povahy prostředí, ve kterém tato nová ekonomika funguje. Dále budeme uvažovat o úloze znalostí při zvýšení inteligence prostředí tak, aby toto inteligentnější prostředí představovalo oporu jednotlivým základním funkcím znalostní ekonomiky.
Domníváme se, že management znalostí je ta strategie, která zvýšení inteligence prostředí může v dohledné době přinést a tím napomoci lepšímu fungování znalostní ekonomiky v moderní společnosti.

2 Charakteristiky znalostní ekonomiky

Znalostní ekonomika využívá informačních a komunikačních technologií k ***vytváření zboží symbolické povahy***. Digitální symboly reprezentují informace, které potřebujeme znát o fyzicky existujícím zboží, abychom mohli provozovat obchod. Tyto elektronické symboly digitalizují lidskou komunikaci, navíc je lze rychle přenášet a přijímat v pravém slova smyslu vždy a všude. Organizace v této nové ekonomice tedy musí být schopny vytvářet a spravovat tyto symboly. Z toho plyne potřeba vytvořit v organizacích prostředí nového typu, které je přirozeně schopno nárokům digitalizace zboží čelit a podporovat s tím související technologie a postupy.
Ve znalostní ekonomice dále ***vzniká snížená závislost organizací na potřebě fyzické koncentrace zdrojů***. Takovéto uspořádání je typické např. pro softwarové firmy, kde třeba vedení firmy sídlí ve Francii, programátoři vytvářející nový software se rekrutují z odborníků pracujících v Indii nebo v Irsku a prodej hotových produktů je prostřednictvím Internetu možný kdekoliv. Současné organizace musí být tedy schopny vytvářet pro své pracovníky pracovní prostředí, podporující jejich rozptýlení po celém světě, aniž by byla nutná jejich fyzická koncentrace do jediné lokality.
Znalostní ekonomika je rovněž typická tím, že ***nemá pevně definované hranice*** [3]. Znalosti překračují firemní, oborové, ba i státní hranice. Potřebné znalosti už nelze najít na jediném místě. Organizace již nejsou natolik závislé na čase a místě, práci v nich lze tedy provozovat z mnoha různých míst. Přenositelnost práce, umožněná velkým rozvojem mobilních technologií dovoluje plnit pracovní úkoly kdekoliv. Z toho vyplývá potřeba rozvíjet a zvyšovat inteligenci netradičních prostředí, doposud pro různé typy znalostních činností nepoužívaných (letiště, nádraží, atd).
Nové technologie umožňují ***transformovat fyzicky existující organizace do virtuálních***, které pracují převážně na základě kontraktů s dodavateli a distributory, a obvykle pronajímají spolupracovníky k plnění specifikovaných úkolů. Takovéto pracovní týmy lze najmout doslovně z celého světa, přičemž jsou schopné pracovat společně aniž by museli být všichni přítomni na stejném místě ve stejný čas. Virtualizace má mnoho výhod, spočívajících zejména ve vysoké pružnosti virtuálních týmů a ve využitelnosti získaných zdrojů, na druhé straně však klade velké nároky na management zejména z hlediska obtížné koordinace všech zúčastněných. Z toho plynoucí nároky na prostředí ve kterém zúčastnění pracovníci působí, jsou zřejmé.
Počítače, počítačové sítě a zejména Internet přinesly pro organizace a instituce nebývalé ***možnosti vzájemného propojení a vytváření partnerství***. Díky síťovému propojení lze ve velkém rozsahu sdílet informace i znalosti, lze vytvářet „ad hoc“

partnerství organizací, spojených ke splnění společných cílů. Tato potřeba partnerství je vyvolána zejména tou skutečností, že dnes lze jenom stěží očekávat, že jediná organizace bude vlastnit veškeré znalosti k tomu, aby úspěšně konkurovala všem ostatním. V organizacích musí být proto vytvořená nová inteligentnější prostředí zaměřené zejména na využití možností, které poskytuje management znalostí.
Vzhledem k možnostem přímého síťového propojení zákazníka s výrobcem, nebo zaměstnance s vedením organizace lze postupně eliminovat vše, co stojí v cestě bezprostřednosti tohoto vztahu. Tuto vlastnost lze nazvat ***eliminací zprostředkovatelů ekonomických aktivit***. Pochopitelně toto přináší další změny prostředí, ve kterém organizace doposud působí.
Znalostní ekonomika umožňuje dovést služby zákazníkům až do stadia ***personalizace***, neboli přizpůsobení služeb či výrobků jedinečným individuálním požadavkům zákazníků. Znamená to, že zákazníci hrají mnohem aktivnější roli ve výrobním procesu, neboť jejich znalosti, informace či myšlenky se stávají součástí procesu specifikace vlastností finálního výrobku či poskytované služby. V důsledku této vlastnosti musí být organizace schopné nepřetržitě přizpůsobovat své kreativní a inovační schopnosti. Zvýšená inteligence prostředí v těchto organizacích by tomuto mohla velice pomoci.
Ve znalostní ekonomice se obchody velice často provozují v ***reálném čase***, nejsou tedy mezi jednotlivými kroky tohoto procesu žádné prodlevy. Úspěšné společnosti se nepřetržitě a rychle přizpůsobují změnám podmínek. Životní cyklus výrobku je mnohem kratší, než v minulosti. Rychlost všech akcí s tímto spojených klade nebývalé nároky na přizpůsobivost firem a na rychlost jejich reakce. Inteligence pracovního prostředí managerů, řídících tyto firmy a nucených vydávat rychlá a co nejkvalitnější rozhodnutí, jim může velice pomoci.
Znalostní ekonomika dala vzniknout ***společenstvím zákazníků***, ve kterých mohou zákazníci komunikovat mezi sebou na lokální nebo i globální bázi. Zákazníci mohou spolu diskutovat a sdílet informace, to vše v reálném čase. Tyto možnosti jsou velmi užitečné zejména v případě potenciálních zákazníků, zajímajících se o nějaký výrobek či o službu V případě jakýchkoliv pochybností není pro ně problém najít na Internetu několik stávajících zákazníků, kteří mají s daným výrobkem, službou, či firmou už nějaké zkušenosti a obvykle se o ně rádi podělí. Mnohdy tato možnost ušetří potenciálnímu zákazníkovi případná zklamání, nebo naopak posílí jeho rozhodnutí vedoucí posléze k uzavření obchodní transakce. Zde lze znovu vidět možnosti pro tvorbu inteligentních prostředí podporujících činnost takovýchto komunit.

3 Management znalostí a inteligence prostředí

Koncept inteligence prostředí (ambient intelligence) patří mezi vůdčí témata výzkumu organizovaného v rámci programů Evropské unie, zejména v jejím 6. rámcovém programu. Tento koncept představuje vizi budoucí znalostní společnosti, ve které je největší důraz kladen na uživatelskou přívětivost, efektivní podporu speciálních i ostatních služeb, opakované využívání zdrojů uživatele, i na podporu interaktivních činností. Tato představa předpokládá, že lidé budou působit v inteligentním prostředí plném intuitivně využitelných uživatelských rozhraní včleněných do objektů,

podporujících obvyklé i méně obvyklé činnosti uživatelů ve snaze pomoci jim řešit případné potíže či problémy. Tato prostředí budou schopna rozpoznat přítomnost konkrétní osoby a přizpůsobit dané prostředí této osobě na míru. Přitom by dané prostředí mělo reagovat prakticky neviditelným, nerušivým způsobem.
Chceme-li této vize dosáhnout, je nutno se zabývat technologickou, sociální a politickou dimenzí problému inteligence prostředí. Všechny tyto dimenze však souvisí se ***znalostním aspektem*** vytváření inteligentních prostředí: jaké znalosti a jak použité pomohou ke zvýšení inteligence konkrétního prostředí?
Management znalostí, jeho funkce a úlohy ve vztahu ke zvýšení inteligence prostředí, ve kterém budou lépe podporovány základní charakteristiky či funkce znalostní ekonomiky je téma, které je nutné podrobněji studovat již v nejbližší budoucnosti, neboť jsme na prahu znalostní společnosti a znalostní ekonomika už částečně funguje. Pro další rozvoj celé společnosti je nutné urychlení i jejího rozvoje.

Poděkování

Příspěvek úzce souvisí s řešením grantového projektu GAČR reg. č. 406/04/2140 KNOMEDIAS.

Reference

1. Stiroh, Kevin, J. Is There A New Economy?, *Challenge, July-August* 1999.
2. Lengnick-Hall, M.L., Lengnick-Hall, C.A. *Human Resources Management in the Knowledge Economy.* Berrett-Koehler Publ., Inc., 2003 San Francisco. ISBN 1576751597
3. Tapscott, D. *Digitální ekonomika: Naděje a hrozby věku informační společnosti*. Computer Press, 1999 Brno. ISBN 80-7226-176-2.

Annotation:

Knowledge Management for Ambient Intelligence Support in Knowledge Economy Environment

It is commonly accepted that our society is on the way from the development phase known as the **information society** into a new phase stressing the exceptional role of knowledge in it. This new knowledge society has to be inevitably based on a new type of economy, gradually known as new, or network economy, possibly better known under the name **knowledge economy**. This knowledge economy can be characterised from various standpoints and aspects, which commonly expect also a change in the environment where this new economy will be in effect. In the paper we stress the idea that this new economy effectiveness should be supported also by certain higher degree of ambient intelligence.

Implicitní znalosti v relačních modelech a jejich odkrývání topologickou analýzou pojmových map

Karel Mls

1Katedra informačních technologií, FIM, UHK Univerzita Hradec Králové,
Rokitanského 62, 500 02, Hradec Králové

Karel.Mls@uhk.cz

Abstrakt. Pojmové mapy jsou využívány pro znázorňování znalostních modelů a umožňují následnou strukturální analýzu. V příspěvku je ukázána a na konkrétním příkladu demonstrována možnost odvození dalších informací z prostorového uspořádání konkrétní pojmové mapy.

