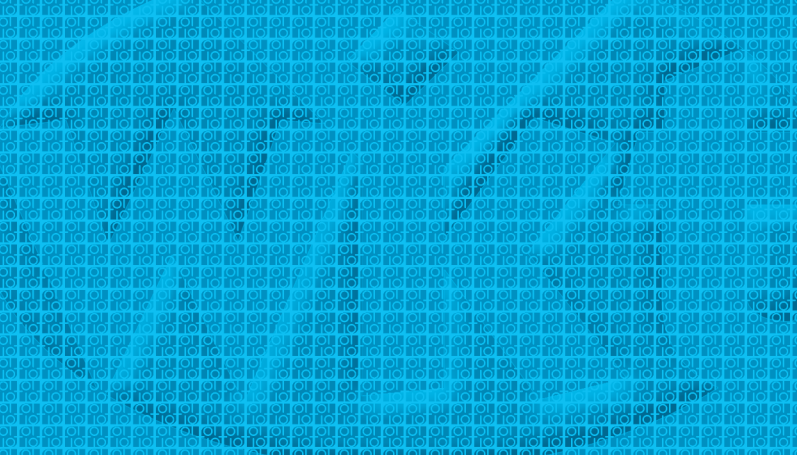
Proceedings in
Informatics and Information Technologies

7th Workshop on Intelligent and Knowledge Oriented Technologies

Mária Bieliková, Marián Šimko (Eds.)



STU
FIIT

Proceedings in
Informatics and Information Technologies

WIKT 2012

7th Workshop on Intelligent and Knowledge Oriented Technologies

Mária Bieliková, Marián Šimko (Eds.)

WIKT 2012

7th Workshop on Intelligent
and Knowledge Oriented Technologies

Proceedings

November 22-23, 2012
Smolenice, Slovakia

WIKT 2012 was organized by:

- Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava
- Centre for Information Technologies, Faculty of Electrical Engineering and Informatics, Technical University of Košice
- Institute of Informatics, Slovak Academy of Sciences, Bratislava
- Informatics Development Foundation

Editors

Mária Bieliková, Marián Šimko

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
{name.surname}@stuba.sk

The workshop was supported by

- The Slovak Research and Development Agency under the contract No. APVV-208-10, TraDiCe – Cognitive travelling in digital space of the Web and digital libraries supported by personalized services and social networks
- The Slovak Research and Development Agency under the contract No. APVV-0233-10, RIOT – Virtual and constructive modelling and simulation of crowd behaviour in urban environment
- IBM Faculty Award, Innovative Education of Business Analytics for Students, Technical University of Košice

© 2012 The authors listed in the Table of Contents
All contributions reviewed by the Programme Committee and printed as delivered by authors without substantial modifications

Visit WIKT 2012 on the Web: http://wikt2012.fiit.stuba.sk

Executive Editor: Marián Šimko
Copy Editors: Katarína Mršková, Michal Kompan
Cover Designer: Peter Kaminský

Published by
Nakladateľstvo STU
Vazovova 5, Bratislava, Slovakia

ISBN 978-80-227-3812-5

Preface

Intelligent and knowledge-oriented technologies currently affect various areas of human lives. They form an important component of research activity of several research groups active in Slovakia and the Czech Republic. They also constituted a subject of presentations and discussions in the Smolenice Castle, where the Workshop on Intelligent and Knowledge oriented Technologies WIKT was held from the 22nd to the 23rd of November 2012.

This year followed the tradition started in 2006, at the Institute of Informatics, at Slovak Academy of Sciences in Bratislava. A series of workshops during the last six years fostered the creative environment and research by making a forum for exchanging knowledge and creative discussions in the field of intelligent and knowledge oriented technologies in Slovakia. The aim of the workshop WIKT was always to bring together researchers from several research centres in Slovakia and vicinity. After several meetings in Bratislava, Košice, Smolenice and Herľany, WIKT returned to Smolenice.

Main topics of WIKT 2012 workshop were:

- knowledge technologies and their applications,
- information and knowledge modelling, semantic representation,
- analysis and processing of information sources,
- social web and its applications, analysis of social networks,
- personalized web and its applications, recommendation,
- processing of information sources in Slovak language,
- semantic and service oriented architecture,
- reasoning and inference.

Authors sent their contributions in the form of extended abstracts (in Slovak, Czech and English) of the following types:

- research contribution,
- work-in-progress,
- visionary contribution,
- knowledge practices,
- lessons and experiences,
- application paper.

The workshop posed a special challenge for doctoral students who could offer a contribution related to the direction and goals of their dissertation.

WIKT 2012 began with invited talk on Methodological pitfalls of information sciences research by a team around Prof. Steinerová and Prof. Šušol from the

Department of Library and Information Science of Comenius University in Bratislava. The invited talk brought a new perspective to the understanding of cognition and information processing in comparison with how it is perceived in the workplaces with roots in computer science and information technology. It also followed up on the meeting of TraDiCe project (Cognitive travelling in digital space of the Web), which preceded the workshop WIKT.

The workshop WIKT reaffirmed its significance this year, in its seventh edition, again. A total of 47 papers were submitted, most of them as research paper or work-in-progress. 6 papers were submitted to the doctoral section. Each contribution was reviewed by two members of the program committee. The result of the assessment was acceptance of 45 papers in total. 20 of them were accepted for standard presentation in four sections, 19 for short presentations and 5 for doctoral section.

Majority of the papers are written in the native language of the authors, i.e., in Slovak or Czech. The language of the workshop was Slovak and Czech. This fact on the one hand limits the dissemination of the results, but on the other hand it helps in growing professional language skills in the domain of rapidly developing knowledge and web technologies.

25 hours of intensive communication with little sleep. That is how the WIKT can be characterized. In this scale "standard presentation" is a 5 minute long presentation of main ideas in a series of presentations followed by a controlled discussion on all papers in the section, and in a moment followed by an uncontrolled discussion accompanied by delicious drinks and breath taking environment of the Smolenice castle with its nooks – tailor-made for late night discussions. Short presentations were in the form of one minute notification of key ideas, which are further discussed in the section entitled *Discussion club* lasting until the early morning hours. Specific position was assigned to the doctoral section. We are glad that we could provide space for PhD students to consult their research plans with experts from several reputable research centres.

Interest in a workshop WIKT in terms of contributions was a record this year. And not only that. We are very pleased that this year Smolenice Castle was also a record in number of participating research groups from Slovakia and the Czech Republic.

We thank all the authors for interesting contributions initiating fruitful debates. We thank the members of the program committee, who willingly participated in the judging of submissions and discussions about the direction of the workshop. We also thank them for the contribution to the maintenance of high professional level of the event and the fact that they came to the workshop with their research groups. We thank all the members of the organizing committee, who made a considerable effort to turn a picturesque spot in the heart of Central Europe into a two day passionate scientific debate centre and helped to spread the knowledge and collaboration.

Bratislava, November 2012

Mária Bieliková and Marián Šimko

Workshop Organisation

The 7th Workshop on Intelligent and Knowledge Oriented Technologies (WIKT), held on November 22-23, 2012 in Smolenice, was organised by the Slovak University of Technology in Bratislava (and, in particular, its Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies) in collaboration with Centre for Information Technologies, Faculty of Electrical Engineering and Informatics, Technical University Košice and the Institute of Informatics, Slovak Academy of Sciences, Bratislava under considerable support of Informatics Development Foundation.

Programme Committee

Chair

Mária Bieliková, Slovak University of Technology in Bratislava

Members

Ladislav Hluchý, Institute of Informatics, Slovak Academy of Sciences
Daniela Chudá, Slovak University of Technology in Bratislava
Stanislav Krajči, University of P.J.Šafárik Košice
Michal Laclavík, Institute of Informatics, Slovak Academy of Sciences
Marián Mach, Technical University of Košice
Kristína Machová, Technical University of Košice
Pavol Návrat, Slovak University of Technology in Bratislava
Ján Paralič, Technical University of Košice
Viera Rozinajová, Slovak University of Technology in Bratislava
Petr Šaloun, Technical University, Ostrava
Ján Šefránek, Comenius University, Bratislava
Peter Vojtáš, Charles University, Prague

Organizing Committee

Chair

Marián Šimko, Slovak University of Technology in Bratislava

Members

František Babič (Technical University of Košice),
Michal Holub, Michal Kompan, Tibor Krajčovič, Katarína Mršková, Alexandra Zakálová (all from Slovak University of Technology in Bratislava)

Predhovor

Inteligentné a znalostne orientované technológie ovplyvňujú v súčasnosti najrôznejšie oblasti ľudskej činnosti. Tvoria aj významnú zložku náplne činnosti viacerých výskumných skupín pôsobiacich na Slovensku a v Česku. Tvorili aj hlavnú tému prezentácií a diskusií na Smolenickom zámku 22. – 23. novembra, 2012, kde sa konala tvorivá pracovná dielňa o inteligentných a znalostne orientovaných technológiách WIKT 2012.

Tento ročník nadviazal na tradíciu započatú v roku 2006 na Ústave informatiky Slovenskej akadémie vied v Bratislave. Séria pracovných dielní počas šiestich rokov vytvorila tvorivé prostredie pre podporu výskumu najmä prostredníctvom výmeny poznatkov a tvorivých diskusií v atraktívnych oblastiach inteligentných a znalostne orientovaných technológií na Slovensku. Snahou dielne WIKT vždy bolo spájať výskumníkov viacerých výskumných centier v širšom zábere Slovenska. Po niekoľkonásobných stretnutiach v Bratislave, Košiciach, Smoleniciach a Herľanoch sa vrátil WIKT znovu do Smolenít.

Hlavné témy dielne WIKT 2012 boli:

- znalostné technológie a ich aplikácie,
- modelovanie informácií a znalostí, reprezentácia sémantiky,
- analýza a spracovanie informačných zdrojov,
- sociálny web a jeho aplikácie, analýza sociálnych sietí,
- personalizovaný web a jeho aplikácie, odporúčanie,
- spracovanie informačných zdrojov v slovenskom jazyku,
- sémanticky a servisne orientované architektúry,
- usudzovanie a odvodzovanie.

Autori zasielali príspevky v tvare rozšíreného abstraktu v slovenskom, českom alebo anglickom jazyku nasledujúcich kategórií:

- výskumný príspevok,
- work-in-progress,
- vizionársky príspevok,
- znalostné praktiky,
- ponaučenia a skúsenosti,
- aplikačný príspevok.

Špeciálnu výzvu mali študenti tretieho stupňa okolo dizertačnej skúšky, ktorí mohli ponúknuť príspevok o smerovaní a cieľoch svojej dizertačnej práce.

WIKT 2012 začal pozvanou prednáškou na tému Metodologické úskalia výskumov informačných vied kolektívu autorov okolo prof. Steinerovej a prof. Šušola

z Katedry knižničnej a informačnej vedy Filozofickej fakulty Univerzity Komenského v Bratislave. Pozvaná prednáška vniesla nový pohľad do chápania práce s informáciami v porovnaní s tým ako sa vníma na pracoviskách s koreňmi v informatických vedách a informačných technológiách. Zároveň nadviazala na pracovné stretnutie k projektu TraDiCe (Cognitive travelling in digital space of the Web), ktoré predchádzalo dielni WIKT.

Tvorivá dielňa WIKT v tomto roku, v roku svojej siedmej edície, znovu potvrdila svoje opodstatnenie. Celkovo bolo ponúknutých 47 príspevkov, väčšina z nich v kategórii výskumný príspevok alebo work-in-progress. 6 príspevkov bolo prihlásených do doktorandskej sekcie. Každý príspevok posúdili dvaja členovia programového výboru. Výsledkom posudzovania bolo rozhodnutie o prijatí 45 príspevkov, z ktorých 20 bolo prijatých na štandardnú prezentáciu v štyroch sekciách, 19 na krátke prezentácie a 5 do doktorandskej sekcie.

Väčšina príspevkov je napísaná v materinskom jazyku autorov, teda slovensky, resp. česky. Jazyk tvorivej dielne bol slovenský a český, čo síce na jednej strane ohraničuje šírenie výsledkov, ale na strane druhej pomáha pestovaniu odborného jazyka v doméne rýchlo rozvíjajúcich sa znalostných a aj webových technológií.

25 hodín intenzívnej komunikácie s trochou spánku. Aj tak možno charakterizovať WIKT. V tejto mierke „štandardná prezentácia“ znamená krátku 5 minútovú prezentáciu série príspevkov – hlavných myšlienok z nich, nasledovanú riadenou diskusiou, a potom neriadenou diskusiou sprevádzanou príjemnými nápojmi a nádherným prostredím Smolenického zámku so svojimi zákutiami ako stvorenými pre diskusie. Krátka prezentácia bola vo forme minútových oznámení o kľúčových myšlienkach, ktoré sa ďalej diskutovali v rámci sekcie nazvanej Diskusný klub trvajúcej do skorých ranných hodín. Špecifické postavenie mala doktorandská sekcia. Sme radi, že sme mohli poskytnúť doktorandom priestor pre konzultovanie svojich výskumných zámerov s odborníkmi z viacerých renomovaných pracovísk.

Záujem o tvorivú dielňu WIKT v podobe ponúknutých príspevkov bol tento rok rekordným. A nielen to. Sme veľmi potešení tým, že tento rok na Smolenickom zámku bol aj rekordný počet výskumných skupín zo Slovenska a Česka.

Ďakujeme všetkým autorom za zaujímavé príspevky podnecujúce diskusiu. Ďakujeme členom programového výboru, ktorí ochotne participovali na posudzovaní príspevkov a diskusiách o smerovaní tvorivej dielne. A tiež za príspevok k udržaniu vysokej odbornej úrovne celého podujatia aj tým, že na pracovnú dielňu prišli aj so svojimi výskumnými skupinami. Zároveň ďakujeme všetkým členom organizačného výboru, ktorí vynaložili nemalé úsilie na to, aby sa na dva dni jedno malebné miestečko v srdci strednej Európy premenilo na zanietené vedecké diskusie a pomohlo tak v šírení poznatkov a spolupráci.

Bratislava, november 2012

Mária Bieliková a Marián Šimko

Obsah

Pozvaná prednáška

Prepojené dáta

Analýza a spracovanie textu

Model používateľa a spätná väzba

Analýza a modelovanie dát

Smerovanie dizertačných projektov

Diskusný klub

Metodologické úskalia výskumov informačných vied

Jela Steinerová, Jaroslav Šušol, Miriam Ondrišová, Jana Ilavská

Katedra knižničnej a informačnej vedy
Filozofická fakulta, Univerzita Komenského v Bratislave,
Gondova 2, 814 99 Bratislava
{steinerova,susol,ondrisova}@fphil.uniba.sk, ilavska@rec.uniba.sk

Abstrakt. V kontexte základných metodologických prístupov informačnej vedy (kvantitatívne a kvalitatívne, filozoficko-analytický, kognitívny, sociálno-doménový, pluralizmus) sa predstavujú metódy kvalitatívnych prístupov v informačnej vede. Zdôrazňuje sa premyslená koncepcia výskumu, vymedzenie objektu výskumu, výber premenných a vzorky na výskum, etika výskumu. Uvádzajú sa príklady vlastných výskumov realizovaných kvalitatívnymi prístupmi – informačné správanie, relevancia, informačná ekológia. Predstavuje sa metodologická koncepcia nového výskumu so zameraním na doktorandov a ich interakciu s informačnými systémami. Z hľadiska kognitívnych základov informačného správania sa zdôrazňuje význam pojmového mapovania a nástrojov na organizáciu informácií vo vzdelávaní a výskume doktorandov.

Kľúčové slová: metodológia výskumu, kvalitatívne prístupy, informačné správanie, relevancia, informačná ekológia, pojmové mapovanie, informačný model doktoranda

1 Úvod

Metodológia výskumu predstavuje systém teoretických princípov a praktických metód umožňujúcich výskumníkom realizovať výskumný projekt. Všeobecne metodológia určuje uhol pohľadu pri plánovaní výskumu a formulovaní výskumného problému. To sa týka tak tradícií experimentálnych informatických výskumov ako aj sociálnopsychologických tradícií výskumov v informačnej vede.

V informačných vedách sa metodológia výskumu používa ako nástroj na uchopenie objektov reality. Umožňuje formulovať problém a otázky. Obsahuje metódy a postupy súvisiace so získavaním údajov (napríklad meranie, testovanie, pozorovanie, prieskumy), analýzami, syntézami, modelovaním. Môže pritom ísť o exploráciu (napríklad čo je web 2.0), deskripciu (koľko používateľov využíva funkcie určitej webstránky) alebo explanáciu (prečo ľudia využívajú určité elektronické zdroje) či predikciu.

Informačné vedy sú príkladom pluralistického vývoja metodologických tradícií. Využívanie informácií sa od 50. rokov 20. storočia spájalo s nástrojmi výpočtovej techniky. Aj na metodologickej úrovni sa rozvinula systémová (technologická) para-

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 1–12.

digma a používateľská paradigma. Systémová paradigma predpokladala existenciu objektívneho poznania (nadväzovala na pozitivistické tradície filozofie) a riešila najmä problémy prístupu k informáciám a nadväzné funkcie informačných systémov. Používateľská paradigma sa zameriava na kognitívne a sociálne procesy pri spracovaní informácií a informačné správanie. V súčasných prístupoch sa tieto tradície prepájajú práve v zameraní na zložitosť sociotechnických systémov a prepájanie človeka a systémov v rôznych inteligentných funkciách spracovania informácií. Príkladom sú výskumy sociálnej informatiky.

V tomto príspevku stručne vysvetlíme metodologické prístupy informačných vied a naznačíme výhody aj problémy kvalitatívnych prístupov na príklade vlastných výskumov. Predstavíme novú koncepciu modelovania interakcie doktorandov s informačnými systémami a elektronickými zdrojmi.

2 Základné metodologické prístupy informačných vied

Metódy výskumu sa tradične delia na kvantitatívne a kvalitatívne prístupy. Kvantitatívne prístupy využívajú kvantitatívne metódy pri získavaní aj analýzach informácií. Najčastejšie sú to štatistické analýzy údajov z dotazníkových prieskumov, z transakčných logov vo webových službách a systémoch, prípadne bibliometrické analýzy dát z databáz. Výsledky sa často prezentujú v tabuľkách, grafoch a modeloch, validita sa overuje matematickými testami, odkrývajú sa tendencie vývoja. Kvalitatívne prístupy sledujú skúmaný objekt, najčastejšie človeka pri vyhľadávaní a využívaní informácií, z hľadiska konkrétnej situácie. Snažia sa pochopiť javy a udalosti. Výsledkom sú interpretácie, kategórie textov, opisy, príbeh. Validita sa zabezpečuje triangulóciou či konsenzom expertov. Príkladom kvalitatívnych metód sú prípadové štúdie, diskurzná analýza, obsahová analýza, historická analýza, denníky ai. Zvyčajne sa tieto prístupy navzájom kombinujú s dôrazom na riešenie samotného problému. Väčšina problémov má transdisciplinárny charakter (napríklad, čo je relevancia informácií, aké reprezentácie poznania sú efektívne pre inteligentný systém, ako sa ľudia správajú pri využívaní informácií). Často ide o problémy komplexné, preto triedenie metodologických prístupov nie je jednoduché. Napríklad M. Batesová [1] delí metódy výskumov informačného správania podľa typu poznania a cieľov výskumu na „nomotetické“ a „ideografické“. Nomotetické majú cieľ vysvetľovať a predvídať javy a objekty, hľadajú všeobecné zákonitosti, princípy a vzorce, približujú sa metodologickým tradíciám prírodných a technických vied. Príkladom sú systémový prístup, bibliometria, používateľsky orientovaný dizajn systémov. Ideografické prístupy sa snažia odhaliť súvislosti, tendencie, jedinečné fakty, zvláštnosti a budujú na tradíciách humanitných a sociálnych vied. Príkladom môžu byť sociokognitívne prístupy a prístupy sociálnej informatiky (včleňovanie informačných systémov do sociálnych štruktúr a vzťahov).

T. Wilson [28] navrhuje rozlíšiť metodologické prístupy deterministické (koncepcia a štruktúra výskumu je daná vopred) a emergentné (metódy sa vyvíjajú spolu so samotným výskumným procesom). Z hľadiska vzťahu k informačným technológiám sa rozlišujú také prístupy ako technologický determinizmus, humánny determinizmus,

sociálny determinizmus a multidimenziálnosť. Technologický determinizmus predpokladá zlepšovanie informačných procesov prostredníctvom inteligentných technológií, iné prístupy zdôrazňujú úlohu človeka, sociálnych štruktúr alebo komplexitu pohľadov na informačnú spoločnosť.

Metodologické tradície informačnej vedy sa tiež rozčleňujú na filozoficko-analytické, kognitívne, sociálne a pluralistické. Filozoficko-analytické prístupy uplatňujú kritickú reflexiu a analýzy pri vysvetľovaní základných pojmov a koncepcií informácií. Kognitivizmus sa zameriava na myšlienkové a afektívne procesy človeka pri spracovaní a využívaní informácií a v praxi často smeruje k zlepšovaniu rozhraní systémov. Sociálne prístupy zdôrazňujú význam pojmov, dokumentov a domén ako spoločných priestorov pri spracovaní a využívaní informácií. Informácie sú interpretované ako sociálna konštrukcia a rekonštrukcia. V praxi sa to prejavuje v terminológii a nástrojoch organizácie poznania. Pluralizmus počíta s mnohonásobnými prístupmi pri hľadaní významov informácií a ich využívaní.

Z hľadiska výskumných metód sa najčastejšie vyskytujú prieskumy rôzneho typu, napríklad dotazníkové prieskumy, štruktúrované a pološtruktúrované rozhovory, skupinové rozhovory, cielené ťažiskové skupiny, Delphi štúdie. Často sa začínajú pilotnými štúdiami. Z iného pohľadu sa delia výskumné metódy na experimentálne, založené na pozitivistických tradíciách, a pozorovania a interpretácie založené na tradíciách etnografického výskumu. Dôležité sú aj prístupy hodnotenia zdrojov, metódami môže byť napríklad informačný audit, štúdie vplyvu, hodnotenie služieb ako napríklad heuristická evaluácia rozhraní prieskumových systémov, metóda hlasného myslenia, metóda analýzy sociálnych sietí a iné. V informačnej vede sa často využívajú aj bibliometrické a webometrické metódy založené na kvantitatívnych analýzach dokumentov z hľadiska tém, autorov, inštitúcií, krajín naznačujúce aj prepojenia medzi dokumentmi, citácie či vzorce a tendencie vývoja ap. Veľmi častými metódami výskumu vo webovom prostredí a skúmaní informačných systémov sú analýzy transakčných logov (TLA), analýzy interakcií, monitorovanie interakcií. V kvantitatívnych výskumoch často chýba kontext a hlbší pohľad na motiváciu informačného správania a praxe.

3 Kvalitatívne metódy výskumu

Metódy výskumu v informačnej vede majú svoje tradície, existuje viacero metodologických publikácií [2, 5, 27]. Aj pri výskumoch v menšom rozsahu autori zdôrazňujú význam dôkladnej prípravy koncepcie, výber premenných a vzorky na skúmanie a prípravné analýzy predchádzajúcich podobných výskumov.

Kvalitatívne metódy sa vyznačujú uplatňovaním prístupov využívaných v kultúrnej antropológii, etnografii a fenomenológii. Zaujímavým kvalitatívnym prístupom je fenomenografia, ktorá zdôrazňuje interpretácie situácie z hľadiska skúmaného subjektu a pochopenie čo najvyššieho stupňa kontextu. Medzi kvalitatívne výskumy môžeme zaradiť aj prehľady literatúry a analýzy internetových zdrojov.

Dôležitá je kritická analýza predchádzajúcich výsledkov výskumu. V nej sa uplatňujú také metódy ako konceptuálna (filozofická) analýza, historická analýza, obsa-

hová analýza (hodnotenie obsahu dokumentov, pojmy a kategórie), diskurzná analýza (spôsob použitia jazyka v určitých situáciách, skupinách, témach), kognitívne modelovanie. Osobitne sa vyčleňujú meta-analýzy a meta-syntézy kombinujúce výsledky kvantitatívnych či kvalitatívnych výskumov. Úskalia kvalitatívnych metód súvisia najmä so subjektívnosťou interpretácií či validitou výskumu.

Kvalitatívne metódy sa zameriavajú na pochopenie objektov, javov a udalostí. Vo výskume je preto dôležitá interpretácia, kategórie, opisy, (životné) príbehy. Validita sa dosahuje expertným posudzovaním, trianguláciou, viacnásobnými analýzami viacerých expertov. Tu treba pripomenúť, že informačná veda sa často vymedzuje ako sociálna veda s dôrazom na sociálne interakcie pri využívaní informácií. Napríklad sa skúma informačné správanie z hľadiska sociálnych faktorov, ale aj jazyk a kultúra informačných inštitúcií. Poznatky v informačných službách a systémoch majú byť sociálne konštruované, pravdivé a overené informácie (pohľad sociálnej epistemológie Jesseho Sheru). Základnou epistemologickou dichotómiou je pritom určenie subjektívneho a objektívneho v poznaní a určenie vzťahov technológií a sociálnych interakcií. Niektoré metodologické aspekty sa opierajú o východiská subjektívneho idealizmu, objektívneho idealizmu a realizmu. Možným riešením na metodologickej úrovni je vymedzenie informácie na rôznych úrovniach integrácie (biologická, kognitívna, sociálna).

Kvalitatívne prístupy využívajú princípy jedinečnosti, kontextuálnosti, procesuálnosti a dynamiky skúmaných javov (najmä v psychometrických a sociometrických výskumoch). Predmetom výskumu často bývajú ľudia, ale aj iné objekty (dokumenty). Pri záveroch a analýzach záleží na interpretácii, často sa vytvárajú tzv. „interpretačné repertoáre“. Základné typy výskumov obsahujú prípadové štúdie, analýzy dokumentov, terénny výskum kvalitatívny experiment a kvalitatívnu evaluáciu. Najčastejšie metódy výskumu sú pozorovanie a rozhovory - riadené, pološtruktúrované ai. Zo sociálnej psychológie sa v informačnej vede využívajú aj rôzne metódy pripravených „balíkov metód – testov“ (napríklad osobnostné dotazníky, inventáre, diagnostické nástroje, sémantický diferenciál, ai.).

Význam kvalitatívnych prístupov vzrastá práve pri snahách o skúmanie afektívnych a kognitívnych základov využívania informácií. Takéto výskumy informačného správania napokon nachádzajú uplatnenie aj pri modelovaní funkcií a služieb webových systémov (napríklad afektívny informačný manažment, digitálne knižnice pre deti).

V kvalitatívnych výskumoch a analýzach sa často využíva metóda „ukotvenej teórie“ autorov Glasera a Straussa [6], ktorá z množstva údajov v priebehu výskumu buduje kontext a vyvodzuje z údajov vynárajúcu sa teóriu. V procese výskumu sa objavujú také štádiá ako získavanie údajov, ich zachytenie a fixácia (videozáznamy, audiozáznamy), vytvorenie protokolov a databáz, redukcia údajov, analýzy, vyvodenie záverov. Na spracovanie kvalitatívnych údajov sa využívajú aj softvérové nástroje na analýzy textov, kódovanie, vytváranie kategórií (napríklad ATLAS.ti, Ethnograph, Textbase Alpha, NUD*ist ai.) alebo aj na analýzu sociálnych sietí (napr. UCINET ai.) a vizualizáciu kategórií a tvorbu pojmových modelov (napr. CmapTools, [4]).

4 Príklady vlastných výskumov z hľadiska použitých metód výskumu

V našich výskumných projektoch sme sa zameriavali na informačné správanie používateľov akademických knižníc, posudzovanie relevancie doktorandmi, publikačné správanie pedagógov a výskumníkov, informačnú ekológiu akademického informačného prostredia. Išlo o sériu prieskumov s využitím takých metód výskumu ako dotazníkové prieskumy (zamerané na hlbšie procesy využívania informácií), pološtruktúrované rozhovory, skupinové rozhovory, fenomenografické, obsahové a konceptuálne analýzy. V metódach výskumu sme aplikovali aj kategoriálne a frekvenčné analýzy, štatistické testy, pojmové modelovanie a rôzne spôsoby vizualizácie empirických dát (hierarchické kategoriálne modely, pojmové schémy, pojmové mapy).

V rokoch 2002-2004 sme napríklad skúmali informačné správanie používateľov akademických knižníc, zrakovo postihnutých používateľov aj seniorov (v rámci riešenia projektu VEGA 1/9236/02, 2002-2004). Na zber údajov sme použili sériu dotazníkových prieskumov s hlbšími otázkami založenými na škálovaní. Najrozsiahlejší súbor údajov obsahoval vzorku 793 respondentov zo 16 akademických a vedeckých knižníc na Slovensku. Štatistickými analýzami sme identifikovali dva informačné štýly pri využívaní informácií (pragmatický, analytický), dve štádiá pri vyhľadávaní informácií (orientačné, analytické) a vyvodili sme „profil“ priemerného používateľa [20]. Bola spracovaná aj monografia o informačnom správaní vrátane modelov, štruktúry či typológie informačného správania [16].

V ďalších projektoch sme sa orientovali na uplatnenie kvalitatívnych metód vo výskume informačného správania. Skúmali sme proces posudzovania relevancie doktorandmi z FiFUK v rámci projektu VEGA 1/2481/05 (2005-2007). Na zber údajov sme využili pološtruktúrované rozhovory (prieskum názorov 21 doktorandov). Na analýzy sme použili fenomenografické analýzy – viacnásobné pojmové kategorizácie a vyvodili sme model kolektívnej skúsenosti doktorandov pri posudzovaní relevancie. Vizualizovali sme základné vlastnosti relevancie v konceptuálnych mapách a následne sme vyvodili aj model relevancie 2.0. [19, 23]. V rámci projektu sa skúmalo aj publikačné správanie [9, 26].

V ďalšom projekte sme skúmali problematiku informačnej ekológie (projekt VEGA 1/0429/10 - 2010-2011). Tu sme použili viac kvalitatívnych metód výskumu. Išlo o pološtruktúrované rozhovory so 17 manažérmi slovenských a českých univerzít, ďalej prieskum stavu univerzitných repozitárov a nakoniec experimenty so záverečnými prácami v odbore knižničná a informačná veda z hľadiska organizácie informácií – vytvorenie pojmových máp a tematických máp z kľúčových slov. Výsledky sú publikované v správe Informačná ekológia akademického informačného prostredia [22]. Na analýzy sme využili obsahovú analýzu zaznamenaných informácií, výsledky sme vizualizovali v pojmových grafoch (napr. Fig. 1 pojmový model emócií v akademickom informačnom prostredí). Experimentálne pojmové modelovanie sme využili aj pri vytvorení slovníka a pojmových máp k učebnici vyhľadávania informácií a informačného správania [21]. Identifikovali sme aj ekologické informačné stra-

tégie pri vyhľadávaní informácií v rámci sémantického a kolaboratívneho vyhľadávania [17].

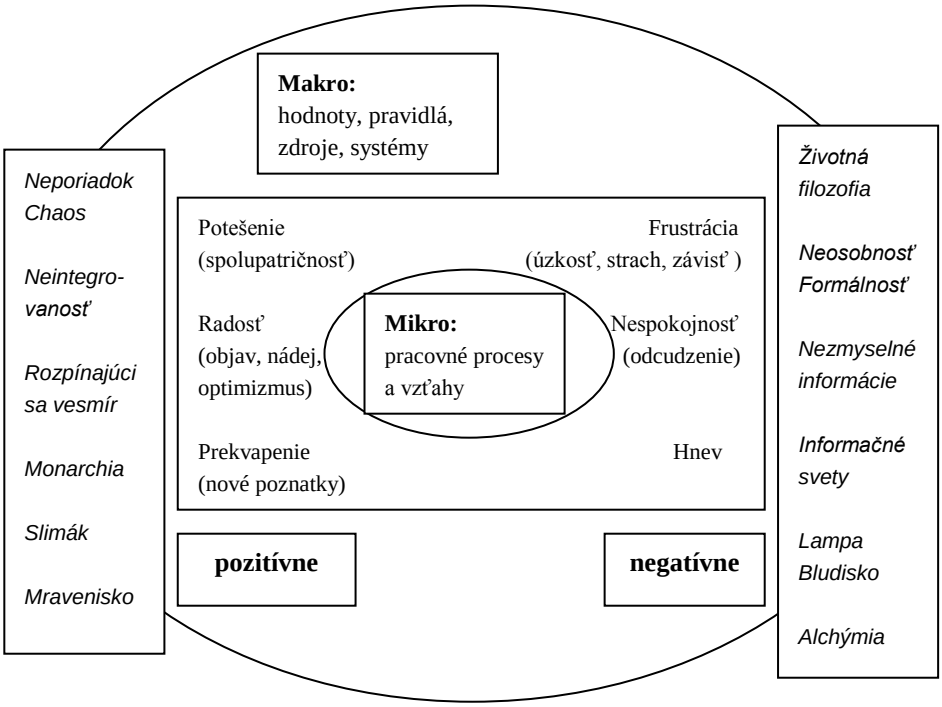


Fig. 1. Pojmový model emócií v akademickom informačnom prostredí.

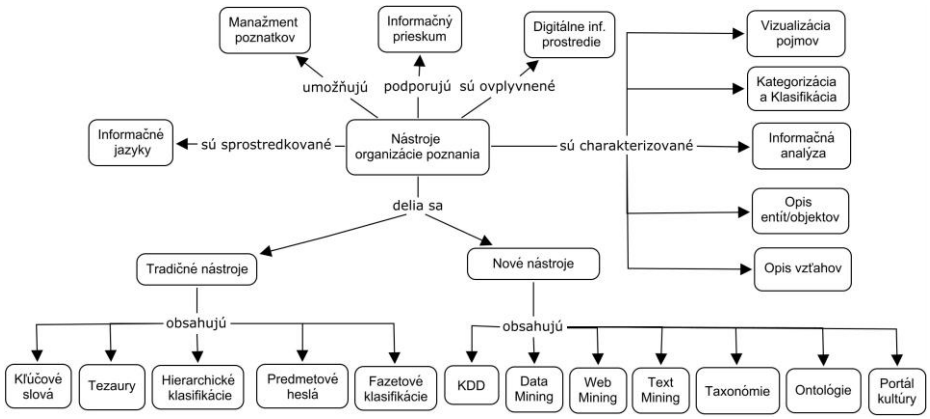


Fig. 2. Pojmová mapa Nové prístupy k organizácii poznania v digitálnom informačnom prostredí,

V experimente sme spracovali pojmové mapy tém vybraných dizertačných a diplomových prác na základe viacnásobných obsahových analýz a pojmového modelovania, ktoré bolo založené na špeciálnej metodike. Pritom sme využili softvérový nástroj CmapTools [4] využívaný vo vzdelávaní a manažmente znalostí. Príklad pojmovej mapy je na Fig. 2.

Z kľúčových slov záverečných prác za obdobie 10 rokov boli vypracované tematické mapy založené na frekvencii výskytu kľúčových slov. Príklad tematickej mapy je na Fig. 3.

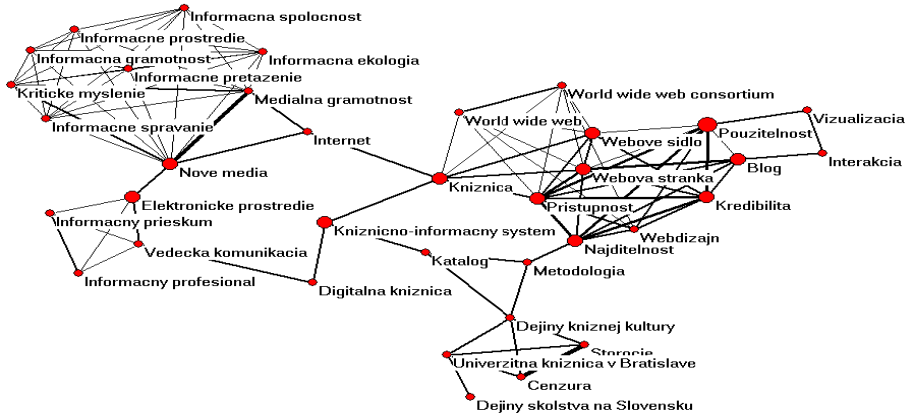


Fig. 3. Tematická mapa záverečných prác v rokoch 2009-2010.

V závere sme vypracovali model informačnej ekológie akademického informačného prostredia, ktorý je znázornený na Fig. 4.

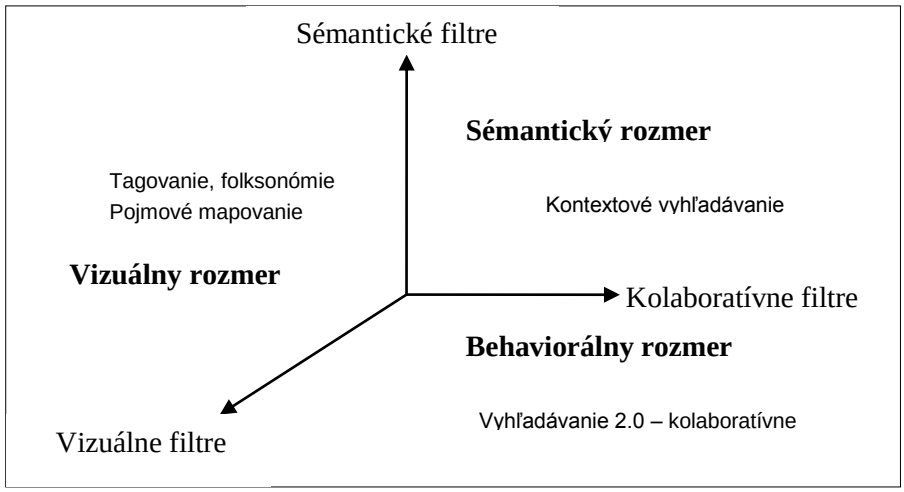


Fig. 4. Model informačnej ekológie akademického informačného prostredia.

Mnoho doktorandských projektov riešených v rámci dizertačných prác na KKIV FiFUK bolo priekopníckych z hľadiska uplatnenia metód výskumu a metodológie. Okrem dotazníkových prieskumov a modelovania [15] treba spomenúť najmä využitie metód hlasného myslenia na skúmanie kognitívnych základov interakcie s inteligentným agentom [7] alebo využitie metódy životného príbehu knihovníkov na skúmanie identity knižničných a informačných profesionálov [14]. Boli spracované aj zaujímavé práce zamerané na historický výskum knižničných fondov či tlačiarskej produkcie využívajúce historické analýzy. Zaujímavé aplikácie kvantitatívnych štatistických metód a modelovania sa objavili v bibliometrických analýzach publikačnej činnosti či webových univerzitných portálov [3, 10]. Kombináciu kvalitatívnych a kvantitatívnych prístupov využila vo svojej práci pri modelovaní kolaboratória pre manažment znalostí Katuščáková [11].

Problémom kvalitatívnych metód je práve vhodný výber vzorky, interpretácia informácií a výsledkov, problematické zovšeobecňovanie, úskalia subjektivizmu. Uplatnenie kvalitatívnych prístupov sa často kombinuje s kvantitatívnymi metódami a experimentmi. Kvalitatívne metódy však výskumníci považujú za ťažšie na realizáciu. Pritom práve princípy kognície a interakcie zmenili paradigmu informačnej vedy. Informačné procesy a systémy sa modelujú s dôrazom na kontext, čo sa prejavuje aj vo výskumoch sociálnych sietí a sociálnych médií. Metodologické semináre pre doktorandov poukazujú na najčastejšie metodologické problémy výskumu doktorandov ako zužovanie záberu problematiky a prepojenie teórie a praxe systémov a služieb. Súčasné aktuálne témy doktorandských projektov sa zameriavajú na nové médiá, kultúrnu produkciu [8], sociálne siete [25], individualizáciu internetu, kredibilitu webových sídiel, modelovanie elektronických diskusných skupín. Z tém informačného správania sú zaujímavé tie, ktoré sa venujú špeciálnym či okrajovým skupinám používateľov (profesionáli – informatici, bezdomovci, sociálne vylúčení ai.).

5 Etika výskumu

V sociálne orientovanom výskume v elektronickom prostredí treba pripomenúť práve aktuálnu otázku etiky výskumu. Výskumníci sú ľudia, ktorí sa môžu dopustiť omylov a pracujú s určitým súborom hodnôt a princípov. Etika vo výskume naznačuje správnosť a nesprávnosť správania, je založená na skupinových normách správania. Pritom už existujú aj štandardy (napr. kódex správania Americkej sociologickej asociácie), ktoré naznačujú štyri základné princípy pri skúmaní ľudí a ich správania: účastníci výskumu nesmú byť poškodení (fyzicky či psychicky), účastníci výskumu nesmú byť nijakým spôsobom zavádzaní či podvádzaní, účasť na výskume musí byť dobrovoľná („informovaný súhlas“), všetky údaje zhromažďované o skúmaných osobách musia byť dôverné (a zabezpečené pred zneužitím). Týmto princípom sa ani vo výskumoch informačného správania nevenovala dostatočná pozornosť. Zdôrazňuje sa však vytvorenie atmosféry dôvery a vzájomného rešpektu medzi výskumníkmi a respondentmi. Osobitne aktuálna je však etika internetového výskumu. Ľahkosť získania množstva údajov o správaní používateľov diskusií, sociálnych sietí, elektronickej pošty prostredníctvom robotov a monitorovacích nástrojov vyvoláva pochybnosti o „plnej in-

formovanosti“ skúmaných osôb. Riešením bývajú práve inštitucionálne revízne rady, ktoré si vyžadujú dôkazy o informovanosti pozorovaných osôb. Mnoho internetových výskumov podľahlo zjednodušenej metodológii (napr. nedôsledná príprava online dotazníkov). Etika výskumu sa ešte viac týka biomedicínskej technológie, ale aj takých metód ako analýza transakcií, často zneužívaná na marketingové a komerčné účely. V tomto zmysle by sa práve informatický výskum mohol inšpirovať pravidlami sociálnovedných výskumov. Mnohí autori podporujú užitočnosť vízie spoločného postupu informačných vied (informatika, informačná veda ai.).