Klíčová slova: pojmová mapa, topologcká analýza, implicitní znalosti

1 Úvod

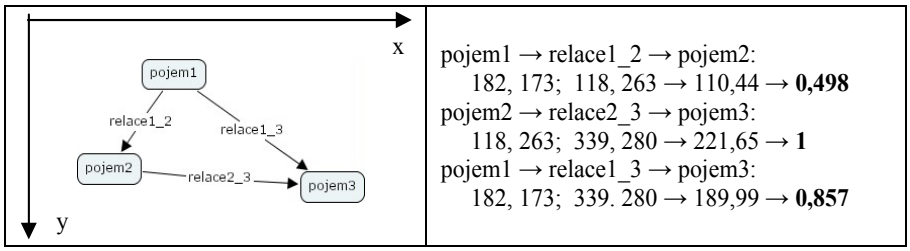
Zatímco data můžeme chápat jako elementární, ověřitelná fakta, znalosti umožňují tato data interpretovat a využívat. Lze tedy konstatovat, že člověk využíval a využívá znalosti pro lepší orientaci ve světě, ve kterém žije. Předpokládáme, že původní přínos znalostí byl založen na schopnosti lepší orientace ve fyzickém prostoru, na schopnosti vracet se na známá místa v krajině (potrava, řeka, tlupa, vesnice, …) a byl pro přežití člověka jako jedince zásadní. Přitom takové vnímání a interpretace "prostoru" je v praxi ve většině případů dvourozměrné, podobně jako mapa krajiny.
Jedním z nástrojů pro formální zachycování a znázorňování znalostí ve formě relačních struktur jsou pojmové mapy, navržené a propagované prof. Novakem [1]. Pojmové mapy se nejčastěji používají jako nástroj pro znázorňování znalostního modelu určitého problému při jeho analýze, vysvětlování, a jako objektivní metoda hodnocení kvality určitých získaných znalostí, například u studentů. V obou případech jsou pro jejich automatizovanou analýzu navrhovány různé metody a algoritmy [2, 3, 4, 5]; tento příspěvek se zabývá návrhem vyhodnocení a využití implicitní informace obsažené ve struktuře zakreslené pojmové mapy.

2 Hypotéza, návrh experimentu

Na základě uvedené analogie s mapováním fyzického prostředí, kde bližší bývá obvykle známější, předpokládáme, že podobné souvislosti lze nalézt i u pojmové mapy abstraktního problému, sestavené expertem, nebo obecně člověkem s určitou znalostí v dané oblasti. Taková pojmová mapa pak obsahuje nejenom znalosti explicitně vyjádřené (pojmy a relace mezi nimi), ale ve vzájemném uspořádání pojmů je obsažena i implicitní znalost - jak konkrétní jedinec vnímá důležitost jednotlivých komponent modelu. Které považuje za prioritní nebo centrální, které naopak za

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 300–303, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

nepodstatné, které logicky blízké, které vzdálené nebo které případně z důvodu neznalosti nepoužije vůbec.
Při práci s pojmovými mapami vycházíme z jejich grafické (grafové) reprezentace, kdy pojmům odpovídají uzly grafu a relace jsou znázorněny orientovanými šipkami mezi uzly. Takovou pojmovou mapu charakterizuje nejenom množina použitých pojmů a relací, ale jako doplňkovou informaci lze získat i souřadnice jednotlivých pojmů vzhledem ke zvolené souřadné soustavě. Absolutní hodnoty těchto souřadnic nejsou pro naše úvahy přímo zajímavé, proto je před vlastní analýzou převedeme na (euklidovské) vzdálenosti a s ohledem na požadavek poměrné porovnatelnosti těchto vzdáleností u různých map tyto vzdálenosti pro každou mapu normalizujeme. Ukázka výpočtu normalizovaných vzdáleností pojmů pro konkrétní pojmovou mapu je na obr.1, kde čísla pod zápisem konkrétní relace znamenají postupně x,y souřadnice pojmů, jejich euklidovskou vzdálenost a výslednou normalizovanou vzdálenost.



Obr. 1. Příklad výpočtu normalizovaných vzdáleností uzlů (pojmů) pojmové mapy ve zvoleném souřadném systému .

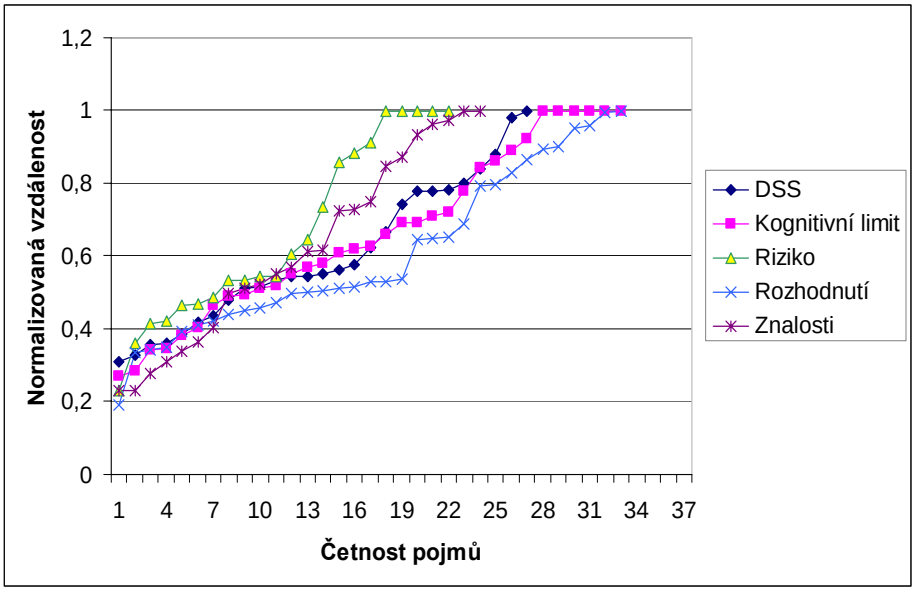
Analýzou získaných pojmových map se pokusíme potvrdit předpoklad existence vztahu mezi úrovní znalosti relace mezi jednotlivými pojmy a jejich prostorovou vzdáleností na zakreslené pojmové mapě. Za tím účelem byl navržen následující postup vyhodnocení dat:

- ze známých souřadnic jednotlivých pojmů spočítat čtvercovou matici jejich normalizovaných vzdáleností
- pro jednotlivé pojmy spočítat četnosti výskytu relací těchto pojmů v daném vzorku pojmových map
- seřadit jednotlivé pojmy podle četnosti výskytu v relacích a porovnat s odpovídajícími hodnotami vzdáleností.

3 Konkrétní příklad

Pro posuzování úrovně znalostí studentů – absolventů předmětu Systémy pro podporu rozhodování – byl do závěrečného testu zařazen úkol sestavit pojmovou mapu z dané množiny pojmů, charakteristických pro studovaný předmět. Studenti dostali za úkol určit a pojmenovat všechny relace mezi stanovenými pojmy, které považují za důležité, a vzniklý model znázornit formou jednoduché pojmové mapy, přičemž nebyli informováni o způsobu dalšího zpracování těchto dat. Následně byly pojmové

mapy převedeny v měřítku 1:1 do elektronické podoby a bylo provedeno jejich vyhodnocení podle navrženého postupu.
V první etapě vyhodnocování bylo zpracováno 37 pojmových map, některé výsledky jsou znázorněny na obr.2 a v tabulce 1.
Obr. 2 ukazuje typické závislosti normalizované vzdálenosti mezi různými pojmy (Manažer – DSS, Manažer – Kognitivní limit, Manažer – Riziko, Manažer – Rozhodnutí, Manažer – Znalosti) na četnosti jejich výskytu v pojmových mapách pro nejčastější relace. Tabulka 1 pak obsahuje souhrn četností všech relací a odpovídajících průměrných normalizovaných vzdáleností pro pojmy "Manažer" a "DSS", z nichž je patrná nepřímá souvislost mezi těmito daty.



Obr. 2. Závislost normalizovaných vzdáleností pojmů na jejich četnosti vzhledem k pojmu „Manažer".

Tabulka 1. Porovnání četností a průměrných normalizovaných vzdáleností dvojic pojmů vzhledem k pojmu „Manažer" (a)) a „DSS" (b))

	Data	Datový sklad	DSS	Kognitivní limit	Manažer	Modely	OLAP	Riziko	Rozhodnutí	Znalosti
a)	3	1	27	33	0	4	10	22	33	24
	0,5647	0,7868	0,6033	0,6619	0	0,6059	0,7452	0,6654	0,6064	0,6181
b)	22	9	0	6	6	29	12	7	16	14
	0,5306	0,6149	0	0,8795	0,6618	0,542	0,6518	0,7813	0,6138	0,5959

4 Závěr, další směry výzkumu

První výsledky analýz ukazují na zajímavé relace v dostupných datech – pojmy, které se v relacích vyskytují častěji jsou v grafickém zobrazení zobrazovány obvykle blíže a naopak - vzdálenosti pojmů, jejichž relace nejsou zmiňovány v pojmových mapách příliš často jsou zpravidla vzdálenější. Také závislost
Další zkoumání bude proto zaměřeno na analýzu většího množství dat a na hledání a exaktní definování souvislostí mezi prostorovým uspořádáním pojmových map a dalšími zjistitelnými daty o jejich tvůrcích.

Reference

1. Novak, J., and Gowin, D. 1984. *Learning How to Learn*. New York: Cambridge University Press.
2. Cañas, A.; Leake, D.; and Maguitman, A. 2001. Combining concept mapping with CBR: Towards experiencebased support for knowledge modeling. In Proceedings of *FLAIRS-2001*. Menlo Park: AAAI Press.
3. Gavalec, M.; Mls, K.: Individual Cognitive Maps and Evaluation of Their Dynamic Changes. *Kognice 2005*, Hradec Králové 2005.
4. Coffey, J. W., R. Hoffman, A. J. Canas and K. M. Ford, A Concept Map-Based Knowledge Modeling Approach to Expert Knowledge Sharing, *IKS 2002- The IASTED International Conference on Information and Knowledge Sharin*g, November 2002, Virgin Islands.
5. Canas, A. J., D. B. Leake, D. C. Wilson, Managing, Mapping and Manipulating Conceptual Knowledge *AAAI Workshop Technical Report WS-99-10: Exploring the Synergies of Knowledge Management & Case-Based Reasoning*, AAAI Press, Menlo Calif, July 1999.