6 Koncepcia aktuálneho výskumu v rámci projektu APVV Tra-Di-Ce

V novom výskume sa orientujeme na skupinu doktorandov ako mladých začínajúcich vedeckých pracovníkov. Cieľom výskumu je identifikovať informačné potreby a informačné správanie študentov doktorandského štúdia na vybraných univerzitách v SR. Zaujímajú nás hlbšie procesy využívania informácií, plánovania výskumných projektov, publikovania či využívania sociálnych sietí. Vzhľadom na ciele aplikujeme vo výskume kvalitatívnu metódu pološtruktúrovaných rozhovorov. Predpokladáme, že na základe analýz predchádzajúcich výskumov, zistených empirických údajov a ich analýz vypracujeme informačný model doktoranda smerom k návrhu nových služieb a systémov s pridanou hodnotou pre túto skupinu používateľov. Naznačíme aj možnosti využitia znalostných technológií pre služby doktorandom.

Základná koncepcia výskumu je znázornená na Fig. 5 a Table 1. Na základe toho je vypracovaný súbor otázok na pološtruktúrované rozhovory. Získavanie empirických údajov prebieha od októbra 2012.

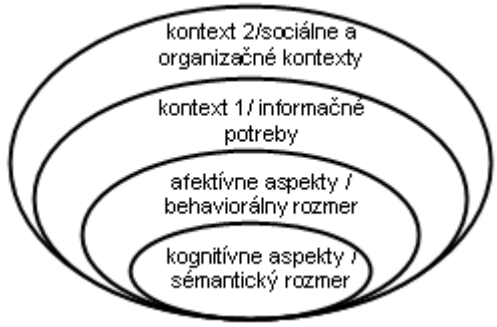


Fig. 5. Základná koncepcia výskumu (vrstvy premenných).

Table 1. Kategórie skúmania informačného správania a informačných potrieb.

výskumné správanie	▪ výber témy ▪ plánovanie výskumného procesu
informačné správanie pri využívaní informácií	▪ informačné stratégie, postupy ▪ náhodné získavanie informácií
získavanie a vyhľadávanie informácií	▪ typy zdrojov ▪ informačný horizont
organizovanie informácií	▪ triedenie zdrojov ▪ nástroje triedenia
sociálne médiá	▪ využívanie ▪ prínos
informačné správanie pri tvorbe	▪ publikovanie ▪ typ zdroja; výber časopisu, vydavateľstva, formy

Možnosti spolupráce sa vynárajú pri získavaní a analýzach empirických údajov – bolo by vhodné využiť špeciálny softvér na kvalitatívne analýzy v slovenskom jazyku. Zaujímavé je aj pokračovať v pojmovom modelovaní a vytváraní tematických máp zo záverečných prác aj v iných odboroch. Do analýz by bolo vhodné zaradiť aj analýzy využívania špeciálnych elektronických zdrojov doktorandmi z hľadiska ich informačného správania. Pri analýzach by sa uplatnili aj softvérové nástroje na dolovanie textov z empirických údajov a najmä na vizualizáciu sémantických kategórií. Ďalšie možnosti spolupráce vidíme pri budovaní digitálnej knižnice pre doktorandov a pri modelovaní komunitného informačného portálu pre doktorandov so službami s pridanou hodnotou.

7 Záver

Predpokladáme, že kombinácia kvalitatívnych metód výskumu so špeciálnymi experimentálnymi metódami informatiky môže viesť k zaujímavému metodologickému dialógu a inšpiráciám. Tzv. veda 2.0 naznačuje, že zaujímavé sú práve transdisciplinárne problémy. V informačnej vede sa týkajú organizácie informácií v digitálnych knižniciach a repozitároch, modelovania informačného správania a informačných potrieb, budovania služieb s pridanou hodnotou pre špeciálne skupiny vo vzdelávaní a výskume. Na sémantickej úrovni je možné prepojiť tradície knižničnej činnosti pri analýzach dokumentov s autorskou organizáciou informácií a inteligentnými technológiami - napríklad pri vytváraní pojmových máp a tematických máp. V nových nástrojoch organizácie poznania sa účinnosť zvyšuje kombináciou hrubšieho obsahového triedenia s podrobnejšou informačnou štruktúrou (pojmové mapy) až po budovanie ontológií. Metodológia výskumu podporuje hľadanie skrytých súvislostí v interdisciplinárnych témach (sociálne siete, sémantické siete, neurónové siete). Metodologická rozmanitosť informačnej vedy môže byť úskalím, ale aj inšpiráciou na riešenie nových širších problémov ako je informačná ekológia, kredibilita, kreativita,

identita, kontext, ale aj nové modely vedeckej komunikácie a metódy manipulácie s digitálnymi objektmi.

Poďakovanie. Príspevok bol vypracovaný v rámci riešenia projektu APVV-0208-10 TraDiCe. Kognitívne cestovanie po digitálnom svete webu a knižníc s podporou personalizovaných služieb a sociálnych sietí.

Literatúra

1. Bates, M. 2005. An Introduction to Metatheories, Theories, and Models. Chapter 1. In Theories of Information Behavior. Ed. By K. Fisher, S. Erdelez, L. McKechnie. Medford (NJ): Information Today, 2005. pp. 1-24.
2. Bawden, D., Robinson, L. 2012. Introduction to Information Science. London: Facet Publishing, 2012. ISBN 978-1-85604-810-1.
3. Bellérová, B. Publikovanie autorov v slovenskom akademickom prostredí z pohľadu bibliometrickej analýzy. In Zborník FiFUK. Roč. 23. Knižničná a informačná veda. Bratislava: Univerzita Komenského, 2011. pp. 67-88.
4. Cañas, A. J. et al. 2004. CmapTools. A Knowledge Modeling and Sharing Environment. In Concept Maps: Theory, Methodology, Technology, Proceedings of the First International Conference on Concept Mapping. Pamplona: Universidad Pública de Navarra, 2004. pp. 125-133.
5. Case, D. 2002. Looking for information: a survey of research on information seeking, needs and behaviour. New York: Academic Press, 2002.
6. Glaser, B. G., Strauss, A. L. 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. New York: Aldine Publishing Company, 1967.
7. Grešková, M. 2009. Interaktivita vyhľadávania ako nástroj znižovania kognitívnej záťaže. In Zborník FiFUK. Roč. 22. Knižničná a informačná veda. Bratislava: Univerzita Komenského, 2009. pp. 55-70.
8. Chudý, A. 2011. Zdieľanie súborov v prostredí slovenských diskusných fór (prípadová štúdia Lamky.net). In Zborník FiFUK. Roč. 23. Knižničná a informačná veda. Bratislava: Univerzita Komenského, 2011. pp. 159-172.
9. Ilavská, J. 2011. Publikačná činnosť ako kritérium hodnotenia výkonov vo vede a výskume. In Zborník FiFUK. Roč. 23. Knižničná a informačná veda. Bratislava: Univerzita Komenského, 2011. pp. 89-103.
10. Jedličková, Ľ. 2010. Aplikácia vybraných metodologických postupov pri webometrickej analýze časti slovenskej univerzitnej siete. In Zborník FiFUK. Roč. 23. Knižničná a informačná veda. Bratislava: Univerzita Komenského, 2011. pp. 105-123.
11. Katuščáková, M. 2011. Manažment znalostí: sociálne aspekty. Žilina: Žilinská univerzita, 2010. ISBN 978-80-554-0244-4.
12. Kollárik, T., Sollárová, E. 2004. Metódy sociálnopsychologickej praxe. Bratislava: Ikar, 2004.
13. Miovský, M. 2006. Kvalitatívní přístup a metody v psychologickém výzkumu. Praha: Grada, 2006. ISBN 80-247-1362-4.
14. Ondriašová, H. 2011. Strategické vytváranie organizačnej a profesijnej identity v prostredí knižníc. In Zborník FiFUK. Roč. 23. Knižničná a informačná veda. Bratislava: Univerzita Komenského, 2011. pp. 125-146.

15. Ondrišová. M. 2009. Inštitucionálny repozitár ako akcelerátor výskumu. In *Zborník FiFUK. Roč. 22. Knižničná a informačná veda*. Bratislava: Univerzita Komenského, 2009. pp. 107-117.
16. Steinerová, J. 2005. *Informačné správanie: Pohľady informačnej vedy*. Bratislava: CVTI SR, 2005. ISBN 80-85165-90-2.
17. Steinerová, J. 2011. Ekologické informačné stratégie - nový prístup k pojmovému modelovaniu. In *Inforum 2011. 17. konferencia o profesionálnych informačných zdrojoch, Praha, 24.-26.5.2011* [online]. Praha: Albertina icome, 2011 [cit. 2012-09-10]. Dostupné na: http://www.inforum.cz/pdf/2011/steinerova-jela.pdf.
18. Steinerová, J. 2011. Metodologické problémy výskumov informačnej vedy. In: *ProInFlow* [online]. 2011, 3(1), pp. 5-18 [cit. 2012-09-10]. Dostupné na: http://ww.proinflow.cz.
19. Steinerová, J. 2011. Premeny relevancie v informačnej vede. In *Knihovna*. 2011, 22(2), pp. 59-70.
20. Steinerová, J. et al. 2004. *Správa o empirickom prieskume používateľov knižníc ako súčasť grantovej úlohy VEGA1/9236/02 Interakcia človeka s informačným prostredím v informačnej spoločnosti.* Bratislava: Filozofická fakulta UK - KKIV, 2004.
21. Steinerová, J. et al. 2010. *Informačné stratégie v elektronickom prostredí: pojmové mapy a slovník* [CD-ROM]. Bratislava: Stimul, 2010. ISBN 978-80-89236-99-2.
22. Steinerová, J. et al. 2012. *Informačná ekológia akademického informačného prostredia*. Záverečná správa z výskumu VEGA 1/0429/10. Editor Jela Steinerová. Autori: Jela Steinerová, Jana Ilavská, Lucia Lichnerová, Miriam Ondrišová, Helena Ondriašová, Linda Prágerová, Martina Haršányiová. Bratislava: Univerzita Komenského, 2012. ISBN 978-80-223-3178-4.
23. Steinerová, J., Grešková, M. a Šušol, J. 2007. *Prieskum relevancie informácií: Výsledky rozhovorov s doktorandmi FiF UK*. Bratislava: CVTI SR, 2007.
24. Steinerová, J., Grešková, M., Ilavská, J. 2010. *Informačné stratégie v elektronickom prostredí.* Bratislava: Univerzita Komenského, 2010. ISBN 978-80-223-2848-7.
25. Šuchová, J. 2011. Online sociálne siete a filtrovanie informácií. In *Zborník FiFUK. Roč. 23. Knižničná a informačná veda.* Bratislava: Univerzita Komenského, 2011. pp. 147-157.
26. Šušol, J. 2009. Publikačné správanie autorov v akademickom prostredí. In *INFOS 2009.* Bratislava: Spolok slovenských knihovníkov, 2009. pp. 178-194.
27. Troll Covey, D. 2002. *Usage and Usability Assessment: Library Practices and Concerns.* Washington, D.C.: Digital Library Federation, 2002. ISBN 1-887332-89-0.
28. Wilson, T. 2002. „Information science“ and research methods. In *Zborník FiFUK. Roč. 19. Knižničná a informačná veda.* Bratislava: Univerzita Komenského, 2002. pp. 63-71.

Prepojené dáta

Navigácia v zjednodušenom LinkedData grafe

Ján Mojžiš, Michal Laclavík

Ústav informatiky, Slovenská akadémia vied,
Dúbravská cesta 9, Bratislava 845 07, Slovenská Republika
mojzis@fiit.stuba.sk, laclavik.ui@savba.sk

Abstrakt. Iniciatíva LinkedData obsahuje rozsiahle vzájomne prepojené štruktúrované dáta. Poznáme síce riešenia na prehľadávanie, vizualizáciu a navigáciu v dátach, chýba však nástroj pre navigáciu v dátach s neznámou ontológiou. Pomocou grafov a sietí môžeme reprezentovať najrôznejšie dáta. Výhodou je zobrazenie relácií (hrán a ciest) medzi entitami (vrcholmi grafu). V tejto práci ukazujeme ako konvertovať LinkedData grafy s ontológiami na zjednodušené LinkedData grafy. Pre navigáciu v zjednodušenom grafe používame nástroj gSemSearch. Ontológie použité na konverziu sú *DBLP*, *Public Contracts Ontology* a *ACM proceedings*. Pri navigácii grafy aj vizualizujeme.

Kľúčové slová: LinkedData, graf, vizualizácia, gSemSearch

1 Úvod

Linked Data je jedným z dôležitých konceptov Sémantického webu. Umožňuje využitie a šírenie štruktúrovaných dát pomocou vytvorenia spojení medzi datasetmi na webe. Dáta zapísané pomocou odporúčaní[1] LinkedData sú význačné svojou schopnosťou byt strojovo čitateľné. Znamená to, že jeho štruktúrované sémantické dáta môžu byt hľadané, čítané (parsované) a analyzované. Jedným z kompatibilných dátových formátov konceptu Linked Data je RDF (Resource Description Framework).

S rozvojom Web 2.0 a sociálnych sietí sa na webe presadila ľahká sémantika vo forme značiek, prepojených ľudí a objektov ktoré tvoria informačné siete s vlastnosťami malého sveta. Podobné vlastnosti má aj web graf alebo aj wikipédia. Pravdepodobne aj graf prepojení LinkedData.

V minulosti sme použili algoritmus šírenia aktivácie a rozhranie gSemSearch [6] pre objavenie vzťahov v sieťach vytvorených z emailov s podobnými vlastnosťami. V tejto práci ukazujeme naše experimenty na objavovaní vzťahov v zjednodušených grafoch LinkedData. Vybrané zdroje LinkedData sú prevádzané do jednoduchých grafov.

[1] http://5stardata.info/

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 15–18.

2 Existujúce riešenia

W. Hu a kol. sa zaoberajú grafovou reprezentáciou LinkedData v [1]. Pre ontológie používajú bipartitné grafy. Jedná sa však len o porovnávanie ontológií za pomoci grafu. Prehľadávanie dát s viacerými ontológiami riešia Gašević a Hatala v ich práci [2]. Na reprezentáciu ontológií používajú systém SKOS.

W. Hu má v porovnávaní pomocou grafov dobré výsledky. Podobne Gašević a Hatala pri získavaní dát z neznámych ontológií. Avšak, vo svojich prácach nenavrhujú neriešenia pre navigáciu (prehľadávanie aj pohyb po dátach) a vizualizáciu.

Pre navigáciu v známej ontológii existujú prehliadače LinkedData [3, 4, 5].

V práci ukazujeme postup, ako možno konvertovať LinkedData na zjednodušený graf tak, aby bola možná navigácia bez znalosti ontológie. Relácie objektov sú vyjadrené v pároch kľúč-hodnota, a sú to hrany medzi dvoma vrcholmi.

3 Prevod LinkedData na graf

Pre získanie grafu z LinkedData s neznámou ontológiou môžeme uskutočniť prevod na jednoduchší graf s netypovými hranami a vrcholmi typu *klúč-hodnota*. Napríklad ontológia RDF/XML používa inštancie, dátové vlastnosti a literály (vlastnosti jednoduchých typov ako String, Integer alebo Date). Bez ohľadu na ich štruktúru môžeme vybrať najvnútornejšie literály. Osvedčilo sa načítanie RDF dát do pamäte (Systém Jena[2]). Následný prevod na zjednodušený graf prechádza všetky objekty, pričom sa vnára do štruktúry, kde hľadá najvnútornejšie hodnoty vlastností typu *literal*. Ku každej dátovej vlastnosti sa ako *kľúč* berie jej typ (rdf:type) a ako *hodnota* sa priradí najreprezentatívnejší[3] literál typu String. Ak ide o vlastnosť ktorá je typu literal, *hodnotou* je samotný literál a *kľúčom* predikát. Vytvorí sa zjednodušený graf, zapísaný dvojicami kľúč-hodnota podobný tomu, ktorý sa využíva v gSemSearch [6]. Prevod rešpektuje hierarchickú štruktúru pôvodných dát RDF/XML. Prevádzané boli RDF dáta ACM Proceedings[4], VSE contracts[5] a DBLP[6].

Výhoda v prevode RDF dát na zjednodušený graf je v tom, že zanedbaním niektorých predikátov sa odstránia hrany, sprehľadníme tak graf a ponúkneme možnosť lepšej navigácie. Relácie sú viditeľnejšie a priamejšie.

---

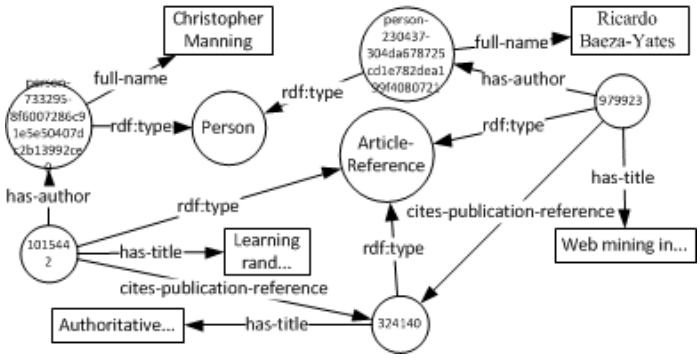
[2] http://jena.sourceforge.net/index.html

[3] Určenie najreprezentatívnejšieho literálu inštancie je definované priamo pre každý typ objektu v ontológií. V budúcnosti by sme chceli tento proces automatizovať.

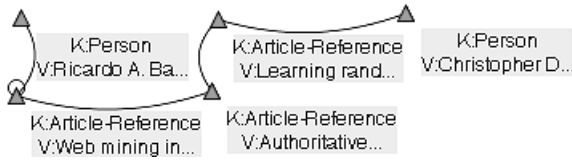
[4] http://acm.rkbexplorer.com/models/dump.tgz

[5] http://keg.vse.cz/ismis2012/data/task4-test.rdf.gz

[6] http://dblp.uni-trier.de/xml/dblp.xml.gz



Obr. 1. Pôvodný RDF graf ACM Proceedings. Obrázok ukazuje spojenie *Ch. Manninga* a *Baeza-Yatesa* cez spoločnú citáciu *Authoritative sources in a hyperlinked enviroment*. Hrany s predikátmi a identifikátory trochu zneprehľadňujú graf.



Obr. 2. Ten istý graf zjednodušený zanedbaním predikátov a identifikátorov z pôvodného grafu. S menším počtom hrán je jasnejšie vidieť spojenie *Ch. Manninga* a *Baeza-Yatesa* cez spoločnú citáciu.

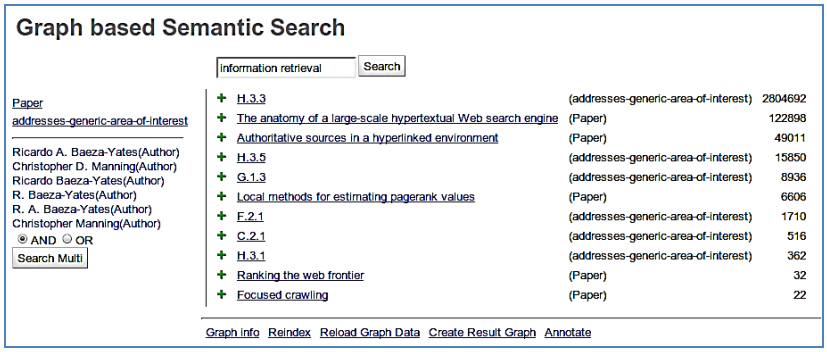
Pre *ACM Proceedings* sme zjednodušili pôvodný RDF graf (obr.1) zanedbaním niekoľkých predikátov. Tým sme sprehľadnili graf (odstránili sa hrany) a spojenie *Ch. Manninga* a *Baeza-Yates* cez spoločnú citáciu *Authoritative sources in a hyperlinked environment* je jasnejšie (obr.2).

Ako nie vhodné na prevod do zjednodušeného grafu sa ukázali dáta *VSE Contracts.* Objekt *Kontrakt* obsahuje dve dátové vlastnosti typu *Organizácia* a pri zjednodušovaní grafu zanedbávame názvy týchto vlastností, keďže zjednodušený graf obsahuje netypové hrany. Preto potom nie je jasné, či meno organizácie v kontrakte je kontraktujúca autorita alebo výherca kontraktu.

DBLP databáza ukázala spojenia len prostredníctvom spoločných autorov pri publikáciách, pretože neobsahuje citácie.

4 Navigácia

Na obrázku 3 je zobrazený výsledok prehliadania dvoch autorov z oblasti Information Retrieval (IR), pre ktorých boli odvodené relevantné články z oblasti IR ako napríklad známy článok o Google vyhľadávači alebo oblasť výskumu H3.3 zameraná na oblasť IR. Kliknutím na článok je možné navigovať sa v grafe cez relevantné objekty (v našom prípade články, autori, oblasti). Je možné označiť aj viacero objektov rôzneho typu a pre ne odvodiť relevantné objekty.



Obr. 3. Prehliadanie zjednodušeného grafu ACM v nástroji gSemSearch.

Takýmto spôsobom je možné odvodiť relevantné články k zaujímavým autorom alebo článkom, kde pomocou šírenia aktivácie v grafe sú relevantné články odvodené pomocou relácií citácií, autorov alebo oblastí.

5 Záver

Zanedbaním niektorých vlastností grafu sme v práci ukázali, ako možno zjednodušiť a následne vizualizovať relácie v grafe. Tiež sme zistili, že nie všetky dáta sú vhodné na prevod na zjednodušený graf (VSE). Znázornili sme odvodzovanie podľa spoločných citácií autorov pre dáta ACM Proceedings. Odvodzovať však je možné aj podľa spoluautorov, záujmových oblastí a i. Náš návrh v porovnaní s doterajšími prácami počíta s prevodom na jednoduchý graf a s jeho vizualizáciou.

Poďakovanie. Táto práca vznikla s podporou projektov TRA-DICE APVV-0208-10 a VEGA 2/0184/10.

Zdroje

1. W. Hu, N. Jian, Y. Qu, Y. Wang, GMO: a graph matching for ontologies, Proceedings of the K-CAP Workshop on Integrating Ontologies, pp. 41-48, 2005.
2. Gašević, D., Hatala, M., Ontology mappings to improve learning resource search, British Journal of Educational Technology, pp. 375-389, 2006.
3. Berners-Lee, T., Y. Chen, L. Chilton, D. Connolly,R. Dhanaraj, J. Hollenbach,, A. Lerer, and D. Sheets, Tabulator: Exploring and Analyzing linked data on the Semantic Web, In Proceedings of the The 3rd International Semantic Web User Interaction Workshop (SWUI06), Nov 2006.
4. Hastrup, T., Cyganiak, R., Bojars, U. Browsing Linked Data with Fenfire, In LDOW 2008: Proc., Linked Data in the Web Workshop at WWW'2008, 2008.
5. Huynh, D., Mazzocchi, S., Karger, David. Piggy Bank: Experience the Semantic Web Inside Your Web Browser, In International Semantic Web Conference (ISWC) 2005, 2005.
6. Laclavík, M., Dlugolinský, Š., Šeleng, M., Ciglan, M., Hluchý, L., Emails as Graph: Relation Discovery in Email Archive, WWW 2012 – Email Workshop, pp. 841-846, 2012.

Využitie prepojených dát v digitálnej knižnici

Michal Holub, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava, Slovensko
{holub,bielik}@fiit.stuba.sk

Abstrakt. V tomto príspevku predstavujeme kombinovaný model používateľa a model domény digitálnych knižníc, ktorý je založený na princípe prepojených dát (angl. Linked Data). Prezentujeme možnosti využitia takýchto dát pri podpore práce v digitálnej knižnici v kontexte vyhľadávania zdrojov pomocou dopytov v prirodzenom jazyku a pri adaptívnom vedení používateľa a odporúčaní vedeckých zdrojov pri orientovaní sa v príslušnej výskumnej oblasti. Oblasť záujmu používateľa digitálnej knižnice reprezentujeme navštívenými článkami a s nimi súvisiacimi metadátami. Tieto informácie zbierame pomocou rozšírenia webového prehliadača zaznamenávajúceho aj akcie vykonané pri prezeraní daného článku. Diskutujeme možnosti identifikácie nových prepojení na základe interakcie používateľov so zdrojmi v digitálnej knižnici, ktorými je možné obohatiť existujúci model domény. Prezentujeme aktuálny stav realizácie modelu, ktorý spracúva metadáta o entitách v digitálnej knižnici (články, autori, konferencie, publikácie) z ACM DL a uchováva ich pomocou trojíc použitím rámca RDF.

Kľúčové slová: digitálna knižnica, prepojené dáta, Linked Data, vyhľadávanie, navigácia, model používateľa, model domény

1 Úvod

Efektívna a pohodlná práca s údajmi v rozsiahlom informačnom priestore, akým sú napr. digitálne knižnice, vyžaduje podporné prostriedky zo strany tohto systému. Patria medzi ne najmä podpora pri vyhľadávaní informácií a podpora pri navigovaní v rámci vybranej problémovej domény [1]. Tieto prostriedky využívajú metódy adaptácie a personalizácie, aby používateľovi uľahčili prehliadanie informačného priestoru. Hoci rozšírené v iných doménach (napr. výučbových systémoch), v súčasných digitálnych knižniciach sú tieto metódy málo používané.

Digitálna knižnica, akou je napr. ACM Digital Library, môže obsahovať záznamy o miliónoch článkov, státisícoch autorov, tisícoch uskutočnených konferencií a vydaných odborných časopisov. Okrem veľkosti priestoru sú ďalšími problémami nejednoznačné mená autorov, viacznačné pojmy opisujúce výskumné oblasti, rôzne úrovne náročnosti článkov, a pod. V takomto informačnom priestore sa používateľ

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 19–22.

bez podpory pri vyhľadávaní a navigácii ľahko stratí, čo znižuje efektivitu jeho práce. Existujúce rozhrania digitálnych knižníc umožňujú:

- základnú formu vyhľadávania pomocou kľúčových slov hľadaných v zadaných parametroch záznamov (nadpis článku, hlavný text, autor, atď.),
- filtrovanie výsledkov podľa spoločných hodnôt týchto parametrov (rok vydania článku, názov konferencie, atď.),
- prehľadávanie záznamov z určitej oblasti.

Chýba však podpora pri vyhľadávaní pomocou rozširovania, odporúčania dopytov a formulovania dopytov v prirodzenom jazyku, ako aj podpora pokročilejších techník prehľadávania, navigovanie používateľa článkami z vybranej domény (ktoré môžu byť navyše usporiadané podľa miery náročnosti textu). Tieto metódy sa už úspešne využívajú napr. vo výučbových systémoch [5], pričom nimi sú:

- odporúčanie ďalších článkov, ktoré by si mal používateľ pozrieť, založené na histórii videných článkov,
- navigovanie používateľa digitálnym priestorom článkov tak, aby získal prehľad z vybranej výskumnej oblasti (berúc do úvahy jeho doterajšie poznatky).

Na realizáciu uvedených scenárov potrebujeme dostatočne podrobný doménový model zachytávajúci vzťahy medzi jednotlivými entitami v digitálnej knižnici, ako bolo prezentované už v [2]. V tomto príspevku opisujeme takýto model domény, namiesto ontológií však využívame ľahkú sémantiku prepojených dát (angl. *Linked Data*), ktorá sa nám javí flexibilnejšia. Vďaka prítomnosti prepojení medzi entitami vieme používateľovi ponúknuť spomínanú podporu pri vyhľadávaní a navigácii.

2 Budovanie doménového modelu s využitím prepojených dát

V súčasnosti využívame ako zdroj dát digitálnu knižnicu ACM Digitral Library. Na získanie dát o entitách uchovávaných v tejto knižnici používame fokusovaného webového pavúka (angl. *web spider*), pomocou ktorého sťahujeme HTML stránky obsahujúce záznamy o jednotlivých článkoch. Tieto stránky následne parsujeme a extrahujeme z nich konkrétne údaje.

Každý článok má v knižnici ACM pridelený unikátny číselný identifikátor. Ten sa skladá z číselného identifikátora publikácie (konkrétny zborník, odborný časopis, atď.) a identifikátora samotného článku, pričom je obsiahnutý v URL stránky s údajmi o tomto článku. Vieme tak stiahnuť a spracovať voľne dostupné metadáta o článkoch.

Po stiahnutí stránky na server z nej extrahujeme všetky dostupné dáta pomocou parsera, ktorý sme za týmto účelom vytvorili. Články spravidla obsahujú aj zoznam referencií na iné články. Pri nich je uvedený ACM identifikátor, vďaka ktorému vieme články prepájať. Problém je len s článkami publikovanými mimo ACM knižnice. Pri rozšírení modelu na iné digitálne knižnice plánujeme identifikovať referencie na základe názvu článku, autorov a názvu publikácie. Ich formát zápisu sa medzi jednotlivými článkami spravidla líši, čo si vyžiada vývoj samostatnej metódy na ich identifikáciu. To už je však samostatný výskumný problém.

Získané dáta uchovávame pomocou trojíc subjekt-predikát-objekt s využitím rámca RDF. Takáto grafová reprezentácia nám umožňuje využívať súvislosti (vyjadrené prepojeniami) medzi entitami na to, aby sme používateľovi odporúčali príbuzné entity (články) a viedli ho na ďalšie témy. Takisto využívame grafový dopytovací jazyk SPARQL na zodpovedanie dopytov používateľa.

3 Budovanie modelu používateľa

Zaznamenávame správanie používateľa pri interakcii s digitálnou knižnicou, pričom z neho zostavujeme jeho model. Primárnym zdrojom údajov sú články, ktoré si používateľ prezeral (každý článok v digitálnej knižnici je zobrazený na samostatnej webovej stránke).

Na každej webovej stránke zároveň zaznamenávame akcie, ktoré na nej používateľ vykonal. Sú nimi čas strávený na stránke, rolovanie stránky, vyznačovanie textu pomocou kurzora myši, kopírovanie údajov do schránky. Z týchto akcií vieme určiť, či stránka používateľa zaujala alebo nie, ako sme ukázali v [3].

Na zaznamenávanie práce používateľa s digitálnou knižnicou využívame rozšírenie webového prehliadača. Toto je naviazané na knižnicu ACM Digital Library, takže na iných stránkach sa správanie nezaznamenáva. Uchovávame len identifikátor používateľa, na základe ktorého však nevieme určiť jeho presnú identitu (ak ju sám pri nastavovaní rozšírenia nezadá). S týmto identifikátorom spájame všetky akcie, ktoré v knižnici urobí (postupnosti navštívených stránok a akcie na nich vykonané).

Toto rozšírenie nám tiež pomáha pri budovaní doménového modelu. Keď používateľ navštívi stránku, ktorú ešte nemáme spracovanú, jeho prehliadač ju pošle na centrálny server. Nemusíme tak zaťažovať digitálnu knižnicu opakovanými dopytmi na danú stránku. Zároveň vieme, ktoré články používateľa zaujímajú, a tak pri sťahovaní smerujeme webového pavúka na príbuzné články (z tej istej publikácie, príp. s podobným ID ako má prezeraný článok).

4 Využitie modelov pri vyhľadávaní a navigovaní

Ako sme už spomínali, dôležitými aktivitami pri práci s digitálnou knižnicou sú vyhľadávanie článkov a navigácia priestorom. Obe tieto činnosti chceme podporiť zapojením prvkov personalizácie a adaptácie.

Pri vyhľadávaní čelíme problému, že často nevieme vyjadriť svoje potreby pomocou kľúčových slov. Ľahšie by sa nám vyhľadávalo pomocou dopytov v prirodzenom jazyku, ktorý však tradičné vyhľadávače nevedia rozpoznať. V dátach reprezentovaných grafom sa dá vyhľadávať napr. pomocou jazyka SPARQL, v ktorom vieme skonštruovať aj zložité dopyty a pomerne jednoducho na ne získať odpoveď. Problémom ale je, že tento jazyk je pre bežného používateľa zložitý.

V súčasnosti pracujeme na metóde, ktorá umožní používateľovi zadať dopyt v polo-prirodzenom jazyku (využívame angličtinu). Tento dopyt bude následne preložený do jazyka SPARQL a vykonaný nad digitálnou knižnicou reprezentovanou pomocou vyššie opísaného doménového modelu.

Pri spracovaní dopytu využívame nástroj na spracovanie prirodzeného jazyka Stanford NLP a databázu WordNet, podobne ako v [6, 7]. Používateľ vidí postup spracovania dopytu a môže ho v prípade potreby preformulovať, na čo ho vyzveme pri zistení nejednoznačnosti. Dopĺňanie informácií od používateľa je využité aj v [4], metóda však umožňuje písať dopyty iba podľa pripravenej šablóny.

Pri podpore navigovania problémovou doménou využívame poznatky používateľa reprezentované kľúčovými pojmami, ktoré extrahujeme z článkov, ktoré čítal. Pri každom pojme uvažujeme aj znalostné skóre (čím viac článkov s daným pojmom používateľ prečítal, tým väčšia by jeho znalosť mala byť). Navrhujeme metódu, ktorá na základe takýchto vstupov, ako aj prepojení medzi jednotlivými článkami, odporučí používateľovi ďalšie články na štúdium, a to v príslušnom poradí. Uplatnenie má najmä pre používateľov, ktorí sú v danej oblasti noví a chcú v nej získať prehľad.

Okrem zjavných prepojení medzi entitami (napr. človek *A* je autorom článku *X*) existujú aj skryté prepojenia, ktoré nie sú hneď zrejmé. Dajú sa však vydedukovať zo správania používateľov. Napr. ak pri riešení určitej výskumnej úlohy prečítam dva články, pravdepodobne spolu súvisia (hoci na seba nemusia odkazovať ani mať spoločných autorov). Práve z aktivity používateľov pri práci v digitálnej knižnici plánujeme extrahovať dodatočné prepojenia a obohacovať nimi doménový model.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0971/11 a APVV-0208-10.

Literatúra

1. Brusilovsky, P., Maybury, M.T. (2002). From Adaptive Hypermedia to the Adaptive Web. In: Communications of the ACM, 45(5), pp. 30-33. ACM Press, New York.
2. Ferran, N., Mor, E., Minguillón, J. (2005). Towards Personalization in Digital Libraries through Ontologies. In: Library Management, 26(4/5), pp. 206-217. Emerald Group.
3. Holub, M., Bieliková, M. (2010). Estimation of User Interest in Visited Web Page. In: Proc. of the 19th Int. Conf. on World Wide Web, pp. 1111-1112. ACM Press, New York.
4. Kaufmann, E., Bernstein, A., Zumstein, R. (2006). Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In: Proc. of the 5th Int. Semantic Web Conference, pp. 980-981. LNCS 4273, Springer.
5. Šimko, M., Barla, M., Bieliková, M. (2010). ALEF: A Framework for Adaptive Web-Based Learning 2.0. In: IFIP Advances in Information and Communication Technology, Vol. 324, pp. 367-378. Springer, Boston.
6. Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J.T. (2011). OWLPath: An OWL Ontology-Guided Query Editor. In: IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, 41(1), pp. 121-136. IEEE.
7. Wang, C., Xiong, M., Zhou, Q., Yu, Y. (2007). PANTO: A Portable Natural Language Interface to Ontologies. In: Proc. of the 4th European Conf. on The Semantic Web: Research and Applications, pp. 473-487. Springer-Verlag Berlin, Heidelberg.

Advanced Email Search in Small Enterprises

Štefan Dlugolinský[1], Michal Laclavík[1], Martin Šeleng[1], Marek Ciglan[1],
Martin Tomašek[2], Ladislav Hluchý[1]

[1]Institute of Informatics, Slovak Academy of Sciences,
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
{stefan.dlugolinsky, laclavik.ui, martin.seleng, marek.ciglan,
hluchy.ui}@savba.sk
[2]InterSoft a.s., Košice, Slovakia
martin.tomasek@intersoft.sk

Abstract. The paper presents an advanced search applicable on email collections, which is being developed in VENIS project and which can help small and micro enterprises to fill the gap of missing legacy CRM, ERP[1] or process management systems. The approach is based on reusing existing infrastructure and data – email collections and shared documents, allowing semantic search and recommendation in order to fulfil business tasks.

Keywords: search, small enterprises, lightweight semantics, email.

1 Introduction

Email communication with its interconnections to other organizational as well as public resources (e.g. LinkedData) can be a valuable source of information and knowledge for business intelligence, knowledge management or better enterprise and personal email search. The future of email [3] is in interconnecting email with other resources, services, data and entities, which are present in email.

This paper describes enterprise entity search based on email and document analysis, which can help small enterprises achieving their business tasks, especially those, which include fulfilling of some information need. A proof of concept implementation is provided on a VENIS[2] project InterSoft use case.

Presented search approach gains from our previous work, particularly from email analysis performed within the Commius project [2] and entity relation discovery discussed in [1]. In the Commius project, several types of NEs (Named Entities) have been extracting from emails, like people, organizations, addresses or contact data, but

[1] CRM - Customer Management System, ERP - Enterprise Resource Planning
[2] http://www.venis-project.eu/

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 23–26.

the extraction of business related entities like products, services or invoice numbers needed to be customized with the help of developers. The point of the entity relation discovery was in constructing semantic trees and networks from extracted entities and in use of a spreading activation algorithm. The paper is focused on combining our previous work with email analysis, indexing, full text and faceted search as well as connection to small enterprise business needs.

2 Use Case

The InterSoft use case is related to the allocation of development resources to various providers and customers at the same time. In such collaboration process, there is currently no information system used either by InterSoft or by the large providers subcontracting to InterSoft. E-mail communication is here the only common method to manage the collaboration as well as exchange and share the electronic materials (documents, software) related to the outsourced projects. This use case identifies three actors: 1. Supplier – SME, which offers a portfolio of simple services; 2. Customer – requires a complex service and in general looks for a large *Provider* of such services; 3. Provider – LE/SME, which provides a complex service and needs a *Supplier* for simple services to build the complex service for a *Customer*.

The use case covers activities such as matching requirements and specification to the profiles of involved parties, contacting the involved parties and can be extended further to formal ordering of services, invoicing and payments. Some of the interoperability activities are shown in chapter 3.3.

3 Early Enterprise Search Prototype

In this section an early enterprise search prototype developed in VENIS project is discussed. First, we describe email parsing, indexing and then search possibilities.

The implemented prototype is able to access several kinds of remote and local email collections. It uses Java Mail API[3] to access email collections by IMAP, SMTP and POP3 protocols. The prototype can access also local mbox and PST collections. PST collections are read using the java-libpst library[4].

A connected email collection is parsed email by email and each email is parsed part by part. If Java Mail API is used, then the part can be a message header (envelope) or a message body part (as defined in RFC 2882 and RFC 1341). If the java-libpst library is used to parse PST collection, then the part stands for a message envelope, body text or message attachment.

There are three types of output being parsed from email parser: *envelope*, *body text* and *attachment* (parsed email part, further in the text). Emails are parsed with the help of Apache Tika[5], which can recognize several file formats and is able to parse file

[3] http://www.oracle.com/technetwork/java/javamail/index.html
[4] https://github.com/rjohnsondev/java-libpst
[5] http://tika.apache.org/

metadata and its textual content. Each of the parsed email parts consists of metadata and textual content provided by Tika except the *envelope* part, which consists only of metadata built from message headers. Attachments are parsed recursively, because some types (e.g. zip archives) can contain other files. If such files are discovered during the recursive attachment parsing, they are treated just like they were a regular attachment, i.e. they are parsed as *attachment* part with metadata and textual content.

Additional lightweight semantic information is put into the *message* and *attachment* part metadata by Ontea, which extracts this information from the related textual content. After the metadata is enhanced by information extraction, the email represented by parsed parts with text and metadata is ready to be indexed.

After an email is parsed, an indexer is called to index the email. There is a special indexer implemented in our prototype, which builds indexing requests to Solr from parsed results. For each email being indexed, the indexer gets on input one *envelope* parsed part, one or more *body text* parsed parts and zero or more *attachment* parts. Parsed results are not directly put into the index, but they are pre-processed first.

There are two document types, which are put into the Solr index. The first is *message* and the second is *attachment*. The *message* document is built from the *envelope* parsed part and all the parsed *body text* parts. The metadata of the *envelope* and *body text* parts is merged into one and so all the *body text* parts textual content is concatenated. The *attachment* document is built from the *attachment* parsed parts by enhancing their metadata by *envelope* metadata.