Annotation:

Implicit knowledge in relational models and its revelation by topological analysis of concept maps

Concept maps are basicaly used for visualization of knowledge models and they enable future structural analysis. There is a possibility of inference of further information from the concept map arrangement presented in this article, together with a practical example.

TL - Komplexní komunikační jazyk pro MAS*

Jaroslav Müller

FEI - katedra informatiky, VŠB - Technická univerzita Ostrava,
jarekmuller@yahoo.co.uk

Abstrakt Oblast multiagentových systémů se neustále rozvíjí, stále však chybí možnost komunikace na úrovni přirozeného jazyka potřebná pro 'inteligentní lidské' rozhraní. Stávající jazyky plní tuto roli pouze ve speciálních případech. Navrhujeme tedy jazyk TL, dobře sémanticky podložený Transparentní intenzionální logikou, vhodný pro multiagentové systémy a analýzu přirozeného jazyka.

1 Úvod

Zpracování přirozeného jazyka respektive jeho logická analýza a multiagentové systémy jsou dva samostatné obory. Avšak jejich spojením by mohla vzniknout velmi perspektivní platforma pro vývoj aplikací schopných nejen komunikace s jinými aplikacemi, ale i s člověkem, který nemusí být zrovna počítačový odborník.

Za tímto účelem navrhujeme jazyk TL, který může sloužit jako komunikační jazyk v MAS a zároveň má dostatečnou vyjadřovací sílu, aby zachytil jakoukoli větu přirozeného jazyka. Tento jazyk má dobrý sémantický základ v Transparentní intenzionální logice (TIL), která poskytuje řadu výsledků a základní metodiku.

2 Komunikace v MAS

Komunikaci v multiagentových systémech standardizuje organizace FIPA[1] [1]. Základním prvkem komunikace mezi agenty je tzv. zpráva. Zpráva může mít v podstatě libovolnou formu a způsob přenosu, ale očekává se, že bude mít přesně definovanou strukturu obsahující určité atributy. Strukturu zprávy specifikuje FIPA ACL v [2]. Z atributů zprávy jsou pro účely tohoto článku relevantní zejména obsah a ontologie. Obsah zprávy nese samotné 'tělo' této zprávy, tedy její sémantickou složku. Obsah je výraz, zapsaný v nějakém jazyku pro obsah

* Tento výzkum je podporován programem „Informační společnost" AV ČR, projekt č. 1ET101940420 „Logika a umělá inteligence pro multi-agentní systémy" a grantem GA ČR č. 401/03/H047 „Logické základy sémantiky a reprezentace znalostí".

[1] FIPA – The Foundation for Intelligent Physical Agents, http://fipa.org

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 304–307, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

zpráv. Tento výraz používá řadu pojmů (identifikátorů), o kterých se předpokládá, že jejich význam agent-příjemce zná nebo je může 'dohledat'. Za tímto účelem nese atribut „ontologie“ název ontologie, která tyto pojmy definuje.

FIPA Abstract Architecture standardizuje, jaké podmínky musí jazyk pro obsah zpráv a ontologie splňovat, ale neomezují multiagentové systémy pouze na jeden konkrétní aparát.

3 Jazyk SL

Jako kandidáta na roli jazyka pro obsah zpráv FIPA poskytuje jazyk SL [3][2].

Jazyk SL navazuje na tradiční logiky, jako jsou predikátová logika prvního řádu (PL1) a modální logiky. Tyto logiky jsou však určeny pouze k vyjadřování tvrzení a nemají prostředky pro zachycení otázek nebo příkazů. Proto je SL rozšířen o některé další operátory.

Přestože má jazyk SL přesně popsanou syntaxi a způsoby užití, jeho sémantika je popsána jen velice neformálně a spoléhá se především na intuitivní porozumění výrazů a na konvence jejich používání. Specifikace jazyka SL [3] žádným způsobem nedefinuje jeho sémantiku, obsahuje pouze kapitolu *Poznámky k sémantice*[3], která stručně intuitivně způsobem popisuje význam jednotlivých operátorů.

Tato neformálnost spolu s jejím důsledkem, že nemůže být definována logika stojící za jazykem SL, by mohla vést k nedorozuměním mezi agenty. Např. už u jednoduchého tvrzení: „Agent a věří, že A a B.“, není zřejmé, jestli z toho plyne, že: „Agent a věří, že B a A.“.

Jazyk SL stojí na základech predikátové logiky prvního řádu, což z něj dělá jazyk vhodný pro popis matematických struktur. Pro popis 'běžných' nematematických věcí by byl vhodnější spíše jazyk postavený na základech obecné (filozofické) logiky.

4 Transparentní intenzionální logika a jazyk TL

Transparentní intenzionální logika (TIL) je postavená na podobných principech jako typovaný lambda kalkul. Základním stavebním kamenem TIL jsou *konstrukce* (analogie lambda termů), které slouží k zachycování tvrzení a označování entit. Každá konstrukce *konstruuje* (nebo v-konstruuje, pokud bereme v potaz valuaci v) nějaký objekt daného typu, nebo je *nevlastní* (v-nevlastní). TIL je tedy parciální logika, která pracuje s parciálními funkcemi.

Samotná Transparentní intenzionální logika '*tak jak je*' není použitelná jako jazyk pro obsah zpráv, a to z několika důvodů:

– Jazyk konstrukcí TILu je neformální. Jedná se spíše o standardní notaci nebo o způsob, jak konstrukce zapisovat, rozhodně to však není přesně specifikovaný formální jazyk. Znamená to tedy definovat syntaxi formálního jazyka pro vyjadřování konstrukcí, nejlépe pomocí bezkontextové gramatiky.

[2] A několik dalších, které jsou ovšem zatím v experimentálním stadiu.
[3] Notes on FIPA SL Semantics (pozn.)

- Jazyk konstrukcí není možné kódovat pomocí ASCII znaků. Často obsahuje řecké symboly, horní i dolní indexy... Toto je sice možné kódovat do ASCII např. pomocí systému LaTeX, ale pro komunikaci mezi počítačovými agenty je to naprosto nevhodné. Je tedy nutné navrhnout způsob, jak zakódovat tyto informace jednoduchým a přehledným způsobem pomocí ASCII znaků.
- Chybí spojení s ontologiemi. Definice trivializace umožňuje trivializovat libovolný objekt, navíc neurčuje způsob jak tento objekt vyjádřit. To není vhodné pro komunikaci v MAS, kde agent, který obdrží zprávu musí být schopen dekódovat veškerou informaci v ní obsaženou.
- Je potřeba standardizovat typovou bázi. Epistemická typová báze není dostatečná, je třeba ji např. obohatit o typ akcí, ale i o jiné typy, které jsou běžné při komunikaci v MAS.
- TIL má typ množin – množiny se zachycují pomocí jejich charakteristických funkcí – ale chybí mu typ seznamů. Přestože i ten by se dal zachytit pomocí funkcí, v komunikaci v MAS se používá tak často, že by ho bylo vhodné zavést přímo.

Jazyk TL řeší tyto nedostatky a přitom zachovává dobré vlastnosti TILu. Výsledkem je jazyk, který splňuje požadavky FIPA na jazyk pro obsah zpráv, a který má transparentní hyperintenzionální sémantiku převzatou z TIL. Tím TL získává také teorii významu a metodu analýzy přirozeného jazyka, které jsou v TIL dobře propracované.

5 Aplikace TL a další výzkum

Pro účely testování a dalších experimentů byl kodek jazyka TL implementován do prostředí pro implementaci multiagentových systémů JADE[4]. JADE sleduje standardy FIPA a je v rámci těchto standardů rozšiřitelný. Jazyk TL je do tohoto nástroje plně zaintegrován a může být používán stejně jako kterýkoli jiný jazyk.

Potenciální využití jazyka TL se neomezuje pouze na komunikaci na úrovni přirozeného jazyka. V multiagentových systémech nachází uplatnění hlavně matematická logika, ale ta je vhodná zejména k popisu matematických struktur. Pro popis reálného světa se příliš nehodí. Přitom většina procesů a komunikace v MAS se týká nematematických oblastí, a matematika se v MAS omezuje většinou na pouhé výpočty. Tady se nabízí jazyk TL, který vychází z obecné (filozofické) logiky.

V dalším ve stručnosti jmenujeme některé možnosti uplatnění jazyka TL:

Analýza znalostí agentů s omezenými prostředky Mnoho logik může zachycovat tvrzení o tom, že nějaký agent má určitou znalost. Tyto bývají často založené na modálních epistémických logikách. Tyto logiky ovšem předpokládají ideálního agenta, který zná všechny důsledky svých znalostí, což není v reálném prostředí možné. Transparentní intenzionální logika, a tedy i TL, poskytuje sémantické prostředky k přesnému popisu agentových znalostí i jeho inferenčních schopností.

[4] `http://jade.cselt.it/`

Inteligentní uživatelské rozhraní Síla multiagentových systémů je v jejich flexibilitě. Důsledkem toho ovšem je to, že není lehké navrhovat uživatelské rozhraní pro takový systém, protože se může poměrně volně měnit. Řešení je vhodné hledat v komunikaci člověk / stroj způsobem, který více odpovídá 'lidskému' způsobu než formuláře. Jazyk TL se hodí k tomuto účelu, protože je navržen analýzu 'lidského' způsobu reprezentace znalostí.

Usuzování v umělé inteligenci V poslední době se čeká alespoň malé množství inteligence od téměř každé softwarové komponenty. TL poskytuje logický aparát pro odvozování důsledků a hledání řešení. Od ostatních logik se liší zejména tím, že díky velké vyjadřovací síle a jemné transparentní sémantice se neomezuje na úzkou oblast a může v sobě zahrnout i aparát jiných logik.

6 Závěr

Komunikace v MAS je různorodá oblast a různé aplikace mohou mít různé požadavky na nástroje. V současné době je k dispozici celá řada jazyků pro obsah zpráv, ale žádný z nich není vhodný pro vyjadřování složitých zpráv, které mohou např. vyjadřovat věty přirozeného jazyka. Multiagentní systémy, které nějakým způsobem zahrnují člověka, se budou muset s tímto problémem vyrovnat (viz. [4, str. 42]), a proto navrhujeme jazyk TL.