A filter is applied on metadata, to filter and rename particular metadata being added into the *message* and *attachment* Solr documents as index fields. It is because not all the metadata fields are required in the index and also because Solr has a strict indexing schema. There can be various filtering and renaming rules defined based on regular expressions. On the other hand, Solr indexing schema allows dynamic fields – fields, which do not have to be declared explicitly, but can be declared by a prefix or suffix pattern. For instance entities extracted by Ontea have suffix "ann_" (i.e. annotation), so we can write a filter to pass entities matching the "ann_.+" pattern into the index as fields. Textual content is put into the index in a special field *text_general*, which is later processed in Solr by standard tokenizer, stop word filter and lower case filter. If the *message* and *attachment* Solr documents are ready, they are sent to Solr and indexed.

Early implementation of a search interface is depicted in Fig. 1. There is a search query input field (labeled *Find:*), which is used to search for documents or emails using full-text search. Dynamic fields, described in chapter 3.2, can be easily used here for full-text semantic search. For example, to find all emails and attachments, where a person Paul is mentioned: ann_Person:Paul*. One can use also numeric dynamic fields like *img_w* and *img_h* and use them to search for particular sized pictures included in email attachments. Search results can be filtered by facets constructed from email envelope metadata. Full-text search is integrated with an entity search, where related entities are displayed by clicking on *Enty Srch* link. Both full-text and entity searches should be combined and integrated even more later, but for now we have separate user interfaces for the full-text and the entity search. Entity search and recommendation have similar features to those used in Email Social

Network Search[6] developed in Commius project and being extended in VENIS [1]. The entity search interface is displayed in Fig. 1 at the bottom. There is a ranked list of recommended people returned on a basis of "Java developers" email context, in which personal skill entities were detected and used in search for the people.

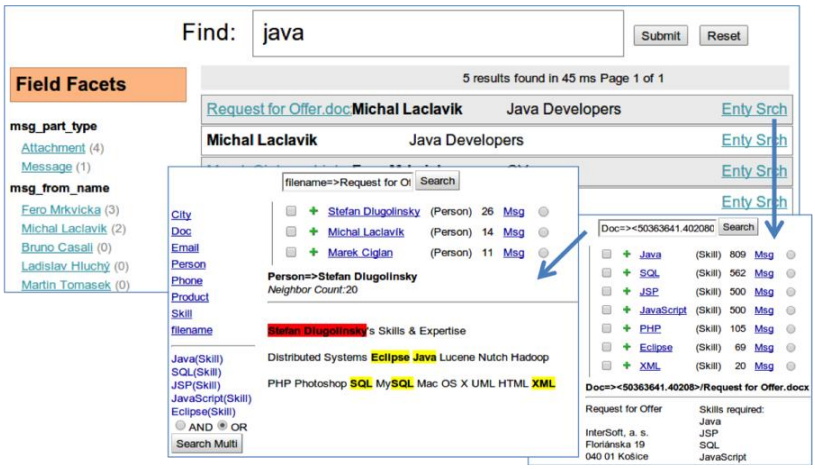


Fig. 1. Prototype Enterprise search user interface.

4 Conclusion

We have provided a proof of concept implementation of the enterprise search, which exploits the lightweight semantic graph and can help with business tasks communicated by email and documents exchanged. This was also tested on the concrete use case coming from VENIS project provided by InterSoft.

Acknowledgments. This work is supported by projects: VENIS FP7-284984, CLAN APVV-0809-11 and VEGA 2/0184/10.

References

1. Laclavík, M., Ciglan, M., Dlugolinský, Š., Šeleng, M. and Hluchý, L.: Emails as Graph: Relation Discovery in Email Archive. In Email2012 workshop, WWW 2012, April 16-20, 2012, Lyon, France, pages 841-846, 2012.
2. Laclavík, M., Dlugolinský, Š., Šeleng, M., Kvassay, M., Gatial, E., Balogh, Z. and Hluchý, L.: Email Analysis and Information Extraction for Enterprise Benefit. In Computing and informatics, 2011, vol. 30, no. 1, p. 57-87. ISSN 1335-9150
3. Faucette, M. (2012): The Future of Email Is Social. White Paper; IBM IDC report; ftp://ftp.lotus.com/pub/lotusweb/232546_IDC_Future_of_Mail_is_Social.pdf

[6] http://ikt.ui.sav.sk/esns/

Categorization based on Company Activities

Martin Repka, Ján Paralič

Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky,
Technická univerzita v Košiciach, Letná 9, 040 01 Košice, SR
{Martin.Repka, Jan.Paralic}@tuke.sk

Abstract. This paper approaches the issue of company categorization as part of understanding task in the data mining process on company networks' data. We describe the resources, machine learning methods and tools for data mining in data of company networks. Using classification algorithms and processes tries to compare several learners for classification models based on company categories in online catalogs and their business activities. In our experiments we will test the hypothesis about similarity of companies within categories mined with respect to their compositional data and with them provided by online catalogue.

Keywords: classification, data mining, rapid miner, company networks, data transformation, data correlation

1 Introduction

Company Network (CN) is such a social network, where actors are represented by companies and connections are representing interactions between them. CNs also contain a lot of nonstructural data as compositional data and temporal data.

In next chapters we will take a closer look on CN particularly on the specifics of compositional features of CN. We will look into possibilities of data mining in compositional attributes of CN and especially with problematic of company categorization. Besides social connections online sources contain a lot of information about companies such as company activities, contact details, etc.

2 Company categorization

Companies are usually registered in various online catalogs. A human is needed to enter a data about company into catalog like selection of category etc. We will create an experiment of data mining technique utilization to aid in fast company categorization process. We find it useful to predict category of company based on compositional data. This will be useful application of data mining, if such an automatically created categorization would fit manually created categorization in online catalogue.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 27–30.

2.1 Data and data sources used in experiment

From *Amadeus* database we retrieved compositional data about 25683 Slovak companies. Using the official number of each company we tried to find company in *Azet* online catalog and we found entries in 15705 cases. From these 15705 companies we selected our example set consisting of companies in 10 largest categories, which represents 2040 companies (ranging from 105 to 389 companies in category).

We decided to aggregate data as seen in Fig. 1 and then to use *RapidMiner* functionality. We first exported data from *ITLIS*[1] project (companies activities originating from *ORSR*) [1,2,3]. Another source of activities is professional company database *AMADEUS*[2], where semi-structured information about company activities can be found [4]. As "human input" we decided to create crawler, which retrieves categorization from popular company catalog *AZET*[3] using company's official number (ICO). Categories in this catalog are created by companies (moderated by administrators) and every company has to input data about itself and manually select categories, in which company actively acts.

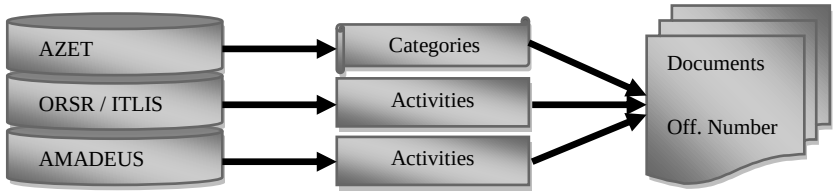


Fig. 1. Data integration, process of creating documents for RapidMiner.

2.2 Document processing in RapidMiner

Each document is tokenized, where text of a document is split into a sequence of tokens using non-letter characters. In next step stopwords and tokens by with length < 4 or > 25 are removed and then stemming is applied. Stopwords are removed by removing every token which equals a stopword from the built-in stopword list. Stemming process stems words by applying stemming algorithms written for the Snowball language. This process is presented on Fig. 2 as "Process Documents" subpart.

3 Classification of documents into categories

For classification we used several model learners. After learning we evaluated their performance using 10-fold cross-validation and we counted the average performance.

Default model. This learner creates a model that will simply predict a default value for all examples, i.e. the average or median of the true labels (or the mode in case of classification). This learner can be used to compare the results of "real" learning

[1] www.itlis.eu
[2] https://amadeus.bvdinfo.com
[3] http://www.azet.sk/#firmy

schemes with guessing. Guessing based on mode values produced results of performance with accuracy: 19.07% +/- 0.15%.

3.1 Used model for classification

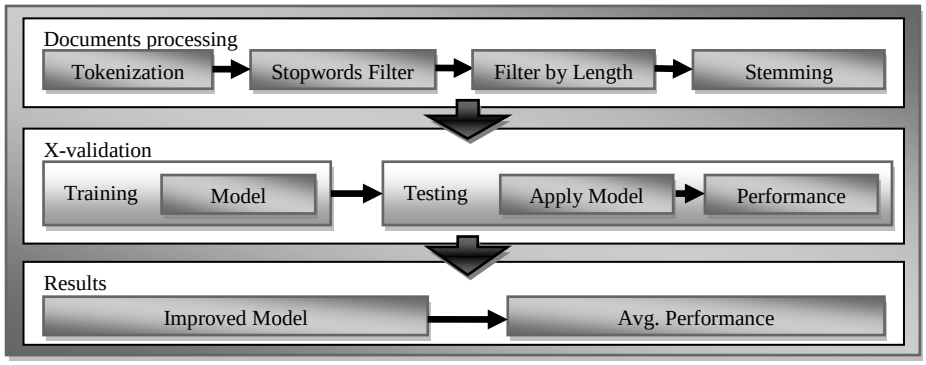


Fig. 2. Process of classification of documents into categories (text mining).

Naive Bayes and Naive Bayes (Kernel). Naive Bayes learner returns classification model using estimated normal distributions. It uses Laplace correction to prevent high influence of zero probabilities. Kernel learner is classification model using estimated kernel densities and it also uses Laplace correction.

Decision Tree. This model learner learns decision trees from both nominal and numerical data. In order to classify an example, the tree is traversed bottom-down. This decision tree learner works similar to Quinlan's C4.5 or CART.

Neural Network (NN) model. This learner learns a model by means of a feed-forward neural network trained by a backpropagation algorithm (multi-layer perceptron). We use one sigmoid type hidden layer of size calculated from the number of classes of the example set. In this case, the layer size is set to *(number of classes + 2)*. The used activation function is the usual sigmoid function and type of the output node is sigmoid for the learning data of a our task. NN was trained for 1000 cycles.

K-Nearest Neighbour (K-NN) Model. Classification with k-NN based on an explicit similarity measure, which uses a k nearest neighbor implementation.

Support Vector Machine (SVM). We used C-SVM with linear type of kernel function from libsvm. In contrast to other SVM learners, the libsvm supports also internal multiclass learning and probability estimation based on Platt scaling for proper confidence values after applying the learned model on a classification data set [5].

3.2 Evaluation and Results

Performance of used learners is summarized in Table 1. Best performance achieved SVM learner (acc.: 79.26 %) with low time complexity, which is very suitable in real time applications. NN model learner achieved also very good results in creating model of accuracy of 78.48 %, but NN learning is very time consuming. Still efficient

model learners were Kernel Naive Bayes, K-NN and Decision Tree model learners with model accuracy over 67 %.

Table 1. Performance (accuracy) of model learners.

Model	Accuracy	Model learned in
SVM (C-SVM-Linear)	79.26% +/- 2.55%	Seconds
Neural Network	78.48% +/- 3.06%	Tens of minutes
Kernel Naive Bayes	75.25% +/- 2.13%	Seconds
K-NN (1-nearest)	69.46% +/- 1.83%	Seconds
Decision Tree	67.16% +/- 7.32%	Minutes
Naive Bayes	46.47% +/- 1.98%	Seconds
Default model (Mode)	19.07% +/- 0.15%	Seconds

4 Conclusions

In this paper we proposed data integration from several sources and experimentally verified correlation of compositional data (company activities) and online categorization. We tried several model learners and compared their time complexity and model accuracy. As the very efficient model learners seems to be SVM, Neural Network and Naive Bayes (Kernel) model. This information could be useful as support to other analyses of company networks like we proposed in [6], [7].

Acknowledgement. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10 (50%) and partially by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grant No. 1/1147/12 (50%).

References

1. Ministry Of Justice Of The Slovak Republic, “Business Register On Internet,” Online at www.orsr.sk, 31-Oct-2011
2. M. Repka, “Analýza určitých typov sociálnych sietí. Písomná práca k dizertačnej skúške.” TU v Košiciach, Fakulta elektrotechniky a informatiky, Košice, Chapter 4, 2011.
3. P. Kostelník, P. Smatana, Itlis, “Transparent information in context,” Online at www.itlis.eu, Accessed at 31-Oct-2011
4. Bureau van Dijk: Amadeus. A database of comparable financial information for public and private companies across Europe. Accessed at 08-Sep-2012. Online at http://www.bvdinfo.com/Products/Company-Information/International/AMADEUS.aspx
5. RapidWiki. Online at http://rapid-i.com/wiki/index.php?title=Main_Page
6. Repka, M.: Component Identification in Company Networks. In SCYR 2012 - 12th Scientific Conference of Young Researchers – FEI TU of Košice, Košice (2012).
7. Repka, M.: Local Structure Analysis of Company Network. In IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics, Košice (2012).
8. Jánošík, T.: Klasifikácia a zhlukovanie aktérov v sociálnych sieťach. Bakalárska práca. Technická Univerzita v Košiciach. Košice (2012).

Doménová závislosť automatickej sumarizácie textu

Róbert Móro, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava, Slovensko
{moro, bielik}@fiit.stuba.sk

Abstrakt. Automatická sumarizácia textu pomáha orientovať sa v rozsiahlych informačných priestoroch. Uplatnenie si nachádza v rôznych doménach - pri vyhľadávaní na webe, v novinových portáloch, digitálnych knižniciach, ale aj vo vzdelávaní či v doméne programovania. Dokumenty v rôznych doménach sa pritom navzájom líšia v dĺžke, štruktúre (a štruktúrovanosti), v type a spôsobe prezentácie informácií. Text môže byť v rôznej miere dopĺňaný informáciami v podobe obrázkov, tabuliek, ukážok zdrojových kódov a pod. V príspevku analyzujeme závislosť výslednej sumarizácie od zvolenej domény a diskutujeme spôsob prispôsobenia nami navrhnutej metódy personalizovanej sumarizácie.

Kľúčové slová: automatická sumarizácia textu, personalizácia, doménová závislosť

1 Úvod

Pri každodennej práci na webe, ale aj v akomkoľvek inom rozsiahlom informačnom priestore, musíme čeliť problému zahltenia informáciami. Jedným z prístupov zameriavajúcich sa na riešenie tohto problému je automatická sumarizácia textu. Hlavnou myšlienkou sumarizácie je poskytnúť krátky súhrn najdôležitejších informácií obsiahnutých v danom dokumente (množine dokumentov), na základe ktorého je možné rozhodnúť, či je daný dokument pre nás relevantný a mali by sme si ho prečítať celý, alebo nie. Sumarizáciu možno pritom využiť v rôznych doménach – od vyhľadávania na webe, cez digitálne knižnice až po doménu programovania a zdrojových kódov.

Výsledkom (extraktívnej) sumarizácie je *extrakt* – výber najrelevantnejších viet pôvodného dokumentu, pričom existuje množstvo metód určenia relevantnosti pojmov (termov) a viet v dokumentoch. Najjednoduchší (a najstarší) spôsob spočíva v určení relevantnosti pojmov na základe frekvencie ich výskytu v dokumente [6]. Uvažujú sa aj ďalšie indikátory relevantnosti, napr. poloha pojmu [3]. Často používanou metódou pri sumarizácii textu je latentná sémantická analýza (LSA) [4], a to najmä vďaka jej schopnosti nájsť dôležité koncepty zachytené v dokumente ako spoločný výskyt viacerých pojmov. Do súhrnu potom vyberáme vety s najväčším zastúpením najdôležitejších takto identifikovaných konceptov [10].

Okrem samotného textu môžeme pri sumarizácii uvažovať aj ďalšie informácie, ktoré nám pomáhajú lepšie identifikovať relevantné pojmy alebo fragmenty doku-

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 31–34.

mentu. Ak medzi tieto informácie zahrnieme aj charakteristiky a potreby používateľa, hovoríme o personalizovanej sumarizácii [2]. Výhodou personalizovanej sumarizácie oproti generickému variantu je, že odráža záujmy, znalosti či ciele konkrétneho používateľa, a tak dokáže lepšie uspokojiť jeho aktuálnu informačnú potrebu.

Navrhli sme metódu personalizovanej sumarizácie, ktorá využíva a navzájom kombinuje viaceré zdroje informácií o používateľovi a doméne [7]. Pri jej experimentálnom overení sme sa zamerali na doménu výučby, avšak navrhnutá metóda je dostatočne všeobecná, aby sme ju mohli použiť a prispôsobiť aj pre iné domény.

2 Metóda personalizovanej sumarizácie

Pri návrhu metódy sme vychádzali z metódy latentnej sémantickej analýzy, pričom sme identifikovali krok konštrukcie matice pojmov a viet ako vhodný pre prispôsobenie a personalizáciu sumarizácie. Konštruujeme teda personalizovanú maticu pojmov a viet, v ktorej sú váhy pojmov vo vetách určené na základe nami navrhnutej váhovacej schémy, ktorá rozširuje klasickú metódu tf-idf:

$$w(t_{ij}) = \sum_k \alpha_k R_k(t_{ij}), \tag{1}$$

kde $w(t_{ij})$ je váha pojmu t_{ij} matice a α_k je lineárny koeficient hodnotiča R_k. Hodnotič R_k je funkcia, ktorá pre zadaný term z množiny kľúčových slov T vráti jeho ohodnotenie (reálne číslo). Takýto návrh je modulárny, pretože umožňuje pridávať a meniť jednotlivé hodnotiče. Navrhli sme sadu *generických* a *personalizovaných* hodnotičov, ktoré využívajú v doméne výučby nami identifikované tri hlavné zdroje personalizácie a prispôsobovania sumarizácie:

- *Konceptualizáciu domény* – model domény zachytáva znalosť expertov v podobe dôležitých konceptov (relevantných doménových pojmov) a vzťahov medzi nimi
- *Znalosti používateľov* – sú modelované vo výučbových systémoch a ich zohľadnenie pri sumarizácii je vhodné predovšetkým, ak uvažujeme ako scenár použitia opakovanie získaných vedomostí
- *Anotácie pridané používateľmi* – zamerali sme sa najmä na zvýraznenia, ktoré indikujú, že dané fragmenty sú nejakým spôsobom pre používateľa dôležité a chcel by sa k nim v budúcnosti vrátiť

Za účelom overenia navrhnutej metódy sme uskutočnili dva experimenty na predmetoch *Funkcionálne a logické programovanie* a *Princípy softvérového inžinierstva* s pomocou výučbového systému ALEF[1]. Kvalitu sumarizácií hodnotili používatelia systému, t.j. študenti na päťstupňovej škále, resp. pomocou priameho porovnania dvoch zobrazených variantov. Okrem toho odpovedali na otázky ohľadom čitateľnosti a zrozumiteľnosti sumarizácií, vhodnosti vybraných viet či celkovej dĺžky súhrnov.

V prvom experimente [7] sme sa zamerali na porovnanie sumarizácie zohľadňujúcej relevantné doménové pojmy s generickým variantom (zohľadňujúcim len frekven-

[1] http://alef.fiit.stuba.sk

ciu pojmov a ich polohu). V druhom experimente [8] sme voči generickému variantu vyhodnotili sumarizáciu zohľadňujúcu zvýraznenia používateľov. Výsledky oboch experimentov naznačili, že zohľadnenie dodatočných informácií, či už v podobe relevantných doménových pojmov alebo v podobe zvýraznení, prináša zlepšenie oproti generickému variantu.

3 Prispôsobovanie sumarizácie zvolenej doméne

Vďaka jednoduchosti navrhnutej metódy je možné ľubovoľne ju rozširovať o nové hodnotiče, navzájom ich kombinovať, a tak prispôsobovať sumarizáciu pre rôzny kontext, doménu a scenár použitia. Každá doména má pritom svoje špecifiká, ktoré treba zohľadniť. Dokumenty z rôznych domén sa môžu odlišovať v dĺžke, štruktúre (a štruktúrovanosti) a v type a množstve fragmentov dopĺňajúcich samotný text, ako sú napr. obrázky, tabuľky (s ich opismi), či časti zdrojových kódov alebo matematických vzorcov. Prispôsobenie metódy sumarizácie zvolenej doméne teda znamená:

- Zvoliť spôsob predspracovania textu
- Identifikovať vhodné zdroje informácií (hodnotičov)
- Nastaviť parametre kombinácie
- Zvoliť dĺžku výstupného súhrnu

Ak uvažujeme napr. doménu programovania a zdrojových kódov, kde by nám sumarizácia mohla pomôcť pri zefektívnení vyhľadávania v zdrojových kódoch [1], predspracovanie textu bude nutne iné ako napr. v prípade novinových či vedeckých článkov alebo výučbových textov. Zatiaľ čo v prípade článkov ide o texty v prirodzenom jazyku, zdrojové texty sú písané v niektorom z programovacích jazykov. Možno tu však nájsť analógie – názvy tried a metód sú v pozícii nadpisov a podnadpisov v texte, kľúčové slová programovacieho jazyka sú zas väčšinou v pozícii stop slov, príkazy sú oddeľované špeciálnymi znakmi podobne ako vety a sú organizované do blokov kódu podobne ako vety do odsekov.

Jedným z dôležitých zdrojov informácií je tak znalosť štruktúry dokumentov v danej doméne. Napr. v doméne výučby, v ktorej sme overovali navrhnutú metódu, nadpis väčšinou dobre vystihuje opisovanú tému, na úvod kapitoly je opísané, čo bude jej obsahom a na záver býva jej zhrnutie. Túto znalosť sme využili na zlepšenie extrakcie relevantných viet pomocou hodnotiča polohy pojmov. V doméne digitálnych knižníc a vedeckých článkov majú dôležité postavenie citácie, ktoré je možné použiť ako jeden z významných zdrojov informácií o sumarizovanom dokumente [9]. V doméne programovania väčšinou názvy metód vystihujú, čo je ich úlohou, podobne dôležité sú vysvetľujúce komentáre. Pritom pojmy v nich obsiahnuté sa môže v tele metódy vyskytovať s nízkou frekvenciou [5]. Je preto vhodné dať im pri sumarizácii vyššiu váhu pomocou hodnotiča polohy pojmov.

Pri experimentálnom overení metódy sme parametre kombinácie ako aj dĺžku sumarizácie nastavovali manuálne. Správne nastavenie pre rôzne dokumenty v rôznych doménach je však náročné, preto je vhodné tento proces automatizovať. Pre každý dokument môžeme určiť množinu čŕt: dĺžka dokumentu, počet odsekov a viet, kategó-

ria, resp. doména, typy netextových fragmentov (obrázkov, tabuliek atď.); ďalšie črty vieme určiť pre jednotlivých používateľov, ktorým sa má sumarizácia personalizovať (úroveň znalostí v doméne, ciele, kontext), ako aj pre jednotlivé hodnotiče – úroveň dôvery (istoty) v daný zdroj informácií. Tieto extrahované črty môžu byť vstupom pre niektorý z algoritmov strojového učenia (napr. rozhodovacie stromy, Bayesove siete), ktorého úlohou bude optimalizovať jednotlivé parametre kombinácie na základe hodnotení výsledných sumarizácií používateľmi.

Takýmto rozšírením nami navrhnutej metódy dosiahneme, že bude schopná dynamicky reagovať na zmenu kontextu alebo domény. Toto je obzvlášť dôležité pri agregácii informácií z rôznych typov zdrojov a ich prezentácii používateľovi v podobe sumarizácie, napr. pri prieskumnom vyhľadávaní. Z hľadiska ďalšieho smerovania plánujeme overiť schopnosť navrhnutej metódy prispôsobiť sumarizáciu niektorej vybranej doméne; ako vhodná a zaujímavá sa na tento účel javí doména zdrojových kódov alebo digitálnych knižníc.

Poďakovanie. Táto práca vznikla vďaka čiastočnej podpore projektov VG1/0971/11 a APVV-0208-10.

Literatúra

1. Bieliková, M. et al.: Webification of software development: General outline and the case of enterprise application development. Proc. of the 3rd World Conf. on Inf. Tech., vol. 4. Procedia Tech. (2012).
2. Díaz, A., Gervás, P.: User-model based personalized summarization. J. of Inf. Processing and Management, vol. 43, no. 6, pp. 1715-1734. Pergamon Press, Tarrytown (2007).
3. Edmundson, H.P.: New methods in automatic extracting. J. of the ACM, vol. 16, no 2, pp. 264-285. ACM Press, New York (1969).
4. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. Proc. of the 24th Int. ACM SIGIR Conf. on Research and Development in Inf. Retrieval, pp. 19-25. Springer, Berlin (2001).
5. Haiduc, S., Aponte, J., Marcus, A.: Supporting program comprehension with source code summarization. Proc. of the 32nd ACM/IEEE Int. Conf. on Software Engineering, pp. 223-226. ACM Press, New York (2010).
6. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. of Research Development, vol. 2, no. 2, pp. 159-165, IBM Corp. Riverton, New York (1958).
7. Móro, R., Bieliková, M.: Personalized text summarization based on important terms identification. Proc. of the 23rd Int. Workshop on Database and Expert Systems Applications, pp. 131-135. IEEE CS Press, Los Alamitos (2012).
8. Móro, R., Bieliková, M.: Combinations of different raters for text summarization. Information Sci. and Tech. Bulletin of the ACM Slovakia, vol. 4, no. 2, pp. 56-58. (2012).
9. Qazvinian, V., Radev, D.R.: Scientific paper summarization using citation summary networks. Proc. of the 22nd Int. Conf. on Comp. Linguistics, pp. 689-696. Association for Computational Linguistics, Stroudsburg (2008).
10. Steinberger, J., Ježek, K.: Text summarization and singular value decomposition. Proc. of Advances in Inf. Systems, pp. 245-254. Springer, Berlin (2005).

Analýza a spracovanie textu

Určovanie kľúčových slov pre odvíjajúce sa príbehy pomocou roja sociálnych agentov

Pavol Návrat, Štefan Sabo

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita, Bratislava
{navrat,sabo}@fiit.stuba.sk

Abstrakt. V tomto článku predstavujeme metódu, ktorá si kladie za cieľ umožniť sledovanie spravodajských príbehov, odvíjajúcich sa na Webe, bez potreby strojového učenia, či analýzy dátového korpusu. Za účelom rozlíšenia jednotlivých spravodajských tém využívame roj sociálne sa správajúcich agentov, ktoré prechádzajú spravodajskými článkami na Webe a vyhľadávajú kľúčové slová, na základe ktorých je možné prepájať súvisiace články. Následne predstavujeme výsledky experimentu s predstavovanou metódou na Webe. Na záver zhŕňame doposiaľ dosiahnuté výsledky a načrtávame ďalší smer nášho úsilia v tejto oblasti.

1 Úvod

Spravodajstvo na Webe je v dnešnej dobe rozšírené a čoraz viac ľudí siaha po Webe ako po hlavnom zdroji informácií. Populárne výrazy, ako „*kríza v EU*“, či „*konflikt v Sýrii*“ si vieme vďaka ich rozšírenosti jednoducho spojiť s reálnymi správami a udalosťami. Čo však robí výraz dostatočne reprezentatívnym, aby ho bolo jednoznačne možné spojiť s určitým príbehom a ako by sme mohli určovať takéto populárne výrazy strojovo? Našou snahou je vytvoriť systém, ktorý bude schopný poskytnúť odpovede na tieto otázky. Za týmto účelom využívame sadu agentov, inšpirovaných sociálnym hmyzom, ktoré prechádzajú jednotlivé články a vyhľadávajú kľúčové slová, ktoré majú potenciál prepájať rôzne články. Takto získavame kľúčové slová, relevantné pre skupiny článkov, ktoré reprezentujú jednotlivé príbehy, alebo udalosti, na rôznej úrovni abstrakcie. Výhodou navrhovaného prístupu je, že nevyžaduje znalosť tematickej štruktúry problémovej oblasti, ani strojové učenie a je možné ho aplikovať aj priebežne na rozširujúcu sa skupinu článkov, bez toho, aby sme mali dostupný celý korpus.

2 Súvisiace práce

Väčšina existujúcich techník pre získavanie kľúčových slov využíva buď učenie s učiteľom, alebo pracuje s korpusom dát. Lee a Kim [1] predstavili metódu, ktorá získava kľúčové slová pre vopred známu sadu spravodajských článkov na základe váhovania podľa TF-IDF. Ak nemáme k dispozícii informácie o tematickej príslušnosti článkov, môžeme štruktúru tém stanoviť na základe klastrovacích algoritmov [2]. Alternatívou k spomínaným prístupom je metóda, ktorú

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 37–40.

navrhol Bohne v [3]. Táto metóda využíva Helmholtzov princíp. Každý dokument je rozdelený na menšie časti a výskyt jednotlivých slov je porovnaný s vopred odhadovanou hodnotou. Prípadná odchýlka vo frekvencii výskytu slov potom indikuje potenciálne kľúčové slovo. Tento prístup je však možné využiť iba pre text v ktorom je výskyt jednotlivých výrazov štatisticky nezávislý, čo neplatí pre spravodajské články.

Metafora včelieho úľa (*Beehive metaphor*, *BM*), ktorú v práci využívame ako model sociálneho správania sa agentov, bola navrhnutá v [4] ako všeobecný model koordinácie agentov v rámci multi-agentového systému. Návrat a Kováčik [5] následne navrhli prístup, ktorý využíva BM ako základ pre webový vyhľadávací stroj. Ďalšie spôsoby využitia BM ponúka [6], kde je načrtnutý spôsob využitia BM pre optimalizáciu funkcií a hierarchickú optimalizáciu.

3 Získavanie kľúčových slov

Za účelom získania kľúčových slov reprezentujúcich odvíjajúce sa príbehy na Webe využívame sadu agentov, inšpirovaných sociálnym hmyzom. Agenty pri zbere údajov napodobňujú správanie včiel medonosných podľa metafory včelieho úľa [4]. Na rozdiel od prístupu popísaného v [5] však ako zdroje nektáru pre včely neslúžia webové stránky, ale kľúčové slová.

Agent môže počas svojho pôsobenia zastávať tri rôzne úlohy, *zber*, *tancovanie* a *pozorovanie*. Každému agentovi je počas inicializácie priradený náhodný článok a zvolí si náhodného kandidáta na kľúčové slovo, ktorého bude následne hodnotiť. Počas zberu agenty navštevujú články a určujú mieru relevancie svojho kľúčového slova. V prípade, že agent nájde článok, vzhľadom na ktorý je nesené kľúčové slovo dostatočne relevantné, môže sa rozhodnúť, že vytvorí prepojenie medzi týmto článkom a pôvodným článkom, z ktorého bolo nesené kľúčové slovo získané. Podrobnejšie ilustruje priebeh zberu Algoritmus 1.

V prípade, že sa agent rozhodne vytvoriť väzbu medzi dvoma článkami, zahajuje fázu tancovania. Tancovanie je signálom, že agent identifikoval podobnosť medzi dvoma článkami na základe kľúčového slova, ktoré aktuálne nesie. V takomto prípade agent preruší zber a bude propagovať nájdenú kombináciu článku a kľúčového slova. Ostatné agenty, ktoré momentálne nenesú žiadne kľúčové slovo sa môžu rozhodnúť, že si takúto kombináciu osvoja. V takomto prípade tento nový agent preberá daný článok a dané kľúčové slovo a snaží sa nájsť ďalšie články, pre ktoré bude toto kľúčové slovo relevantné. Myšlienkou tohto mechanizmu je, že populárne kľúčové slová, u ktorých je predpoklad, že budú relevantné pre širší spravodajský príbeh, dostanú priestor, aby oslovili širšiu skupinu agentov a boli kvalitnejšie analyzované, zatiaľ čo menej relevantné kľúčové slová budú priebežne opúšťané, alebo analyzované iba užšou skupinou agentov. Treťou možnosťou, ako sa agent môže zachovať je, že sa rozhodne svoje kľúčové slovo opustiť, ak pozoruje, že relevancia navštevovaných článkov je nízka. Takýto agent si následne môže osvojiť nové kľúčové slovo z palety aktuálne propagovaných slov, alebo si nové kľúčové slovo zvolí náhodne.

Algoritmus 1 Fáza zberu pre jednu včelu

Require: aktuálny článok a_c, kľúčové slovo key
1: nový článok $a_n \leftarrow selectNewArticle()$
2: $r_c \leftarrow relevance(key, a_c)$
3: $r_n \leftarrow relevance(key, a_n)$
4: $p = min(r_c, r_n)$
5: Rozhodnutie, či opustíme článok s pravdepodobnosťou $1 - p$.
6: **if** chceme opustiť článok **then**
7: $status \leftarrow$ „sledovanie“ Opúšťame článok a začíname sledovať.
8: **else**
9: Rozhodnutie, či budeme tancovať s pravdepodobnosťou p.
10: **if** chceme tancovať **then**
11: $newConnection(a_c, a_n)$ Prepojíme články.
12: $status \leftarrow$ „tanec“ Začíname tancovať za aktuálny článok a kľúčové slovo.
13: **else**
14: Pokračujeme v zbere.
15: **end if**
16: **end if**

4 Experiment

Za účelom overenia správania sa navrhovaného modelu sme vykonali experiment, v rámci ktorého sme počas deviatich dní zbierali kľúčové slová zo spravodajského webu Reuters[1]. Každý deň sme vykonali nezávislý zber a pre jednotlivé kľúčové slová sme sledovali počty článkov, pre ktoré boli dané kľúčové slová identifikované ako relevantné. Dĺžka každého zberu bola obmedzená na 100 iterácií a použité parametre modelu BM boli $BIOR = 0$, $BISB = 100$, $MDT = 4$ a $OT = 2$. Charakteristiku a význam parametrov modelu je možné nájsť v [5].

Počas deviatich dní sme identifikovali 298 jedinečných kľúčových slov, pričom 99 z toho boli vlastné podstatné mená, čo predstavuje 33.22%. Pre každý článok mohlo byť identifikovaných viac kľúčových slov. Na základe počtu článkov, pre ktorých bolo kľúčové slovo identifikované, sme kľúčové slová zoradili podľa popularity. Prehľad najpopulárnejších kľúčových slov relevantných pre spravodajské príbehy znázorňuje Tabuľka 1, spolu so súhrnným počtom článkov n^k, pre ktoré bolo kľúčové slovo k identifikované a percentuálne zastúpenie kľúčových slov v celej získanej vzorke n^k/N.

Môžeme pozorovať, že medzi najlepšie hodnotenými kľúčovými slovami majú vlastné podstatné mená výrazne lepšie zastúpenie, než v celej získanej vzorke, čo zodpovedá intuitívnemu predpokladu, že vlastné podstatné mená sa často používajú ako identifikátory, ktoré je možné spojiť s konkrétnymi udalosťami v priestore spravodajských príbehov. Je potrebné dodať, že v malom počte boli identifikované aj všeobecne populárne výrazy, ktoré nie je možné jednoznačne spojiť so spravodajským príbehom, ako „*oil*“ a „*economy*“, avšak zďaleka nedosahovali popularitu kľúčových slov prezentovaných v Tabuľke 1.

[1] http://www.reuters.com/

Tabuľka 1. Najpopulárnejšie kľúčové slová.

k	n^k	$n^k/N[\%]$	*keyword*	n^k	$n^k/N[\%]$	*keyword*	n^k	$n^k/N[\%]$
Syria	189	7.32	shooting	64	2.48	attack	47	1.82
Euro	79	3.06	China	63	2.44	trial	38	1.47
Apple	74	2.86	Colorado	55	2.13	murder	33	1.28
Egypt	83	3.21	court	54	2.09	Libor	29	1.12
Afghan	61	2.36	Samsung	50	1.94	Aleppo	26	1.01

5 Záver

Cieľom prezentovanej metódy je získať zo spravodajského webu kľúčové slová, ktoré sú relevantné pre aktuálne sa odvíjajúce spravodajské príbehy. Pre tento účel využívame sadu agentov, ktorí prechádzajú článkami a vyhľadávajú vyhovujúce kľúčové slová na základe ich porovnávania so sadou článkov, ktoré pri svojej ceste navštívia. Najlepšie kľúčové slová sú propagované a vyhodnotené dôkladnejšie, zatiaľ čo nezaujímavé kľúčové slová sú opustené. Z vykonaných experimentov sa javí, že navrhovaný prístup je schopný získavať kľúčové slová relevantné pre spravodajské príbehy, bez nutnosti predchádzajúceho učenia, alebo dostupnosti celého korpusu článkov. Nevýhodou tohto prístupu je, že uprednostňuje kľúčové slová spojené s dlhodobými a všeobecnými príbehmi, ako je „*Syria*“, alebo „*Euro*“, pričom nezohľadňuje aktuálnosť článkov. Menšie udalosti sa teda presadzujú horšie, aj keď môžu byť krátkodobo populárne. Toto je v súčasnosti predmetom ďalšej práce, pričom snahou je zaviesť hierarchický pohľad a štrukturovať príbehy podľa úrovne všeobecnosti kľúčových slov.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0971/11 Získavanie, spracovanie, vizualizácia textových inf. na základe analýzy relácií podobnosti, projektom VG1/1221/12 Pokročilé metódy v evolúcii softvéru: varianty, kompozícia a integrácia a projektom APVV–0208–10 Kognitívne cestovanie po digitálnom svete webu a knižníc s podporou pers. služieb a soc. sietí.

Literatúra

1. Lee, S., Kim, H.J.: News keyword extraction for topic tracking. In: Netw. Comp. and Adv. Inf. Management, 2008. NCM '08. 4th Int. Conf. Vol. 2. 554–559 (2008)
2. Turney, P.D.: Learning algorithms for keyphrase extraction. Information Retrieval **2** 303–336 (2000)
3. Bohne, T., Rönnau, S., Borghoff, U.M.: Efficient keyword ext. for meaningful doc. perception. In: Proc. of the 11th ACM symp. on Doc. eng. ACM 185–194 (2011)
4. Navrat, P.: Bee hive metaphor for web search. Communication and Cognition-Artificial Intelligence **23**(1-4) 15–20 (2006)
5. Navrat, P., Kovacik, M.: Web search engine as a bee hive. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. WI '06, Washington, DC, USA, IEEE Computer Society 694–701, (2006)
6. Navrat, P., Jelinek, T., Jastrzembska, L.: Bee hive at work: A problem solving, optimizing mechanism. In: Nature Bio. Ins. Comp. World Congress 122 –127 (2009)

Recognizing, Classifying and Linking Entities with Wikipedia and DBpedia

Milan Dojchinovski[1,2] and Tomáš Kliegr[2]

[1] Web Engineering Group, Faculty of Information Technology,
Czech Technical University in Prague
milan.dojchinovski@fit.cvut.cz
[2] Department of Information and Knowledge Engineering,
Faculty of Informatics and Statistics,
University of Economics, Prague, Czech Republic
tomas.kliegr@vse.cz

Abstract. In this paper we present system for entity recognition and classification. Entity candidates are recognized in the input text with a JAPE grammar. Hypernym for the entity is discovered from Wikipedia article describing the entity also with a JAPE grammar; this hypernym is the classification result. Both the entity and its hypernym are cross-linked with their representation in DBpedia and the result is published in the machine-readable NIF format.

Keywords: Named Entity Recognition, Linked Data, Wikipedia

1 Introduction

In this paper we present our system for entity recognition and classification. First, using a rule-based JAPE grammar [1] entity candidates are extracted from the input text. For each candidate its hypernym is discovered by our previously developed Targeted Hypernym Discovery (THD) algorithm [3] from the full text of Wikipedia articles describing the entity. Each entity candidate and its hypernym are cross-linked with their representation in DBpedia. Finally, the extracted entities and hypernyms are published in the machine-readable NIF format and contributed to the Linked Data cloud.

The system is implemented as a Web 2.0 applicationon top of the GATE framework[1]; a REST API is also exposed. The architecture of our tool is illustrated in Figure 1. The tool is available at http://ner.vse.cz/thd.

2 Entity Candidate Extraction

The first phase of our algorithm accepts free text and outputs a list of *named entities* and what we call *common entities*. As a named entity we consider each

[1] http://gate.ac.uk

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 41–44.

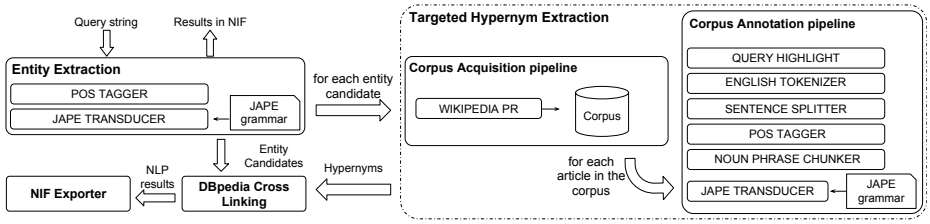


Fig. 1. Architecture overview.

match of the following JAPE pattern: `NNP+`, where NNP is a proper noun. An example of a named entity is "Diego Maradona". Common entities are recognized with pattern `JJ* NN+`, where JJ is an adjective and NN is a noun. An example of a common entity is "footballer" or "hockey player".[2]

3 Targeted Hypernym Discovery

The hypernym discovery approach proposed here is based on the application of hand-crafted lexico-syntactic patterns. Lexico-syntactic patterns were in the past primarily used on larger text corpora with the intent to discover all word-hypernym pairs in the collection. On this task, the state-of-the-art algorithm of Snow [5] achieves F-measure of 36 %. However, with THD we apply lexico-syntactic patterns on a *suitable document* with the intent to extract *one hypernym* at a time. This approach is more successful – e.g. [4] report F1 measure of 0.851 with precision 0.969.

The hypernym returned by the THD algorithm can be considered as the type of the entity represented by the noun phrase extracted from the text. The details relating to the THD algorithm can be found in [3].

4 Linking Entities and Hypernyms with DBpedia

Existing approaches often exploit Wikipedia through DBpedia – either through its SPARQL interface or the RDF dumps. In our approach, we introduce a new way of discovering links to DBpedia by utilizing the Wikipedia Search API, which provides access to Wikipedia's Lucene-based full text search.[3] Since each article in Wikipedia is represented as a resource in DBpedia, we first search for an article describing a hypernym or entity in Wikipedia and the search result is used to map the Wikipedia article URL to a DBpedia URL describing the hypernym or entity.

The advantage of our approach over other algorithms is that Wikipedia search uses a PageRank-like algorithm for determining the importance of the article in addition to the textual match with the query.

[2] Note that both "*" and "+" in JAPE grammar are greedy operators.
[3] http://www.mediawiki.org/wiki/Extension:Lucene-search

5 Example and NIF Export

The web-service interface of our THD tool provides results in the RDF-based NLP Interchange Format (NIF) [2] which utilize the String Ontology[4] (`str` prefix) as well as the Structured Sentence Ontology[5] (`sso` prefix).

Referring to the example in Listing 1, the *input plain text* received for processing on line 1 is considered as a *context* formalized using the OWL class **str:Context** (lines 1-3), within which entity candidates and their hypernyms need to be retrieved. Entity candidate, here string "Diego Armando Maradona Franco" (the second entity "Argentina" is omitted for space reasons), is treated as an offset-based string within the context resource (lines 4-10). Using the **sso:oen** the entity is connected with its representation in DBpedia (line 7). Finally, the hypernym is attached to the entity as its type (line 11).

```
1  :offset_0_100_Diego+Armando+Maradona+Franco+is+from+Argentina.
2    rdf:type str:Context ;
3    str:isString "Diego Armando Maradona Franco is from Argentina." ;
4  :offset_0_29_Diego+Maradona
5    rdf:type str:String ;
6    str:referenceContext :offset_0_100_Diego+Armando+Maradona+Franco+is+from+
         Argentina. ;
7    sso:oen dbpedia:Diego_Maradona ;
8    str:beginIndex "0" ;
9    str:endIndex "29" .
10   str:isString "Diego Armando Maradona Franco" ;
11 dbpedia:Diego_Maradona rdf:type dbpedia:Manager .
```

Listing 1. Excerpt of a NIF export.

6 Experimental Evaluation

The experimental evaluation was performed on a manually created *"Czech Traveler Dataset"*, based on textual annotations from a travelogue-like photo collection. There are 101 named entities and 85 common entities. The experiments were run for the THD algorithm and three other state-of-the-art tools: DBpedia Spotlight (DB), Open Calais (OC) and Alchemy API (ALC). We used the NERD tool (http://nerd.eurecom.fr) to access these systems.

The correctness of results were manually evaluated by the first author and rechecked by another researcher with PhD from the text mining domain according to the following guidelines. **Entity recognition**: Named entity or common entity was considered as correctly extracted only if there was a full textual match – extracting "Maradona" instead of "Diego Maradona" would be considered an error. **Classification**: the distance of the hypernym was not considered as long as it was correct (*manager* is as good a hypernym for Maradona as *person*). **Entity linking** – the target of the link must describe the entity given context

[4] http://nlp2rdf.lod2.eu/schema/string
[5] http://nlp2rdf.lod2.eu/schema/sso

(the documentary "dbpedia:Maradona by Kusturica" would be considered as incorrect replacement for "dbpedia:Manager" in Listing 1).

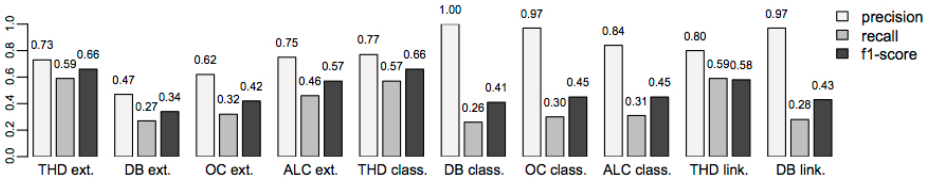


Fig. 2. Evaluation of extraction, classification and linking of Named Entities.

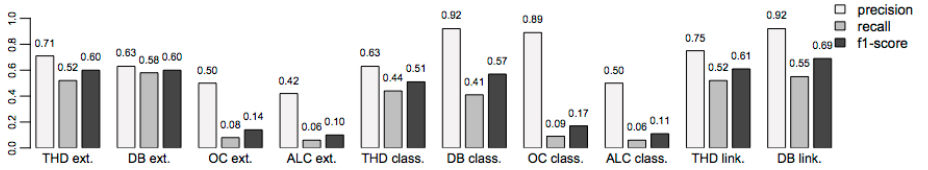


Fig. 3. Evaluation of extraction, classification and linking of common entities.

The experimental results presented in Fig. 2 and Fig. 3 show that our tool produced almost consistently better results in all tasks than the other three tools. It was overtaken only in the task of common entity linking by DBpedia Spotlight with *f-score* 0.69 (as compared to 0.61 for THD). Open Calais did not provide any entity links, while Alchemy API very few (only 6).

Our THD tool is available at http://ner.vse.cz/thd.

Acknowledgements. This research was supported by the European Union's 7th Framework Programme via the LinkedTV project (FP7-287911), by the grant of the Czech Technical University in Prague (SGS12/093/0HK3/1T/18) and by the grant GAČR P202/10/0761.

References

1. H. Cunningham, D. Maynard, and V. Tablan. JAPE - a Java Annotation Patterns Engine (Second edition), Dep. of Com. Sc., Uni. of Sheffield, 2000. Tech. rep., (2000)
2. S. Hellmann, J. Lehmann, and S. Auer. Towards an ontology for representing strings. In *Proc. of the EKAW 2012*, (LNAI). Springer, (2012)
3. T. Kliegr, *et al.* Combining image captions and visual analysis for image concept classification. In *Proc. of the 9th Int. Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, MDM '08, ACM, USA, 8–17 (2008)
4. B. Litz, H. Langer, and R. Malaka. Sequential supervised learning for hypernym discovery from Wikipedia. In *Knowledge Disc., Knowlege Eng. and Knowledge Manag.*, vol. 128 of *Comm. in Comp. and Inf. Sc.*, Springer-Verlag, 68–80 (2011)
5. R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Inf. Proc. Systems*, 17, MIT Press, 1297–1304 (2005)

Identifikácia previazania slov prostredníctvom ich rozloženia v dokumente

Tomáš Kučečka

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave, Ilkovičova 3, 842 16, Bratislava
kucecka@fiit.stuba.sk

Abstrakt. Pri určovaní podobnosti textových dokumentov je často potrebné identifikovať spoločne sa vyskytujúce slová nad daným korpusom dokumentov. Napríklad pri hľadaní podobnosti na základe témy alebo kľúčových slov. Prostredníctvom vzájomne sa vyskytujúcich slov vieme totiž lepšie reprezentovať tému dokumentu. V tomto príspevku navrhneme nový prístup k hľadaniu vzájomne sa vyskytujúcich slov. Tento prístup je založený na multinomiálnom rozdelení frekvencie výskytu slov nad ich pozíciami v texte. Porovnávaním dvoch rozdelení chceme určiť, či dané slova sú, respektíve nie sú previazané. Možnosti využitia nášho prístupu vidíme aj v prípade zisťovania, či sa jedná o slovo charakteristické pre celý dokument alebo len pre jeho konkrétnu časť – lokálne a globálne slová. Kým globálne slová predstavujú tému dokumentu, lokálne predstavujú jeho jednotlivé podtémy.

1 Úvod

Identifikovanie témy textových dokumentov napísaných v prirodzenom jazyku patrí k často rozoberaným problémom v oblasti vyhľadávania informácií. Používateľ sa napríklad môže pri vyhľadávaní dokumentov obmedziť len na určitú tému a vyhľadávať len v rámci tejto témy. Ďalším možným využitím je automatizované priraďovanie článkov recenzentom na základe ich problematiky.

Pri zhlukovaní textových dokumentov ide o proces zoskupovania dokumentov do skupín na základe ich vzájomnej podobnosti, pričom počet skupín ako aj ich opis nie je vopred známy (rozdiel oproti klasifikácii). Často sa tu stretávame s problémami ako veľkosť dátovej množiny alebo priradenie opisu vytvoreným zhlukom.

Princípom segmentácie textových dokumentov je zase rozdelenie dokumentu na segmenty (úseky), pričom každý úsek by mal predstavovať uzatvorený celok z hľadiska ním rozoberanej témy. Pri segmentácii dokumentov sa môžeme riadiť len na základe obsahu daného dokumentu (angl. *single document segmentation*) alebo môžeme pritom využívať aj ostatné dokumenty v danom korpuse (angl. *multi document segmentation*).

V súčasnosti existuje množstvo prístupov k zhlukovaniu alebo segmentácii textových dokumentov, ktoré využívajú vo svojom procese aj informácie o spolu sa vyskytujúcich slovách v korpuse dokumentov. My v tomto článku predstavíme nový prístup

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 45–48.

k odhaľovaniu prepojení medzi slovami. Tento prístup je založený na porovnávaní pravdepodobnostných rozdelení výskytov slov nad ich pozíciami v dokumente.

Tento článok je členený nasledovne. V sekcii 2 uvádzame existujúce prístupy v oblasti identifikácie spolu sa vyskytujúcich slov. Sekcia 3 opisuje nami navrhnuté riešenie k danej problematike. Zhrnutím tohto článku je sekcia 4.

2 Súvisiaca práca

K určovaniu podobnosti textových dokumentov na základe ich témy môžeme pristupovať rôzne. Napríklad zhlukovaním frekventovaných množín termov alebo použitím pravdepodobnostných prístupov. Všeobecne platí, že podobnosť medzi dvoma dokumentmi je o to vyššia, čím viac kľúčových slov majú tieto dokumenty spoločné. V prípade, ak dva dokumenty navyše zdieľajú dvoj a viacslovné frázy, ich celkovú podobnosť to len zvyšuje.

Identifikovanie spolu sa vyskytujúcich slov je kľúčové pri hľadaní sémantickej podobnosti medzi slovami s rôznou syntaxou. Prostredníctvom sledovania spolu sa vyskytujúcich slov dokážeme odhaliť skryté vzťahy medzi slovami (ich previazanie). K existujúcim prístupom k odhaľovaniu sémantickej previazanosti patrí metóda LSA [1] (latentná sémantická analýza). LSA využíva SVD [2] (singulárny rozklad matíc) prístup pre rozklad matice, ktorej riadky sú slová a stĺpce dokumenty, na súčin troch matíc. Získanú maticu singulárnych čísel následne upravíme a získané matice opäť vynásobíme. Výsledkom celého procesu je nový priestor vyjadrený cez lineárne nezávislé vektory, ktorý obsahuje nové sémantické previazania medzi slovami.

K existujúcim prístupom k identifikovaniu témy dokumentu patria pravdepodobnostné prístupy pLSA [3] a LDA [4], ktoré na základe rozloženia slov v dokumente priradia každému dokumentu množinu tém, každú s danou pravdepodobnosťou. Oba tieto prístupy vychádzajú zo zmesového modelu a porovnávajú rozloženie slov v dokumentoch na základe ktorého vyhodnocujú ich vzájomnú podobnosť z hľadiska témy. V práci [5] autori určujú podobnosť textových dokumentov na základe porovnávania multinomiálnych rozdelení slov nad dokumentmi. Ich cieľom je zredukovať počet dokumentov na porovnávanie prostredníctvom výberu podmnožiny podozrivých dokumentov. Tento prístup bol aj inšpiráciou pre nami navrhnuté riešenie.

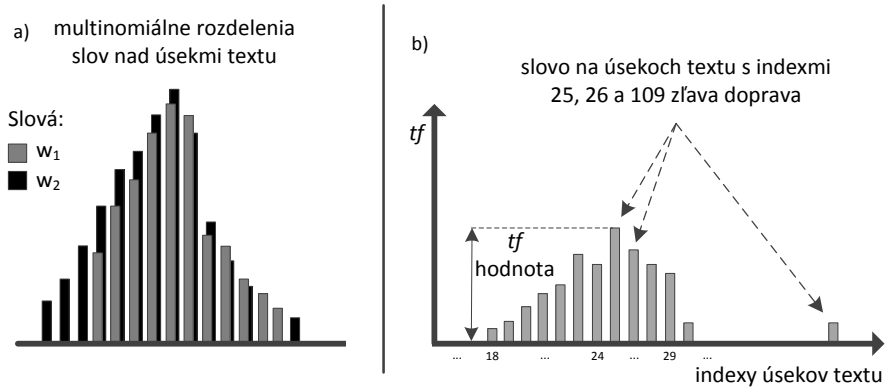
3 Navrhnuté riešenie

Pre identifikáciu slov ktoré spolu súvisia sme sa rozhodli sledovať ich pozície v rámci textového dokumentu a vzájomne porovnávať frekvencie ich výskytu na týchto pozíciách. Celkový postup nášho riešenia je nasledujúci:

1. Predspracovanie dokumentu. Pri predspracovaní sa použijú klasické metódy ako odstraňovanie stop slov, lematizácia, stemovanie a nahradzovanie synoným.
2. Rozdelenie textu dokumentu na úseky dĺžky *k*, kde *k* vyjadruje maximálny počet slov, ktorý môže daný úsek obsahovať. Po prekročení tohto počtu bude úsek na najbližšom konci vety ukončený a ďalší úsek bude začínať na začiatku nasledujú-

cej vety. Každý dokument z korpusu tak po skončení tohto procesu bude rozdelený na množinu úsekov dĺžky *k*, pričom jednotlivé úseky sa nebudú prekrývať.

3. Extrakcia kľúčových slov z textu na základe *tf-idf* váhovania. Snahou je pre daný dokument vybrať čo najväčšiu množinu kľúčových slov, čo môže predstavovať aj stovky slov v závislosti od dĺžky dokumentu. Ďalej v tomto dokumente používame pojem slovo vo význame kľúčové slovo.
4. Určia sa frekvencie výskytu jednotlivých slov v daných úsekoch. Následne získame multinomiálne rozdelenie slov nad úsekmi textov v dokumente. Pre lepšie pochopenie pozrite obrázok 1.
5. Porovnávaním rozdelení dvoch slov Jensenovou–Shannonovou (JS) [6] mierou podobnosti zistíme, či ide o slová vyskytujúce sa v rámci dokumentu spoločne na rovnakých úsekoch textu, alebo nie.



Obrázok 1. Multinomiálne rozdelenia frekvencie výskytu slov nad úsekmi textu. a) zobrazuje príklad dvoch rôznych slov, ktorých rozdelenia sú si podobné. b) je príkladom multinomiálneho rozdelenia slova *w*. Slúži pre lepšiu ilustráciu a pochopenie navrhnutého prístupu.

Predpokladáme, že vyššie opísaný prístup je možné využiť na odhaľovanie vzájomne sa vyskytujúcich slov v danom korpuse dokumentov. Po porovnaní distribúcie dvoch slov Jensenovou–Shannonovou mierou podobnosti by sme mali vedieť povedať, či sa jedná o slová, ktoré sa spolu vyskytujú často.

Z hľadiska identifikácie témy a podtémy v dokumente plánujeme využiť vyššie spomínaný prístup. Pred samotným opisom tohto prístupu ale najskôr vysvetlíme pojmy ako lokálna a globálna téma. Globálna téme predstavuje celkovú tému dokumentu, to znamená tému, ktorá je charakteristická pre celý dokument. Takáto téma sa môže v dokumente vyskytovať len raz. Lokálne témy predstavujú jednotlivé podtémy dokumentu. Jeden dokument nemusí obsahovať ani jednu lokálnu tému, ale globálnu tému musí obsahovať práve jednu. Globálnu tému charakterizujú globálne kľúčové slová, lokálnu tému lokálne kľúčové slová.

Predpokladáme, že na základe výpočtu rozptylu a strednej hodnoty rozdelení jednotlivých slov by sme mohli získať informácie o ich vzťahu k celkovému obsahu dokumentu. Napríklad, slová s vysokým rozptylom v rámci úsekov textu sa budú skôr

vzťahovať k celkovej téme dokumentu. Naproti tomu, slová s malým rozptylom ale vyššou strednou hodnotou sú charakteristické len pre určitú časť dokumentu. Ak sa takýchto slov nájde v dokumente viacero (slov s podobným rozdelením), ktorých rozptyl je nízky a naopak, stredná hodnota je vysoká, tak ich môžeme považovať za lokálne. Naopak, slová s vysokým rozptylom budeme považovať za globálne.

Získané vzťahy medzi slovami a ich zaradenie ku lokálnym alebo globálnym chceme následne využiť pri segmentácii a zhlukovaní obsahu textového dokumentu. Momentálne je ale naším aktuálnym cieľom na základe experimentov zistiť, či vyššie uvedeným prístupom dokážeme identifikovať opísané vzťahy.

4 Záver a budúca práca

Celkovo sme predstavili nový prístup, ktorý by podľa nášho názoru bolo možné využiť pre skvalitnenie segmentovania a zhlukovania textových dokumentov. Tento prístup je zameraný na identifikovanie vzájomne sa vyskytujúcich slov a na zaradenie daného slova do lokálnej alebo globálnej skupiny.

Navrhnutý prístup je založený na porovnávaní multinomiálnych rozdelení frekvencie výskytu slov nad úsekmi textu v rámci dokumentu. Pre porovnanie dvoch rozdelení plánujeme použiť Jensenovú–Shannonovú mieru podobnosti.

Navrhnuté riešenie plánujeme v najbližšej dobe experimentálne overiť a stanoviť najvhodnejšie parametre, akými sú hlavne hraničná stredná hodnota a rozptyl rozdelenia v prípade posudzovania lokálnosti a globálnosti slova a dĺžka úseku textu.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0971/11 a APVV-0208-10.

Literatúra

1. Deerwester, S. et al. (1990). Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science, 1990.
2. Dian I. Martin, John C. Martin, Michael W. Berry, and Murray Browne. 2007. Out-of-core SVD performance for document indexing. Appl. Numer. Math. 57, 11-12 (November 2007), 1230-1239.
3. Hofmann, T. (1999). Probabilistic latent semantic indexing. In: SIGIR '99 Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, vol. 34, no. 1-3, pp. 50-57, 1999.
4. Blei, M.D. – Ng, Y.A. – Jordan, I.M. (2003). Latent Dirichlet Allocation. In: The Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003.
5. Cedeño, B.A. – Rosso, P. – Benedí, M.J. (2009). Reducing the Plagiarism Detection Search Space on the Basis of the Kullback-Leibler Distance. In: Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '09), ACM, pp. 523-534, 2009.
6. Dagan, I., Lee, L., and Pereira, F. C. N. (1999). Similarity-based models of word cooccurrence probabilities. Mach. Learn., 34(1-3):43–69.

Extrakcia metadát zo zdrojových kódov

Tomáš Kramár, Mária Bieliková

Ústav informatiky a softvérového inžinierstva,
Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave
{kramar, bielik}@fiit.stuba.sk

Abstrakt. V mnohých doménach sa modelovanie znalostí zakladá na ľahkej sémantike vo forme rôznych druhov metadát, extrahovaných z textových dokumentov. Extrakcia metadát je pomerne dobre preskúmanou oblasťou a v súčasnosti existuje množstvo metód na extrakciu z textov v prirodzenom jazyku. Pri modelovaní znalostí programátorov sa ako vhodný zdroj metadát javia zdrojové kódy programov, na ktorých pracujú. V porovnaní s extrakciou metadát z textov v prirodzenom jazyku má však extrakcia metadát zo zdrojových kódov iné špecifiká a obmedzenia. V tomto príspevku analyzujeme špecifiká extrakcie metadát zo zdrojových kódov, prezentujeme návrh metódy na extrakciu metadát a uvádzame experimentálne výsledky.

1 Úvod

V posledných rokoch sledujeme v mnohých doménach odklon od tradičných, ťažkých reprezentácií znalostí vo forme ontológií smerom k ľahším a flexibilenejším modelom, založeným na metadátach. Ťažké reprezentácie sľubujú pokročilé vlastnosti (napríklad odvodzovanie znalostí), daňou za ich použitie je však ťažkopádnosť práce s nimi, zložitosť napĺňania a definície, a aj horší výkon pri väčšom množstve dát. Ľahké formy reprezentácie znalostí ponúkajú predovšetkým flexibilitu, jednoduché napĺňanie dát a škálovateľnosť spracovania.

Pod metadátami je možné chápať množinu dát, ktoré opisujú iné dáta a predstavujú redukovaný pohľad na pôvodné dáta, pričom zachovávajú pôvodný význam. Rozlišujeme niekoľko druhov metadát, napríklad: *(i.)* kľúčové slová, ktoré sú podmnožinou najdôležitejších slov z pôvodného textu; *(ii.)* pojmy, ktoré sú na rozdiel od kľúčových slov viac-slovné; *(iii.)* pomenované entity, ktoré reprezentujú entity reálneho sveta, napríklad mená osôb, miest a p. a *(iv.)* značky, ktoré predstavujú slová, ktoré sa nemusia nachádzať v pôvodnom texte; tieto značky môžu byť dodané ručne, používateľmi, alebo aj automatizovanými nástrojmi.

Ako zdroj metadát sa najčastejšie používajú doménové objekty. Napríklad [3] modeluje záujmy používateľa metadátami, extrahovanými z textov webových stránok, ktoré daný používateľ prezerá. V [2] sa zasa modelujú znalosti študentov na základe metadát, extrahovaných z materiálov, ktoré študenti v rámci výučbového systému čítajú.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 49–52.

V príspevku sa venujeme extrakcii metadát zo zdrojových kódov programov. Cieľom je modelovanie znalostí programátora pomocou ľahkej sémantiky – metadát, extrahovaných z dokumentov, s ktorými sa programátori stýkajú najčastejšie a tými sú zdrojové kódy a následné uľahčenie práce programátora automatizovanými metódami [1]. Kým na extrakciu metadát z textov v prirodzenom jazyku existuje množstvo metód, extrakcia metadát zo zdrojových kódov je pomerne nepreskúmaná oblasť. Dôležitou otázkou, na ktorú sa snažíme v tejto práci odpovedať je to, aké sú špecifiká zdrojových kódov oproti textom v prirodzenom jazyku a či a ako je možné na extrakciu metadát použiť štandardné postupy.

2 Zdrojové kódy a metadáta

Extrakcia metadát zo zdrojových kódov má oproti extrakcii textu z prirodzeného jazyka niekoľko špecifík.

Zdrojový kód je tvorený dvomi prvkami s výrazne odlišnou štruktúrou: samotného zdrojového kódu a komentárov, pričom aj komentáre môžu byť odlišného typu: môže sa jednať o zakomentovaný kód, alebo môže ísť o komentár k časti zdrojového kódu, napísaný v prirodzenom jazyku. Celkovo sa teda v zdrojovom kóde môžu nachádzať tri typy prvkov, z ktorých každý je potrebné spracovávať iným štýlom. Zakomentovaný kód je možné ignorovať, jedná sa zrejme o mŕtvy kód, ktorý nemá žiadnu informačnú hodnotu a v programe zostal z dôvodov prípadného návratu k nemu. Komentáre ku kódu sú písané v prirodzenom jazyku a je z nich možné extrahovať metadáta štandardnými metódami.

Štruktúra zdrojového kódu je iná ako štruktúra prirodzeného jazyka. Štatistické metódy na extrakciu metadát z prirodzeného jazyka sú založené práve na tejto štruktúre, napríklad metóda TF.IDF [4] vychádza z predpokladu, že kľúčové informácie sa v danom texte opakujú často a zároveň je ich koncentrácia vyššia ako v ostatných dokumentoch. Dôležitou otázkou, na ktorú sa pokúšame v práci odpovedať je, či tento predpoklad platí aj pre zdrojové texty programov.

V neposlednom rade obsahuje zdrojový kód množstvo šumu vo forme prirodzených konštrukcií programovacieho jazyka, napr. operátory priradenia, porovnania a rôzne iné syntaktické konštrukcie.

3 Aplikácia metódy TF.IDF na zdrojové kódy

Pri extrakcii metadát zo zdrojových kódov programov používame štandardnú metódu TF.IDF. Metóda TF.IDF priraďuje skóre každému elementu z dokumentu na základe jeho výskytu v danom dokumente a v ostatných dokumentoch: $tfidf(t, d, D) = tf(t, d) \times idf(t, D)$, kde tf je frekvencia výskytu kľúčového slova t v dokumente d a idf je inverzná frekvencia výskytu kľúčového slova t v korpuse dokumentov D.

Vzhľadom na definíciu TF.IDF pri jej aplikácii postupujeme tak, že text dokumentu rozdelíme na jednotlivé kandidátne kľúčové slová, vypočítame TF.IDF pre každé kandidátne kľúčové slovo samostatne a odfiltrujeme kľúčové slova s nižšou hodnotou TF.IDF ako je vopred stanovená hranica.

Pri aplikácii TF.IDF je však potrebné vykonať dve rozhodnutia:

- *Ako rozdeliť zdrojový kód na kandidátne kľúčové slova?* Vzhľadom na povahu zdrojového kódu, kde sa nachádzajú rôzne nealfanumerické syntaktické konštrukcie, ktoré nechceme považovať za kľúčové slová, získame množinu kľúčových slov tokenizáciou na alfanumerické sekvencie. To nám umožní odfiltrovať syntaktické konštrukcie a zároveň ponechať slová, ktoré boli v zdrojovom kóde súčasťou programových reťazcov.
- *Ako zvoliť korpus dokumentov?* Výber korpusu dokumentov je veľmi dôležitou časťou metódy TF.IDF, pretože pomáha odfiltrovať opakujúce sa slová. Napríklad spojka "ale" sa v prirodzenom jazyku vyskytuje v dokumentoch často, preto bude mať vysokú hodnotu tf. Tým, že sa vyskytuje často aj v iných dokumentoch bude mať nízku hodnotu *idf* a vo výsledku teda aj nízku hodnotu TF.IDF. Podobný efekt chceme docieliť aj pre často sa opakujúce konštrukcie v zdrojových kódoch, napríklad *class*, *while*, alebo *for*. Je preto zrejmé, že podobne ako pri spracovaní prirodzeného textu je vhodné mať separátny korpus pre každý prirodzený jazyk, je vhodné mať separátny korpus pre každý samostatný programovací jazyk.

Navrhli sme takýto postup pre extrakciu kľúčových slov:

- Najskôr je potrebné vytvoriť korpus dokumentov. Ten vytvárame z množiny zdrojových kódov, pre ktoré je známy ich programovací jazyk. Pre každý dokument z množiny vykonáme rozdelenie na zdrojový kód a komentáre. Komentáre pridáme do samostatného korpusu, na zdrojové kódy aplikujeme tokenizáciu a takto predspracovaný dokument vložíme do korpusu. Tento dokument zároveň použijeme na natrénovanie *Naive-Bayes* klasifikátora.
- Pri exktrakcii kľúčových slov zo zdrojového kroku vykonáme v prvom kroku rozdelenie na zdrojový kód a komentáre. Na zdrojový kód aplikujeme tokenizáciu. Následne aplikáciou *Naive-Bayes* klasifikátora detegujeme jazyk dokumentu a vyberieme správny korpus pre zdrojový kód. S použitím korpusu pre komentáre a korpusu pre daný programovací jazyk získame TF.IDF skóre pre každé kandidátne kľúčové slovo a odfiltrujeme všetky kľúčové slová, ktorých skóre je nižšie ako je priemerné skóre všetkých kandidátov.

Overenie tohto postupu aplikácie na zdrojové kódy bolo vykonané doménovými expertmi, ktorí dostali na posúdenie zdrojový kód a z neho extrahované kľúčové slová. Doménoví experti boli zároveň autormi projektu, z ktorého sme vybrali zdrojové kódy pre experiment. Výsledky experimentu sú zosumarizované v tabuľke 3.

Extrakcia kľúčových slov zo zdrojových kódov pomocou metódy TF.IDF dosahuje úspešnosť medzi 70% až 80%, čo podľa nás postačuje na potreby modelovania znalostí programátora. Toto vyhodnotenie tvrdo penalizuje každé nesprávne kľúčové slovo, vždy sa však jedná iba o subjektívny pocit doménového experta. Navyše, pri reálnej aplikácii a modelovaní používateľa, začnú s pribúdajúcim množstvo metadát v modeli počtom prevažovať tie dobré a tie zlé budú potlačené do úzadia.

Tabuľka 1. Hodnotenie metódy extrakcie kľúčových slov doménovými expertmi.

doménový expert	# dokumentov	kľúčových slov	akceptovaných	% akceptovaných
#1	10	173	125	72%
#2	10	158	113	71%
#3	10	165	137	83%

4 Zhodnotenie

Extrakcia metadát zo zdrojových kódov sa od extrakcie z prirodzeného jazyka líši najmä rozdielnou štruktúrou a stavbou dokumentov. V tejto práci sme ukázali, že aj napriek tejto rozdielnosti, je možné na extrakciu metadát použiť štandardné metódy, ktoré sa používajú na extrakciu metadát z textov v prirodzenom jazyku.

Naša práca však neponúka definitívnu odpoveď a skôr predstavuje len prvý krok v ďalšom výskume. Ďalšou dôležitou otázkou, na ktorú treba odpovedať, je otázka výberu korpusu. Teda či ako korpus dát pre výpočet IDF zvoliť všetky zdrojové kódy v danom programovacom jazyku, alebo len zdrojové kódy z daného projektu, alebo korpus vyberať dynamicky. Nemenej dôležitá je aj otázka spracovania komentárov a rovnako výber vhodného korpusu pre výpočet IDF. Jedným z možných ďalších výskumných smerov je aj extrakcia metadát inými metódami, ako je TF.IDF.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0971/11, APVV-0208-10 a vznikla vďaka podpore v rámci OP Výskum a vývoj pre projekt ITMS: 26240220039, spolufinancovaný ERDF.

Literatúra

1. Bieliková, M., et al.: Webification of software development: General outline and the case of enterprise application development. In: Procedia Technology, 3rd World Conf. on Inf. Tech. vol. 4. to appear
2. Šimko, M., Barla, M., Bieliková, M.: Alef: A framework for adaptive web-based learning 2.0. In: Reynolds, N., Turcsányi-Szabó, M. (eds.) Key Competencies in the Knowledge Society, IFIP Advances in Information and Communication Technology, vol. 324, pp. 367–378. Springer Boston (2010)
3. Kramár, T., et al.: Disambiguating search by leveraging the social network context based on the stream of user's activity. In: Proc. of the 18th Int. Conf. on User Modeling, Adaptation, and Personalization. pp. 387–392. Springer (2010)
4. Roelleke, T., Wang, J.: Tf-idf uncovered: a study of theories and probabilities. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 435–442. SIGIR '08, ACM, New York, NY, USA (2008)

Označovanie kódu značkami s informáciou o kontexte

Dušan Zeleník, Mária Bieliková

Ústav Informatiky a Softvérového Inžinierstva
Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave,
Ilkovičova 3, 842 16 Bratislava
{zelenik, bielik}@fiit.stuba.sk

Abstrakt. Kód, ktorý vývojár vytvára je ďalej predmetom ďalších akcií spojených so životným cyklom softvéru. Proces jeho vytvárania často krát pomáha určiť príčiny zlyhania, či naopak uľahčuje prácu ďalším. Označovaním kódu značkami, ktoré hovoria o podmienkach, za ktorých kód vznikol môžu čiastočne pomôcť pri automatickom odhaľovaní chýb v kóde. Nehovoríme o syntaktickej analýze kódu, avšak o obohacovaní metadát o kóde. Tieto značky môžu byť atribútmi programátora (úroveň znalosti, nálada), ktorý ho vytváral, alebo stav prostredia (blízkosť termínov projektu). Množstvo z týchto značiek pritom vieme generovať automaticky použitím záznamov z nástrojov pre manažment zdrojových kódov, alebo z pozorovania programátora priamo pri práci. V tejto práci prezentujeme možnosti generovania značiek a uvádzame príklady využitia týchto značiek v praxi.

Keywords: značky, metadáta, kód, kontext

1 Úvod

Tvorba softvéru je sprevádzaná okolnosťami, ktoré vplývajú na výstup. Na programátora môžu vplývať okolnosti z prostredia v ktorom sa nachádza, ale aj programátor sám vnáša do kódu, ktorý píše svoj vlastný stav. Vplyvy, ktoré vieme pozorovať má význam zaznamenávať priamo ku vytvorenému kódu. To môže pomôcť napríklad pri odhaľovaní chýb v tomto kóde. V tomto prípade nejde o hľadanie chýb formou testovania, či analýzou kódu [4]. Ide skôr označovanie potenciálne chybných častí, ktoré boli vytvorené v strese, v časovej tiesni, či bez skúsenosti programátora.

Označovanie kódu značkami, ktoré hovoria o tom ako bol kód vytvorený je jeden so spôsobov ako vytvárať ďalej využiteľné metadáta [1]. Značky môžeme generovať priamo ku štruktúre kódu, až na najjemnejšie jednotky, ktoré sú riadky. Tak s kódom pracujú aj nástroje na manažment kódu a verziovanie. V našej práci sa venujeme značkám na úrovni súborov. Takže nezachádzame do hlbšej analýzy štruktúry kódu. Značky priraďujeme k súborom, v ktorých je editovaný kód. Jeden taký súbor pritom môže získať viac značiek rôzneho typu. Taktiež jeden súbor môže byť označený viacerými značkami s pôvodom od rôznych programátorov.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 53–56.

Značka, s ktorou pracujeme má hlavne svoju platnosť a súbor. Ostatné atribúty sú dané typom značky. Tu nepredpokladáme žiadne obmedzenie na ďalšie atribúty. Pravidlom pri značkách, ktoré využívame je však hodnota, reprezentovaná reálnym číslom z intervalu <0,1> a typ. Príkladom typu a hodnoty značky, ktorá bola vygenerovaná pre opis stavu prostredia môže byť *doma:1*. Táto značka hovorí, že kód bol editovaný doma s vysokou pravdepodobnosťou. Takáto reprezentácia nás neobmedzuje v pridávaní nových typov, i keď pracujeme s vopred definovanou množinou typov, ktorými automaticky označujeme kód.

2 Vplyvy na proces tvorby kódu

Tvorba kódu je často ovplyvnená externými vplyvmi na programátora, ale aj programátorom samotným. Tieto vplyvy sa môžu prejaviť ako nedbanlivo napísaný kód v prípade programátora neskúseného v danej oblasti. Prípadne i u skúseného, ktorý však pracuje pod stresom. Podnet vplyv nazývame kontext, v ktorom je kód vytváraný. V tejto časti sa venujeme kontextu programátora a kontextu prostredia ako hlavným dvom množinám typu kontextu. Programátor má svoje vlastné atribúty, ktoré môžu byť použité v značkovaní kódu. Z prostredia ďalej vieme určovať skôr všeobecné informácie. Určili sme niekoľko typov, ktoré sme schopní zaznamenávať.

- *Skúsenosť*. Množstvo kódu doposiaľ vytvoreného v danom projekte.
- *Produktivita*. Množstvo kódu v porovnaní s ostatnými.
- *Stereotyp*. Tvorba kódu počas obvyklých časov, alebo neštandardne.
- *Návraty*. Opakované editovanie kódu.
- *Čas*. Kedy bol kód vytvorený. Počas dňa, týždňa, sviatkov
- *Počasie*. Aké počasie práve bolo v mieste tvorby kódu.
- *Miesto*. Kde sa programátor nachádzal. V práci, doma, niekde inde.

Tieto typy sme schopní zaznamenávať využívaním záznamov z nástroja pre správu kódu (v našom prípade Git) a externých služieb (ip2location.com a wunderground.com pre určenie počasia). Jednotlivé značky potom generujeme využívaním vlastných knižníc, pracujúcich s časom a IP adresami.

Uvedomujeme si rôznu mieru vplyvu kontextu. Chceme však vytvárať prostredie, kde môžeme generovať rôzne typy značiek a hľadať súvislosti medzi tvorbou kódu a kontextom. Preto začleňujeme i zdanlivo nevýznamný kontextový typ ako počasie.

3 Značkovanie kódu informáciou o kontexte

Princípom automatického generovania je využívanie záznamov z nástroja na správu kódu. To znamená, že pre konkrétny projekt poznáme identifikátor programátora a jeho aktivitu v podobe akcie *commit*. Takáto akcia v sebe nenesie informáciu o samotnej práci. Poznáme iba kedy programátor považoval nejakú časť svojej práce za ukončenú a na čom danú prácu urobil. Poznáme zmeny, ktoré vytvoril, pričom ich hlbšie neanalyzujeme.

3.1 Metóda automatického generovanie značiek

Naša metóda spočíva v priraďovaní značky k súboru. Automaticky aktualizujeme zdroj informácií ako klon úložiska manažmentu zdrojových kódov projektu. Aktualizácia pritom prebieha s každou novou zmenou, čiže takmer v reálnom čase.

Po aktualizácií klonu úložiska sa spúšťajú analýzy každej novej akcie *commit*. Spätne sa tiež analyzuje minulosť programátora a pripravujú sa údaje pre nové značky. Už priradené značky sa však neaktualizujú, tie sú platné k stavu, ktorý bol aktuálny v čase ich vzniku.

Ďalšími údajmi pre generovanie značiek sú výstupy z externých služieb, ktoré môžu napríklad obohatiť kód o počasie, ktoré bolo práve aktuálne. Tieto služby nie sú viazané na správu zdrojových kódov. Ich vyvolanie však prebieha priamo po aktualizácií úložiska, keďže značky generujeme vzhľadom na akcie *commit*.

Posledným krokom našej metódy je generovanie samotnej značky. Pre jeden súbor sa s každou naviazanou akciou *commit* spája viacero značiek, v závislosti od vopred definovanej množiny kontextových typov, ktoré chceme zaznamenávať.

3.2 Využitie značiek

Využitie plánujeme ako nástroj na predpovedanie chybného kódu. To znamená, že plánujeme generovať ďalšie značky, ktoré hovoria o kóde, či môže byť chybný [3]. Pritom nerobíme analýzu kódu, pre hľadanie pachov, chýb, či neefektívnych konštrukcií [2]. Predpovedáme chyby, ktoré vznikajú kvôli nepriaznivým podmienkam.

Pre takúto predpoveď potrebujeme získať informácie o vplyvoch. Trénovanie predpovedí sa vieme uskutočniť využívaním výstupov o hlásených chybách, alebo aj z revízií, či opráv. Následne trénujeme väzby medzi kontextom a chybovosťou.

V našej súčasnej práci pracujeme iba so značkami, ktoré sme vygenerovali. Hľadáme vzory o spoluvýskytoch značiek, ktoré potom overujeme s reálnymi programátormi a hypotetickými otázkami. Spoluvýskyt značiek odhaľujeme štatisticky v projektoch. Hľadáme n-tice značiek, ktoré sa často spolu vyskytujú pri akciách a zisťujeme, či takéto situácie nastávajú u reálnych programátorov. Jedná sa napríklad o *návraty* ku kódu počas *večerných hodín*, alebo o *produktivitu* v *pracovných ďnoch.*

3.3 Vyhodnotenie správnosti značkovania

Hlavným prínosom tejto práce je generovanie značiek a teda obohacovanie kódu novými metadátami. Vyhodnotenie takéhoto značkovania a jeho správnosti prakticky nemá význam, keďže iba získavame nové informácie. V našej práci sa však snažíme odhaľovať korelácie v podobe častých spoluvýskytov značiek. To znamená, že hľadáme niektoré vzory, ktoré môžu existovať.

Zaujímavejším spôsobom vyhodnotenia správnosti značkovanie je preto konfrontácia objavených vzorov s reálnym programátorom. V našom vyhodnotení sme oslovili 5 programátorov, ktorým boli predložené často opakujúce sa vzory. Programátori následne z ich subjektívneho pohľadu posúdili tieto vzory a označili ich ako správne, alebo nesprávne. Taktiež sme programátorom položili otázku, či nie je vzor bežný pre

iných programátorov. Tak eliminujeme výnimočnosť jednotlivca, ktorá samozrejme môže nastávať. Výsledky dotazníka sú prezentované v nasledujúcej tabuľke.

Tabuľka 1. Ohodnotenie top 10 objavených vzorov.

	Správnosť (subjektíne)	*Správnosť (objektívne)*
Percentuálna úspešnosť	66%	80%

4 Záver a budúca práca

V tejto práci sme prezentovali najmä spôsob automatického generovania značiek ku kódu pomocou externých služieb a nástroja na manažment zdrojových kódov. Venovali sme sa kontextu, ktorý bol platný v dobe vytvárania kódu.