Jazyk TL je postaven na dobrých sémantických základech Transparentní intenzionální logiky, která už se v oblasti logické analýzy přirozeného jazyka osvědčila, a je navržen tak, aby odpovídal specifikacím FIPA.

Reference

1. Foundation for Intelligent Physical Agents: FIPA Abstract Architecture Specification. `http://fipa.org/specs/fipa00001/` (2000)
2. Foundation for Intelligent Physical Agents: FIPA ACL Message Structure Specification. `http://fipa.org/specs/fipa00061/` (2000)
3. Foundation for Intelligent Physical Agents: FIPA SL Content Language Specification. `http://fipa.org/specs/fipa00008/` (2000)
4. Luck, M., McBurney, P., Shehory, O., Willmott, S.: Agent Technology: Computing as Interaction A Roadmap for Agent-Based Computing. AgentLink (2005)

Annotation:

TL - Complex communication language for MAS

Technologies based on multi-agent systems are being continualy developt. Still means of communication on the level of natural languages, which is needed for design of 'intelligent human' interaction, is missing. Actual languages can be used for this purpose only in special cases. Therefor we propose the language TL, which is semantically based on Transparent Intensional Logic, and it can be used by multi-agent technologies for natural language analalysis.

Textmining in the Grid Environment

Martin Sarnovský

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice,
Letná 9, 041 20, Košice, Slovakia

Martin.Sarnovsky@tuke.sk

Abstract. This paper presents the fusion of two approaches - Grid computing and text mining. GridMiner is a system developed at University of Vienna, and it is a framework for knowledge discovery process in the distributed Grid environment. JBOWL is a framework for text mining and information retrieval being developed at Technical University in Košice. In this paper the integration of document preprocessing services into the GridMiner system is proposed.

1 Introduction

Grid is a technology, that allows from geographically distributed computational and memory resources create an universal computing system with extreme performance and capacity [1]. System has the features of global worldwide computer, where all of the components are connected together via Internet. For the user it appears like a common workstation, but some segments of a solved task are computed in different parts of system. The main advantage of the grid is high efficiency of using technological capacity. The early Grid efforts started during the middle 1990's as the projects linking supercomputing sites. Main objective was to provide computational resources to high performance applications. Nowadays the Grid projects are built on protocols and services provided by the Globus Toolkit [2] that enables applications to handle distributed computing resources as s single virtual machine.
Motivation of this work is to use the Grid computational capabilities for text mining tasks. Some of the methods are time-consuming and use of the Grid infrastructure can bring significant benefits. Implementation of text mining techniques in distributed environment allows us to access different geographically distributed data collections and perform text mining tasks in parallel/distributed fashion.

2 GridMiner

GridMiner is a service based software architecture for distributed data mining in the Grid environment. This architecture is being developed on top of the Globus 3.0 toolkit, using OGSA-DAI for access to Grid Data Sources. This technology is being tested on medical application addressing treatment of brain injury victims.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 308–311, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

The aim of GridMiner project is to give to an user a tool of knowledge discovery process in the distributed Grid environment. The system provides also a graphical user interface that hides the complexity of the Grid, but still offers the possibility to interfere during the execution, control the task and visualize the results.

2.1 GridMiner architecture

2.1.1 Grid layer

GridMiner supports dynamic workflow concept, it means that user can compose workflow according to its individual needs. All services are integrated into a workflow, which is executed by the Dynamic Service Composition Engine (DSCE) [3]. Dynamic Service Composition Language (DSCL) was developed as a new specification for dynamic service composition, it is based on XML notation. DSCL allows the description of a workflow consisting of various Grid services and specification of their parameters. DSCE is implemented as a Grid service and can be controlled interactively by a client.
The data integration in the GridMiner is based on the wrapper mediator, which allows integrating heterogeneous relational databases, XML databases and comma separated value files into one homogeneous virtual data source [4]. The data mining process is supported by several services able to perform data mining tasks. Data mining services includes sequential, parallel and distributed implementation of data mining algorithms.

2.1.2 Web layer

Knowlege base is storing the information needed by the other components in the process of the knowledge discovery. It consists of ontologies, that are describing data domain, sources and activities. It also contains metadata that holds information about data in the data sources. Results of datamining tasks are also stored and explicit knowledge is generated as a application of results on the domain ontology.

Service configuration is implemented as a set of web applications able to interact with the user in the process of preparing data mining task. These web applications allows to configure the services, set the parameters and prepare workflow parameters.

The main purpose of DSCE Client is to bridge the Web and Grid environments.

2.1.3 User Environment

The GUI allows user to construct workflow and visualize the results of the task. It lies in the client environment, what can be any operating system supporting Java. This GUI allows easy integration of new data mining (and text mining) or data preprocessing services into the Grid.

3 JBOWL

JBOWL (Java Bag-Of-Worlds Library) is a software system developed in Java for support of information retrieval and text mining [5]. The system is being developed as open source with the intention to provide an easy extensible, modular framework for pre-processing, indexing and further exploration of large text collections, as well as for creation and evaluation of supervised and unsupervised text-mining models. JBOWL supports the document preprocessing, building the text mining model and evaluation of the model. In this work I focused on the preprocessing phase, which I implemented as a Grid service. JBOWL is able to preprocess collections of text documents with a flexible set of available preprocessing techniques. Vector space representation of text documents is supported and various preprocessing filters are implemented.

4 Implementation

4.1 Textmining and Grid Services

The third version of the Globus Toolkit is based on Grid Services. A Grid Service is simply a Web Service with a lot of extensions that make it adequate for a grid-based application. Grid Services, which are basically Web Services with improved characteristics and services.

My main task was to enhance the existing GridMiner system with the text mining support. GridMiner is a service-based system, so I decided to integrate the text mining support as a Grid service using JBOWL library. As a first step I selected the document preprocessing phase to implement. Since JBOWL already has implemented the tools for document preprocessing, my main goal was to design a service interface and simple client for that service. All features of document preprocessing was used, including various implemented filters and weighting schemes.

Service is implemented on top of the Globus Toolkit 3.0, which is a standard for developing Grid applications. The input parameters are specified by the client, output is represented by the vector space representation of text document collection. Representation is stored in the PMML (Predictive Model Markup Language) text model form, PMML in its latest specification includes the standard for building the model of text document collection. It is an XML extension and model in PMML format can be easily visualized in the GUI.

In order to implement the text mining support in the GridMiner's GUI, the client was designed as a web service. Simple JSP page was designed and it serves as a simple configurator of the input parameters for the service. Input parameters includes the document collection location and set of parameters selecting the weighting scheme.

This client can be simply enhanced with another service configuration settings including the selection of filters etc.

5 Conclusions and Future Work

In this paper I described my research effort, which focuses on the application and extension of the existing GridMiner system to knowledge discovery in text databases. Integration of the text mining services into the GridMiner system opens ways to a plenty of various possibilities for building the distributed text mining scenarios. Using the Grid as a platform it is possible to access different distributed document collections and perform various text mining tasks. Service oriented architecture allows the extensions of the system simply by adding different services for various algorithms. These algorithms can run in parallel fashion with different settings. Another way how to use the Grid infrastructure is parallelization of the algorithms. Some methods such as induction of the decision trees can be simply modified.

Acknowledgements

The work presented in this paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the project "Document classification and annotation for the Semantic web" No. 1/1060/04 and by the German-Slovak research project DAAD No. 8/2004 "Text Mining for Metadata Extraction and Semantic Retrieval".

References

1. I. Foster, C Kesselman, Computational Grids, The Grid - Blueprint for a new Computing Infrastructure, Morgan Kaufmann, 1999
2. The Globus Project, Globus toolkit 3.0. http://www.globus.org
3. P. Brezany, I. Janciak, A. Woehrer, A Min Tjoa - GridMiner: A Framework for Knowledge Discovery on the Grid - from a Vision to Design and Implementation Cracow Grid Workshop, Cracow, December 12-15, 2004
4. A.Woehrer, P. Brezany, A Min Tjoa - Novel mediator architectures for Grid information systems, Journal for Future Generation Computer Systems - Grid Computing: Theory, Methods and Applications, January 2005
5. Bednar, P., Butka, P., Paralic, J.: Java Library for Support of Text Mining and Retrieval. In Proceedings of Znalosti 2005, pp. 162-169, Stará Lesná (2005)

First considerations for applying AI techniques in LISp-Miner extension

Sebastian Schutte

University of Freiburg, Center for Cognitive Science,
Friedrichstr. 50, 79098 Freiburg

Schutte@cognition.uni-freiburg.de

Abstract. This paper deals with possible and already implemented extensions for the LISp-Miner system. The recent implementation comes along with a few drawbacks in usability for inexperienced data miners, and a complex setup procedure for large mining tasks. This paper discusses various possible and already tested extensions of the software to simplify and accelerate the setup. Techniques from the field of machine learning and artificial intelligence are discussed for this purpose.

Key words: LISp-Miner, Expert System, Setup Wizard,

1 Introduction

The LISp-Miner system is an academic software system for KDD. It contains several GUHA procedures [1]. The basic idea of GUHA methods is to offer all relevant true patterns, hidden in the analyzed data, see [1].

The procedure that this paper will deal with is the so called 4ft-Miner, a GUHA procedure to mine association rules [2]. Mining rules of the form $\varphi \approx \psi / \chi$ is a calculation intense process and defining a proper setup for a mining run is a complicated procedure. A setup for this mining task requires a set of relevant attributes (φ, ψ and χ in the formal definition above) and a set of defined states, these attributes can be in. This set of states is called 'categories' (within the LISp terminology). Defining these categories and choosing a set of categories to be computed afterwards, are the key tasks of a setup procedure.

The recent 4ft-Miner implementation provides a graphical user interface that enables a qualified user to create a fitting setup for a data mining trail in reasonable time.

A problem, that inexperienced users suffer from, is the necessity to define different categories from the attributes that are to be tested in a data mining run. Due to the fact that there is no setup procedure, that would guide the user through the process, the high number of possible choices can be confusing for the inexperienced user. A large data mining run on the other hand makes it necessary to perform similar setup steps for a number of times, depending on the number of attributes that are taken into account.