Poukázali sme na možnosť využitia takýchto značiek napríklad pre označovanie potenciálne chybného kódu, nad rámec syntaktickej, či sémantickej analýzy samotného kódu. Overovali sme však súčasný stav, kde hlavne generujeme značky a analyzujeme niektoré vzory, ktoré objavujeme. Programátori potvrdili vzory, ktoré sme objavili, čo znamená, že generovanie značiek je správne a má význam objavovať vzory. Taktiež sa otvára priestor pre budúcu prácu v podobe trénovania modelu pre predpoveď chybovosti kódu využívaním týchto zdrojov. Nie je pritom jedinou možnosťou využívať vzory správania masy, ale vieme objavovať vzory správania jednotlivca, či skupiny podobných programátorov.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0971/11, APVV-0208-10 a vznikla vďaka podpore v rámci OP Výskum a vývoj pre projekt ITMS: 26240220039, spolufinancovaný ERDF.

Literatúra

1. Bieliková, M., et al.: Webification of Software Development: General Outline and the Case of Enterprise Application Development. In Proceedings of 3rd World Conf. on Inf. Tech., Vol. 4. To appear.
2. Li, J., Ernst. D Ernst.: CBCD: cloned buggy code detector. In Proceedings of the 2012 International Conference on Software Engineering (ICSE 2012). IEEE Press, Piscataway, NJ, USA, 310-320, (2012).
3. Rástočný, K.., Bieliková, M.: Maintenance of Human and Machine Metadata over the Web Content. In: M. Grossniklaus and M. Wimmer (Eds.): Current Trends in Web Engineering (ICWE 2012), LNCS, Springer-Verlag, Berlin Heidelberg (2012).
4. Schumacher, J., Zazworka, N., Shull, F., Seaman, C., Shaw., M.: Building empirical support for automated code smell detection. In Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement(ESEM '10). ACM, New York, NY, USA, Article 8, (2010).

Model používateľa a spätná väzba

Vplyv schopností hráčov na úspešnosť hier s účelom

Jakub Šimko, Mária Bieliková

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave, Ilkovičova 3, 842 16, Bratislava
{jsimko, bielik}@fiit.stuba.sk

Abstrakt. Hry s účelom (angl. *games with a purpose*) sa v rámci domény čerpania z davu (angl. *crowdsourcing*) stali často používaným spôsobom riešenia strojovo ťažkých úloh, zároveň vhodných na riešenie človekom. Tieto prístupy sa opierajú o redundanciu úloh, aby zabezpečili kvalitu riešení ktoré poskytujú. Otvorenou výskumnou otázkou je hľadanie spôsobov ako túto redundanciu zmenšovať a šetriť tak prostriedky (prácu hráčov). Hry s účelom zároveň vo všeobecnosti vykazujú neschopnosť spracúvať špecifické úlohy, pretože nerozlišujú rôzne schopnosti hráčov a väčšinovým hlasovaním, či konsenzom produkujú neužitočné riešenia. V príspevku skúmame fenomén rozdielnych schopností hráčov hier s účelom. Analyzujeme záznamy existujúcej hry, poukazujeme na rozdiely v užitočnosti jej hráčov a pomocou syntetických experimentov demonštrujeme zlepšenie výkonu hry po zapojení rozlišovania vplyvu hráčov na riešenie úloh na základe ich predchádzajúcej užitočnosti.

Keywords: Hra s účelom, čerpanie z davu, model používateľa, web so sémantikou

1 Úvod

Zber informácii, prieskumy, vytváranie praktických a výskumných aplikácii či experimentálne overovanie nových prístupov mnohokrát zahŕňa úlohy, ktoré nemožno riešiť automatickými prostriedkami. Tieto úlohy musia riešiť ľudia. Časť z týchto úloh je pritom závažná skôr svojou kvantitou než náročnosťou individuálnej úlohy. Typickým príkladom pre oblasť webu so sémantikou, či manažment znalostí je zber metadát k multimediálnym zdrojom, či napĺňanie báz znalostí. Na riešenie tohto typu úloh sa často využívajú prístupy, ktoré sa namiesto malého množstva expertov (ktorých vyžaduje riešenie zložitých úloh) opierajú o veľké skupiny laikov, tzv. dav (angl. *crowd*). V procese nazývanom „čerpanie z davu“ (angl. *crowdsourcing*) riešia jeho členovia jednoduché zadané úlohy, najčastejšie výmenou za malú finančnú odmenu [3]. Kvalitu výsledkov zabezpečuje redundantné riešenie jednej úlohy viacerými riešiteľmi a uplatňovanie zhody (konsenzu) na rovnakých riešeniach.

Atraktívnou podoblasťou čerpania z davu sú tzv. hry s účelom (angl. *games with a purpose*), ktorých hlavnou črtou je, že finančnú odmenu pre riešiteľov nahradzujú zábavou, čím sa stávajú ekonomicky výhodnými. Riešenie zadanej úlohy sa zaobaľuje ako súčasť hry: kto chce hru vyhrať, je nútený produkovať riešenia danej úlohy (často

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 59–62.

bez toho, aby o tom vedel). Typickým príkladom hry s účelom je *ESP game*, kde sa dvaja hráči snažia zhodnúť na slove charakterizujúcom zadaný obrázok [1].

Hry s účelom, napriek širokému využitiu, nevyužívajú „prácu" hráčov vždy efektívne. Vyplýva to z ich základného mechanizmu: využívajú redundantné riešenie úloh na zabezpečenie kvality riešení. Ak by však hra na prvý krát vybrala na riešenie úlohy hráča, o ktorom s veľkou mierou istoty vie povedať, že ju vyrieši správne, redundanciu by bolo možné čiastočne, alebo úplne eliminovať a šetriť prácu hráčov.

S tým súvisí neschopnosť hier s účelom riešiť špecifické úlohy, teda také úlohy, ktoré väčšina hráčov nedokáže riešiť vôbec (napr. určovanie druhov rastlín na fotografiách) [4]. A opäť, ako možné riešenie sa ponúka schopnosti hráčov zisťovať a prideľovať im na riešenie čo najvhodnejšie úlohy.

Prístupy v oblasti čerpania z davu sa s týmito problémami snažia vyrovnať prostredníctvom explicitného zisťovania schopností účastníkov (dotazníky) či ich tréningom [3]. Niečo také však v prostredí hier s účelom možné nie je – hráči niečo také dobrovoľne nepodstúpia (absentuje finančná odmena). Implicitné zisťovanie schopností hráčov priamo počas hry a ich následné využívanie však môže predstavovať riešenie. V našom výskume sme sa preto snažili odpovedať na tieto otázky:

- Ako sa schopnosťami líšia jednotliví hráči? Líšia sa hráči na základe svojej reálnej užitočnosti? Koreluje s touto užitočnosťou participácia hráčov na konsenze pri filtrovaní správnych riešení?
- Pomôže účelu hry, ak budeme na základe zmeraných schopností uvažovať dôveryhodnosť ich riešení? Zvýši sa správnosť? Stačí na prípadné zlepšenie uvažovať mieru participácie na konsenze (ktorá sa dá merať už za behu hry) alebo je nutné hľadať iné spôsoby zisťovania reálnej užitočnosti hráčov?

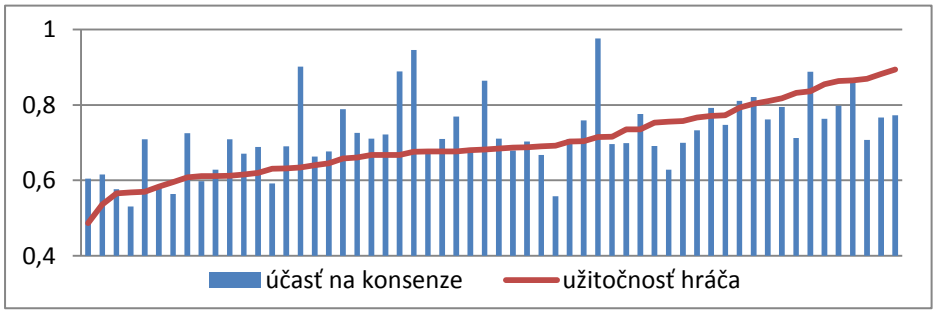
2 Meranie úrovne schopností hráčov

Pre nájdenie odpovedí sme analyzovali záznamy našej existujúcej hry s účelom *PexAce*, ktorá slúži na získavanie anotácii k obrázkom [4]. Základným (medzi)výstupom hry je zoznam „návrhov" hráčov na priradenie textových značiek (angl. *tags*) k obrázkom. Ak abstrahujeme od konkrétnej hry, na trojicu *hráč-obrázok-značka* sa môžeme pozerať ako na trojicu *hráč-úloha-riešenie*, ktorú môžeme nájsť aj v iných hrách s účelom [1,2]. Zoznam týchto trojíc je potom vždy vstupom do poslednej fázy tvorby finálnych riešení úloh v hre: filtrovaním riešení na základe konsenzu viacerých hráčov.

Za pomoci zoznamu trojíc z hry PexAce a za pomoci zlatého štandardu, v našom prípade zoznamu obrázkov so správne priradenými značkami (vytvorenom expertmi) sme pre každého hráča tejto hry vypočítali dve hodnoty (Obr. 1):

- *Užitočnosť*. Predstavuje priame porovnanie návrhov hráča s objektívnou pravdou zlatého štandardu a vystihuje tak úroveň jeho schopností pre riešenie danej úlohy. Definovaná je ako podiel počtu správnych a počtu všetkých návrhov hráča.
- *Účasť na konsenze*, definovaná ako podiel počtu návrhov, v ktorých sa hráč zhodoval aspoň s jedným hráčom oproti všetkým jeho návrhom. Táto hodnota nemusí

odrážať „schopnosti“ hráča presne (viacero hráčov sa môže mýliť rovnako), je však merateľná už za behu hry (nevyžaduje zlatý štandard).



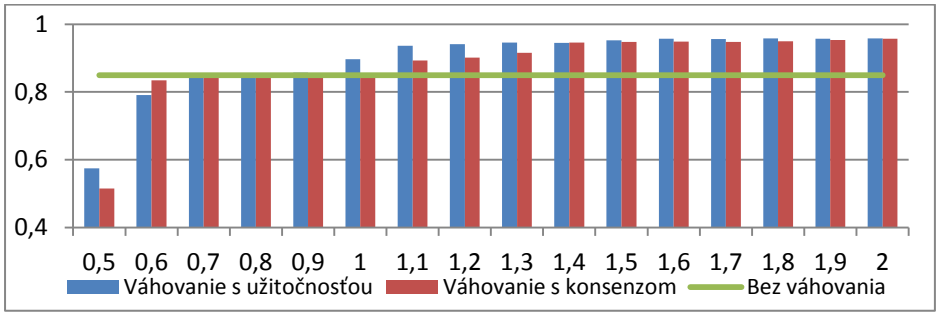
Obr. 1. Porovnanie *užitočnosti* jednotlivých hráčov s ich *účasťou na konsenze* s inými hráčmi.

V prípade, že by existovala dostatočne veľká korelácia medzi *užitočnosťou* a *účasťou na konsenze*, bolo by možné účasť na konsenze merať pre každého hráča z jeho predchádzajúcich akcií a použiť ju na ovplyvňovanie „sily hlasu“ hráča v budúcnosti. Keďže sme predpokladali lineárnu závislosť týchto dvoch hodnôt, mieru korelácie sme vypočítali pomocou Pearsonovho korelačného koeficientu: 0,496. Tento výsledok ukazuje, že pokiaľ existuje príčinná závislosť medzi uvedenými faktormi, bude len čiastočná. Aj to však môže pomôcť pri zvyšovaní efektívnosti hier s účelom.

3 Aplikovanie znalosti o schopnostiach hráčov

Základný proces filtrovania návrhov riešení sme obohatili o váhovanie hráčov na základe „schopností“. Použili sme najskôr hodnoty *užitočnosti* potom *účasti na konsenze*. Oba faktory (v rozsahu od 0 po 1) boli pre každého hráča transformované na *váhy hlasov*. Zároveň bol určený parameter *prah*, ktorý určoval platnosť konsenzu hráčov: pokiaľ bol súčet váh hlasov presadzujúcich jedno riešenie vyšší ako prah, riešenie bolo prehlásené za platné. Optimálnu výšku *prahu* sme odhadli ako dvojnásobnú priemernú *užitočnosť* hráča (v našom prípade 2 x 0,7 = 1,4), tzn. dvaja priemerní hráči mohli presadiť spoločné riešenie podobne ako v originálnom prístupe.

S pomocou tohto kľúča sme nanovo prefiltrovali výsledné značky k obrázkom a porovnaním so zlatým štandardom sme vypočítali presnosť výsledkov. Experimentovali sme pritom s viacerými hodnotami *prahu* (bežiacich od 0,5 do 2,0) Výsledky môžeme vidieť na grafe na obrázku 2, kde sú hodnoty presnosti pre obe metódy váhovania porovnané so základnou presnosťou, dosiahnutou bez zapojenia váhovania. Ako prvá s narastajúcim prahom vzrastá presnosť pri váhovaní na základe *užitočnosti*, neskôr sa na jej úroveň dostáva aj váhovanie na základe *konsenzu*. Obidve hodnoty sa pritom dostali nad hranicu pôvodného výsledku o sľubných 10%, čo pri už aj tak vysokých hodnotách presnosti považujeme za veľmi dobré.



Obr. 2. Presnosť výsledkov hry pri použití váhovania hlasov na základe užitočnosti a konsenzu.

4 Výsledky a budúca práca

Táto práca je prvým krokom k zapojeniu nových techník „modelovania hráčov“ do hier s účelom, ktorých sa zúčastňujú hráči s rôznymi schopnosťami a tým pádom aj užitočnosťou pre daný účel hry. Ukázali sme, že už pri použití najjednoduchšieho spôsobu odhadovania schopností hráčov – implicitne získavanom podiele *účasti na konsenze* – sa zvýšila presnosť značiek priraďovaných v hre k obrázkom. Hoci tento spôsob nedosiahol tak dobrý výsledok ako aplikácia reálnej *užitočnosti*, aspoň s ňou do určitej miery koreluje a možno ho počas behu hry merať bez ďalších vstupov a následne využívať.

Ďalším krokom v tomto výskume bude rozlišovanie jednotlivých úloh v hre. Doteraz sme sa na účel hry pozerali ako na celok a aj schopnosti hráčov sme vzťahovali na všetky úlohy spoločne. Úlohy v hre sú však často rozdelené na poddomény (napr. tematické celky obrázkov) a je pravdepodobné že aj schopnosti jedného hráča sa pre ne budú rôzne. Na mieste je teda hľadanie spôsobov ako tieto skupiny úloh odhaľovať a ako zisťovať schopnosti hráčov ich riešiť.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0675/11 a APVV-0208-10.

Literatúra

1. Ahn, Lv, Dabbish L.: Designing games with a purpose. Communications of the ACM 51. 58-67 (2008).
2. Ho, C., Chang, T., Lee, J., Hsu, J.Y., Chen, K.: KissKissBan: a competitive human computation game for image annotation. SIGKDD Explor. Newsl. 12, 1, 21-24 (2010).
3. Sabou, M. Bontcheva, K., Scharl, A.: Crowdsourcing research opportunities: lessons from natural language processing. In Proc. of the 12th Int. Conf. on Knowledge Management and Knowledge Technologies (i-KNOW '12). ACM, New York, NY, USA, (2012).
4. Šimko, J., Bieliková, M.: Personal image tagging: a game-based approach. In Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS '12). ACM, New York, NY, USA, 88-93 (2012).

Model používateľa priamo vo webovom prehliadači

Márius Šajgalík, Michal Barla, Mária Bieliková

Ústav informatiky a softvérového inžinierstva, Fakulta informatiky
a informačných technológií, Slovenská technická univerzita v Bratislave,
Ilkovičova 3, 842 16, Bratislava, Slovensko
{sajgalik, barla, bielik}@fiit.stuba.sk

Abstrakt. Webový prehliadač ako ho v súčasnosti poznáme a používame častokrát predstavuje náš hlavný pracovný nástroj. V dôsledku toho možno z histórie prehliadania webu a aktivity pri prehliadaní jednotlivých stránok, či používaní rôznych webových aplikácií odhaliť aj zlomok našej osobnosti a záujmov. Túto skutočnosť sme využili pri návrhu a realizácii personalizačnej platformy MePersonality. Kľúčovým konceptom je presun monitorovania a spracovania údajov o používateľovi zo servera bližšie k používateľovi – priamo do jeho prehliadača, kde sa vytvára model používateľa. Na základe tohto modelu vieme v rámci našej platformy vytvárať množstvo personalizačných rozšírení, ktoré umožňujú prispôsobenie konečnej podoby prehliadaných webových stránok.

Kľúčové slová: personalizácia, model používateľa, personalizačná platforma, kolaboratívne rozšírenia, webový prehliadač

1 Úvod

Väčšina súčasných webových aplikácií sa snaží zbierať údaje používateľov o ich správaní, záujmoch, zvykoch, cieľoch, potrebách a rôznych ďalších charakteristických vlastnostiach, ktoré by neskôr vedeli využiť na vytvorenie určitého profilu používateľa – jeho modelu. Účelom je vedieť sa používateľovi čo najlepšie prispôsobiť a umožniť mu lepší prístup k množstvu informácií, aby bola jeho práca efektívnejšia a pohodlnejšia.

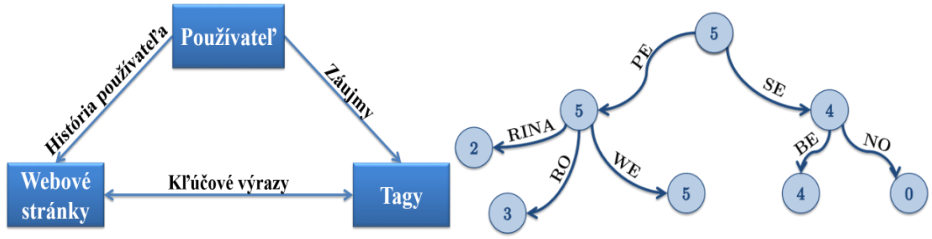
V našom projekte sa zameriavame na prácu vo webovom prehliadači ako základnom pracovnom nástroji používateľa. Cieľom je umožniť personalizáciu webového obsahu a tak zlepšiť, zjednodušiť a zefektívniť prácu používateľa. Dôležitý rozdiel oproti iným populárnym personalizačným riešeniam predstavuje decentralizované a zároveň distribuované riešenie. Riešenie využíva napríklad aj také kľúčové informácie, ku ktorým vzdialené servery nemajú prístup (napr. používateľova detailná história prehliadania webu, samotná aktivita používateľa ako práca s kartami prehliadača, či pohyby myšou). Aby bola takáto realizácia na klientovi reálne použiteľná a škálovateľná, už od začiatku návrhu sme sa sústredili i na výber vhodných dátových štruktúr a algoritmov.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 63–66.

2 Modelovanie používateľa a personalizácia

Modelovanie používateľa a personalizácia predstavuje rozsiahlu oblasť súčasného výskumu (napr. [1] a [2], dobrý prehľad možno nájsť v [4]). Keďže vyhľadávanie na webe je v súčasnosti jeden z najčastejších spôsobov prehliadania webu, venuje sa mu zvýšená pozornosť aj v rámci personalizácie (napr. [3], [5], [6], [8]).

Účelom modelovania používateľa je vytvoriť taký model, v ktorom budú zachytené charakteristické vlastnosti používateľa na základe ktorých mu vieme ďalej prispôsobovať prezeraný webový obsah. Základ modelu používateľa je v našom riešení [7] reprezentovaný v podobe jeho záujmov, ktoré extrahujeme v podobe kľúčových výrazov z navštívených webových stránok. Model záujmov používateľa (obrázok 1 vľavo) spolu s jeho indexovacími dátovými štruktúrami (modifikované verzie koreňového stromu – obrázok 1 vpravo), ktoré tvoria tzv. indexovač, sme navrhli s cieľom dosiahnuť optimálnu časovú a pamäťovú zložitosť pre všetky potrebné operácie, aby bol pripravený na reálne použitie v praxi.



Obr. 1. Model záujmov používateľa (vľavo) a modifikovaný koreňový strom (vpravo).

Hlavným účelom indexovača je ukladať položky, ktoré sa skladajú z výrazu (tag, alebo URL adresa) a jeho relevancie. Vloženie, vymazanie a vyhľadanie prvku podľa jeho indexovaného výrazu má lineárnu časovú zložitosť, čo je optimálne. V prípade nášho indexovača vieme navyše efektívne vyhľadať prvých k najrelevantnejších prvkov s priemernou dĺžkou m, čo je rovnako optimálne v čase . Výsledky teoretickej analýzy sme overili priamo v prehliadači niekoľkými experimentmi na reálnych scenároch (tabuľka 1), v ktorých sme porovnali časovú zložitosť nášho indexovača oproti vstavanému JS objektu (najprv sú uvedené výsledky nášho indexovača, v zátvorke vstavaného JS objektu, údaje sú v mikrosekundách).

Tab. 1. Porovnanie výkonu indexovača v rôznych prehliadačoch.

Prehliadač	Vloženie	Vyhľadanie podľa výrazu	Vyhľadanie 10 najrelevantnejších prvkov	Vyhľadanie 50 najrelevantnejších prvkov
Chrome 17	1,77 (0,57)	0,53 (0,04)	200 (8650)	560 (8650)
IE 9	4,42 (2,37)	3,54 (1,4)	750 (34610)	1540 (34610)
Firefox 11	9,33 (1,7)	8,06 (0,56)	1090 (13840)	3410 (14480)

V týchto testoch sme použili kolekciu 4204 ováhovaných tagov, ktoré boli extrahované našou personalizačnou platformou MePersonality z webových stránok reálnej histórie prehliadania. Všetky výsledky testov sú uvedené ako priemer zo 100 behov vykonaných na notebooku Dell Vostro 1700 s procesorom 2 GHz T7250 Intel Core 2 Duo pod operačným systémom Windows 7 64-bit.

3 Platforma MePersonality – možnosti a potenciál

Uvedený návrh sme realizovali v rámci platformy MePersonality[1]. Personalizácia sa vykonáva prostredníctvom personalizačných rozšírení, ktoré realizujeme v podobe jednoduchých skriptov (v jazyku JavaScript), čím je umožnené rozširovať a personalizovať naše riešenie podľa potreby a vkusu používateľa. V rámci personalizačných rozšírení sme navrhli a realizovali:

- Možnosť použitia externých knižníc.
- Vykonávanie požiadaviek na zdroje z iných domén (angl. cross-origin requests).
- Možnosť využívať perzistentnú databázu (databázové API).
- Možnosť využívať personalizačné API (napr. najrelevantnejšie záujmy, najrelevantnejšie webové stránky v rámci záujmov) a pristupovať k histórii používateľa.
- Možnosť využívať API extrakcie kľúčových slov.
- Možnosť využívať komunikačné API (kolaborácia viacerých používateľov).

S využitím tejto funkcionality sa nám otvárajú takmer neobmedzené možnosti personalizácie a vďaka dostupnej funkcionalite aj oveľa pohodlnejšia práca pri vývoji týchto rozšírení. Platforma je navyše v rámci súčasných možností nezávislá od webového prehliadača, takže vývojár sa nemusí sústrediť na problémy s kompatibilitou a môže sa plne sústrediť na hlavný cieľ svojej práce, ktorým je personalizácia.

Komunikácia umožňuje kolaboráciu skupín podobných používateľov, čo prispieva k oveľa väčším možnostiam personalizácie. Keďže poznáme aj lokálne záujmy používateľa, vieme vytvárať tzv. záujmové skupiny, ktoré nevyužívajú iba používateľove globálne záujmy, resp. celú jeho históriu, ale ktoré sú zamerané na určitú doménu v rámci modelu používateľa.

Na platforme MePersonality sme vytvorili aj experimentálne personalizačné rozšírenie, ktoré loguje vyhľadávané výrazy a odporúča výsledky ostatných používateľov. MePersonality používalo približne 15 študentov po dobu 4 mesiacov. V tabuľke 2 môžeme vidieť porovnanie najhľadanejších slov z obdobia krátko po začatí experimentu a súčasnosti. MePersonality predstavuje distribuovanú multi-agentovú personalizačnú platformu, ktorá je jednoducho rozšíriteľná a umožňuje prispôsobovať správanie a vzhľad webových stránok na základe vytváraného modelu používateľa. Riešenie je jednoduché na nasadenie v podobe rozšírenia do webového prehliadača. V rámci experimentu sme navyše získali nielen dáta o vyhľadávaní na webe, ale vytvorili sme celý nový zdroj týchto dát, ktoré nám neustále pribúdajú.

[1] Domovská stránka projektu - http://vm08.ucebne.fiit.stuba.sk/MePersonality/index.html

Tab. 2. Najvyhľadávanejšie slová (vľavo v začiatkoch experimentu, vpravo v súčasnosti).

Slovo	Počet vyhľadávaní
ruby	96
latex	72
model	65
android	60
emotion	56

Slovo	Počet vyhľadávaní
ruby	189
to	136
javascript	77
bratislava	77
sql	75

Táto platforma má veľký potenciál i pre realizáciu ďalšieho výskumu. Ako decentralizované a zároveň distribuované riešenie vykonáva personalizáciu priamo na koncovom klientskom zariadení s možnosťou komunikácie s ostatnými klientmi, čím umožňuje kolaboráciu viacerých používateľov pre lepšiu a nielen obsahovo bohatšiu personalizáciu. Vďaka jej otvorenosti ju možno rozširovať prakticky akokoľvek – vylepšovanie extrakcie kľúčových výrazov, rozširovanie modelu používateľa, kolaboratívne odporúčanie na webe, realizácia distribuovaných experimentov a pod.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0675/11, 0208-10 a vznikla vďaka podpore v rámci OP Výskum a vývoj pre projekt ITMS: 26240220039, spolufinancovaný ERDF.

Referencie

1. Barla, M. Towards Social-based User Modeling and Personalization. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 1 (2011) pp. 52-60
2. Barla, M., Tvarožek, M., Bieliková, M. Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-based Systems. Computing and Informatics, Vol. 28, No. 4, 2009, pp. 399-427.
3. Bennett, P., Radlinski, F., White, R., Yilmaz, E. Inferring and using location metadata to personalize web search. Proc. of the 34th int. conf. ACM SIGIR on Research and development in Information Retrieval (SIGIR '11), Beijing, China, 2011, pp. 135-144.
4. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A. User Profiles for Personalized Information Access. The Adaptive Web: Methods and Strategies of Web Personalization.: Springer Berlin / Heidelberg, 2007, Vol. 4321, pp. 54-89.
5. Matthijs, N., Radlinski, F. Personalizing web search using long term browsing history. Proc. of the fourth ACM int. conf. on Web search and data mining (WSDM '11), Hong Kong, China, 2011, pp. 25-34.
6. Sugiyama, K., Hatano, K., Yoshikawa, M. Adaptive web search based on user profile constructed without any effort from users. Proc. of the 13th int. conf. on World Wide Web (WWW '04), New York, NY, USA, 2004, pp. 675-684.
7. Šajgalík, M. Decentralised User Modelling and Personalisation. In Proc. of the 8th Student Research Conference 2012 in Informatics and Information Technologies Bratislava, Vol. 1, Nakladateľstvo STU Bratislava, 2012, pp. 175-180.
8. Teevan, J., Dumais, S., Horvitz, E. Personalizing search via automated analysis of interests and activities. Proc. of the 28th annual int. conf. ACM SIGIR on Research and development in information retrieval, Salvador, Brazil, 2005, pp. 449-456.

Predicting Student Performance Utilizing Matrix Factorization

Štefan Pero

Institute of Computer Science,
Faculty of Science, University of Pavol Jozef Šafárik, Košice, Slovakia
stefan.pero@student.upjs.sk

Abstract. Recommender systems attempt to predict items for a user in which he might be interested. One of their aims is learning user preferences and recommend products for any user. Recently, they are also aplied in e-learning for recommending learning objects (e.g. subjects) to students. This paper presents state-of-the-art matrix factorization techniques which can be used not only for recommendation any products for users, but also in the educational data domain. We formulate the problem of predicting student performance as a recommender system problem.

Keywords: recommender system, matrix factorization, predicting student performance

1 Introduction

Recommender systems are largery used nowdays in many areas, especially in the e-commerce, to generate items of interest for users [1] [2]. Recently, they are also useful in e-learning for a recommending resources for students (papers, books, hyperlinks, ...).

On the other side, educational data minnig has also been taken into account recently. Since the universities want to improve their educational quality and success of students, the usage of data mining methods in higher education is more attractive for university managers. It helps also students to improve their performance.

Moreover, many universities are very focused on assessment, thus, the pressure on teach to the test leads to a significant amount of time spending for preparing and taking standardized tests, so any advances which allow us to reduce the role of standardized tests hold the promise of increasing deep learning [4].

In educational data mining viewpoint, a good model which accurately predicts student performance could replace some current standardized tests. Many papers have been published about student performance predition, but most of them address to classification, regression methods, neural networks or support vector maschines. Recommender systems, especially matrix factorization have been not very used in this area [5].

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 67-70.

This work proposes a approach which applies recommender system techniques for predicting student performance and using predicted results for improving rating of students. We use real datasets collected during the last two years from the course Programming Algorithms and Computation taught by our Institute of Computer Science.

2 Problem formulation

Computer-aided tutoring systems allow students to solve some problems (exercises) with a various tools that can be useful for students and help automate some annoying tasks, provide some hints and feedback to the students. Such systems can profit from anticipating student performance (e.g. selecting the right combination of exercises, choosing an appropriate level of difficulty). The problem of student performance prediction means to predict the likely performance of a student for some exercises. Computer-aided tutoring systems also allow to collect a rich amount of additional informations about students behaviour (interaction with system and his past successes and failures) and about tasks (difficulty of task and success to solve).

Formally, let S be a set of students IDs, T be a set of tasks IDs and $p \subset R$ be a performance measure, then $D \subset (S \times T \times p)$ is the triple data collected from computer-aided system. In our dataset is p collected during the teaching and means current score of a one task of any student. Given $s \in S$ and $t \in T$, our problem is to predict p. Obviously, in a recommender system content, s, t and p would be *user*, *item* and *rating*, respectively. The recommender system task is the rating prediction. In our case the main aim is then framework learns more about (the difficulty of) its tasks and about the users' expected performance (and thus where help might be appropriate).

3 Recommender systems techniques

First strategy is content filtering approach which creates profile for each user or product to characterize its attributes. For example, user profile might consist of his personal attributes or his answers to questions and movie profile could include attributes such as genre, author, time, etc. The profiles allow programs to associate users with matching products.

3.1 Matrix factorization

The most successful realization of latent factor models are based on matrix factorization models which map students and tasks to a joint latent factor space of some dimensionality. High correspondence between student and task factors leads to a recommendation. These models have become popular in recent years by combining good scalability with predictive accuracy.

Recommender systems use different types of input data which are placed in the matrix with one dimension representing users and the second dimension representing items of interest.

The goal of matrix factorization techniques is to approximate original matrix X as a product of two much smaller matrices:

$$\mathbf{D} \approx \mathbf{WH}, \tag{1}$$

where $\mathbf{W}$ is an $\mathbf{I} \times \mathbf{K}$ and $\mathbf{H}$ is a $\mathbf{K} \times \mathbf{J}$ matrix (I, J are number of students respectively tasks and K is number of latent factors) [3]. Each row i in matrix W is a vector containing K latent factors describing the student i and each row j in matrix H is a vector containing K latent factors describing task j [2]. Let $w_{i,k}$ and $h_{j,k}$ are elements of W and H, respectively, then performance p given by user i and product j is predicted by:

$$\hat{p}_{i,j} = \sum_{k=1}^{K} w_{i,k} h_{j,k} = (WH^T)_{i,j} \tag{2}$$

The main issue of this technique is how to find optimal parameters (elements of) W and H given a criterion such as root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{ij \in \mathcal{D}^{test}} (p_{i,j} - \hat{p}_{i,j})^2}{\mid \mathcal{D}^{test} \mid}} \tag{3}$$

4 Illustrative example

To illustrate our approach we present a small example. Our purpose is to cluster students into groups with respect to their success in solving the tasks. The key problem is that not every student have solved each task. We will proceed as follows. We predict the success of students in tasks that they have not solved yet (empty values in the input table).

The table in the left side of the figure 1 contains five students who have performed several tasks. The total marks that can be scored are 9; 3 marks for each task. Table values represent students' marks for tasks. The matrix represents a relation between students and tasks.

Figure 1 shows an example of how we can factorize the students and tasks. After the training phase with $K = 2$ latent factors (Factor1 and Factor2), we get the student-factor matrix W and the factor-task matrix H. Suppose we would like to predict Tom's (3rd row) performance for the task 2 (2nd column). The prediction is performed using equation 2

$$\hat{p}_{32} = \sum_{k=1}^{K} w_{3k} h_{2k} = 1.80 * 1.20 + 0.13 * 1.08 = 2.30.$$

Since the predicted value should be included in the input matrix, we round this value to one of the values 1, 2 and 3. We have predicted Tom will achieve

D	Task1	Task2	Task3
Katie	3	3	
Dean	2	2	1
Tom	2	**?**	2
Sam		2	
Adam	2	3	2

W	Factor1	Factor2
Katie	1.18	1.52
Dean	1.24	0.31
Tom	**1.80**	**0.13**
Sam	0.91	1.57
Adam	1.68	0.49

H	Task1	Task2	Task3
Factor1	0.98	**1.20**	0.85
Factor2	1.35	**1.08**	0.54

Fig. 1. Factorization of the input matrix D.

average results in task 2. Similarly, we can predict the performance of other students in tasks which they have not yet done.

Now, we would like to cluster students according to their results in solving the tasks.

5 Conclusion

We propose an approach which uses recommender system techniques for educational data mining, especially in predicting student performance. We also propose how to map the educational data to user/item in recommender systems.

In the future work, we will validate this approach, we will compare recommender system techniques with traditional regression methods such as logistic regression by using educational data.

Acknowledgements. This work was supported by the grants VEGA 1/0832/12 and VVGS-PF-2012-22 at the Pavol Jozef Šafárik University in Košice, Slovakia.

References

1. Y. Koren, R. Bell, C. Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer Society, (2009)
2. N. Thai-Nghe, L. Drumond, T. Horváth, Krohn-Grimberghe, A. Nanopoulos, L. Schmidt-Thieme. Factorization Techniques for Predicting Student Performance. Workshop on Knowledge Discovery in Educational Data (KDDinED 2011). Held as part of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, (2011)
3. G. Takács, I. Pilászy, B. Németh.Investigation of Various Matrix Factorization Methods for Large Recommender Systems KDD Netflix workshop, (2008)
4. M. Feng, N. Heffernan, K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses, User Modeling and User-Adapted Interaction 243266, ISSN 0924-1868, (2009)
5. N. Thai-Nghe, L. Drumond, Krohn-Grimberghe, L. Schmidt-Thieme. Recommender System for Predicting Student Performance. 1stWorkshop on Recommender Systems for Technology Enhanced Learning, Procedia Computer Science, (2009)

Vplyv charakteristík jednotlivcov na priebeh spolupráce

Ivan Srba, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita,
Ilkovičova 3, 842 16, Bratislava, Slovensko
{srba,bielik}@fiit.stuba.sk

Abstrakt. Na základe dát zozbieraných počas dlhodobého kombinovaného riadeného a neriadeného experimentu analyzujeme ako dokážu rôzne typy charakteristík používateľov (kolaboratívne, študijné a osobnostné) ovplyvniť nadväzovanie a priebeh spolupráce pri riešení stanovených úloh. Výsledky tejto analýzy plánujeme následne aplikovať počas navrhovania modelov systémov pre podporu efektívnej spolupráce.

Kľúčové slová: spolupráca, charakteristiky, vzdelávanie

1 Modely systémov pre podporu spolupráce

Súčasný web charakterizuje prívlastok sociálny. Je to spôsobené predovšetkým tým, že sociálny softvér sa stal populárny v širokom spektre rozličných používateľov webu. Pri vzájomnej interakcii a spolupráci sa tak stretávajú používatelia s odlišným sociálnym kontextom. Používatelia sú navzájom rôzni svojimi charakteristikami, cieľmi, ktoré sledujú a tiež aktivitami, ktoré vykonávajú, aby svoje ciele dosiahli. To umožnilo vznik nového fenoménu, ktorým je navrhovanie modelov systémov pre podporu spolupráce používateľov pri dosahovaní svojich cieľov. Toto smerovanie je možné sledovať predovšetkým v doméne kolaboratívneho vzdelávania (angl. *Computer Supported Collaborative Learning, CSCL*) a v doméne kolaboratívnej práce (angl. *Computer Supported Collaborative Work, CSCW*).

Efektívnosť spolupráce používateľov v malých skupinách záleží od mnohých atribútov, ktoré ovplyvňujú jej kvalitu a dosiahnuté výsledky počas všetkých etáp životného cyklu skupín [1]. Pri súčasnom stave poznania však nevieme jednoznačne určiť, akú rolu majú tieto atribúty pri nadväzovaní a realizovaní spolupráce. Aby bolo možné nájsť odpoveď na túto otázku, treba navrhovať nové metódy a nástroje pre analyzovanie a modelovanie vzájomnej interakcie [2]. Takýto prístup ku skúmaniu spolupráce používateľov sa označuje ako dialogický a je založený na myšlienke, že spolupráca pri riešení úloh je sociálne organizovaná aktivita [3]. Predmetom analýzy sú skupiny používateľov, ktorí spolupracujú, aby dosiahli spoločný cieľ. Kľúčovými konceptmi je sprostredkovanie znalostí, vlastný prínos, sociálne zručnosti, kolaboratívne nástroje a vzájomná interakcia cez tieto nástroje. Tieto koncepty predstavujú

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 71–74.

jednotlivé aspekty a otvorené problémy, v ktorých je možné podporiť spoluprácu používateľov.

Prvá etapa životného cyklu skupín, ktorou je rozdeľovanie používateľov do skupín, má významný vplyv na nasledujúci priebeh spolupráce. Vytvorenie vhodnej skupiny však nie je postačujúce pre dosiahnutie skutočne efektívnej spolupráce a je potrebné venovať pozornosť aj návrhu kolaboratívneho prostredia, v ktorom spolupráca prebieha. Z tohto dôvodu sme vyvinuli webový vzdelávací systém *PopCorm* (*Pop*ular *Co*llaborative Platfo*rm*) [5], ktorý bol použitý aj pre získanie dát opísaných v tomto príspevku. Vzhľadom na sociálny charakter dialogického prístupu sme sa rozhodli v týchto dátach analyzovať vplyv charakteristík používateľov na priebeh spolupráce.

2 Analýza vplyvu charakteristík na priebeh spolupráce

Pri analýze vplyvu charakteristík používateľov na priebeh spolupráce vychádzame z dlhodobého kombinovaného riadeného a neriadeného experimentu v oblasti kolaboratívneho vzdelávania. Za skúmané charakteristiky sme si zvolili kolaboratívne správanie študentov, ich študijné výsledky a osobnostné charakteristiky.

2.1 Definovanie a štatistika experimentu

Dlhodobý kombinovaný riadený a neriadený experiment [4] sme realizovali v rámci predmetu Princípy softvérového inžinierstva (PSI), ktorý sa prednáša počas letného semestra v druhom ročníku na FIIT STU v Bratislave. Riadená časť experimentu prebiehala formou koordinovaných stretnutí, počas ktorých mali študenti možnosť spolupracovať na úlohách súvisiacich s prednášaným učivom. Neriadená časť experimentu prebiehala popri individuálnom vzdelávaní študentov.

Experimentu sa celkovo zúčastnilo 110 študentov, ktorí boli zaradení opakovane do 254 skupín. Na riešení krátkodobých úloh spolupracovali priemerne 11 minút v skupinách skladajúcich sa z dvoch alebo troch členov. Celkovo sme počas experimentu zaznamenali 3763 aktivít, pričom jednotlivé aktivity boli automaticky odvodené na základe odoslaných správ v semištruktúrovanej diskusii.

Výsledky spolupráce študentov sme vyhodnotili prostredníctvom ôsmich dimenzií. Prvých sedem zodpovedalo kritériám úspešnej spolupráce určených na základe psychologických štúdii (napr. rovnomerné rozdelenie úloh, plynulosť spolupráce). Posledná ôsma dimenzia bola určená na základe manuálneho hodnotenia vypracovaného výsledku pedagógom.

2.2 Definovanie charakteristík

Za skúmané charakteristiky študentov sme si zvolili tri navzájom odlišné oblasti charakteristík. C*harakteristiky kolaboratívneho správania* sme odvodili automaticky na základe analýzy spolupráce študentov počas riešenia úloh. Pre každého študenta sme zvolili tri najtypickejšie aktivity, ktoré používal počas spolupráce. Príkladmi takýchto aktivít je navrhnutie lepšieho riešenia, upozornenie na chybu alebo navrhnutie akcie.

V rámci *študijných výsledkov* sme sa zamerali na vážený študijný priemer (VŠP) a výsledky dosiahnuté na niektorých relevantných predmetoch. Pre získanie *osobnostných charakteristík* vyplnili študenti psychologický dotazník *Big Five*, ktorý bol následne vyhodnotený za pomoci psychológov. Výstupom z dotazníka je určenie osobnostných charakteristík študentov v rámci päťfaktorového modelu osobnosti (neurotizmus, extraverzia, otvorenosť, príjemnosť a svedomitosť).