Many setup choices the user makes repeatedly in the process are guided by basic statistical assumptions the user has in mind. Most of the steps, leading from the general expectation of the user to the resulting setup-choices are context-freely describable.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 312–323, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

Embedding them in an implemented wizard system follows quite naturally. This papers will discuss, how to simplify the setup for mining association rules in an essential step: The setup of so-called 'coefficients', which are necessary to create reasonable sets of literals for the mining run. The described techniques will be adapted to simplify the setup of the attribute-categories in a similar way.

2 Basic conceptual outlines of the proposed extensions

To perform a long lasting contribution to the usability of the LISp-Miner software, the upcoming and already tested changes correspond to the following principles

- **Guidance**
  Explaining the purpose of the different setup steps enables the user to understand the underlying mathematical basis of the 4ft-Miner procedure. It also teaches a basic intuition for the dependencies between the setup steps. Above all, time consuming mistakes in the setup order can be avoided to prevent time consuming searches for possible sources of the problem.

- **Automation**
  Many decisions, the user makes during the configuration of his data mining run are guided by context-freely describable assumptions about the statistical scale-type of the data, the expected number of resulting association rules and the time consumption of the process. These assumptions can be mapped into the architecture of an expert system, that performs calculations, which are equivalent to those, the user would have done mentally.

3 A first approach: The Coefficient Setup Wizard

As a practical example for a working setup wizard, a rule based system for the initial coefficient setup was implemented. Like other LISp-Modules, it accesses the metabase directly via SQL to change or update specific values. Basis of the updates that the system performs in specific metabase tables are rules that are stored in a file, that can be automatically generated from a set of rule sheets.

The rule sheets can be generated in a blank or in a filled in version, based on a properly configured script. After the user (or the script) filled in the sheet, a machine readable file is generated out of it, that encodes the user choices and the chosen setup-decision for any case. This file is loaded when the wizard starts up. After performing a user dialog concerning the type of the data, the wizard calculates the case number from the dialog and searches his rule set for the same number. After finding the number, the system accesses the metabase to set the coefficients as the decision sheet allows it.

An example of a decision sheet is shown in figure 1. Due to the users interest in small values, the left-cut coefficient was chosen for this case. This coefficient-type with the length 1 to 3 leads to the automated generation of literals that consist of the first three categories of a given attribute.

```
---------------------------------------------------------------------------
SMALL SET of rules, FAST processing, NOMINAL data,
5 - 10 CATEGORIES, interested in SMALL VALUES.
0.2.1.5.0
Coefficient type:
[] subset
[] interval
[x] left cut
[] right cut
[] cuts
[] one category
Coefficient length:
=1-3
---------------------------------------------------------------------------
```

Fig. 1. Example of a rule sheet that associates the users intrest in small values with the left-cut coefficient type and the number of categories to the length 1 to 3.

For the user, the setup procedure appears more structured and simplified. Instead of navigating through a large number of possible actions, provided by the graphical interface, he is guided through a coherent dialog. The questions to be answered in this dialog refer to scale type of the present data and the general expectation of the user, concerning the number of association rules to be found.
The current setup procedure can be described visually in the decision tree diagram on the right side below. To make the proper decisions, the user has to navigate through a large set of possible decisions. The graph on the left hand side shows the simplified procedure.

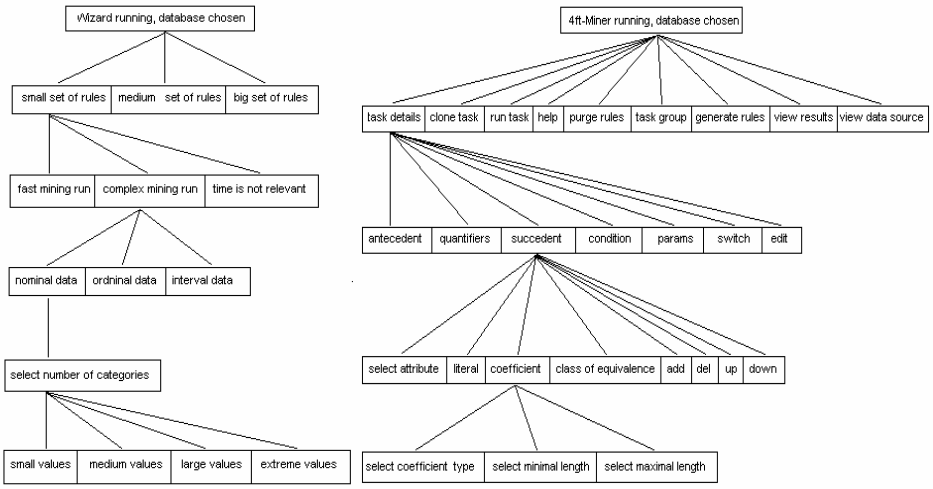


Fig. 2. Comparison of decision pathes for to the coefficient setup.

Like the example above, all future improvements of the setup procedure of LISp-Miner software will follow the conceptual principles of comprehensive guidance and automation.

4 Outlook

The proposed extensions for the 4ft-Miner Software are still in a conceptual state. The implemented coefficient wizard has prototype character. Nevertheless, the conceptual direction for the upcoming extensions is clear. Above a rule based initial setup, an optimization cycle could be added to the software. The user could be asked after the association rules were mined, if he is satisfied with the resulting number of rules or the discovered knowledge. If he is not, an other expert system could optimize the setup.

Even techniques from the toolkit of unsupervised machine learning could be taken into account. It is possible for example to perceive the sequence of choices made by the user as a 'chromosome' in term a genetic programming. This chromosome consists of different genes, which are discrete states, just as the user choices are. One could use an genetic algorithm to optimize this chromosome according to a properly defined fitness function, that might be adapted to the particular user desire. If the user wants more rules, for example the high-number-of-rules variable in the fitness function could get a higher priority for the calculation of the fitness. With a few PC's that could calculate the fitness of different setups in parallel, an optimization could be achieved, without any contribution of mental capacities. If an extension like this could reach a state of frequent practical use is questionable. Nevertheless, the LISp-Miner environment remains a challenging field for merging the computing power of modern machines with the creative and context-depending data mining focus of a scientific user, see also the project EverMiner [3].

4 References

1.
2. Hájek, P.: (guest editor): *International Journal of Man-Machine Studies*, vol. 10, No 1 (special issue on GUHA), 1978.
3. Rauch, J. - Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Lin, T. Y. - Ohsuga, S. - Liau, C. J. - Hu, X. - Tsumoto, S. (ed.): *Foundations of Data Mining and Knowledge Discovery*. Berlin: Springer, 2005, pp. 211 – 232
4. Rauch, J. - Šimůnek, M.: GUHA Method and Granular Computing. In: HU, Xiaohua, LIU, Qing, SKOWRON, Andrzej, LIN, Tsau Young, YAGER, Ronald R., ZANG, Bo (ed.). *IEEE Granular Computing 2005*. pp. 630–635.

Algebra of Approximate Pattern Matching Problem

Václav Snášel, Marian Mindek

Department of Computer Science, FEI, VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba

{Vaclav.Snasel, Marian.Mindek}@vsb.cz

Abstract. In this paper we classify approximate pattern matching problems using algebraic means. We construct an algebra of pattern matching automata in which finite automata are the elements and operation applied to them correspond to the creation of new pattern matching problems. We present several such operations and describe some identified properties of the algebra defined in this way.

Keywords: approximate pattern matching, finite automata, algebra, Hamming, Levenshtein distance, DIR, DIRT

1 Motivation and Introduction

Exist a number of conventional string matching algorithms. These algorithms are used to finding a certain string in a large body of text. Two of the better known algorithms are the Boyer Moore and the Knuth-Morris-Pratt algorithms [Ba88]. These algorithms are used in the problem of detecting the occurrence of a particular substring, called the pattern, in another string, called the text. The conventional string matching algorithms are only able to match patterns, where they exist completely, that is unbroken and with no irrelevant data in between any letters. This problem occur practically for example in genetics and biochemistry. The ability to match a deoxyribonucleic acid (DNA) pattern into a long DNA sequence is something for which computers have been used and is an essential part of current research. However, many of the resources used in this process are too expensive and generally use a brute force approach to pattern matching.

One of the complications that is faced is that DNA patterns are not always found perfectly in a DNA sequence. The pattern could be split in the sequence by irrelevant and unimportant or mutated data.

Approximate string matching extends the conventional string matching ideas. Here the algorithms need to be able to cope with errors in the strings. This is thus more relevant to the research than conventional matching algorithms. Navarro [Na01] describes the problem as find a pattern in a text, allowing a limited number of errors in the matching. Where m is the length of the pattern, n is the length of the text and k is the maximum number of allowed errors. But there are a number of different methods that have been developed in order to handle errors and we propose a formal description useful for all these problems for better using of known algorithm.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 316–319, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

1.1 Introduction

Melichar and Holub [MH97] also deal with pattern matching problems classification in their work. However, their approach is different. They describe pattern matching problems using 6 criteria and therefore they can locate them in 6D space. They show how to construct a finite automaton for each point of the space. It is possible to start from simple automaton which performs an exact match and transform into a more sophisticated one. This inspired us to create an algebra of finite automata and to define operations with them which would correspond to transformations resulting in different points of the 6D space.

We wanted to construct operations for each of the six axes of the space which would generate automata corresponding to the points already identified. Then we could apply these operations several times and identify the properties of the algebra defined by the set of finite automata and these operations.

2 Definition of the algebra of pattern matching automata

[MH97] introduced construction *R(X, k)*, *DIR(X, k)* and *DIRT(X, k)*. These construction correspond to the creation of a nondeterministic automaton *X'* from the automaton *X* which accepts a string $P=p_1p_2....p_n$. Automaton *X'* accepts only those string *P'* with the value of *P* to *P'* distance equivalent to k while using distance *R*, *DIR* or *DIRT*. Distance *R(P,P')* is called Hamming distance and is defined as the minimum number of symbol replacement operations in string *P* required for the conversion of string *P* into string *P'*. Distance *DIR(P,P')* is called Levenshtein distance and is defined as the minimum number of operations of symbol deletion (*D*), insertion (*I*) or replacement (*R*) in *P* required for the conversion of string *P* into string *P'*. Finally distance *DIRT(P,P')* is called generalized Levenshtein distance and use *DIR* and next operation transpose (*T*).

Automaton *X'* is called *R*-trellis, *DIR*-trellis or *DIRT*-trellis as the case may be. Their construction is described, for example, in [MH97] or [HO96].

When defining the operations in our algebra, we are not restricted to the set of automata which perform an exact match, but we define operations *R*, *DIR* and *DIRT* for any finite automaton *X* in a similar way as Mužátko in his generalization of regular expression matching automata in [MU96]. Since distance *DIR* (*DIRT*) corresponds to any combination of operation of deletion *D*, insertion *I*, or replacement *R* (and transpose *T*), it is possible to define each of these operations independently.

Definition: Let $X = (\Sigma, Q, \delta, q_0, F)$ be a finite automaton and let $k \in N$. The result of operations *D(X, k)*, *I(X, k)*, *R(X, k)*, or *T(X, k)* is an automaton *X'* which will be derived from automaton *X* through the following steps. Automaton X' will contain k+1 clones of automaton X.

1. The states of automaton *X'* will be labeled $q_{i,j}$ where *i* is the sequence number of the clone and *j* is the sequence number of the state inside the original automaton *X*. But if is used operation *T*, new state which is created is labeled before states $q_{i,j}$ and next state is labeled $q_{i,j+1}$.

2. All transitions defined in the original automaton X will remain included in all its clones.
3. Error transitions will be added into automaton X' according to one of the following operations: D,I,R,T
4. Start state of automaton X' is state $q_{0,0}$. The alphabet of the automaton is the extended alphabet Σ'. The set of final states is the union of final states in all the clones: $F=F_0 \cup F_1 \ldots F \cup F_k$.

Definition: Operation $D(X)$, $R(X)$, $I(X)$ and $T(X)$ correspond to operations $D(X,1)$, $R(X,1)$, $I(X,1)$ and $T(X,1)$ respectively.

Now we are ready to define the algebra of pattern matching automata.

Definition: The algebra of pattern matching automata is algebra A = (A, D, I, R, T, *, +, •, ⊕), where A is a set of finite automata, ⊕ is an operation of merger[1], D is an operation of deletion, I is an operation of insertion, R is an operation of replacement, T is an operation of transpose, * is an operation of closure, + is an operation of union, • is an operation of concatenation

2.1 Properties of the algebra of pattern matching automata

We have not defined an equivalent of the *DIR* (*DIRT*) construction in our algebra. It is not really necessary. The corresponding *DIR* (*DIRT*) operation in our algebra will result from suitable application of operation ⊕ to automata $D(X)$, $R(X)$, $I(X)$ and $T(X)$.

The correctness of such a definition ensues from the behavior of operation ⊕. In a similar way is also possible to define the following. $DI(X) = D(X) \oplus I(X)$, $DR(X) = D(X) \oplus R(X)$, $RI(X) = R(X) \oplus I(X)$, $DT(X) = D(X) \oplus T(X)$, ...

In string $w = w_0 r_0 w_1 r_1 \ldots w_k r_k w_{k+1}$, where $r_i \in \Sigma$, is accepted by automaton X, then automaton $DIRT(X,k)$ accepts string $w = w_0 r'_0 w_1 r'_1 \ldots w_k r'_k w_{k+1}$, where $r'_i \in \Sigma'$ (where Σ' over the set Σ as $\Sigma \cup \Sigma^- \cup \{\varepsilon\} \cup \{\Gamma\}$.). If $r'_0 = \Gamma$(whole set), then having read string $w_0 r'_0$ using transition from automaton $I^k(X)$, automaton $DIRT(X,k)$ reaches state $q_{w0r'0}$ which correspond to the state of automaton X after the acceptance of word $w_0 r_0$. Similarly, it is possible to demonstrate that for any segment of string w', automaton $D^k(X) \oplus R^k(X) \oplus I^k(X) \oplus T^k(X)$ reaches the state corresponding to the equivalent segment of string w. It means with that string w' the automaton will be into a final state.

Let us suppose that string w is accepted by automaton X. The string accepted by automaton $D^k(X) \oplus R^k(X) \oplus I^k(X) \oplus T^k(X)$ will contain k+1 segments separated by symbols from set Σ'. It means that this string will be accepted by automaton $DIRT(X,k)$ too.

---

[1] Let A be set of automaton using the same alphabet Σ and the same start state q_0. Then for the automaton $X = (\Sigma, Q^x, \delta^x, q_0, F^x) \in A$ and $Y = (\Sigma, Q^y, \delta^y, q_0, F^Y) \in A$ we will define the automaton $X \oplus Y$ in the following way: $X \oplus Y = (\Sigma, Q^x \cup Q^y, \delta^x \cup \delta^y, q_0, F^x \cup F^y)$
Note: $\delta^x \cup \delta^y$ for our propose means a union of relations.

3 Program

During the phase of building the hypotheses we carried out some experiments using our own program. In this program we implement nondeterministic finite automata and operations with such automata.

We used that method of simulation of the nondeterministic finite automaton in a deterministic way. The usage of symbols from the extended alphabet over the original alphabet prevented the rapid increase of states and transitions which would otherwise become inevitable during multiple application of the operation to the original string searching automaton.

The aim of the implementation was to verify our hypotheses and which was why we didn't it pay any special attention to the effectiveness of the implementation. In case where the speed of the algorithm is one of the main criteria, it is possible to use a different type of implementation as described, for example by Mohri in [MM95].

4 Conclusion

In this paper we classify approximate pattern matching problems using algebraic means. A further step of our research could be extension of the computing power of nondeterministic finite automata through the application of the other models, such as multiple automata, and finding whether our approach are more suitable for approximate pattern matching.

References

[Ba88] S. Baase. *Computer Algorithms: Introduction to Analysis and Design*. Addison Wesley, Massachusetts, second edition 1988, chapter 5.

[Cr03] M. Crochemore, *Reducing space for index implementation,* Theoretical Computer Science, 292-1 2003.

[HO96] J. Holub. Reduced Nondeterministic Finite Automaton for Approximate String Matching. Proceedings of the Prague Stringology Club Workshop 1996.

[MH97] B. Melichar, J. Holub. *6D Classification of pattern Matching Problems*. Proceedings of the Prague Stringology Club Workshop 1997.

[MM95] M. Mohri. *Marching Patterns of an Automaton.* Combinatorial Pattern Matching, 6th Annual Symposium, CPM 95, Espo, Finland, Springer Verlag 1995

[MU96] P. Mužátko. *Approximate Regular Expression Matching*. Proceedings of the Prague Stringology Club Workshop 1996.

[Na01] G. Navarro. *A Guided Tour to Approximate String Matching.* ACM Computing Surveys, 33(1) 2001

OntoFarm: kolekce paralelních ontologií pro experimenty s nástroji sémantického webu

Ondřej Šváb, Vojtěch Svátek

Katedra informačního a znalostního inženýrství, Vysoká škola ekonomická v Praze, nám. W. Churchilla 4, 130 67 Praha 3

xsvao06@vse.cz, svatek@vse.cz

Abstrakt. Sémantické mapování a distribuované odvozování prožívají v současné době v oblasti sémantického webu rozkvět. Velké množství nástrojů si žádá také dostatečné množství vhodných (byť testovacích) dat. V tomto případě se jedná o větší počet ontologií z jedné domény, které by obsahovaly i množství axiomů a neměly jen povrchní strukturu. Vyplnit tuto mezeru je snahou kolekce ontologií OntoFarm, jejíž ontologie mají společnou doménu pořádání konferencí. V tomto příspěvku se věnujeme průběhu vzniku této kolekce, experimentování s dostupnými sémantickými nástroji nad ontologiemi z kolekce a popisu možnosti využití těchto nástrojů pro praktické úkoly předvedenými na prvotní verzi demonstrační aplikace.

Klíčová slova: ontologie, sémantický web, mapování

1 Úvod

Koncepce sémantického webu je do značné míry založena na znovupoužití ontologií jakožto konceptuálních modelů s relativně bohatou formální sémantikou. Distribuovaný a neřízený charakter (nejen běžného, ale i sémantického) webu ovšem předznačuje, že pro tutéž věcnou oblast (doménu) bude zpravidla existovat ontologií větší množství. 'Sémantické' aplikace či agenti budou muset být schopni různorodé modely *integrovat*. Pro tento účel byla v poslední době vyvinuta celá řada nástrojů; přehled mnohých z nich lze nalézt v [3] nebo na http://www.ontologymatching.org. Nástroje lze zhruba rozdělit do dvou hlavních typů. Prvním z nich jsou nástroje pro automatické mapování ontologií, které k sobě přiřazují potenciálně ekvivalentní koncepty (ev. i relace) ze dvou či více ontologií. Druhým jsou nástroje, které na kolekci již (ručně či automaticky) namapovaných ontologií aplikují odvozovací postupy analogické odvozování nad ontologií jedinou; typickým příkladem je distribuované tablové usuzování [4].

Zdá se ovšem, že (v posledních dvou letech až překotný) vývoj mapovacích a odvozovacích nástrojů předbíhá dobu z hlediska dostupnosti doménových ontologií, na které by bylo možné nástroje (byť v testovacím režimu) aplikovat. "Reálné" ad hoc ontologie, např. shromážděné v *DAML repository*[1] či vyhle-

[1] http://www.daml.org/ontologies/

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 320–323, ISBN 80–248–1001–8. VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

dané pomocí vyhledávače *Swoogle*[2] jsou vesměs málo strukturované, nevelké rozsahem, a navíc lze obtížně nalézt větší počet ontologií pro tutéž doménu; k obdobnému závěru dospěla i studie [7]. S 'paralelními' ontologiemi pro tutéž doménu se operuje v aktivitách typu *Ontology Alignment Contest*[3] a *Ontology Alignment Evaluation Initiative*[4]. I tam se však jedná o ontologie s poměrně povrchní strukturou (terminologické taxonomie, webové katalogy), případně o modely vzniklé simulací, které slouží pouze pro 'kalibrační' účely.