2.3 Výsledky analýzy

Najprv sme sa zamerali na skúmanie vplyvu charakteristík kolaboratívneho správania na priebeh a výsledky spolupráce, a to konkrétne ako veľmi súvisia jednotlivé aktivity s kvalitou výsledku. S manuálnym hodnotením pedagóga relatívne najviac korelovalo napísanie pochvaly (korelačný koeficient 0,28), navrhnutie akcie (0,23) a upozornenie na chybu (0,20). Sledovali sme tiež počet študentov, ktorí využívali jednotlivé typy aktivít: najviac študentov využívalo akceptovanie akcie (počet študentov 55), napísanie pochvaly (47) a navrhnutie akcie (40). Z týchto hodnôt môžeme odvodiť, že k *pozitívnemu výsledku spolupráce študentov prispievala ich samoregulácia*. Bez potreby explicitnej kontroly pedagóga boli študenti schopní samostatne riadiť svoju spoluprácu, upozorňovať na prípadné nedostatky vo vypracovaní úloh a vylepšovať tak svoje riešenie. Kladne hodnotíme aj zistenie, že študenti sa dokázali vzájomne motivovať napísaním pochvaly za dobre vypracovaný príspevok do riešenia zadanej úlohy.

Z kolaboratívnych charakteristík sme vyhodnotili aj do akej miery koreluje manuálne hodnotenie pedagóga so siedmimi automaticky vypočítavanými dimenziami. Relatívne najviac koreluje plynulosť spolupráce (0,35), udržovanie vzájomného porozumenia (0,18), argumentovanie a dosahovanie konsenzu (0,18) a rovnomernosť rozdelenia úloh (0,16). Na základe týchto poznatkov môžeme vyvodiť niekoľko záverov. Úspešnejšie boli tie skupiny, v ktorých si študenti dokázali rozdeliť zadanú úlohu na niekoľko menších častí a následne ich vypracovali približne s rovnomerným podielom a aj rovnomerne v čase. Navyše ku kvalite dosahovaného výsledku prispieva kritické hodnotenie vytváraného výsledku a snaha vytvárať obsah tak, aby bol zrozumiteľný pre všetkých členov skupiny. Predovšetkým vplyv dimenzie argumentovania a dosahovania konsenzu hodnotíme veľmi pozitívne, pretože študenti sa počas riešenia úloh snažili vyjadrovať svoj súhlas, príp. nesúhlas s navrhovanými vylepšeniami riešenia. Tento záver je v kontraste s výsledkom niektorých podobných výskumov, v ktorých mali študenti tendenciu sa vyhýbať kritickému hodnoteniu.

Zo študijných výsledkov sme sa zamerali na analýzu korelácie medzi váženým študijným priemerom (VŠP) a dosahovaným hodnotením študentov. Priemerné hodnotenie študenta na úlohu a VŠP korelovalo mierne negatívne (-0,25), naopak, celkové hodnotenie (súčet hodnotení za všetky úlohy) korelovalo s VŠP mierne pozitívne (0,24). To znamená, že lepší študenti (s nižším VŠP) vyriešili síce menej úloh, ale dosahovali v nich lepšie výsledky v porovnaní s ostatnými študentmi. Slabší študenti s vyšším VŠP dosahovali vyššie celkové hodnotenie tým, že vyriešili väčší počet úloh aj keď s nižšou kvalitou. Na tento výsledok mala vplyv aj motivácia, ktorou bolo získanie bonusových bodov pre študentov, ktorí sa aktívne zapojili do experimentu.

3 Záver a ďalšia práca

Na základe aktuálnych výsledkov analýzy vplyvu kolaboratívnych a študijných charakteristík sme odvodili niekoľko zaujímavých výsledkov. Samostatnosť študentov pri riešení úloh poukazuje, že v oblasti koordinovania spolupráce nie je potrebné explicitné zasahovanie pedagóga do priebehu spolupráce. Na druhej strane systém môže automaticky poskytovať podporu nielen v oblasti problémovej domény, ale môže aj usmerňovať samotný priebeh vzájomnej interakcie tak, aby sa študenti naučili efektívnejšej spolupráci. Pri dosahovaní tohto cieľa môžeme nadviazať na výskum, ktorý je realizovaný na našej fakulte [6].

Schopnosť študentov rozdeliť si prácu na niekoľko menších častí a spolupracovať na nich paralelne s cieľom vytvárať zrozumiteľný obsah aj pre ostatných členov skupiny má zase dôležité implikácie pre oblasť návrhu techník kolaborácie. Dostupné nástroje musia umožňovať nezávislú prácu študentov s možnosťou vzájomného kritického pripomienkovania a zlepšovania vytváraného obsahu.

Z analýzy študijných výsledkov vyplýva dôležité zistenie pre oblasť vytvárania skupín. Pri vyberaní členov skupiny je potrebné zohľadniť rozsah ich vedomostí a odborné zameranie, ktoré vieme odvodiť na základe výsledkov dosiahnutých v ostatných predmetoch.

V analýze plánujeme ďalej pokračovať skúmaním vplyvu osobnostných charakteristík, ktoré majú potenciál mať signifikantný vplyv na nadväzovanie a priebeh spolupráce. Výsledky tejto analýzy následne plánujeme použiť ako základ pre návrh metód pre podporu spolupráce v stanovených oblastiach.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0675/11, APVV 0208-10.

Literatúra

1. Daradoumis, T., Guitert, M., Giménez, F., Marquès, J. M., Lloret, T.: Supporting the Composition of Effective Virtual Groups for Collaborative Learning. In *Proc. of the Int. Conf. on Computers in Education (ICCE '02)*. IEEE Computer Society, Washington, DC, USA, 2002, 332-336.
2. Dillenbourg, P., Baker, M., Blaye, A., O'Malley, C.: The evolution of research on collaborative learning. In P. Reimann & H. Spada (Eds.), *Learning in humans and machines: Towards an interdisciplinary learning science*. Oxford, UK: Elsevier, 1996, 189-211.
3. Ludvigsen, S., and Mørch, A.: Computer-supported collaborative learning: Basic concepts, multiple perspectives, and emerging trends. In *The Int. Encyclopedia of Education*, 3rd Edition, edited by B. McGaw, P. Peterson and E. Baker, Elsevier (in press), 2009.
4. Srba, I., and Bieliková, M.: Encouragement of Collaborative Learning Based on Dynamic Groups. In A. Ravenscroft, S. Lindstaedt, C. D. Kloos, & D. Hernández-Leo (Eds.), *Lecture Notes in Computer Science* (Vol. 7563/2012, pp. 432–437). Springer, Berlin, 2012.
5. Srba, I. Collaboration Support by Groups Creation in Educational Domain. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 4, No. 2 (2012) 62-64.
6. Tvarožek, J. Bootstrapping a Socially Intelligent Tutoring Strategy. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 1 (2011) 33-41.

GAIN: Analysis of Implicit Feedback on Semantically Annotated Content

Jaroslav Kuchař[1,2], Tomáš Kliegr[2]

[1] Web Engineering Group, Faculty of Information Technology,
Czech Technical University in Prague
jaroslav.kuchar@fit.cvut.cz
[2] Dep. of Information and Knowledge Engineering,
Faculty of Informatics and Statistics, University of Economics Prague
tomas.kliegr@vse.cz

Abstract. The trend in application development is to provide a personalized interface. The availability of the user preference level associated with user actions is the key for the personalization process. This paper describes a "work-in-progress" framework for deriving user preference from actions performed on semantically annotated objects – be it web pages or TV news. Preference level is computed using supervised learning with genetic programming from implicit feedback, which might be time on page for the web domain, or the user engagement level for the TV domain. We provide tool called GAIN (General Analytics INterceptor) covering the whole approach at wa.vse.cz.

Keywords: implicit feedback, analytics, semantics, rules

1 Introduction

Personalized interfaces are featured in an increasing number of software applications. Personalization is typically used to customize user interface or displayed content with the intent to increase the efficiency of work of a particular user.

An important part of the personalization process is obtaining the information on user preference. There are two main input channels: implicit and explicit feedback. In this paper we will focus primarily on implicit feedback – tracking and interpreting interactions and behaviour of users. While there are many tools supporting this task, most of them are domain specific: consider e.g. Google Analytics or Piwik (`piwik.org`) for the web domain. There is an abundance of proprietary solutions for other domains, such as multimedia applications.

According to our observation, the problem addressed in personalization tends to be algorithmically similar for most domains. In this paper we propose a "work-in-progress" General Analytics INterceptor (GAIN) tool for capturing and preprocessing user actions.To prove the domain independent character of our tool, we deployed it to a travel agency website (as an extension of Google Analytics Tracking code) and GAIN is being incorporated within the LinkedTV EU Project to a TV player to obtain TV usage data.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 75–78.

2 Architecture

This section describes the four main modules that comprise the architecture of our framework. Since GAIN is a general analytics tool, we use a generic terminology fitting not only web analytics, but also other areas incl. the TV domain (ref. to Table 1).

Table 1. Generic Terminology.

general	web	brief definition
object	pageview	entity, which can be interacted with
session	visit-session	series of user actions
user	visitor	the one who interacts
preference level	conversion	explicit preference level
interaction	event	user feedback
attribute	custom variable	semantic description of object

Tracking module: is responsible for capturing interactions, information is sent in the form of events to the application module. This module is integrated into the user interface of the application that is tracked. We support two implementations of this module: modified Google Analytics (GA) tracking code (`ga.js`) for the web domain and a REST API for other domains. Both cover the same functionality. GA tracking code is easy to integrate and provides simple functions for tracking actions in web-based or javascript-supporting environments.
Application module: contains main application logic and exposes API for importing and exporting data. We decided for the Node.js platform (`http://nodejs.org/`) which uses event-driven, non-blocking I/O model. This platform is lightweight, very efficient and scalable suitable for data-intensive applications and scalable services. The application module processes interactions from multiple tracking modules. After initial processing, the data are passed to the storage module.
Storage module: is implemented as a NoSQL document-oriented database MongoDB (`http://www.mongodb.org/`). The key advantage of this solution is a schema-less design, which fosters the attachment of semantic description to objects, with which the user interacted.
Aggregation module: processes the data sent by the Tracking module, which were preserved by the Storage module. Standard aggregation functions provided such as number of interactions per day, average number of interactions per session, are implemented as Map/Reduce tasks. Interactions can be also aggregated using custom operators working on the basis of hand-coded rules or supervised learning (see Sec. 3). The difference compared to other solutions is that semantic annotation of content is also processed. These annotations can be sent by the tracking module simultaneously with each interactions or they can be added during the aggregation process.

3 Preference Computation and Aggregation

The data output by existing tracking systems are typically unsuitable for direct processing by mainstream machine-learning algorithms. One reason is data sparsity, since the distance between objects or their values is not sensitive to the semantics of the domain. The second reason is that the interactions of individual users tend to be of uneven length (number of pageviews per session in the web domain). Without special preprocessing, such input cannot be consumed by most mainstream machine-learning algorithms.

Our framework implements an aggregation step which addresses both these problems. The data can be semantically enriched either immediately during the data collection phase, or during the aggregation task. Multidimensional semantic (taxonomical) description of tracked objects are processed along with implicit user feedback to the lower-dimensional output representation, with a tabular form suitable for analysis with mainstream data mining algorithms.

This section covers the aggregation module, which is the the main component for analyzing implicit feedback. Due to limited space, we will focus only on user preference: its computation and use for dimensionality reduction.

Preference Computation
Our framework defines three ways of computing preference value.

Genetic Algorithm: uses symbolic regression to learn algebraic function from training data consisting of user interactions with objects, each interaction being assigned a preference level. The output of the function is a preference level for each object. E.g. for the web domain, the input of the function are click-streams of converted visitors.[3] The function is learnt so that if applied on individual pages (objects) in the clickstream, the page with conversion has the highest value. Details can be found in [2,3]. The disadvantage of this approach is that it requires a large amount of training data annotated with explicitly expressed preference levels.

Heuristically defined rules: each rule assigns level of preference for each microconversion type.[4] Microconversions have the advantage that they are generated typically more frequently than the "full" conversions. On the other hand, manual definition of these rules is resource-intensive and possibly error-prone.

Association rule learning: a learning process similar to the genetic algorithm, a training set with explicitly defined preference levels is required. For rule learning, we plan to use the web service interface to the LISp-Miner system (`lispminer.vse.cz`) developed within the I:ZI Miner project (`www.izi-miner.eu`). The function learnt by the genetic algorithm is able to assign level of preference for each instance. In contrast, association rules generally cover only part of the space, but the confidence is higher since each rule needs to meetp re-

[3] Converted visitor explicitly manifested their preference, typically by buying one of the objects in the clickstream.

[4] Microconversion is an interaction, which hints at user's positive level of preference for the object on which it was captured. An example of a micro conversion is the user adding an item to the shopping cart.

defined confidence and support thresholds to appear on the output. Examples of such association rules are *bookmark=yes and topic=Sport* $\rightarrow$ $preference(1)$ or *shopping_cart_insert=yes* $\rightarrow$ $preference(2)$. The disadvantage of association rules is that they are not learnt in a way that would ensure their consequents to be "additive". For example, if the antecedents of both above-listed rules match, it is not correct to sum their outcomes ($preference = 3$).

Aggregation
The output of the previous step is the level of preference for each interaction. These values are subsequently used for aggregation. The aggregation granularity is either *object*, *session* or *user* level.

Consider the example for the session level. Each object is described with a three-dimensional vector, where the first two dimensions correspond to the content (here *location*, *activity*), and the third to the level of user preference. The three objects vectors: $(Berlin, Football, 0.4)$, $(Berlin, Tennis, 0.4)$ and $(Vienna, Tennis, 0.6)$.

A straightforward approach is to select the object with maximum preference level obtaining $(Vienna, Tennis)$ as the aggregation result on the session level.

However, since it is often the case that none of the objects is most preferred by the user in all the attributes, an option is to assign a sum of preference values to each attribute value ($Berlin = 0.8, Football = 0.4, Tennis = 1.0, Vienna = 0.6$) and use these to construct the ideal object by selecting the best value in each dimension yielding $(Berlin, Tennis)$. Note that irrespective of the number of the input objects, the result is always a fixed-sized vector (here of length 2).

4 Conclusion and Future Work

There is an ongoing work on integration of the GAIN tool with the I:ZI Miner system. Following the acknowledged limitations of applying the ready-made association rule mining algorithms in the aggregations step, as a future we would like to investigate rule-based approached proposed specifically for the preference learning domain, such as [1]. GAIN demo is available at `wa.vse.cz`.

Acknowledgements. This work was supported by the Czech Technical University grant SGS12/093/OHK3/1T/18, by the University of Economics, Prague by grant IGA 26/2011 and by the EC project FP7-287911 LinkedTV.

References

1. K. Dembczynski, W. Kotlowski, R. Slowinski, and M. Szelag. Learning of rule ensembles for multiple attribute ranking problems. In J. Fürnkranz and E. Hüllermeier, (eds.) *Preference Learning*, pp. 217–247. Springer, (2010)
2. T. Kliegr. Representation and dimensionality reduction of semantically enriched clickstreams. In *Proceedings of the 2008 EDBT Ph.D. workshop*, Ph.D. '08, pp. 29–38. ACM, New York, NY, USA, (2008)
3. J. Kuchař and I. Jelínek. Scoring Pageview Based on Learning Weight Function. *International Journal on Information Technologies and Security*, 2(4), 19–28 (November 2010)

Implicit Feedback in Recommender Systems

Ladislav Peska, Peter Vojtas

Department of Software Engineering
Charles University in Prague
Malostranske namesti 25, Prague, Czech Republic
{peska | vojtas}@ksi.mff.cuni.cz

Abstract. In this paper, we present our vision and some initial experiments on how to anticipate significance, similarity and polarity of various types of (preferably implicit) user feedback. Throughout the area of e-commerce recommenders, we can observe the similar patterns in user behavior (e.g. page-view, amount of scrolling, time on page, purchasing etc.). Although such implicit feedback is often used in both commerce recommenders and scientific experiments, there is yet no agreement on how should we combine various implicit factors or how to measure their relevance.
We propose a model of recommending system based on multiple feedback factors, where importance, combinations and preference of feedback values are dynamically set based on a success metrics. We show several basic methods how to learn our model. Our preliminary experiments supported our ideas and shown interesting differences in evaluation results of various feedback.

Keywords: User preference, implicit feedback, recommender systems

1 Introduction, motivation, contribution and related work

Recommending on the web is both an important business sector and an interesting research topic. The amount of data on the web grows continuously and even a single e-commerce site contains more data, than user is willing to view. The search engines were adopted to fight information overload but they are helpful only if user knows what he/she is looking for. Traditionally the most of the research time and effort in the recommending area was spent on the explicit user rating based experiments and single success metrics (RMSE). However the user ratings are often too rare to provide reasonable output alone and RMSE does not necessarily reflect real-world success metrics like increase in purchases or revenues for the e-commerce.

While moving to the implicit feedback, where user behavior is recorded without user cooperation, we may receive abundant amount of data, but the link between feedback and user preference becomes less clear, especially if we record various

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 79–82.

feedback factors (e.g. time on page, amount of scrolling, mouse clicks, etc.). We face problems like: *Which user actions should be recorded*, *how strong is the link between user feedback, user preference and success metrics* or *how to combine together results from various user actions.*

Motivation example. We focus on an e-commerce site employing personalized object recommendation – e.g. a *travel agency*. On such site we can record several types of user's feedback such as *page-views*, *time spent on page*, *purchasing* related actions, etc. Each of these factors is believed to be related to the user's preference on the object, but the relation is often noisy or biased. Our goal is to identify good factors and combine them in order to improve recommendations.

The **main contribution** of our work will be:

- Proposing new recommender system with self-adjusting model of user preferences based on multiple implicit feedback factors.
- Proposing basic methods to learn the model features.
- Preliminary "feasibility" study on an off-line dataset from the travel agency.

Related work. The area of recommender systems has been extensively studied recently and it is outside the scope of this paper to provide an in-depth study. We suggest Adomavicius and Tuzhilin [1] or Konstan and Riedl [4] papers for overview. Our approach is based on the work of Eckhardt [2] where we have adopted the idea of evaluating various object attributes according to their correlation with the user ratings and transformed it into the evaluating of various feedback factors according to our success metric. Important for our research is also work of Kiessling et al. [3] on the Preference SQL system. The Preference SQL is an extension of SQL language allowing user to specify directly preferences (or so called "soft constraints") and to combine them in order to receive best objects. We use the three described combination operators: *Prior to (hierarchical)*, *Ranking* and *Pareto* in our proposed model for combining implicit factor preferences.

Our previous experiments on evaluating implicit feedback are published in [5].

2 Self-adjusting learning model for user preferences

In our previous work e.g. [5], we have evaluated implicit factors recommendation over the business success metrics. We have divided the experiments into the two phases. In the first phase the implicit factors were examined separately and in the second phase (based on the findings from phase one) we tried various ways how to combine feedback together. However such approach is impracticable in the production recommender. Our key goal was then to develop system where the evaluating of each feedback factor can be done "on the run" and the results can simultaneously or at least iteratively influence the recommending algorithm.

The Figure 1 illustrates our approach: to create recommendations we first evaluate each feedback value separately for arbitrary fixed object and user with the *Pref()* method and receive list of local user to object preferences based only on the single feedback factor value. Then we combine local preferences together into the global

preference via *@()* method according to the feedback factor importance *Imp()*. Afterwards we can use any recommending algorithm suitable for user rating (k-nearest neighbors, matrix factorization etc.).

At the same time we receive data about how we are fulfilling our success metrics (e.g. number of sales). From the relation between feedback factors and the success metrics we infer the importance of each factor and the local preferences on the factor values. Those data are then used as parameters for *Pref()* and *@()* methods.

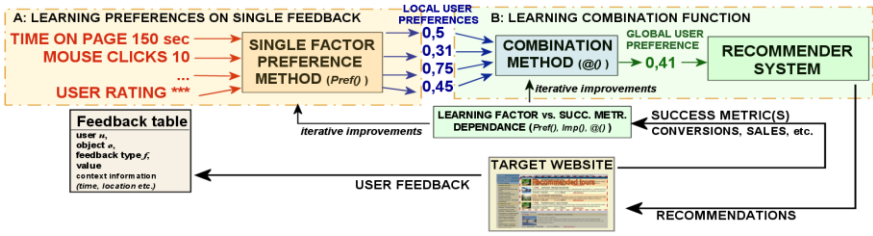


Fig. 1. Overview of the self-adjusting model of user.

Learning methods and experiments. We have to learn *Pref()*, *Imp()* and *@()* methods for each feedback factor. Learning is based on positive examples. These examples are evidences of success in our goals e.g. user is purchasing an object.

The model is updated periodically either after amount of time or after exceeding amount of new positive examples. For each feedback factor F we first create distribution of feedback value vs. positive examples. We cluster the feedback value range into C (closest possible to) equipotent clusters to decrease noise. The clustering must respect, that *Factor* records with the same value have to be in the same cluster C. Figure 2 shows such histograms from our off-line experiment with travel agency dataset and number of purchases as the success metric.

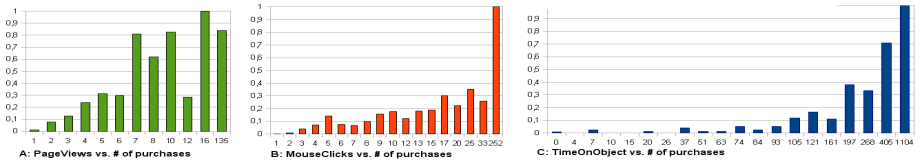


Fig. 2. Histograms of page views (A), mouse clicks (B) and time on object (C) factors against amount of purchases (data linearly normalized into [0, 1] interval).

Learning *Pref()* method. To each cluster c we assign value Pos_C as ratio between positive examples and all records in this cluster. We can then define the preference of feedback value $Pref_C()$ as Pos_C normalized into [0, 1] interval.

Learning *Imp()* method. *Imp()* function reflects importance of the factor F. The factor is more important if it more divides its values into preferred and unprefered. We define *Imp()* as inversion of standard deviation of Pos_C values of current factor F.

Learning *@()* method. For purpose of learning aggregation function, we have adopted *Combination operators* from Preference SQL. We interpret them in this way:

- *Prior to* operator is clearly stating that one factor is more important than others. The final ordering or rating is computed lexicographically.
- *Ranking* operator introduces a tradeoff between the feedback factors e.g. weighted average, multiplications (both factors crucial) etc.
- *Pareto* operator addresses equally important factors or those where we are not sure about their relation.

The @() method should use some of those operators with respect to the importance of each factor *Imp()*. The figure 3 illustrates this approach on our experimental data.

Feedback factor	StdDev	Imp()	@()
MouseClicks	0,0872	11,47	(27*Pref(*TimeOnObject*) + 20*Pref(*PageViews*)) ***PRIOR TO*** Pref(*MouseClicks*)
TimeOnObject	0,0373	26,81	
PageViews	0,0491	20,37	

Fig. 3. *Imp()* and *@()* methods for travel agency data. *@()* constructed as follows: if the difference in consecutive *Imp()* results exceeds 40%, use Prior to, otherwise weighted average.

3 Conclusions and future work

Our main goal was to present new recommending system using self-adjusting model of user preferences based on multiple feedback types. We have described the overview of such system and basic methods how to learn the user model.

Our proposed system is currently in early stages of evaluation and implementation, so immediate future work includes finishing implementation, benchmarking and tuning the system on several datasets.

Acknowledgements. This work was supported by the grant SVV-2012-265312 and GACR 202-10-0761.

References

1. Adomavicius G ; Tuzhilin A. Toward the next generation of recommender systems. IEEE T KNOWL DATA EN Vol: 17 Issue: 6 Pages: 734-749.
2. Eckhardt, A.: Similarity of users' (content-based) preference models for Collaborative filtering in few ratings scenario. Expert Syst. Appl. 39(14): 11511-11516 (2012).
3. Kießling, W.; Endres, M. & Wenzel, F. The Preference SQL System - An Overview. IEEE Data Eng. Bull., 2011, 34, 11-18.
4. Konstan, J.; Reiedl, J.: Recommender systems: from algorithms to user experience. User Model. User-Adapt. Interact. 22(1-2): 101-123 (2012).
5. Peska, L.; Vojtas, P.: Evaluating Various Implicit Factors in E-commerce. To appear in *RUE 2012*, http://ir.ii.uam.es/rue2012/papers/rue2012-peska.pdf

Analýza a modelovanie dát

Využití ontologií k popisu vizuálních rysů dokumentu

Martin Milička, Radek Burget

Fakulta informačních technologií,
Vysoké učení technické v Brně
{imilicka,burgetr}@fit.vutbr.cz
http://www.fit.vutbr.cz

Abstrakt. Cílem této práce je seznámit čtenáře s možným použitím ontologií k popisu vizuálních rysů webových dokumentů. Tento přístup zavádí nový způsob popisu a zpracování vizuálních informací, které jsou na webu publikovány. Základ je tvořen ontologií, která umožňuje popis pomocí obecného grafu. Díky ní a získaným vizuálním rysům (konkrétním individuím) je z RDF dat sestaven model dokumentu. Nad takto definovaným popisem dokumentu je pak možné provádět dotazování nebo sofistikovanější zpracování uložené informace.

Klíčová slova: ontologie, web, vizuální rysy, získávání znalostí, model dokumentu

1 Úvod

Vzhledem k tomu, že byly webové dokumenty prioritně navrženy pro člověka a zpracování dokumentů pouze na základě zdrojového kódu nepřinášelo očekávané výsledky, výzkum se v této oblasti musel začít orientovat na strojové zpracování zohledňující vizuální vzhled.

Aby bylo možné provést zpracování dokumentu tak, jak jej vidí člověk, dokument musí být ze zdrojového kódu vyrenderován. Je to z toho důvodu, že ne všechny informace o vzhledu jednotlivých elementů dokumentu jsou explicitně definovány ve zdrojovém kódu. Spousta z nich se tedy určuje až v okamžiku renderování dokumentu.

Z vyrenderovaného dokumentu lze sestavit jeho model, který obsahuje kompletní vizuální informaci. Model dokumentu můžeme popsat několika způsoby. Obvykle se dokument popisuje stromovou strukturou s notací, která odpovídá HTML. Další možností je zavedení abstraktnější notace k HTML jakou například prezentuje Burget v [2] a nebo Alpuente a Romero v [1].

Vzhledem k tomu, že stromová struktura není schopna pokrýt všechny vazby, které do modelu vnáší vizuální rysy, stojí za zvážení využití obecnější struktury, kterou je graf. Implementací grafové struktury jsou ontologie. Ty umožňují vícenásobné vazby a tak i přesnější popis skutečnosti.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 85–88.

2 Příbuzný výzkum

V úvodní kapitole jsme se dozvěděli, že model renderovaného dokumentu se nejčastěji reprezentuje stromovou strukturou. Největší motivací pro tuto reprezentaci je struktura HTML. O jejím použití se můžeme přesvědčit v publikovaných článcích [1,2,4].

Hlubším zkoumáním konstrukce modelu dokumentu jsme zjistili, že stromová struktura nemusí být vždy dostačující. Existují případy, které není možné precizně popsat s ohledem na reálný vzhled dokumentu a vizuální vazby mezi jeho jednotlivými částmi. Příkladem můžou být tabulková data, kdy chceme mít u konkrétní buňky informaci o názvu jejího sloupce, případně naopak. Dalším případem můžou být informace o vzájemných vizuálních vazbách rozmístěných elementů. K reprezentaci takových vícenásobných vazeb mezi prvky dokumentu může být použit obecný graf. Konkrétní implementací takového grafu jsou *ontologie*.

V současné době můžeme nalézt ontologie, které se snaží obsahy dokumentů popisovat. Jednou z takových ontologií je SALT ontologie [5], která se převážně orientuje na dokumenty s lineární strukturou, jako jsou například vědecké články. Základ této ontologie se dá využít k návrhu nové ontologie, která umožní popisovat obecné dokumenty s jejich vizuálními rysy.

V současné době nám není známa žádná ontologie, která by kromě samotného obsahu uvažovala taky vizuální rysy dokumentu.

3 Vizuální rysy

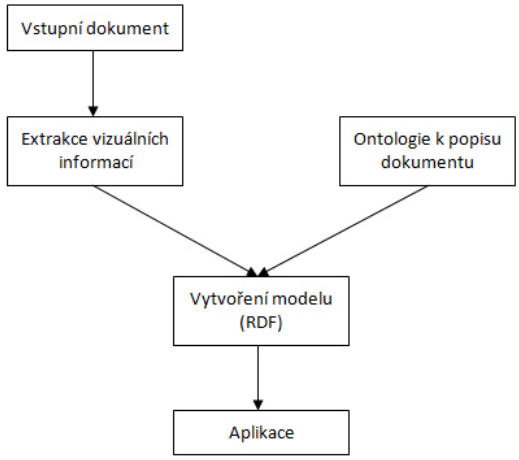
Jak již bylo zmíněno dříve, zpracování dokumentu založené na vizuálních rysech umožňuje pracovat s daty dokumentu tak, jak jej vnímá jeho čtenář. Burget a Burgetová v článku [3] představili ucelenou kategorizaci vizuální rysů, které má smysl zpracovávat. Zavedli kategorie rysů pro použitý font (font features), prostorové rysy (spatial features), rysy textu (text features) a rysy barev (colour features). Každá taková kategorie seskupuje několik příbuzných rysů, jež bychom chtěli do nově vzniklé ontologie začlenit.

Vizuálním rysy umožňují preciznější zpracování dokumentu. Díky nim je možné jednodušeji klasifikovat jednotlivé části dokumentu a tak odstranit šum, který může způsobit nepřesnost výsledků nebo případně identifikovat specifický obsah.

4 Model zpracování

Na obrázku 1 můžeme vidět schéma navrhovaného zpracování webových dokumentů.

K tomu, aby bylo možné sestrojit model dokumentu podle dané ontologie, je nutné provést jeho předzpracování (získání vizuálních informací). Součástí fáze předzpracování je načtení HTML kódu včetně přilinkovaných kaskádových stylů



Obrázek 1. Schéma navrhovaného zpracování dokumentu.

a jejich následné renderování. Tento způsob umožňuje získat přesné vizuální informace, které nejsme ze zdrojového HTML a jeho kaskádových stylů schopni bez renderování získat. Příkladem může být získání jednoznačné pozice bloku v dokumentu, případně jeho vztah k jiným částem dokumentu. Renderování dokumentu nám tedy zajistí, že získáme přesné hodnoty vizuálních rysů. V kontextu ontologií jsou tyto hodnoty nazývány *individua*. Ty následně vstupují do procesu vytvoření modelu dokumentu podle definované ontologie.

Definice ontologie je dalším očekávaným vstupem v procesu generování modelu dokumentu. Umožňuje popsat dokument tak, jak jej vnímá člověk. Žádoucí je, aby se při návrhu nové ontologie využívaly definice již existujících ontologií. Za tímto účelem jsme se rozhodli použít zmíněnou SALT ontologii [5], jež by nám měla návrh nové ontologie usnadnit. Jak již bylo zmíněno dříve, tato ontologie definuje třídy pro lineární popis dokumentů, převážně publikací. To znamená, že nám umožní částečně pokrýt požadavky na popis textového obsahu dokumentu.

Samozřejmostí je, že bude nutné definovat nové třídy, které nám umožní zpracovávat nejenom lineární struktury dokumentů. Nedílnou součástí definice nové ontologie bude taktéž specifikace nových atributů tříd. Bude se jednat o atributy z kategorií zmíněných v kapitole 3, které představil Burget a Burgetová v [3].

Ze vstupního dokumentu, který je předzpracován tak, aby obsahoval jenom individua, a z definované ontologie vznikne model dokumentu. Ten může být reprezentován díky RDF[1], které se stalo základním jazykem pro popis ontologií.

Příkladem tříd, které budou definovány v ontologii, může být *oblast dokumentu* nebo *prvek dokumentu*. Mezi těmito třídami se pak budou definovat vlastnosti jako je *jeNad, jePod, máVelikost, atd.*

[1] Resource Description Framework

Vzhledem k tomu, že v tomto článku není prostor pro detailní definici nové ontologie, byl zde proveden pouze nástin možného řešení.

5 Možnosti použití

Nově definovaná ontologie umožní nad dokumentem popsaným pomocí RDF dat provádět mnohem sofistikovanější dotazy zohledňující vizuální rysy. Nemalou výhodou tohoto přístupu je fakt, že existuje celá řada volně dostupných nástrojů[2], které jsou schopny vyhodnocovat dotazy nad sémantickými daty.

Abychom ukázali sílu navrženého modelu, provedeme jeho integraci do renderovacího stroje CSSBox[3]. Ten v současné době, stejně jako všechny dosud známé renderovací stroje, pracuje se stromovým modelem dokumentu. Očekáváme, že navržený přístup by mohl pozitivně ovlivnit výsledky námi prováděných výzkumů.

Díky vícenásobným vztahům mezi elementy dokumentu budeme schopni získat přesnější znalosti o dokumentu, které se v případě stromového popisu musí někdy zanedbávat.

Model dokumentu, který je popsán pomocí ontologie, je možné taktéž uplatnit v oblasti porovnávání dokumentů založeném na vizuálních rysech.

Poděkování. This work was supported by the research programme MSM - 0021630528, the BUT FIT grant FIT–S–11–2 and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

Reference

1. Alpuente, M., Romero, D.: A Visual Technique for Web Pages Comparison. Electron. Notes Theor. Comput. Sci. 235, 3–18 (April 2009)
2. Burget, R.,: Automatic document structure detection for data integration. In: Business Information Systems. pp. 391–397. LNCS 4439, Springer Verlag (2007)
3. Burget, R., Burgetová, I.: Automatic annotation of online articles based on visual feature classification. International Journal of Intelligent Information and Database System 5(4), 338–360 (2011)
4. Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: Extracting content structure for web pages based on visual representation. In: Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications. pp. 406–417. APWeb'03, Springer-Verlag, Berlin, Heidelberg (2003)
5. Groza, T., Handschuh, S.: Salt document ontology. Dostupné na: http://salt.semanticauthoring.org/ontologies/sdo `[online]` (Srpen 2012)

[2] http://en.wikipedia.org/wiki/Semantic_reasoner
[3] http://cssbox.sourceforge.net/

Towards Conceptual Structure Verification of Linked Data Vocabularies

Vojtěch Svátek[1], Miroslav Vacura[2], Martin Homola[3], Ján Kľuka[3]

[1] Dept. of Information and Knowledge Engineering, University of Economics, Prague,
W. Churchill Sq.4, 130 67 Prague 3, Czech Republic
svatek@vse.cz
[2] Department of Philosophy, University of Economics, Prague,
W. Churchill Sq.4, 130 67 Prague 3, Czech Republic
vacuram@vse.cz
[3] Department of Applied Informatics, Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia
{homola,kluka}@fmph.uniba.sk

1 Introduction

Linked Data (LD) principles are increasingly exploited when publishing structured data in the WWW. Powerful mashups can for example be easily built over public SPARQL endpoints providing encyclopaedic resources such as DBpedia, government data resources (such as public spendings), or geographical resources.

One of the advantages of LD w.r.t. proprietary Web-enabled databases is their adherence to publicly available and widely shared vocabularies; notorious examples are FOAF (for personal profiles), Dublin Core (for bibliographic metadata), Music Ontology (MO; for recordings and other musical information), or GoodRelations (GR; for e-commerce data). One of the important prerequisites to smooth adoption of individual vocabularies, and, in particular, to matching multiple vocabularies (needed when integrating datasets that subscribe to different vocabularies) seems to be their conceptual consistency.[1]

Our proposal, outlined further in the paper, addresses consistency in terms of a so-called deep (conceptual) model that is 'hidden behind' the surface representation of a vocabulary. The crucial assumption is that the actual RDF/OWL representation (i.e., the *surface model*) often only roughly approximates the structure of the implicit conceptual model (i.e., the *deep model*) that faithfully captures the real-world state of affairs. To illustrate this problem, let us consider a 'simple fact' in RDF stating that a company's business is repairing, relying on the GR vocabulary (indicated by the 'gr' prefix):

ex:MyCompany gr:hasBusinessFunction gr:Repair .

While the subject and the object of this fact are, syntactically, both individuals in OWL DL terms, their deep conceptual types are strikingly different: while ex:MyCompany

[1] Not to be confused with logical consistency; unlike ontologies designed primarily for reasoning tasks, typical vocabularies (even if formally adhering to the OWL standard) contain too few complex axioms to become logically inconsistent.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 89–92.

is truly an individual real-world object (ontologically said, a 'particular'), gr:Repair represents a quality (ontologically said, a 'universal') that can be assigned to many individuals. It is not hard to realize that the same conceptual relationship can be expressed (assuming an alternative GoodRelations vocabulary prefix, say, 'gr-alt') as

ex:MyCompany rdf:type gr-alt:RepairCompany .

The latter representation is much more faithful to the deep model, clearly indicating that an individual is assigned to a class (in other terms, declared to have some universal quality). While, in reality, we can rarely transform the structure of a widely used vocabulary (with many datasets referring to it) in the indicated way, we could at least label the entities of an existing vocabulary with the 'conceptual distinctions' of the deep model. For example, labelling the range of the gr:hasBusinessFunction property as a 'conceptual class' may help to avoid assigning to it (in some dataset) real-world individuals as instances. While in the example above the distinction seems obvious, it is not always straightforward to distinguish between a true individual and a conceptual class based on the name of the entity only (cf. examples in Sect. 3).

There are other works [5, 3, 2] aiming to cope with discrepancies and limitations of the surface LD models, most prominently OntoClean [4], which differs from our approach in its focus on annotating classes and repairing their taxonomic relationships.

In the rest of the paper, we first sketch the structure of the deep model. As a practical use case, we look at a fragment of the Music Ontology vocabulary, and show how the deep model can warn us of possible conceptual discrepancies when the vocabulary is merged with or mapped to another one. We also discuss our plans for practical use of the deep model approach for detecting problems in vocabularies and their mappings.

2 Deep Model for Linked Data Vocabularies

To avoid confusion with the surface models, we will prefix the primitives from the deep model with $\mathcal{D}$-, and partly even use distinct terms. The first two building units of deep models will be $\mathcal{D}$-objects and $\mathcal{D}$-classes.

$\mathcal{D}$-object refers to a real-world object, which can be tangible (such as people, animals, things, etc.) or intangible (various abstract entities such as topics, processes, etc.). It is analogous to the notion of *individual* in the surface model, and $\mathcal{D}$-objects also correspond to surface individuals in the majority of cases (e.g., the surface individual ex:MyCompany from the introduction maps to a $\mathcal{D}$-object in the deep model).

$\mathcal{D}$-class refers to a real-world class. Therefore it is usually a set of $\mathcal{D}$-individuals instantiating the same concept or sharing a common property. For simplicity, it also covers the notion of *quality* (e.g., red color), which is ontologically slightly different but plays the same role in the model structure.

The single key distinction between $\mathcal{D}$-objects and $\mathcal{D}$-classes is, respectively, that between particulars (which never have instances) and universals (which possibly may have instances). In the surface model, however, $\mathcal{D}$-classes may be reflected either as true RDFS/OWL classes, but sometimes also as surface individuals (e.g., gr:Repair as seen in the introduction).

Let us now have a look at relationships between entities. $\mathcal{D}$**-relationship** refers to a (particular) relationship between two entities (for instance, an individual is produced by some producer, or a person owns some individual). Its universal counterpart is $\mathcal{D}$**-relation**, which refers to a real world conceptual relation, a class of relationships.

Finally we enrich the model with data values and associated constructs. The particular called $\mathcal{D}$**-valuation** refers to an assignment of a data value to some entity (e.g., some person has height of 199 cm). Its universal counterpart is $\mathcal{D}$**-attribute**, referring to real world valuations of the same (usually quantitative) property.

Although in practice these relationships possibly involve more than two individuals, we focus on binary relationships, to keep the deep model aligned with RDF/OWL. Thus, $\mathcal{D}$-relationships ($\mathcal{D}$-valuations) are analogous to the surface notion of *object (data) property assertions*, and $\mathcal{D}$-relations ($\mathcal{D}$-attributes) are analogous to *object (data) properties*.

Note that the entities which take part in $\mathcal{D}$-relationships ($\mathcal{D}$-valuations) are not only $\mathcal{D}$-objects but also $\mathcal{D}$-classes as we show in practical examples in the next section.

3 Use Case: Deep Model and the Music Ontology Vocabulary

To demonstrate a practical use of the deep model, we now have a look on the Music Ontology vocabulary with the purpose of 'deep disambiguation' of selected constructs. Using this vocabulary we are able to express that the CBS 1992 CD release of Yo-Yo Ma's performance of J. S. Bach's 'Six Cello Suites' is an album:

```
ex:CBS1992Cd_YoYoMa_JSBach_SixCelloSuites
    mo:release_type   mo:album .
mo:album   rdf:type   mo:ReleaseType .
```

At the surface level, ex:CBS1992Cd_YoYoMa_JSBach_SixCelloSuites and mo:album are individuals and mo:ReleaseType is a class. Considering the deep model, however, we see that while the first surface individual is a $\mathcal{D}$-object, mo:album is in fact a $\mathcal{D}$-class (as is mo:ReleaseType). This is because mo:album can have instances, as documented by the example—ex:CBS1992Cd_YoYoMa_JSBach_SixCelloSuites is its instance. What we learn from the deep model is that the surface individual mo:album is of a specific kind (it is a $\mathcal{D}$-class) and it needs to be given special attention: e.g., it is of little sense to assign some data attributes to it as it does not stand for any real object.