Vyplnit tuto mezeru je snahou kolekce ontologií OntoFarm, jejíž ontologie mají společnou doménu pořádání konferencí. V tomto příspěvku se věnujeme průběhu vzniku této kolekce, experimentování s dostupnými sémantickými nástroji nad ontologiemi z kolekce a nástinu možností pro praktické úkoly. Kolekce ontologií v projektu OntoFarm stále vzniká. Materiál v tomto článku zachycuje stav projektu v listopadu 2005; o něco ranější stav byl popsán v [6]. Ve velmi blízké době se počet ontologií více než zdvojnásobí.

2 Tvorba startovní kolekce ontologií

Volba oblasti *pořádání konferencí* pro tvorbu *ontologií* je motivována všeobecnou obeznámeností v komunitě znalostních inženýrů s okolnostmi vážícími se na pořádání konferencí. Neméně důležitým byl pro výběr právě této oblasti i fakt, že pořádání konferencí má několik společných rysů s problematikou *obchodní činnosti*. Máme na mysli především důležitost faktoru času (časové závislosti jednotlivých událostí, vývoj smyslu konceptů v čase), omezování přístupu podle rolí atd. Oblast *pořádání konferencí* se však na rozdíl od oblasti *obchodní* vyznačuje větší dostupností potřebných informací pro zachycení pomocí *ontologií*.

V současné době kolekce obsahuje čtyři *ontologie*. Tři menší (kolem 30 tříd a 30 relací), které vznikly na základě dokumentace a práce s různými *softwarovými nástroji* pro podporu pořádání konferencí. *Ontologie* byly vytvářeny nezávisle, se snahou zahrnout co nejvíce smysluplných *axiomů* odrážejících danou oblast tak, jak ji viděli tvůrci softwarového nástroje. Při tvorbě ontologií byla dodržována doporučení typu zaměření modelování (např. zpracování recenzí, registrace účastníků, kategorie příspěvků apod.) a soustředění se na konceptualizaci dané domény a ne daného nástroje. Čtvrtá ontologie, která patří již mezi středně veliké (80 tříd, 30 relací) s množstvím *axiomů*, byla vytvořena podle *EKAW konference*[5] spolupředsedou jejího příštího konání.

3 Experimenty

Smyslem tvorby kolekce *OntoFarm* je poskytnout pro sémantické nástroje dobrý (řádně zdokumentovaný) a vhodný (větší počet tříd, relací a axiomů) ontologický materiál. V budoucnu plánujeme nad kolekcí zkoušet vlastní nástroje

[2] http://swoogle.umbc.edu/
[3] http://co4.inrialpes.fr/align/Contest/
[4] http://oaei.inrialpes.fr/2005/
[5] http:/ekaw.vse.cz/

(např. automatické mapování ontologií pomocí bayesovských sítí). V současné době z nástrojů, které jsou určeny pro *distribuované odvozování* jsme využívali systém *DRAGO* [5]. Ten umožňuje jednak odvozování nad jednotlivými ontologiemi tzv. lokálně, kdy získáme *lokální taxonomii*, a jednak nad ontologií, která je obohacena prostřednictvím *sémantického namapování* na jinou (či více jiných) *ontologií*. Mapování se skládá ze *zdrojové* a *cílové ontologie* (mapování je jednosměrné z pohledu cílové ontologie) a příslušných pravidel tzv. *bridge rules* (operátory: $\sqsubseteq$, $\sqsupseteq$, $=$). V systému *DRAGO* je implementován přístup, kdy úkol výpočtu *distribuované taxonomie* je výsledkem složení *lokálních taxonomií* (jakožto výsledku *standardního lokálního tablového algoritmu*). Díky sémantickému mapování lze dosáhnout *obohacení struktury ontologie*. Jako příklad lze uvést odvození vztahu podřazenosti mezi koncepty `Chair_PC` a `Member_PC`, který původně v ontologii nebyl obsažen.

Dále jsme nad kolekcí zkoušeli metody obsažené v *Ontology Alignment API* [2]. *Alignment API* nabízí čtyři metody pro mapování entit mezi ontologiemi. Nad kolekcí ontologií *OntoFarm* jsme zkoušeli dvě metody: *EditDistNameAlignment* a *SubsDistNameAlignment*. Mezi automaticky namapovanými dvojicemi bylo v našem případě například i `Chairman = Chair_PC` pro první metodu s mírou jistoty 0,625, pro druhou 0,344 a `title = hasTitle` s 0,625 u první a 0,769 u druhé metody. Výstupní mapování je v jazyce *RDF*, kde jsou uvedeny i míry jistoty. Možné je také zvolit pro serializaci jazyk *OWL*[6] či jazyk *C-OWL* (Context OWL), který je možné přímo použít jako vstup do systému *DRAGO*. Před automatickým mapováním je možné zvolit *prahovou hodnotu* míry jistoty.

4 Demonstrační aplikace

První funkcí *demonstrační aplikace* je nalezení nejvíce podobné ontologie ontologii zadané na vstupu. Vstupní ontologie modeluje konkrétní konferenci a výsledek by tak mohl poskytnout tip na vhodný softwarový nástroj pro právě tuto konferenci. Nynější verze využívá *Ontology Alignment API*, díky které získáme míry jistoty namapovaných dvojic. Z nich pak počítáme 'podobnost' dvojice ontologií jako aritmetický průměr přes jednotlivé míry jistoty. Tento způsob výpočtu ma jisté nedostatky např. upřednostňuje menší ontologie, které pokrývají byť třeba jen malou část ontologie vstupní. Pro možnost porovnávání ontologií se chystáme vyzkoušet i jiné nástroje jako např. *SimPack* [1].

Druhá funkce zahrnuje práci s oběma nástroji (*DRAGO* i *OWL Alignment API*). *Zdrojová ontologie* nám prostřednictvím *sémantického namapování* může dopomoci odhalit možná vylepšení ve struktuře *ontologie cílové* a potažmo v nástroji samém. Nejprve se provede automaticky namapování pomocí *OWL Alignment API*, které vrátí sadu namapovaných dvojic s mírou jistoty vyšší než je na vstupu zadaná prahová hodnota. V druhém kroku se nahrají obě ontologie do systému *DRAGO* společně se sémantickým mapováním, zkontroluje se *konsistence* a odvodí se *lokální* a *distribuovaná taxonomie*, které jsou vypsány na

[6] http://www.w3.org/2004/OWL/

výstup. Pro příští verzi uvažujeme o *automatickém porovnání* těchto taxonomií a upozornění na odchylky a zvláštnosti.

5 Závěr

V současné době je kolekce *OntoFarm* v počátečních fázích. Počet zahrnutých *ontologií* se však v brzké budoucnosti přibližně více než dvojnásobně zvýší. Rovněž tak demonstrační aplikace je ve stádiu zrodu. Tvorba *paralelních ontologií*, které modelují stejnou oblast a obsahují množství *axiomů*, může pomoci v případech testování nástrojů a podpořit tak i jejich další vývoj. Co se týče následujícího vývoje *demonstrační aplikace*, chceme do ní zapojit i další nástroje (v budoucnu i vlastní) se záměrem je využít pro *automatické mapování*.

Poděkování

Projekt je částečně podporován projektem 6. rámcového programu EU (Network of Excellence, IST FP6-507482) "Knowledge Web", a interním grantem Vysoké školy ekonomické v Praze č. 26/05 "Rozvoj metod a nástrojů ontologického inženýrství".

Reference

1. Bernstein A., Kaufmann E., Kiefer C., Bürki C., SimPack: A Generic Java Library for Similarity Measures in Ontologies, August 2005. (techreport).
2. Euzenat J. An API for ontology alignment (version 1.3).
3. Kalfoglou Y., Schorlemmer M., Sheth A., Staab S., Uschold M. (eds.). Proceedings of the Dagstuhl Seminar 04391 on Semantic Interoperability and Integration.
4. Serafini L. and Tamilin A., Local Tableaux for Reasoning in Distributed Description Logics Workshop on Description Logics (DL2004), CEUR-WS, 2004.
5. Serafini L. and Tamilin A., DRAGO: Distributed Reasoning Architecture for the Semantic Web. ESWC'05, Springer-Verlag, 2005.
6. Šváb O., Svátek V., Berka P., Rak D., Tomášek P., OntoFarm: Towards an Experimental Collection of Parallel Ontologies. Poster Session at ISWC 2005.
7. Tempich Ch. and Volz R.. Towards a benchmark for Semantic Web reasoners – an analysis of the DAML ontology library. EON2003 Workshop at ISWC 2003.

Annotation:

OntoFarm: collection of parallel ontologies for experiments with Semantic Web tools

Currently, there is a boom in branches of Semantic Web such as ontology mapping and distributed reasoning. Growing number of tools from these fields also requires sufficient amount of suitable data. In the case of ontology mapping and distributed reasoning, developers are interested in larger amount of ontologies from the same domain. These would be richer in terms of structure, classes, and properties. This is a gap which would be ideally fill up with initiative nicknamed OntoFarm, which aims to grow such a collection. This contribution deals with creation, experiments and some practical demonstration over this colletion.

Privacy Preserving Data Mining

Ondřej Výborný

KD Group, Faculty of Informatics, Masaryk University in Brno,
Botanická 68a, 60200, Brno, Czech Republic

xvyborny@fi.muni.cz

Abstract. Because huge amounts of data are available nowadays, data mining is more important for dealing with it than ever before. However, a large part of that data is sensitive and cannot be processed without taking privacy into account. This issue is not easy to solve, but very important. We must avoid the misuse of private data from mobile phones, bank accounts, etc. We also need tools for legal processing of such data, because in matters of privacy in laws there are very strict requirements. We present here an overview of what privacy in data mining means. We also show common protocols used in data mining that preserves privacy and issues that are important in current research.

1 Introduction

Nowadays we can (and have to) process more and more data that is almost ubiquitous. Because of this phenomenon data mining (DM) is taking more important role within information technology. But the pure DM is often not enough for solving situations when dealing with the sensitive data (bank accounts, data from mobile phone networks, etc.). We have to preserve the privacy and the security of the information, of the owner of such information and also of the data miner. When dealing with sensitive data, many problems are caused by laws and their strict requirements for processing such data. Considering that we have to use the privacy preserving data mining (PPDM [3]).