There are other insights that can be learned here: mo:ReleaseType is in fact a specific type of $\mathcal{D}$-class—its instances are again classes (and should not be $\mathcal{D}$-objects). Moreover, the regular surface property mo:release_type represents a specific kind of $\mathcal{D}$-relation, namely *instantiation*, more typically represented by rdf:type. Note that a relationship between a $\mathcal{D}$-object and a $\mathcal{D}$-class is not always an instantiation; e.g.,

```
ex:YoYoMa   mo:primary_instrument   mo-mit:Cello .
```

clearly represents a regular $\mathcal{D}$-relationship, although the surface individual mo-mit:Cello represents a $\mathcal{D}$-class (many physical musical instruments are celli). These cases are not yet clearly recognized by our model, and are subject to our ongoing investigation.

4 Next Steps: Evaluation, Tool Support and Practical Application

Ongoing work concerns systematic mapping of LD vocabularies to the deep model structures, in order to evaluate the plausibility of the approach. We have already built a nearly-complete mapping for MO, GR, and FOAF, in the form of annotation of individual surface entities (in particular, classes, individuals, and property ranges) with 'meta-properties' corresponding to deep model primitives.

While this initial work was carried out without a dedicated tool support, we are, in parallel, developing an *annotator tool*—a Protégé plugin that would allow to create annotations (referring to a 'deep conceptual annotation' ontology) for each vocabulary, either generically or with respect to the use of the given vocabulary entity in a particular dataset, and to store them in a dedicated annotation space. On this basis we can extend the analysis to further vocabularies, on which we could test the degree of inter-annotator agreement (assuming shared annotator guidelines that are also under design).

Among the intended practical applications of this effort on the Semantic Web, we foresee assistance in *visual inspection* of vocabularies and datasets (summaries) via in-viewer transformation among different surface representations of the same deep structure (using tools such as PatOMat [6]). The annotations of entities with deep conceptual distinctions may also be used by reasoners, to detect some of the modeling discrepancies as sketched in Sect. 3. Finally, deep annotations would contribute to consistent *vocabulary mapping* [1].

Acknowledgements. We are grateful to Aldo Gangemi and Alan Rector for feedback to our initial ideas. This work was partially supported from the CSF grant no. P202/10/1825 (PatOMat—Automation of Ontology Pattern Detection and Exploitation), from the Slovak VEGA project no. 1/1333/12, and from the Czecho-Slovak bilateral mobility project no. SK-CZ-0208-11 (APVV agency) and no. 7AMB12SK020 by (AIP ČR agency).

References

1. Bizer, C., Schultz, A.: *The R2R Framework: Publishing and Discovering Mappings on the Web.* In: Proc. COLD'10. 1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai, November 2010.
2. Gangemi, A.: *SuperDuper Schema: an OWL2+RIF DnS pattern.* In: Deep Knowledge Representation Challenge Workshop at K-CAP'11.
3. Glimm, B., Rudolph, S., Völker, J.: *Integrated metamodeling and diagnosis in OWL 2.* In: Proc. ISWC'10.
4. Guarino, N., Welty, C.: An Overview of OntoClean. In: Staab, S., Studer, R., eds.: *The Handbook on Ontologies*, Springer-Verlag, pp. 151–172.
5. Sunagawa E., Kozaki K., Kitamura Y., Mizoguchi R.: *Role organization model in Hozo.* In: Proc. EKAW'06.
6. Šváb-Zamazal, O., Svátek, V., Iannone, L.: *Pattern-Based Ontology Transformation Service Exploiting OPPL and OWL-API.* In: Proc. EKAW'10.

Aproximácie drsných množín

Ľubomír Antoni, Stanislav Krajči

Ústav informatiky, Prírodovedecká fakulta,
Univerzita P. J. Šafárika v Košiciach, Jesenná 5, 040 01 Košice
lubomir.antoni@student.upjs.sk, stanislav.krajci@upjs.sk

Abstrakt. Analýza drsných množín pracuje s objektovo-atribútovým modelom, v ktorom modelujeme nerozlíšiteľnosť objektov využitím relácie ekvivalencie. Pomocou tejto relácie definujeme hornú a dolnú aproximáciu, ktoré využívame pri rozhodovacích problémoch. Ďalej skúmame, či tieto zobrazenia dolnej a hornej aproximácie spĺňajú vlastnosti operátora uzáveru. Tieto poznatky ilustrujeme vhodnými príkladmi.

Kľúčové slová: relácia nerozlíšiteľnosti, horná a dolná aproximácia

1 Úvod

Pojem drsných množín prvýkrát uviedol Pawlak [3], [4] v roku 1982. V [2] sú uvedené aplikácie drsných množín v medicíne (analýza obrazov pre medicínske aplikácie), ekonomike, financiách a obchode (ohodnotenie rizika, správanie spotrebiteľov), životnom prostredí (ochrana, globálne otepľovanie), inžinierstve (analýza signálov a obrazu), softvérovom inžinierstve, rozhodovacích procesoch, sociálnych vedách, molekulárnej biológii, chémii.

V nasledujúcej kapitole zavádzame pojem drsných množín a ilustrujeme horné a dolné aproximácie na príklade objektovo-atribútovej tabuľky áut, resp. pacientov. V ďalšej časti prezentujeme vlastnosti dolných a horných aproximácií.

2 Drsné množiny

Uvažujme objektovo-atribútový model (B, A, J), kde B je neprázdna množina objektov, A je neprázdna množina atribútov a J je zobrazenie, pre ktoré platí $\mathrm{Dom}(J) = B \times A$. Na tomto modeli definujeme nerozlíšiteľnosť objektov ako reláciu:

Definícia 1 *Nech $Y \subseteq A$. Potom Y-reláciou nerozlíšiteľnosti objektov nazývame reláciu*

$$R_Y = \{\langle b_1, b_2\rangle \in B \times B : (\forall a \in Y) J(b_1, a) = J(b_2, a)\}. \tag{1}$$

Táto relácia je reflexívna, symetrická, tranzitívna, teda je reláciou ekvivalencie. To nás oprávňuje celú množinu objektov rozdeliť na triedy ekvivalencie tvaru

$$[b]_{R_Y} = \{c \in B : \langle b, c\rangle \in R_Y\}. \tag{2}$$

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 93–96.

Množinu všetkých tried ekvivalencií na množine B s reláciou nerozlíšiteľnosti R_Y budeme označovať B/R_Y.

Príklad 1 *Zoberme si množinu ôsmych typov áut. Predpokladajme, že majú rôznu farbu, rôzny tvar svetiel a rôznu veľkosť. Pre každý typ auta sú hodnoty jeho atribútov v nasledujúcej objektovo-atribútovej tabuľke.*

	farba	tvar	veľkosť
1	Č	△	V
2	M	□	M
3	Č	◯	V
4	M	◯	M
5	Ž	△	M
6	Ž	□	M
7	Č	◯	V
8	Ž	◯	V

Množinu všetkých objektov tvorí $B = \{1, 2, 3, 4, 5, 6, 7, 8\}$, *množinu všetkých atribútov tvorí* $A = \{f, t, v\}$. *Uvažujme reláciu nerozlíšiteľnosti objektov pre atribút farba:*

$$\begin{aligned} R_{\{f\}} = &\{\langle 1,1\rangle, \langle 1,3\rangle, \langle 1,7\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 3,7\rangle, \langle 7,1\rangle, \langle 7,3\rangle, \langle 7,7\rangle\} \cup \\ &\cup \{\langle 2,2\rangle, \langle 2,4\rangle, \langle 4,2\rangle, \langle 4,4\rangle\} \cup \\ &\cup \{\langle 5,5\rangle, \langle 5,6\rangle, \langle 5,8\rangle, \langle 6,5\rangle, \langle 6,6\rangle, \langle 6,8\rangle, \langle 8,5\rangle, \langle 8,6\rangle, \langle 8,8\rangle\}. \end{aligned}$$

Trieda ekvivalencie, ktorá obsahuje napríklad objekt 3 *pre* $R_{\{f\}}$ *je:*

$$[3]_{R_{\{f\}}} = \{1, 3, 7\}.$$

Množina všetkých ekvivalencií pre $R_{\{f\}}$ *má tvar*

$$B/R_{\{f\}} = \{\{1, 3, 7\}, \{2, 4\}, \{5, 6, 8\}\}.$$

Použitím relácie nerozlíšiteľnosti môžeme rozdeliť celú množinu objektov na disjunktné triedy ekvivalencie (základné stavebné blok). Zjavne však vieme nájsť aj takú podmnožinu objektov (napr. $\{1, 2, 3\}$), ktorá nie je triedou ekvivalencie, v takom prípade však vieme takúto podmnožinu aproximovať pomocou základných stavebných blokov zhora a zdola. Preto v ďalšom definujeme pojmy dolnej a hornej aproximácie množiny.

Definícia 2 *Nech* $X \subseteq B$ *a* R_Y *je relácia nerozlíšiteľnosti objektov. Dolnou aproximáciou množiny* X *nazývame zobrazenie* $\underline{R_Y} : 2^B \to 2^B$

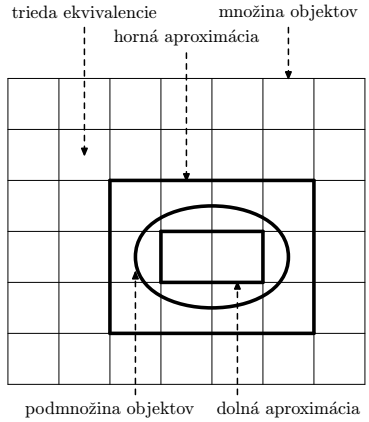
$$\underline{R_Y}(X) = \{x \in B : [x]_{R_Y} \subseteq X\}.$$

Definícia 3 *Nech* $X \subseteq B$ *a* R_Y *je relácia nerozlíšiteľnosti objektov. Hornou aproximáciou množiny* X *nazývame zobrazenie* $\overline{R_Y} : 2^B \to 2^B$

$$\overline{R_Y}(X) = \{x \in B : [x]_{R_Y} \cap X \neq \emptyset\}.$$

Na nasledujúcom obrázku ilustrujeme jednoduchú predstavu dolnej aproximácie a hornej aproximácie, ktoré dostaneme zo základných definovateľných podmožín. Elipsa znázorňuje množinu X, ktorú aproximujeme zhora a zdola. Dvojicu $\left(\underline{R_Y}(X), \overline{R_Y}(X)\right)$ nazveme drsnou množinou pre $X \subseteq B$ a $Y \subseteq A$.

Obr. 1. Grafické znázornenie hornej a dolnej aproximácie.



Príklad 2 *Pohybové skóre dolných končatín sa meria na 50 stupňovej škále. Hodnota 50 znamená optimum, hodnoty od 30 do 50 predikujú tendenciu schopnosti pohybu u človeka. Dôležitú úlohu tu však zohráva aj vek. Aproximujme skupiny pacientov, ktorí boli schopní aktívneho pohybu z hľadiska ich veku a pohybového skóre podľa údajov uvedených v nasledujúcej tabuľke.*

	vek	skóre	pohyb
1	16-30	50	ÁNO
2	16-30	0	NIE
3	31-45	1-25	NIE
4	31-45	1-25	ÁNO
5	46-60	26-49	NIE
6	16-30	26-49	ÁNO
7	46-60	26-49	NIE

Z tabuľky vyberiem len tie objekty, ktoré majú v poslednom stĺpci hodnotu ÁNO. Teda $X = \{1, 4, 6\}$. *Množina všetkých ekvivalencií pre reláciu nerozlíšiteľnosti, ktorá zohľadňuje vek a skóre má tvar*

$$B/R_{\{v,s\}} = \{\{1\}, \{2\}, \{3, 4\}, \{5, 7\}, \{6\}\}.$$

Dolnú aproximáciu nájdem podľa definície tak, že vyberiem všetky triedy ekvivalencie, ktoré sú celé podmnožinou X*:*

$$\underline{R_{\{v,s\}}}(X) = \{1, 6\}.$$

Hornú aproximáciu nájdem podľa definície tak, že vyberiem všetky triedy ekvivalencie, ktorých prienik s množinou X *je neprázdny:*

$$\overline{R_{\{v,s\}}}(X) = \{1, 3, 4, 6\}.$$

3 Vlastnosti hornej a dolnej aproximácie

Postupne uvádzame prirodzenú vlastnosť umiestnenia množiny medzi jej dolnou a hornou aproximáciou, distributívnosť hornej aproximácie nad zjednotením a dolnej aproximácie nad prienikom, vlastnosti týkajúce sa monotónnosti a kompozície dolnej a hornej aproximácie. Dôkazy týchto vlastností uvádzame v [1].

Lema 1 *Nech $X, X_1, X_2 \subseteq B$ a $Y \subseteq A$. Potom platí:*

a) $\underline{R_Y}(X) \subseteq X \subseteq \overline{R_Y}(X)$,
b) $\overline{R_Y}(X_1 \cup X_2) = \overline{R_Y}(X_1) \cup \overline{R_Y}(X_2)$,
c) $\underline{R_Y}(X_1 \cap X_2) = \underline{R_Y}(X_1) \cap \underline{R_Y}(X_2)$.
d) *Ak* $X_1 \subseteq X_2$, *tak* $\overline{R_Y}(X_1) \subseteq \overline{R_Y}(X_2)$.
e) *Ak* $X_1 \subseteq X_2$, *tak* $\underline{R_Y}(X_1) \subseteq \underline{R_Y}(X_2)$.
f) $\underline{R_Y}(\overline{R_Y}(X)) = \overline{R_Y}(X)$.
g) $\overline{R_Y}(\underline{R_Y}(X)) = \underline{R_Y}(X)$.

Ďalej definujme zobrazenie $\mathrm{cl}_{R_Y} : \mathrm{P}(B) \to \mathrm{P}(B)$ ako zloženie zobrazení $\overline{R_Y}$ a $\underline{R_Y}$. Ukážeme, že takto získané zobrazenie tvorí operátor uzáveru.

Lema 2 *Nech $X, X_1, X_2 \subseteq B$, $Y \subseteq A$ a nech $\mathrm{cl}_{R_Y}(X) = \underline{R_Y}(\overline{R_Y}(X))$. Potom:*

a) $X \subseteq \mathrm{cl}_{R_Y}(X)$,
b) *ak* $X_1 \subseteq X_2$, *tak* $\mathrm{cl}_{R_Y}(X_1) \subseteq \mathrm{cl}_{R_Y}(X_2)$,
c) $\mathrm{cl}_{R_Y}(X) = \mathrm{cl}_{R_Y}(\mathrm{cl}_{R_Y}(X))$.

4 Záver

V tomto článku sme uviedli príklad využitia drsných množín pri rozhodovacích procesoch. Príklad s autami slúži na rozdelenie množiny objektov na disjuktné množiny analogicky ako pri zhlukovacích algoritmoch, v príklade s pacientmi navyše modelujeme hornú a dolnú aproximáciu. Niektoré vzťahy teórie drsných množín s teóriou topologických priestorov, prípadne s logikou naznačuje Vlach v [5], [6] a [7]. Príkladom softvérového systému založeného na aplikáciách drsných množín je nástroj Infobright.

Literatúra

1. Ľ. Antoni, S. Krajči. Drsné množiny a formálny kontext. In: Informačné Technológie - Aplikácie a Teória: 17-21. september 2012, Monkova Dolina (SK), 9–16 (2012)
2. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: a tutorial. In: Pal, S. K., Skowron, A. (Eds.), Rough-Neural Computing: Techniques for Computing with Words, Cognitive Technologies, Springer-Verlag, Heidelberg 3–98 (2004)
3. Pawlak, Z.: Rough sets Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, (1991)
4. Pawlak, Z.: Rough sets. Int. J. of Computer and Infom. Sciences 341–356 (1982)
5. Vlach, M.: Topologies of approximation spaces of rough set theory. Advances in Soft Computing Vol. 46, Springer-Verlag, Berlin Heidelberg, Germany 176–186 (2008)
6. Vlach, M.: Mathematical structures of rough sets. Proceedings of the 27th international conference Mathematical Methods in Economics, Kostelec, Czech Republic, September 335–340 (2009)
7. Vlach, M.: Links between extended topologies and approximations of rough set theory. Multicriteria Decision Making and Fuzzy Systems, Shaker Verlag, Aachen, Germany, 13–22 (2006)

Reduction of Computation Times of GOSCL Algorithm Using Sparse-based Implementation

Peter Butka *

Department of Cybernetics and Artificial Intelligence,
Technical University of Košice, Letná 9, 04001 Košice, Slovakia
peter.butka@tuke.sk

Abstract. In this paper we provide information about the comparison of time complexity of standard and sparse-based implementation of GOSCL algorithm. It is an incremental algorithm for the creation of Generalized One-Sided Concept Lattices, which is related to the algorithms from well-known Formal Concept Analysis area, but with the possibility to work with different types of attributes.

Keywords: one-sided concept lattices, time complexity, sparse data

1 Introduction

Formal Concept Analysis (FCA, [4]) is conceptual method for data analysis successfully applied to the domains such as information retrieval, data/text mining and knowledge discovery. While classic FCA approach is based on the binary relations (object has/has-not the attribute), there are natural examples of object-attribute models for which relationship between objects and attributes are represented by fuzzy relations. A special case of fuzzy FCA is so-called one-sided concept lattice (see [5] for reference), where usually objects are considered as a crisp subsets and attributes obtain fuzzy values. All recently known approaches allow only one type of structure for truth degrees. However, it is reasonable to consider object-attribute models with different truth value structures for their attributes. In [1] we have introduced generalization of one-sided concept lattices (GOSCL) based on the Galois connections (more details on generalization of fuzzy concept lattices can be found in [6],[7]) in order to provide the algorithm, which is able to work different types of attributes.

While, in general case, the complexity of FCA-based algorithms is exponential, we have shown in [2] that for fixed attributes structure, the computational complexity of algorithm becomes linear according to the number of input objects. In case of sparse input data tables, where object-attribute models have many of their values equal to zeros, the reduction of computation times can rapidly

* This work was supported by the Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030)(50%) and by the Slovak Research and Development Agency under contract APVV-0208-10 (50%).

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 97-100.

increase with the higher sparseness, as we have shown in [3]. In this paper we will show that for specialized sparse-based implementation computation times are even better.

The paper is organized as follows. In the next section we will provide necessary details for introducing the GOSCL algorithm. Then we will provide information regarding time complexity for sparse inputs with standard and sparse-based implementations of GOSCL algorithm.

2 Algorithm GOSCL

A 4-tuple $\big(B, A, \mathcal{L}, R\big)$ is said to be a *generalized one-sided formal context* if the following conditions are fulfilled:

(1) B is a non-empty set of n objects and A is a non-empty set of m attributes.
(2) $\mathcal{L} : A \to \mathsf{CL}$ is a mapping from the set of attributes to the class of all complete lattices. Hence, for any attribute a, $\mathcal{L}(a)$ denotes the complete lattice, which represents structure of truth values for attribute a.
(3) R is generalized incidence relation, i.e., $R(b, a) \in \mathcal{L}(a)$ for all $b \in B$ and $a \in A$. Thus, $R(b, a)$ represents a degree from the structure $\mathcal{L}(a)$ in which the element $b \in B$ has the attribute a.

Let $\big(B, A, \mathcal{L}, R\big)$ be a generalized one-sided formal context. Then a pair of mappings $\uparrow$: $2^B \to \prod_{a\in A} \mathcal{L}(a)$ and $\downarrow$: $\prod_{a\in A} \mathcal{L}(a) \to 2^B$ defined as $\uparrow \big(X\big)(a) = \bigwedge_{b\in X} R(b, a)$ and $\downarrow (g) = \{b \in B : \text{ for each } a \in A,\ g(a) \leq R(b, a)\}$ forms a Galois connection between 2^B and $\prod_{a\in A} \mathcal{L}(a)$. Moreover, the set $\mathcal{C}\big(B, A, \mathcal{L}, R\big)$ with defined partial order forms a complete lattice. This lattice is called *generalized one-sided concept lattice*. It was proved in [1] that any Galois connection between 2^B and $\prod_{a\in A} \mathcal{L}(a)$ can be obtained by suitable generalized one-sided formal context, hence GOSCL have the same universality property as classical FCA. The main idea of the presented algorithm is to create the set of all intents corresponding to the Galois connection $(\uparrow, \downarrow)$. For $b \in B$ put $R(b)$ an element of $\prod_{a\in A} \mathcal{L}(a)$ such that $R(b)(a) = R(b, a)$ (represents b-th row in data table). Let 1_L denote the greatest element of $L = \prod_{a\in A} \mathcal{L}(a)$, i.e., $1_L(a) = 1_{\mathcal{L}(a)}$ for all $a \in A$.

As we have shown in [3], standard implementation can be used also for sparse data inputs with the dramatically lower computation times. It is based on reduction of comparisons of intents with the new rows during the incremental steps. In order to achieve even lower computation times for sparse inputs we have decided to implement sparse-based version of GOSCL. This has been achieved by the usage of sparse-based: 1.) representation (implementation and usage of sparse-based representation of input data rows and vectors related to concepts); 2.) implementation of $\downarrow$ operation(sparse-based implementation of function, which goes back to objects); 3.) implementation of meet operation (sparse-based implementation of lattice-based meet operation $\wedge$).

For the specific representation we have used sparse-based representation of object-attribute models implemented in Java. The implementation of $\downarrow$, used

Algorithm GOSCL

Require: generalized context $\langle B, A, \mathcal{L}, R \rangle$
Ensure: set of all concepts $\mathcal{C}\langle B, A, \mathcal{L}, R \rangle$
1: $L \leftarrow \prod_{a \in A} \mathcal{L}(a)$ ▷ Direct product of attribute lattices
2: $I \leftarrow \{1_L\}$ ▷ $I \subseteq L$ will denote set of intents
3: $\mathcal{C}\langle B, A, \mathcal{L}, R \rangle \leftarrow \emptyset$
4: **for all** $b \in B$ **do**
5: $I^* \leftarrow I$ ▷ I^* represents "old" set of intents
6: **for all** $g \in I^*$ **do**
7: $I \leftarrow I \cup \{R(b) \wedge g\}$ ▷ Generation of new intent
8: **end for**
9: **end for**
10: **for all** $g \in I$ **do**
11: $\mathcal{C}\langle B, A, \mathcal{L}, R \rangle \leftarrow \mathcal{C}\langle B, A, \mathcal{L}, R \rangle \cup \{(\downarrow (g), g)\}$
12: **end for**
13: **return** $\mathcal{C}\langle B, A, \mathcal{L}, R \rangle$ ▷ Output of the algorithm

within the cycle in step 11, is based on the iterations through reduced indexes for two sparse vectors. The implementation of meet operation $\wedge$ is also used often within GOSCL algorithm. It is used in step for generation of new intents during incremental run (within cycle for every new object b, step 7). The most important is that meet of any elements in lattice with zero is always zero, i.e., only two non-zero elements with same indexes should be processed.

3 Experiments

The generation of sparse input data table for our experiments is based on the sparse factor $s \in [0, 1)$, which indicates the ratio of zeros (as a real number between 0 and 1) in data table. In order to analyze the usage of GOSCL algorithm for text-mining problems, where sparseness of inputs is even more evident, we have produced the randomly generated input formal contexts, where sparseness is based on the real dataset for text-mining (Reuters ModApte dataset part with 7769 documents). For our purposes we have used non-restrictive preprocessing, which leads to representation with 22500 attributes. The average sparseness s of the rows in this data table is more than $0,998$.

During our experiments we have analyzed more aspects of time complexity of standard and sparse-based implementation. We have found that sparse-based implementation is able to work more easily with many attributes due to its specialized operations on sparse vectors, while standard implementation grows much faster according to the number of attributes. Hence, it is not hard to assume that computation times for fixed number of attributes and sparseness will differ dramatically for sparse-based and standard implementation in its growing factor. For example, this fact is shown on Figure 1 (left part), where we can see data for different implementations and number of objects. The benefits of sparse-based implementation are evident. While processing of 50 objects needs more than

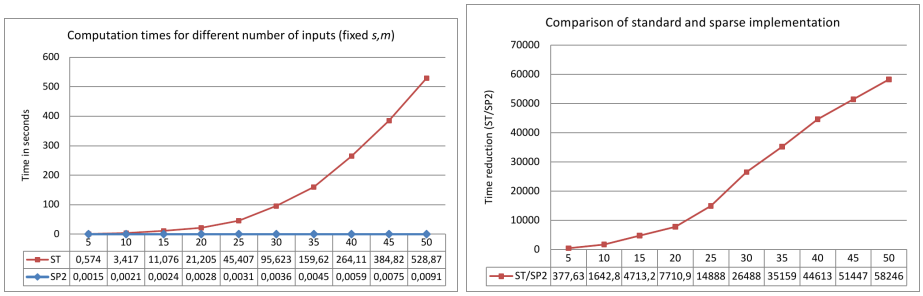


Fig. 1. Comparison of standard (ST) and sparse-based implementation (SP2).

500 seconds for standard implementation, sparse-based version is able to process same number of documents in less than 0.01 second. Figure 1 (right part) also shows the ratio of reduction between standard and sparse-based implementation. As we can see, the ratio is even better for growing number of objects.

When we have used standard implementation for very sparse inputs, more than 500 seconds were needed in order to process 50 objects (with 22500 attributes). Sparse-based implementation proved its strong reduction effect, so we are able to move forward in the processing of more objects during similar times. In our last experiments we have tried to process more inputs in order to find how many objects can be processed in similar times to standard implementation. Sparse-based implementation is able to process 1000 objects in 533 seconds. As we can see, now it is possible to work with hundreds or thousands of objects instead of tenths. Next, we want to extend the current implementation in parallel/distributed way, which can help in analysis of even larger collections, with the vision to process hundred thousands of documents in the future.

References

1. Butka, P., Pócs, J.: Generalization of one-sided concept lattices. Computing and Informatics (to appear)
2. Butka, P., Pócsová, J., Pócs, J.: On Some Complexity Aspects of Generalized One-Sided Concept Lattices Algorithm. In: Proceedings of SAMI 2012, Herľany, Slovakia, 231–236 (2012)
3. Butka, P., Pócsová, J., Pócs, J.: Experimental Study on Time Complexity of GOSCL Algorithm for Sparse Data Tables. In: Proceedings of SACI 2012, Timisoara, Romania, 101–106 (2012)
4. Ganter, B., Wille, R.: Formal concept analysis: Mathematical foundations, Springer, Berlin (1999)
5. Krajči, S.: Cluster based efficient generation of fuzzy concepts. Neural Netw. World 13(5), 521–530 (2003)
6. Pócs, J.: Note on generating fuzzy concept lattices via Galois connections. Inform. Sci. 185(1), 128–136 (2012)
7. Pócs, J.: On possible generalization of fuzzy concept lattices using dually isomorphic retracts. Inform. Sci. 210, 89–98 (2012)

Modelovanie šírenia emócií v dave

Peter Lacko

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave,
Ilkovičova 3, 842 16 Bratislava
lacko@fiit.stuba.sk

Abstrakt. Pri modelovaní správania sa davu je šírenie emócii dôležité, keďže práve ono zabezpečuje prenášanie informácií správania medzi jednotlivcami, ktorí sa potom správajú ako skupina. Je zrejmé, že jednotlivec sa v dave správa inak. Osamotený demonštrant má podstatne menšiu motiváciu správať sa agresívne, ako demonštranti, ktorí sú súčasťou davu. Tendencie správania sa celého davu potom určuje práve šírenie emócií v dave. Môže to byť napríklad hnev, pri ktorom sa môže pokojná demonštrácia zmeniť na agresívnu vzburu. V tomto príspevku analyzujeme možné prístupy k šíreniu emócií v dave založené na modelovaní interakcií prebraných priamo zo socialno-psychologických teórií, ako aj model inšpirovaný epidemiologickým šírením infekcie.

Kľúčové slová: dav, šírenie emócií, multiagentový systém, model správania sa, charakteristiky davu

1 Úvod

Pri modelovaní správania sa davu je práve šírenie emócii veľmi dôležité, keďže práve ono zabezpečuje prenášanie informácií správania medzi jednotlivcami, ktorí sa potom správajú ako skupina. Je zrejmé, že jednotlivec sa v dave správa inak, ako keby bol osamotený. Osamotený demonštrant má podstatne menšiu motiváciu správať sa agresívne, ako demonštranti, ktorí sú súčasťou davu a teda cítia sa byť anonymizovaný. Tendencie správania sa celého davu potom určuje práve šírenie emócií v dave. Môže to byť napríklad panika, kedy pri požiari začnú ľudia utekať, alebo aj hnev, pri ktorej sa môže pokojná demonštrácia zmeniť na agresívnu vzburu.

Šírenie emócií teda predstavuje komunikačný kanál davu, ktorým sa emócie, ktoré priamo ovplyvňujú správanie, prešíria k jednotlivcom v dave.

2 Modelovanie šírenia emócií v dave

V tejto sekcií porovnáme dva prístupy šírenia emócii v dave. Prvý (VU model [2, 7]) je založený na modelovaní interakcií prebraných priamo z socialno-psychologických teórií. Druhý model - Drupinal [5, 7] je inšpirovaný epidemiologickým šírením, kde

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 101–104.

interakcia s emočne infikovanými jedincami zvyšuje pravdepodobnosť emočnej nákazy jedinca.

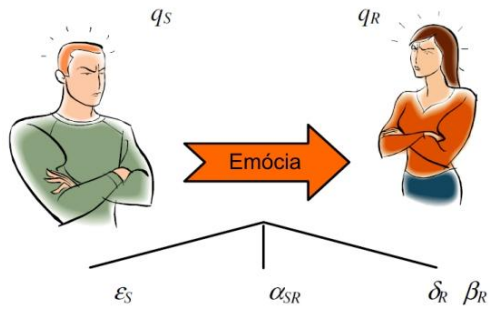
2.1 VU model

Model predstavený v roku 2009 Bosse et al. [2, 3, 7] zabezpečuje to, že emócia skupiny sa mení smerom k váženému priemeru emócií jednotlivcov v skupine.

Emocionálne šírenie je modelované pomocou nasledovných parametrov (obr. 1) [3]:

- q_S - aktuálny stav emócií odosielateľa S
- q_R - aktuálny stav emócií prijímateľa R
- ε_S - sila vyjadrovania/expresivita vysielateľa
- α_{SR} - sila kanálu medzi S a R
- δ_R - otvorenosť/senzitivita prijímateľa
- β_R - tendencia prijať emóciu prijímateľom

Všetky hodnoty sú čísla v intervale <0, 1>. Tieto parametre boli odvodené z teórie navrhnutej v [1], čo je teoretický základ modelu.



Obr. 1. Aspekty šírenia emócií (prebrané z [3]).

V každom časovom kroku si každý agent vypočíta priemerný prenos emócií od každého relevantného agenta. Konkrétne od agenta *S* vysielajúceho emóciu prijímaciemu agentovi *R*. Sila prijatej emócie bude:

$$\gamma_{SR} = \varepsilon_S \alpha_{SR} \delta_R$$

potom sila prijatej emócie v skupine je:

$$\gamma_R = \sum_{S \in G \setminus \{R\}} \gamma_{SR}$$

Je zrejmé, že silnejší kanál, silnejšia expresivita odosielateľa, väčšia otvorenosť prijímateľa vedú k silnejšiemu prenosu emócie. Emočná hladina agenta konverguje k váženému priemeru hodnoty emócie v skupine.

Výhodou tohto modelu je, že aj napriek tomu, že priamo nedefinuje modelovanie sily šírenia emócie na rôzne vzdialenosť, úpravou parametra sily kanálu sa tento efekt dá ľahko docieliť. Nevýhodou modelu je, že napríklad strach sa nikdy nevytratí, emócia v skupine konverguje k váhovanému priemeru. Riešenie vidíme v kombinácií VU modulu šírenia emócií s modelom správania PECS [6].

2.2 Durupinar Model

Model, ktorý navrhol Durupinar [5] je inšpirovaný šírením epidémií [4] - existuje podobnosť medzi šírením nákazy a šírením emócií. Durupinar implementuje dva stavy - emočne infikovaný a neinfikovaný.

Každý agent j pri inicializácií získa náhodnú prahovú hodnotu T_j z vopred stanoveného log-normálneho rozdelenia. V každom časovom kroku, pre každého agenta je náhodne vybraný iný agent z príslušnej skupiny. Ak je tento agent infikovaný, vygeneruje náhodnú dávku d_i z vopred stanoveného log-normálneho rozdelenia, a odovzdá ju pôvodnému agentovi. Ak agent nie je infikovaný, potom je dávka 0. Každý agent si pamätá históriu posledných k dávok ktoré dostal:

$$D_j(t) = \sum_{t'=t-k+1}^{t} d_i(t')$$

Ak celkový súčet všetkých dávok agenta v histórii prekročí jeho prahovú hodnotu ($D_j(t) > T_j$), agent sa dostáva do infikovaného stavu. To spôsobí, že emočná úroveň sa nastaví na hodnotu 1,0 s exponenciálny útlmom smerom k 0, kde sa agent zase vráti do stavu neinfikovaný.

Pre potreby simulácie šírenia emócií v dave je potrebné aj tento model rozšíriť tak, aby zohľadňoval vzdialenosť medzi jednotlivcami. Riešením je zúžiť množinu výber možných jedincov len do určitej vzdialenosti, prípadne upraviť veľkosť dávky d_i podľa vzdialenosti.

3 Zhodnotenie

V tomto príspevku sme opísali dva modely šírenia emócií v dave. Pri modelovaní správania sa davu je práve šírenie emócii veľmi dôležité, keďže práve ono zabezpečuje prenášanie informácií správania medzi jednotlivcami, ktorí sa potom správajú ako skupina.

Model Durupinar má pre naše použitie jednu zásadnú nevýhodu. Strach jednotlivca v veľkej skupine kľudných ľudí klesá rovnako, ako v skupine vystrašených ľudí [7]. Myslíme si, že takúto situácie by model, ktorý použijeme, mal riešiť lepšie.

Pre potreby projektu APVV-RIOT sa nám javí použitie VU modelu v kombinácií s modelom správanie PECS ako vhodnejšie, keďže úprava modelu VU pre analýzu veľkého davu je jednoduchá a je možné ľahko pridať aj zmenu ovplyvňovania emócii v závislosti od vzdialenosti medzi jedincami.

Poďakovanie. Táto práca bola čiastočne podporená projektom APVV 0233-10, VG1/0675/11.

Literatúra

1. Barsade, S.G.: The ripple effect: Emotional contagion and its influence on group behavior. Administrative Science Quarterly 47(4), 644-675 (2002).
2. Bosse T., Duell R., Memon Z. A, Treur J., Wal C. N. V. D. A Multi-Agent Model for Mutual Absorption of Emotions. In Proceedings 23rd European Conference on Modelling and Simulation ECMS-09, 2009.
3. Bosse, T., Duell, R., Memon, Z. A., Treur, J., & Wal, C. N. V. D. (n.d.). A Multi-agent Model for Emotion Contagion Spirals Integrated within a Supporting Ambient Agent Model, 48-67.
4. Dodds P. S., Watts D. J. Universal behavior in a generalized model of contagion. Physical Review Lette rs, 92(21):218701–1–218701–4, 2004.
5. Durupinar, F.: From Audiences to Mobs: Crowd Simulation with Psychological Factors. PhD dissertation, Bilkent University, Dept. Comp. Eng, 2010.
6. Schmidt, B.: Human Factors in Simulation Models, 19th European conference on modelling and simulation, ECMS 2005, (2005).
7. Tsai, J., Bowring, E., Marsella, S., & Tambe, M. (n.d.). Modeling Emotional Contagion, MABS'11 12th International Workshop on Multi-Agent-Based Simulation. Taipei, Taiwan. May 2-6, 2011.

Smerovanie dizertačných projektov

Spätná väzba používateľov v odporúčaní

Martin Labaj[1]

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava, Slovensko
labaj@fiit.stuba.sk

Abstrakt. V príspevku sa zameriavame na spätnú väzbu používateľov v personalizovaných systémoch. Spätná väzba je v týchto systémoch mimoriadne dôležitý prvok, nakoľko umožňuje modelovať používateľa – jeho vedomosti, preferencie k obsahu a pod., ako i vyhodnocovať vytvorený alebo vytváraný systém. Našim cieľom je zlepšenie získavania, spracovania aj využitia spätnej väzby, špecificky v systémoch na odporúčanie. Náš výskum sa zameriava na tri hlavné oblasti: implicitná spätná väzba (správanie používateľov pri prehliadaní webu, paralelné prehliadanie), získavanie explicitnej spätnej väzby vo forme vhodnej pre konkrétneho používateľa (škála a rozsah hodnotenia) a vo vhodných okamihoch prostredníctvom adaptívnych otázok (získavanie hodnotení k objektom, ako i k vyhodnocovaniu personalizovaných systémov).

Kľúčové slová: implicitná spätná väzba, explicitná spätná väzba, paralelné prehliadanie, škálovanie a normalizácia hodnotení, systémy na odporúčanie

1 Úvod

Súčasné webové systémy sú často adaptovateľné alebo adaptívne, čiže adaptujú svoj výstup aktuálnemu používateľovi. Oba typy systémov sa spoločne označujú ako personalizované systémy [11]. Kým adaptovateľné systémy využívajú explicitný vstup od používateľa, aké informácie a ako chce vidieť, adaptívne systémy na základe aktivít používateľa odvodia jeho znalosti, ciele a obsah prispôsobia automaticky. Personalizované systémy sú teda závislé na vstupoch od používateľov – na spätnej väzbe.

Implicitná spätná väzba je založená na pozorovaní akcií používateľa, ktoré vykonáva pri bežnej práci – napríklad čas strávený na stránke, kliknutie na položku alebo jej preskočenie – používateľ hodnotí svojimi akciami. Explicitnú spätnú väzbu poskytuje používateľ vedome, napríklad na škále hodnotení zvolí svoje hodnotenie.

Jedným z častých typov personalizovaných systémov sú systémy na odporúčanie. Výstupom takéhoto systému je odporúčanie obsahu (tovaru v elektronickom obchode, príkladov vo výučbovom systéme), ktorý by pre používateľa mohol byť užitočný. Systémy na odporúčanie vychádzajú z troch typov údajov [9]: *objekty* (napríklad po-

[1] Školiteľ: prof. Mária Bieliková, Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 107–110.

ložky obchodu), *používatelia* (reprezentovaní svojimi preferenciami, vedomosťami, cieľmi – modelom používateľa) a *transakcie* (interakcie medzi používateľmi a objektmi, napríklad ktorým výučbovým textom sa venovali). Spätná väzba je využitá na modelovanie záujmov používateľa, na modelovanie domény – objekty sú v niektorých prístupoch modelované pomocou spätnej väzby ostatných používateľov [1], ako i na vyhodnotenie, či sa používatelia riadia vytváranými odporúčaniami.

Spätná väzba má význam aj v procese tvorby adaptívnych systémov [8]. V skorých fázach umožňuje tvorcom získavať požiadavky používateľov a prispôsobovať tomu systém a algoritmy a následne sledovať spokojnosť používateľov.

2 Implicitná spätná väzba pri prístupe k informáciám na Webe

Implicitnú spätnú väzbu môžeme rozdeliť do troch kategórií vzhľadom na objekt záujmu [7]: *skúmanie* (čas zotrvania, použitie, ...), *retencia* (uloženie odkazu, vytlačenie, ...) a *referencia* (odpoveď na správu, odkaz na stránku, ...). Okrem často sledovaných indikátorov (klikanie, čas, posúvanie stránky) sa teda nahliada na prácu používateľa s webovými zdrojmi ako s celkami. Súčasné prehliadače podporujú paralelné prehliadanie viacerých stránok súčasne v záložkách (angl. *tab*). Takéto správanie používateľov možno skúmať zo záznamov aktivity z rozšírenia prehliadača a následných pohovorov s účastníkmi [4]. Prehliadanie v záložkách je rozšírené a to v rôznych scenároch: používatelia otvárajú viaceré výsledky vyhľadávača a postupne ich prehliadajú, otvoria do záložiek stránky (napríklad výučbové objekty) ako pripomienky, ktorým sa budú venovať neskôr, porovnávajú rôzne stránky navzájom, atď.

Priame využitie v personalizácii je však obmedzené kvôli zachytávaniu týchto údajov – rozšírenie prehliadača umožňuje pokryť iba skupinu používateľov, ktorí si rozšírenie nainštalujú a zachytávanie aktivity na serveri dokonca toto správanie nevidí vôbec. Preto bolo paralelné prehliadanie v odporúčaní iba odhadované, napríklad pomocou úloh, ktorým sa používateľ môže venovať a predpokladom, že v záložkách pracuje na rôznych úlohách [3], čo ale nie je jediný scenár prehliadania v záložkách.

V našom výskume sme najskôr navrhli model zachytávajúci prehliadanie používateľa a algoritmus na tvorbu tohto modelu z údajov zachytávaných skriptami vloženými vo webových stránkach navštevovaných používateľom. Na rozdiel od rozširovania prehliadača sme zahrnuli do získavania údajov všetkých používateľov systému. Model sme aplikovali v adaptívnom vzdelávacom systéme ALEF [2] na analýzu práce používateľov počas učenia sa – 56 % používateľov aktívne využíva paralelné prehliadanie a identifikovali sme prepínanie z otázok a cvičení na učebné texty pre pomoc pri ich riešení [6]. Zároveň sme v trasách používateľov objavili vzťahy medzi objektmi nezachytené v doménovom modeli, napríklad medzi cvičením na najdlhší podzoznam v jazyku Prolog a cvičením na maximálnu hĺbku podzoznamu.