In the first section, we discuss the meanings of a term *privacy* in the DM and in the second section we show some basic protocols used in the PPDM. Then we outline current issues of research in the PPDM and finally we make conclusions with respect to possible future of the PPDM and our own research[1].

2 Privacy in Data Mining

Privacy (from the data mining point of view) is not a simple feature. It has many different aspects which have to be considered in different situations. There are three basic views on privacy, which we will discuss in this section.

[1] This work has been partially supported by the Grant Agency of the Czech Republic under the Grant No. MSM0021622418.

© J. Paralič, J. Dvorský, M. Krátký (Eds.): Znalosti 2006, pp. 324–327, ISBN 80–248–1001–8.
VŠB–Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2006.

2.1 Data Point of View

There are two basic ways of ensuring privacy of data in the data mining process. The first way is to exclude sensitive data from the database (DB) before we do the DM. This could be done by modifying or trimming the sensitive raw data in the DB. There are several techniques, such as perturbation (adding noise), blocking (deleting data from the DB), aggregation (or merging of parts of the DB), swapping (some values are simply swop) or sampling (releasing only part of the DB for the DM) that are used for such data modification.

The second way is to exclude sensitive knowledge that could be possibly mined from the DB. In this case we may want to exclude only a part of the mined information or the whole result, containing sensitive data. The second approach is much more difficult, because we need to find all possible relations between the data in our DB that can lead to revealing the sensitive data to user and exclude all data necessary to avoid this leakage (and this is not an easy task).

2.2 Owners Point of View

When we have a DB and we want to offer its data for DM purposes, there are two possibilities. Firstly, we can make the whole DB available for users (we do not need to keep the data secret). In this case we have no problems with privacy and users can do almost anything they want with the data. Secondly, we may need to keep the data secret but still offer the possibility of doing the DM (this means that any user can do the DM on our data, but the DB itself is not available [6]). This approach is reasonable when, e.g., we want to charge users for every access to our DB (because this cannot be done with the former approach, when the DB is available for a free use).

2.3 Users Point of View

When users want to do DM on a DB they do not own, they may not want to reveal their queries to the DB owner (this is in some cases also sensitive information - imagine that you want to search some medical DB for information about your health without showing your queries to other people).

Another issue is the anonymity of users. In some cases, we may want to be anonymous. This issue is connected more generally to security than to PPDM so we will not deal with it here.

3 Basic Protocols

Current PPDM approaches are focused mainly on cryptography-based techniques, which are based on Secure Multiparty Computations (SMC). SMC is a computation between 2 or more parties based on their inputs. The goal is to conduct such computation without revealing anything more than the result and

the particular input for each party. In this way, we can compute, e.g., secure sum, size of intersection, union or scalar product.

There also exist other approaches for solving some PPDM problems such as heuristic-based techniques (adaptive modifications, minimization of utility loss, etc.) or reconstruction-based techniques (reconstruction of original data distribution from the randomized data). We think that these techniques are useful only for some special problems and will not describe them here minutely.

Many algorithms used nowadays are built on several basic protocols, such as Oblivious Transfer Protocol (OTP) or Private Information Retrieval (PIR) and we will briefly show their main ideas here.

3.1 OT protocol

Oblivious Transfer protocol is a secure protocol between two parties. One party (the server) holds a secret bit b and the second party (the user) can, at the end of the computation, learn this bit b with probability $1/2$ or learn nothing. The main thing is that the server has no knowledge about which of these two events has happened and the user does not know anything about other bits in the DB. This is a simple example of 1-out-of-2 OT protocol. There are also other modifications (1-out-of-N OT, distributed OT, etc.) of this protocol that are used for solving various security problems.

3.2 PIR

Private Information Retrieval protocols [2] allow users to retrieve information from a DB while keeping their query private. It can be seen that these protocols are similar to OT protocols. The difference is that in the PIR protocol a user can learn more values than just the one requested. A trivial example of the PIR protocol is sending the whole DB to the user. However, this approach has the communication complexity $c(n) = n$, where n is the size of the DB. A PIR protocol with $c(n) < n$ is called *non-trivial* and lots of such protocols are designed using several identical DBs (a user makes different requests on such databases and reconstructs the desired information from results).

4 Current PPDM

A large area aimed at by researchers nowadays is improving the efficiency of current privacy preserving algorithms, in particular the communication cost, which is a bottleneck for many PPDM algorithms [1]. There is also a need for improving current algorithms in other ways. For example Yang, Zhong and Wright in [8] propose the privacy preserving classification of customer data without lost of accuracy, which is a very common drawback of PPDM algorithms.

PPDM can also be useful for processing the data from mobile phone networks [5] that are a huge source of information suitable for the DM.

Another area of current research is distributed PPDM [4]. This is a field in which we will always find new challenges (this could be caused by its close connection to the network research).

There are also attempts to adapt PPDM for stream data. This is useful when dealing with large data (network flows, mobile phone data, etc.). Ostrovsky and Skeith in [7] developed a novel way for secret collecting of certain documents in a stream of documents.

5 Conclusions

PPDM as a subject of research is merging knowledge and methods from data mining and security (cryptography). This is a very useful connection because real life issues affect privacy with respect to data more often than ever before. Strictness of laws and requirements on tools used for manipulating private data (bank accounts, medical data, mobile phone data, etc.) are very high and tools that can satisfy them are in demand. This implies that research in this area is quite forward-looking. We can also see a big need for implementations of current PPDM algorithms and for their integrating into real life applications.

In our future work, we want to focus on the utilization of PPDM techniques for new types of data, such as data from mobile phone operators and on possible use of secure environments, such as Java Cards, for solving privacy issues related to the DM. We are also interested in use of DM for discovering possible threats in the security area (network flows, user behaviour, spam, etc.).

References

1. Agrawal S., Krishnan V. and Haritsa J. R. On Addressing Efficiency Concerns in Privacy-Preserving Mining. *In Proceedings of the 9th International Conference on Database Systems for Advanced Applications (DASFAA-2004).*
2. Beimel A. Private Information Retrieval. `http://www.cs.bgu.ac.il/~beimel/Research/PIR.html`
3. Clifton C. Privacy Preserving Data Mining. Tutorial, *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 24, 2003, Washington, D.C.
4. Clifton C., Kantarcioglu M., Vaidya J., Lin X. and Zhu M. Y. Tools for Privacy Preserving Distributed Data Mining. *In SIGKDD Explorations, 4(2)*: 28-34 December 2002.
5. GeoPKDD - Geographic Privacy-aware Knowledge Discovery and Delivery. `http://www.di.unipi.it/~rinziv/prin/index.php`
6. Laur S., Lipmaa H. and Mielikäinen T. Private Itemset Support Counting. *In proceedings of ICICS '05*, volume 3783 of LNCS, pages 97–111, Beijing, China, 2005. Springer-Verlag.
7. Ostrovsky R. and Skeith W. Private Searching On Streaming Data. *25th International Cryptology Conference (Crypto 2005).*
8. Yang Z., Zhong S. and Wright R. N. Privacy-Preserving Classification of Customer Data without Loss of Accuracy. *In Proceedings of the Fifth SIAM International Conference on Data Mining*, Newport Beach, CA, April 21-23, 2005.

Rejstřík autorů (Author Index)

Aubrecht, Petr, 37

Babič, František, 260
Babik, Marian, 48
Balogh, Zoltán, 151, 288
Bednár, Peter, 159, 191
Beránek, Ladislav, 264
Berka, Petr, 183
Bieliková, Mária, 237
Bína, Vladislav, 167
Blaťák, Jan, 130
Bundzel, Marek, 60
Burda, Michal, 175
Bureš, Vladimír, 1, 268
Butka, Peter, 71

Čech, Pavel, 1

Dolog, Peter, 6
Dvorský, Jiří, 83
Džbor, Martin, 13

Gatial, Emil, 151, 288
Gurský, Peter, 95

Habala, Ondrej, 48
Habiballa, Hashim, 106
Hluchý, Ladislav, 48, 151, 288

Jelínek, Jiří, 272, 276

Kasanický, Tomáš, 60
Kejkula, Martin, 37
Keprt, Aleš, 280
Kliegr, Tomáš, 284
Kosek, Jiří, 5
Kováč, Michal, 118
Křemen, Petr, 37
Kuchař, Tomáš, 118
Kunčar, David, 276
Kuzmin, Alexander, 118

Laclavík, Michal, 151, 288
Laš, Vladimír, 183

Mach, Marián, 292
Machová, Kristína, 191
Martinovič, Jan, 83, 199
Maták, Valentín, 191
Matyska, Luděk, 233
Mikulecký, Peter, 296
Mindek, Marian, 316
Mládková, Ludmila, 207
Mls, Karel, 300
Monika Žáková, 37
Müller, Jaroslav, 304

Návrat, Pavol, 237
Nováková, Lenka, 37
Novosád, Tomáš, 199

Ponce, Daniela, 215
Popelínský, Luboš, 130
Přibil, Jiří, 276

Radvanský, Martin, 139
Ralbovský, Martin, 118
Rauber, Andreas, 26
Rauch, Jan, 37

Sarnovský, Martin, 308
Schutte, Sebastian, 312
Sklenák, Vilém, 248
Sklenář, Vladimír, 139
Smrž, Pavel, 252
Snášel, Václav, 83, 199, 316
Sojka, Petr, 243
Strossa, Petr, 248
Svátek, Vojtěch, 256, 320

Šefránek, Ján, 29
Šimůnek, Milan, 37
Štěpánková, Olga, 37
Šváb, Ondřej, 320

Vlček, Lukáš, 223
Výborný, Ondřej, 324

Editoři:	Ján Paralič, Jiří Dvorský, Michal Krátký
Název:	Znalosti 2006, sborník příspěvků
Místo, rok, vydání:	Ostrava, 2006, 1.
Počet stran:	344
Vydavatel:	VŠB–Technická univerzita Ostrava, 17. listopadu 15, 708 33 Ostrava–Poruba
Tisk:	TiskServis Jiří Pustina Petra Křičky 24, 702 00 Ostrava 1
Náklad:	110 kusů

Neprodejné.