3 Forma a okamih explicitného hodnotenia používateľmi

Explicitnú spätnú väzbu vo forme hodnotení môže používateľ poskytovať v rôznych formách – od unárnych („vhodné“), binárnych („vhodné“/„nevhodné“) až skalárnych

hodnotení s rôznym rozsahom (1-3, 1-4 alebo 1-5 bodov), až po hodnotenie voľným textom. Tradične sa pritom v systéme hodnotí len jedným spôsobom, no aktuálny výskum ukazuje, že personalizovaním nástrojov (škál) pre používateľov získame kvalitnejšie hodnotenia [5]. Pri spracovaní získaných hodnotení je potrebné vedieť porovnať hodnotenia rôznych používateľov. Ak jeden používateľ neustále hodnotí priemerne (2-3 body) a raz udelí vysoké hodnotenie (5 bodov), toto vyššie hodnotenie môže mať väčší význam, než od používateľa, ktorý hodnotí aj iné objekty vysoko (4-5 bodov). Pri využívaní rôznych typov škál a rozsahov pre rôznych používateľov a pri možnosti/nemožnosti neutrálneho hodnotenia (párny/nepárny počet možností) je potrebné vedieť hodnotenia porovnať na rovnakú škálu.

Pri získavaní spätnej väzby pre vyhodnocovanie systému sa využíva viacero metód [11]: analýza logov, dotazníky, rozhovor s používateľom po práci, komentovanie akcií používateľom. Prostredníctvom týchto foriem môže byť spätná väzba získavaná buď v riadenom experimente v laboratóriu alebo menej často aj v neriadenom, kde používatelia pracujú prirodzene vo svojom prostredí. V takomto prípade je spätná väzba kvalitnejšia a výpovednejšia, pretože používatelia pracujú smerom k svojim prirodzeným cieľom, no nie všetky formy je možné použiť týmto spôsobom.

Ako vhodné sa javí adaptívne dopytovanie spätnej väzby, kedy systém pokladá používateľovi pre neho prispôsobené otázky (na hodnotenie objektov, na hodnotenie systému, hľadanie jeho záujmov) počas práce v systéme. Pre tento účel sme navrhli metódu na získavanie spätnej väzby od používateľa vo vhodných okamihoch, pri ktorých používateľ nebude vyrušovaný a zároveň bude mať v čerstvej pamäti predchádzajúcu činnosť, čím získame kvalitnejšiu, presnejšiu spätnú väzbu a vo väčšom množstve než napríklad pomocou dotazníka po skončení činnosti v systéme alebo v situáciách, v ktorých by používatelia inak nehodnotili. (Používateľ nevyužil žiadne odporúčanie. Nevšimol si ho alebo si položky prezrel, ale sú pre neho nevhodné?)

Metódu adaptívnych otázok sme overili v systéme ALEF v rámci troch experimentov. Pri pýtaní hodnotenia v momente zmeny používateľovej aktivity zisťovanom na základe pohľadu používateľa odmietli používatelia odpovedať len v 7 % otázok, na rozdiel od 33 % otázok pri dopytovaní v náhodných časoch. Zároveň sme získali spätnú väzbu od o 14,7 % viac používateľov, než koľko vyplnilo dotazník po skončení práce. Pri pýtaní hodnotenia po dočítaní sumarizácie výučbového objektu a začatí skalárneho hodnotenia bolo adaptívnou metódou získaných o 29,8 % viac hodnotení než pri ponechaní vloženia doplnkového hodnotenia iba na používateľovi.

4 Záver

Vo viacerých experimentoch sme overili základný prístup v sledovaných oblastiach – v spôsobe prehliadania Webu používateľom a vo forme a čase explicitného hodnotenia. Identifikovali sme otvorené miesta s potenciálom vylepšenia navrhnutých metód.

V oblasti paralelného prehliadania pracujeme na rozpoznaní a kategorizovaní aktuálnej situácie – otváranie zdrojov v záložkách môže predstavovať viaceré formy retencie aj referencie v implicitnej spätnej väzbe. Používateľ môže otvorením ďalšieho zdroja vytvoriť pripomienku, ktorej sa bude venovať neskôr (retencia) alebo vyjadro-

vať vzťah zdrojov (referencia). Predpokladáme, že zlepšeným modelovaním paralelného prehliadania používateľa prinesieme kvalitnejšie personalizované odporúčanie.

Adaptívne získavanie spätnej väzby rozširujeme o ďalšie vhodné situácie pre zobrazovanie dopytov, ako aj o adaptívnosť formy hodnotenia zahrnutím adaptívnej škály, ktorá bola nezávisle aplikovaná v systéme ALEF [10]. Očakávame získavanie kvalitnejšej a početnejšej relevantnej spätnej väzby od používateľov než pri tradičnom hodnotení závislom od iniciatívy používateľa.

Získavanie a spracovanie spätnej väzby v načrtnutých otázkach samozrejme nie je obmedzené na odporúčanie v doméne vzdelávania. Navrhované metódy a postupy plánujeme preniesť aj do iných adaptívnych systémov, respektíve všeobecne na Web.

Poďakovanie. Táto práca bola čiastočne podporená projektom VEGA VG1/0675/11, APVV 0233-10 a vznikla vďaka podpore v rámci OP Výskum a vývoj pre projekt ITMS: 26240220039, spolufinancovaný ERDF.

Literatúra

1. Adomavicius, G., and Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. In: IEEE Transactions on Knowledge and Data Engineering, 17(6). pp. 734–749.
2. Bieliková, M., Šimko, M., Barla, M., Chudá, D., Michlík, P., Labaj, M., Mihál, V., and Unčík, M. (2010). ALEF: Web 2.0 Principles in Learning and Collaboration. In: e-Learning´10, Proceedings of the International Conference on e-Learning and the Knowledge Society. Riga, Riga Technical University, pp. 54–59.
3. Bonnin, G., Brun, A., and Boyer, A. (2010). Towards Tabbing Aware Recommendations. In: Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia – IITM ’10. New York, ACM Press, pp. 316–323.
4. Dubroy, P., and Balakrishnan, R. (2010). A Study of Tabbed Browsing Among Mozilla Firefox Users. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI ’10. New York, ACM Press, pp. 673–682.
5. Gena, C., Brogi, R., Cena, F., and Vernero, F. (2011). The Impact of Rating Scales on User’s Rating Behavior. In: Proceedings of the 19th International Conf. on User Modeling, Adaption, and Personalization – UMAP ’11. Heidelberg, Springer-Verlag, pp. 123–134.
6. Labaj, M. (2012). Modeling Parallel Web Browsing Behavior for Web-based Educational Systems. ICETA 2012 (prijaté, odoslané do tlače).
7. Oard, D., and Kim, J. (1998). Implicit Feedback for Recommender Systems. In: AAAI Workshop on Recommender Systems. pp. 81–83.
8. Paramythis, A., Weibelzahl, S., and Masthoff, J. (2010). Layered Evaluation of Interactive Adaptive Systems: Framework and Formative Methods. In: User Modeling and User-Adapted Interaction, 20(5). pp. 383–453.
9. Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (Eds.). (2011). Recommender Systems Handbook. Boston, MA, Springer. p. 871
10. Šteňová, A. (2012). Feedback Acquisition in Web-Based Learning. In: 8th Student Research Conference in Informatics and Information Technologies. Bratislava, STU, pp. 139–144.
11. Van Velsen, L., Van Der Geest, T., Klaassen, R., and Steehouder, M. (2008). User-centered Evaluation of Adaptive and Adaptable Systems: A Literature Review. In: The Knowledge Engineering Review, 23(03), pp. 261–281.

Využitie dolovania dát v oblasti výroby polyamidových vláken

Alexandra Lukáčová*

Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky,
Technická univerzita v Košiciach, Letná 9, 042 00 Košice
Alexandra.lukacova@tuke.sk

Abstrakt. Vzhľadom k dosiahnutým pokrokom v systémoch zameraných na zber a spracovanie dát, stáva sa čoraz viac samozrejmosťou aplikovanie dolovania dát i vo výrobe s cieľom získavať nové znalosti, ktoré pomôžu vylepšiť výrobný proces. Táto práca prezentuje analýzu možnosti riešenia problému chemickej spoločnosti vyrábajúcej polyamidové vlákna, ktorá má záujem spoznať, ktoré vlastnosti majú vplyv na výslednú kvalitu vlákna. Použité budú pritom tak merané vlastnosti vstupnej suroviny, ako i hodnoty procesných parametrov z prebiehajúcej výroby.

Kľúčové slová: dolovanie dát, výrobný proces, kvalita výroby

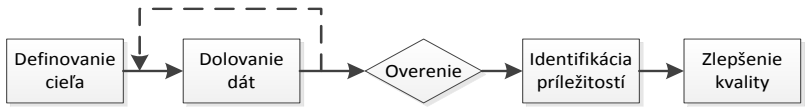
1 Úvod

Vo výrobe vzniká veľký počet interakcií, ktoré spolu s rôznymi procesnými podmienkami majú výrazný vplyv na finálnu kvalitu produktu. Riadiaci pracovníci sa mnohokrát pri vyskytnutí problému rozhodujú len na základe intuície, ktorá vychádza z ich dobrého poznania dát a danej domény. Keď by sme ale dokázali takéto udalosti detegovať, umožnili by sme rýchlejšie rozhodovanie sa, a prispeli tak i k dosiahnutiu signifikantného ekonomického prínosu. Preto sa vzhľadom k pokrokom v systémoch zameraných na zber dát stáva čoraz viac samozrejmosťou aplikovať techniky objavovania znalostí (KDD – Knowledge Discovery in Databases). KDD predstavuje iteratívny poloautomatický proces objavovania a verifikácie vzorov v obrovskom množstve údajov [1].

Neustále zlepšovanie kvality je v oblasti riadenia kvality výroby dôležitou podmienkou, pričom bežne používaným spôsobom na dosiahnutie cieľa je použitie metodológie DMAIC (Definovať – Merať – Analyzovať – Zlepšiť - Kontrolovať). Tá je však podľa He s kolektívom [2] vo výrobe, kde prebiehajú komplikované výrobné procesy, nedostatočná, pretože identifikovať a definovať konkrétny problém nie je jednoduché. Namiesto toho navrhol znalostne založený model, kde prvý krok predstavuje definovanie cieľa.

* Školiteľ: prof. Ing. Ján Paralič, PhD., Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 111–114.



Obr. 1. Znalostne založený model na zlepšenie kvality [2].

2 Súvisiace práce

Aplikovanie algoritmov dolovania dát vo výrobnej oblasti má svoje počiatky v 90-tych rokoch. Irani [3] vtedy vo svojej práci použil generalizovaný ID3 algoritmus, ktorý bol uplatnený vo viacerých oblastiach polovodičovej výroby. Zaujímavý postup pri sledovaní kontinuálneho procesu uviedol Nomikos [4]. Vo svojej práci zostavil predikčný model, ktorý sumarizuje vzťah medzi procesnými parametrami a časom. Ten následne použil na procesné monitorovanie, a to pomocou odchýlok predikcie od aktuálnej pozorovanej hodnoty. Jednu zo súčasných prác predstavil autor Bakir [5], zaoberajúci sa identifikovaním najvplyvnejších premenných, ktoré spôsobujú chybovosť konkrétneho výrobku vo výrobe tureckej liatinovej spoločnosti. Ten použitím algoritmu C5.0 vygeneroval desať užitočných pravidiel a určil hraničné hodnoty parametrov vhodné na použitie optimalizácie procesu výroby liatiny. Podobnú aplikáciu dolovania dát z údajov o výrobe tkanín poskytnutých tureckou firmou špecializujúcou sa na výrobu kobercov vytvoril Ciflikli [6], kde použitím algoritmu C4.5 navrhli model, ktorého presnosť predstavovala 72,811%. Zaujímavú štúdiu vytvoril taktiež Martinéz-de-Pisón s kolektívom [7]. Tá sa týkala dolovania asociačných pravidiel z časových radov dát výrobného procesu, pričom jej cieľom bolo vysvetlenie dôvodov zlyhaní pri zinkovaní oceľových cievok a ich budúcemu sa vyvarovaniu.

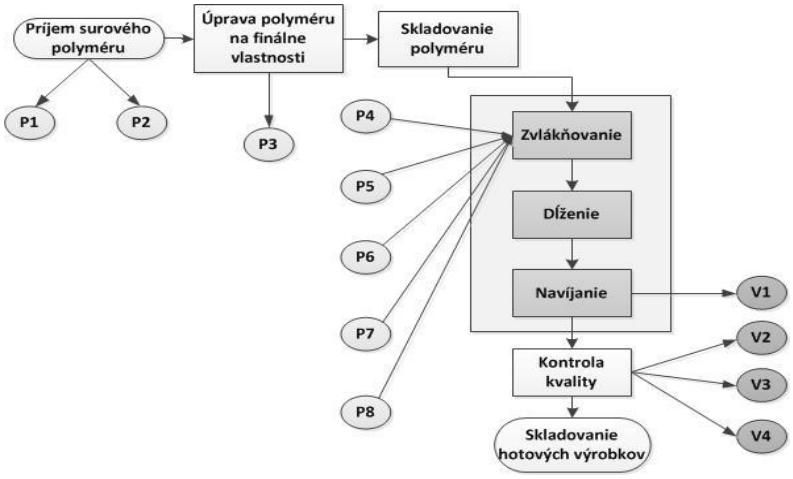
3 Prípadová štúdia

Cieľom sledovaného podniku, chemickej spoločnosti zaoberajúcej sa výrobou polyamidových vláken, je záujem redukovať výrobu s nízkou kvalitou.

V predošlej práci [8] sme sa zaoberali vplyvom vlastností vstupnej suroviny, pričom sme ale do úvahy nebrali procesné podmienky. Spoločnosť vo výrobe však momentálne zaznamenáva celkovo okolo 17 000 parametrov, na ktoré využíva software od firmy Wonderware. Taktiež využíva vlastné riešenie na sledovanie kvality vlákna "ADCS" a niekoľko riešení pripravených v tabuľkovom procesore EXCEL pre manuálne sledovanie vstupných a výstupných parametrov. Jedná sa teda o obrovské množstvo dát, ktoré sa využívajú len na sledovanie on-line stavu stroja, pričom nikto zatiaľ nesledoval aké veľké sú vplyvy jednotlivých parametrov a aké sú ich možnosti využitia na predikciu výslednej kvality výrobkov.

Aby sme ale dokázali zvýšiť kvalitu výrobku, je nevyhnutné najprv pochopiť vzťahy medzi vlastnosťami vstupnej suroviny – polyméru, procesnými parametrami a konečnými fyzikálno-mechanickými vlastnosťami finálneho produktu – vlákna. Polyamidy patria medzi umelo vytvorené organické vlákna vyrobené na báze syn-

tetických polymérov. Táto vstupná surovina – granulát sa považuje za základný stavebný kameň bezporuchového procesu výroby. Pri nej budeme sledovať hodnoty dvoch parametrov (P1, P2) surového polyméru dodaného dodávateľom. V procese ďalej nasleduje sušenie a úprava polyméru na finálne vlastnosti. Tu nás bude zaujímať hodnota parametra (P3), meraná vlastnosť takto pripraveného polovýrobku. Ten pokračuje do fázy zvlákňovania, kde budeme sledovať vplyv piatich procesných parametrov (P4-P8), zvolených doménovým expertom ako najvýznmanejšími. Ďalej nasleduje dĺženie a navíjanie, odkiaľ vychádza prvý výstupný atribút (V1), týkajúci sa počtu chybovosti vlákna. Navité vlákna následne v laboratóriu prechádzajú kontrolou kvality fyzicko-mechanických vlastností, pričom podľa doménového experta sú dôležité tri výstupy (V2-V4).



Obr. 2. Proces výroby polyamidového vlákna.

Celý proces, ako už bolo vyššie spomenuté, je kontinuálny a od začatia zvlákňovania už pripraveného polyméru po navinutie prvého vlákna na cievku netrvá viac ako päť minút. Tento čas je pre nás dôležitý, aby sme mohli následne identifikovať udalosti výroby nekvalitného vlákna. Hodnoty jednotlivých procesných parametrov sú numerického charakteru a do databázy sa zaznamenávajú každých desať sekúnd.

4 Záver

Cieľom môjho výskumu je návrh vhodných metód dolovania dát vo výrobnom prostredí, pričom sa budem zameriavať na procesné dáta v kombinácii s kľúčovými atribútmi kvality. Na začiatku bude potrebné diagnostikovať udalosti, pri ktorých nastala nízka kvalita vlákna a objavenie ich sekvenčných vzorov v časových radoch procesných údajoch. Následne bude pokračovať vytvorenie systému na podporu rozhodovania pri signalizácii alarmov. Navrhnuté a implementované metódy budem testovať na reálnych dátach z chemickej spoločnosti vyrábajúcej polyamidové vlákna,

ktorá má záujem spoznať, ktoré vlastnosti vstupnej suroviny a procesných parametrov majú vplyv na výslednú kvalitu vlákna.

Poďakovanie. Táto práca bola podporená KEGA grantom MŠVVaŠ SR č. 065TUKE-4/2011 (30%), VEGA grantom MŠVVaŠ SR č. 1/1147/12 (30%), ako aj projektom Rozvoj Centra informačných a komunikačných technológií pre znalostné systémy (kód ITMS projektu: 26220120030) na základe podpory operačného programu Výskum a vývoj financovaného z Európskeho fondu regionálneho rozvoja (40%).

Literatúra

1. Paralič J.: Proces objavovania znalostí v databázach. In: Objavovanie znalostí v databázach, pp. 1–11. Elfa, Košice (2003).
2. He S., He Z., Wang G., Li L.: Quality improvement using data mining in manufacturing processes, In: Data mining and knowledge discovery in real life applications, February 2009, I-Tech, Vienna, Austria. p. 438. ISBN 978-3-902613-53-0.
3. Irani K.B., Cheng J., Fayyad U.M., et al.: Applying machine learning to semiconductor manufacturing, IEEE Expert 8 (1) (1993), pp. 41–47.
4. Nomikos, P, MacGregor, J.F.: Multivariate SPC charts for monitoring batch processes. In: Technometrics, February 1995, Volume 37, No. 1.
5. Bakır B., Batmaz I., Gunturkun F. A., Ipekci I. A., Koksal G.: Defect cause modeling with decision tree and regression analysis.‖ In Proceedings of XVII international conference on computer and information science and engineering, 8–10 December 2006. Cairo: World Enformatica Society (Vol. 16, pp. 266–269).
6. Ciflikli C.: Implementing a data mining solution for enhancing carpet manufacturing productivity. In Knowledge-based systems, 23 (2010), pp. 783-788.
7. Martinéz-de-Pisón F. J, Sanz A., Martinéz-de-Pisón E., et al.: Mining association rules from time series to explain failures in a hot-dip galvanizing steel line. Computers and industrial engineering, Volume 63, 2012, pp. 22–36.
8. Lukáčová A.: A review of data mining applications in manufacturing. In: SCYR 2012: 12th Scientific Conference of Young Researchers: Proceeding from conference: May 15th, 2012 Herľany, Slovakia. Košice: FEI TU, 2012. pp. 64–67. ISBN 978-80-553-0943-9.

Údržba ľahkej sémantiky reprezentovanej informačnými značkami: koncept a ciele

Karol Rástočný*

Ústav informatiky a softvérového inžinierstva, Fakulta informatiky
a informačných technológií, Slovenská technická univerzita v Bratislave,
Ilkovičova 3, 842 16, Bratislava, Slovensko
karol.rastocny@stuba.sk

Abstrakt. Systémy vytvárajú množstvo metadát, ktoré obsahujú hodnotné informácie o obsahu informačných priestorov. Tieto medadáta sú však závislé od dynamického obsahu rozsiahlych informačných priestorov (napr. webu), ktorý opisujú. Každá zmena obsahu tak môže viesť k obmedzeniu platnosti metadát o tomto obsahu. Metadáta tak nestačí iba vytvárať, ale je potrebné ich neustále udržiavať. Vzhľadom k pomerne veľkému množstvu rôznych typov metadát, ktoré môžu vyžadovať rôzne prístupy k údržbe, sme sa v našej práci zamerali na informačné značky – opisné metadáta s určeným významom voči označenému obsahu, ktorými je možné vytvoriť ľahký sémanticky opis obsahu informačného priestoru. V článku zdôvodňujeme význam údržby metadát, definujeme pojem informačná značka a opisujeme koncept údržby ľahkej sémantiky reprezentovanej informačnými značkami.

Kľúčové slová: metadáta, informačná značka, údržba, ľahká sémantika

1 Úvod

Ontológie sa stávajú štandardnou, strojovo-spracovateľnou formou reprezentácie znalostí a informácií, pričom umožňujú reprezentovať informácie vo viacerých úrovniach formálnosti a expresívnosti, od jednoduchých slovníkov až po všeobecnú logiku [1]. V súčasnosti sú však možnosti ontológií výrazne obmedzené najmä ich kompletnosťou a pomerne veľkým množstvom chýb [2]. Tieto problémy sú vo výraznej miere zapríčinené skutočnosťou, že súčasné výpočtové zdroje nám neumožňujú vytvárať ontológie plne automaticky (niektoré znalosti, napr. axiómy, nevieme odvodiť bez univerzálnej indukcie [3]). Z tohto dôvodu systémy vytvárajú rôzne metadáta, často vo forme ľahkých ontológií [4, 5] (ontológií, ktoré neobsahujú axiómy [6]), ktorými vytvárajú ľahký sémantický opis informačného priestoru.

Automaticky vytvárané metadáta sú však priamo vytvárané z obsahu informačného systému a často sa na tento obsah priamo odkazujú (v prípade ontológií najmä prostredníctvom sémantických anotácií [7]). Zmeny v tomto obsahu tak priamo a ne-

* Školiteľ: Prof. Mária Bieliková, Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 115–118.

priamo ovplyvňujú aj metadáta, ktoré sú z tohto obsahu vytvorené alebo ho opisujú. Problémom však je, že editáciou obsahu nedochádza priamo k úprave metadát a čo vedie k obmedzeniu platnosti samotných metadát.

Okrem problému úpravy obsahu informačného priestoru, platnosť metadát ovplyvňuje aj čas – zastarávanie informácie. Existujúce systémy riešia tieto problémy najčastejšie zmazaním všetkých metadát po zmene dokumentu, prípadne obmedzeniu platnosti metadát na konkrétnu verziu dokumentu [8, 9]. Pri tomto riešení však dochádza k strate informácií, ktoré sú obnovené až po ich opätovnom vytvorení, nezávisle od toho, či boli zmazané metadáta dotknuté zmenou v dokumente. Súčasné riešenia tak nie sú vhodné pre rozsiahle informačné systémy (napr. web), ktoré vyžadujú vhodnú efektívnu metódu údržby metadát, ktorá zabezpečí dostupnosť informácií v podobe metadát aj po úprave obsahu v informačnom priestore. Keďže existuje pomerne veľké množstvo rôznych typov metadát [10], ktoré vyžadujú odlišné prístupy k ich údržbe, navrhujeme metódu automatickej údržby informačných značiek.

2 Informačné značky

Pod označením *informačné značky* rozumieme triedu opisných metadát s definovaným vzťahom voči označenému obsahu. Formálne, informačná značka je trojica definovaná:

- *Typom* – určuje typ a význam informačnej značky;
- *Ukotvením* – identifikuje označený informačný artefakt;
- *Telom* – reprezentuje štruktúrovanú informáciu, ktorej štruktúra zodpovedá typu informačnej značky.

Väčšina informačných značiek je vytváraná systémami na základe obsahu informačných priestorov a interakcie používateľov s týmto obsahom, pričom systémy využívajú rôzne techniky počítačového učenia a crowdsourcingu. Príkladmi takýchto informačných značiek sú identifikované koncepty, kategorizácia a podobnosť obsahu a záznamy o počte zobrazení. Informačné značky však môžu byť vytvárané aj priamo používateľmi a to prostredníctvom hodnotenia a štruktúrovaných anotácií, vytváraných prostredníctvom vyplnenia preddefinovaných formulárov [11, 12].

Na informačné značky je možné pozerať aj ako na graf, ktorého uzly sú tvorené označenými informačnými artefaktmi a telami informačných značiek, pričom hrany medzi nimi sú definované typmi a ukotvením informačných značiek. Keďže uzly grafu nie sú len jednoduché slovné frázy a hrany nadobúdajú rôzne typy, môžeme konštatovať, že informačné značky sú druhom neformálnej, klasifikačnej, ľahkej ontológie [6]. Ontológia tvorená informačnými značkami má presne definovanú štruktúru informačných značiek prostredníctvom ich typov a umožňuje definíciu hierarchií v informačnom priestore (napr. informačná značka, ktorá je ukotvená k triede v zdrojovom kóde, má typ „part-of“ a jej telo obsahuje meno menného priestoru). Neumožňuje však definíciu tried označených informačných artefaktov. Tieto vlastnosti zaradzujú z hľadiska formálnosti a expresivity informačné značky medzi neformálne hierarchie a schémy databáz [1].

3 Údržba informačných značiek

Informačné značky predstavujú stavebný prvok ľahkej sémantiky informačných priestorov, ktorý je prirodzený pre systémy a z pohľadu anotácií aj pre samotných používateľov. Informačné značky však neriešia problém údržby, ale rozdeľujú tento problém na dve časti – údržbu tela a údržbu ukotvenia. Zároveň definovaním typu a štruktúry metadát zavádzajú vyššiu formálnosť, ktorá napomáha zrozumiteľnosti informácií uložených v metadátach a tým aj ich údržbe.

Zložitosť údržby ukotvenia informačných značiek je závislá od deskriptora popisujúceho umiestnenie informačnej značky v rámci dokumentu. V prípade jednoduchých deskriptorov (napr. index prvého a posledného písmena označenej oblasti v texte) je potrebné tieto deskriptory upravovať vždy keď dôjde k zmene dokumentu, ku ktorému sú značky ukotvené, pričom je potrebné analyzovať zmeny vykonané v dokumente. Výhodou týchto deskritptorov je jednoduchosť ich použitia z hľadiska koncových systémov, ktoré nemusia implementovať zložité algoritmy na vyhodnotenie deskriptorov. Opačná situácia nastáva v prípade zložitejších robustných deskritptorov, ktorých kvalita degraduje voči zmenám v dokumente pomaly a preto ich nie je potrebné udržiavať. Na druhej strane, však vyžadujú implementáciu pomerne zložitých algoritmov (často v podobe pravdepodobnostného vyhodnocovania podobnosti textov), ktoré vyhodnotia deskriptor a určia polohu informačnej značky v dokumente [13].

V súčastnosti môžeme považovať problém údržby ukotvenia za vyriešený, pričom ostáva na tvorcoch systému, aby zvolili vhodný deskriptor. Problém údržby tela informačných značiek však ostáva stále otvorený. Tento problém je najbadateľnejší najmä v prípade rozsiahlych dynamických informačných priestorov (napr. webu), pre ktoré nie je možné garantovať úplnú konzistentnosť informačných značiek. Vhodná metóda automatickej údržby informačných značiek by však mohla v relatívne krátkom čase identifikovať neplatné informačné značky a pokúsiť sa ich opraviť. My využívame dve techniky na identifikáciu a opravu neplatných informačných značiek:

- *Techniky strojového učenia* – keďže väčšina informačných značiek je vytváraná prostredníctvom deterministických algoritmov, je možné vytvoriť algoritmus strojového učenia, ktorý bude sledovať zmeny v informačných značkách po úprave označených dokumentov, pričom bude schopný identifikovať neplatné informačné značky a vykonať potrebné zmeny v neplatných informačných značkách. V prípade, že vyhodnotí informačné značky ako neopraviteľné, označí ich za neplatné, vďaka čomu nebude dochádzať k chybnému využitiu neplatných informačných značiek koncovými systémami;
- *Crowd computing* – techniky využívajúce dav neumožňujú tak rýchle reakcie na zmeny v označenom obsahu ako techniky strojového učenia, ale pri dostatočne veľkom dave (systémov vytvárajúcich a spracúvajúcich značky) môžu dosahovať presnejšie výsledky. Pri identifikácii neplatných informačných značiek umožňujeme systémom priamo ohlásiť neplatné informačné značky. Zároveň sledovaním prístupu systémov k informačným značkám vieme identifikovať „nezaujímavé“, potenciálne neplatné informačné značky. Dav môže byť využitý aj na samotnú opravu informačných značiek, ktoré algoritmus strojového učenia nevedel opraviť.

Navrhovanú metódu automatickej údržby informačných značiek plánujme overiť v rámci projektov PerConIK [14] a TraDiCe, v ktorých jednotlivé systémy zdieľajú získané informácie o dokumentoch v informačných priestoroch projektov prostredníctvom informačných značiek.

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0675/11, APVV-0233-10 a vznikla vďaka podpore v rámci OP Výskum a vývoj pre projekt ITMS: 26240220039, spolufinancovaný ERDF.

Literatúra

1. Wong, W., Liu, W., Bennamoun, M.: Ontology Learning from Text: A Look back and into the Future. ACM Computing Surveys. 44, 36 (2011).
2. Sabou, M., Gracia, J., Angeletou, S., Aquin, M., Motta, E.: Evaluating the Semantic Web: A Task-based Approach. Proc. of the 6th Int. the Sem. Web and 2nd Asian Conf. on Asian Sem. Web Conf. pp. 423–437. Springer-Verlag, Berlin, Heidelberg (2007).
3. Hutter, M.: Open Problems in Universal Induction & Intelligence. Algorithms. 3, 879–906 (2009).
4. Drumond, L., Girardi, R.: A Survey of Ontology Learning Procedures. In: de Freitas, F.L.G., Stuckenschmidt, H., Pinto, H.S., Malucelli, A., and Corcho, Ó. (eds.) WONTO'08. p. 12. CEUR-WS.org (2008).
5. Hazman, M., R. El-Beltagy, S., Rafea, A.: A Survey of Ontology Learning Approaches. Int. Journal of Computer Applications. 22, 36–43 (2011).
6. Giunchiglia, F., Zaihrayeu, I.: Lightweight Ontologies. In: Liu, L. and Özsu, M.T. (eds.) Encyclopedia of Database Systems. pp. 1613–1619. Springer US, Boston, MA (2009).
7. Tallis, M.: Semantic Word Processing for Content Authors. Proc. of the 2nd Int. Conf. on Knowledge Capture KCAP 2003 on Knowledge Markup and Sem. Annotation. p. 7. Citeseer (2003).
8. Plimmer, B., Chang, S.H., Doshi, M., Laycock, L., Seneviratne, N.: iAnnotate□: Exploring Multi-User Ink Annotation in Web Browsers. Proc. of the 8th Australasian Conf. on User Interface - Volume 106. pp. 52–60. Australian Computer Society, Inc. (2010).
9. Sanderson, R., Van de Sompel, H.: Making web annotations persistent over time. Proc. of the 10th Annual Joint Conf. on Dig. Lib. - JCDL'10. pp. 1–10. ACM Press, NY (2010).
10. Gill, T., Gilliland, A.J., Whalen, M., Woodley, M.S.: Introduction to Metadata. Getty Publications, Los Angeles (2008).
11. Zheng, Q., Booth, K., McGrenere, J.: Co-authoring with Structured Annotations. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems - CHI '06. p. 131. ACM Press, NY, USA (2006).
12. Sarkas, N., Paparizos, S., Tsaparas, P.: Structured Annotations of Web Queries. Proc. of the 2010 Int. Conf. on Management of Data - SIGMOD '10. p. 771. ACM Press, NY, USA (2010).
13. Phelps, T.A., Wilensky, R.: Robust intra-document locations. Computer Networks. 33, 105–118 (2000).
14. Bieliková, M., Návrat, P., Chudá, D., Polášek, I., Barla, M., Tvarožek, J., Tvarožek, M.: Procedia Technology Webification of Software Development: General Outline and the Case of Enterprise Application Development. Procedia Technology, 3rd World Conf. on Inf. Tech. 4, 5 (2012). (to appear)

Personalizované vyhľadávanie v zdrojových kódoch

Eduard Kuric[1]

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave,
Ilkovičova 3, 842 16 Bratislava
kuric@fiit.stuba.sk

Abstrakt. Personalizované vyhľadávanie je v prostredí Webu dobre preskúmané, avšak v doméne zdrojových kódov ide o relatívne nový fenomén. V tomto príspevku sa zameriavame na vyhľadávanie v zdrojových kódoch, pričom diskutujeme možnosti personalizovaného vyhľadávania a otvorené problémy. Úspech vyhľadávania v zdrojových kódoch je často ovplyvnený kognitívnymi bariérami, ktorým programátori čelia. Diskutujeme, ako je možné tieto bariéry eliminovať s cieľom pomôcť programátorovi pri vyhľadávaní relevantnej informácie k cieľovej úlohe.

Kľúčové slová: personalizované vyhľadávanie, zdrojový kód, znalosti programátora, spätná väzba, kolaboratívne filtrovanie

1 Vyhľadávanie v zdrojových kódoch

Vývoj softvéru patrí medzi jednu z najkreatívnejších činností vôbec. Programátori každý deň čelia (novým) úlohám v nových doménach. Robia rozhodnutia, ktoré si vyžadujú širokú škálu informácií o softvérových projektoch, na ktorých pracujú. Kvalita ich rozhodnutí často závisí od ich znalostí, skúseností, zručností a úsudkov.

Bolo publikovaných niekoľko štúdií o identifikácii problematických otázok, ktorým programátori najčastejšie čelia [6]. Inými slovami, aké informácie potrebujú o zdrojových kódoch na vykonanie (počas vykonávania) cieľovej úlohy a ako pristupujú k získavaniu potrebných informácií. Tieto štúdie poskytujú konkrétne poznatky o aspektoch vývoja softvéru, t.j. aké informácie programátori hľadajú a prečo ich hľadajú. Štúdie ukázali, že programátori pri písaní nového zdrojového kódu hľadajú často definície funkcií, existujúce komponenty s rovnakou, alebo podobnou funkcionalitou k cieľovej úlohe (za účelom znovupoužitia, príp. vykonania zmien).

Aby mohol programátor nejakú časť zdrojového kódu znovupoužiť, resp. vykonať zmeny, najskôr potrebuje nájsť zodpovedajúci (relevantný) kód. Inými slovami, keď programátor čelí nejakej úlohe, najskôr jej musí porozumieť, t.j. čo je jej cieľom a čo potrebuje nájsť. Následne vytvorí (naformuluje) dopyt, ktorý môže pozostávať z množiny kľúčových slov, príp. zo špecifikácie v nejakom formálnom jazyku. Toto

[1] Školiteľka: prof. Mária Bieliková, Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 119–122.

vedie často ku „strate informácií“, pretože programátori nie sú vždy schopní exaktne porozumieť problému (aké informácie majú hľadať), resp. nie sú schopní vytvoriť správny (zodpovedajúci) dopyt. Na získanie (vyhľadanie) relevantných komponentov k dopytu je potrebná ich indexácia. Tento proces však tiež vedie často ku strate informácií. Problém straty informácií je jednou z príčin výskumu v tejto oblasti.

Vo všeobecnosti sa proces identifikácie komponentov (korešpondujúcich špecifickej funkcionalite) označuje ako lokalizácia konceptov. Cieľom je nájsť komponenty, v ktorých sú implementované (požadované) koncepty. Koncepty sú zvyčajne kľúčové slová extrahované z názvov tried (rozhraní), funkcií (metód), identifikátorov, termy extrahované z komentárov obsiahnutých v zdrojových kódoch, externej dokumentácie, a špecifikácií úloh prepojených s ich implementáciami.

Existuje niekoľko prístupov na lokalizáciu konceptov v zdrojových kódoch [2]. Tieto sú založené na rôznych spôsoboch analýzy kódu. Zvyčajne sa delia na statické, dynamické a hybridné. V statických prístupoch je zdrojový kód spracovávaný ako „textový“ dokument. Používajú sa známe metódy z oblasti vyhľadávania informácií v prirodzenom jazyku, pričom program nemusí byť úplný (vykonateľný). Dynamické prístupy vyžadujú vykonateľný zdrojový kód. Sú založené na sledovaní vlastností a analyzovaní správania sa predmetného programu počas jeho vykonávania.

Hoc sú tieto prístupy dôležité, nemenej dôležité je preskúmať, čo motivuje programátorov formulovať dopyty, resp. aké znalosti potrebujú. Na strane druhej, ich (aktuálne) znalosti môžu byť základom toho, čo chcú nájsť, t.j. je žiaduce personalizované vyhľadávanie na základe ich vedomostí (skúseností) a aktuálnej (cieľovej) úlohy.

2 Určenie zámeru programátora

Empirické štúdie (napr. [7]) ukazujú, že programátori majú rôznu úroveň znalostí o softvérových systémoch, ktoré sa vyznačujú rôznorodou, resp. zložitou funkcionalitou. Programátori čelia kognitívnym bariéram, ktoré majú vplyv na lokalizáciu konceptov. Často majú nedostatok znalostí o systémoch, t.j. nie sú si vedomí existencie komponentov, ktoré môžu byť pre nich potenciálne užitočné. Ďalším problémom sú tzv. konceptuálne medzery, napr. programátor chce nájsť metódu na vykreslenie kružnice (*drawCircle*), avšak v systéme je implementácia pod názvom *drawOval*.

Kvôli konceptuálnym medzerám a nedostatočnej znalosti o softvérových systémoch, programátori nie sú schopní formulovať jednoznačné dopyty. Inými slovami, môžu chcieť nájsť koncepty (komponenty obsahujúce dané koncepty), ktoré poznajú vágne, a o ktorých predpokladajú, že existujú v informačnom priestore (v implementácii softvérového systému), avšak nie sú schopní vytvoriť dobre definované dopyty.

Pomocou explicitnej spätnej väzby sme schopní čiastočne redukovať konceptuálne medzery programátorov. Platí to za predpokladu, že programátor dokáže naformulovať vhodný počiatočný dopyt, ktorý postupne upresňuje na základe reformulácií a spätnej väzby (výsledkov). Avšak, veľa programátorov nie je schopných vytvoriť dobre definovaný dopyt pri ich prvom pokuse lokalizovať relevantný komponent, t.j. v mnohých prípadoch zadajú „chudobný“ dopyt (čo do počtu konceptov). Následne sa programátor musí zaoberať otázkou, či je daný dopyt nesprávny, dokonca, či v sku-

točnosti existuje v softvérovom systéme to, čo naozaj hľadá. Explicitná spätná väzba nedokáže pomôcť programátorovi, ak je jeho počiatočný dopyt chudobný [3].

Iným možným riešením je kolaboratívne spresnenie dopytu. Cieľom je navrhnúť programátorovi koncepty, príp. priamo komponent na základe vyhľadávacích aktivít iných programátorov v podobných (rovnakých) situáciách lokalizácie konceptov, t.j. na podporu cieľového programátora využívame skúsenosti iných programátorov [7].

Uvažujme situáciu, keď nejaký programátor našiel požadovaný komponent a následne ho znovupoužil. Na základe modelovania jeho vyhľadávacích aktivít a nízkoúrovňových aktivít vykonaných vo vývojom prostredí (znovupoužitie nájdeného komponentu), dokážeme zistiť (rozpoznať) podobné (rovnaké) situácie v lokalizácii konceptov. Ak dokážeme identifikovať podobné situácie (z reformulácie dopytu, príp. spätnej väzby), potom dokážeme ponúknuť programátorovi vhodné dopyty. Týmto spôsobom dokážeme cieľovému programátorovi pomôcť v prípade, že má znalostné medzery o softvérovom systéme, a čo viac, poodhaliť koncepty, o ktorých existencii si nebol vedomý. V ideálnom prípade, dokážeme rozpoznať rovnaké situácie a odkázať programátora priamo na komponent obsahujúci hľadané koncepty.

S týmto súvisí niekoľko otvorených problémov, a to, ako modelovať a reprezentovať vyhľadávacie aktivity programátora a ako určiť (odvodiť) podobné (rovnaké) situácie na základe počiatočného dopytu, jeho reformulácií a nízkourovňových aktivít programátora vo vývojovom prostredí. Otvorenou výzvou zostáva, ako navrhnúť mechanizmus, ktorý podporí programátora pri vyhľadávaní relevantných informácií za predpokladu, že nie je schopný vytvoriť vhodný počiatočný dopyt. Je potrebné tiež počítať s konceptuál-nymi medzerami a nedostatkom znalostí programátorov o softvérových systémoch.

3 Identifikácia uznávaného zdrojového kódu

Nástroje na vyhľadávanie v zdrojových kódoch pomáhajú programátorom nájsť a znovupoužiť komponenty. Na podporu týchto aktivít však nepostačuje implementovať jednoduchý „plnotextový“ vyhľadávač, ale treba uvažovať reputáciu zdrojového kódu. Inými slovami, okrem relevancie výsledkov je dôležité aj ich „renomé“.

Keď sa programátori rozhodujú, ktoré komponenty zahrnú do riešenia, zvyknú hľadať stanoviská (názory, skúsenosti) iných programátorov o potenciálnych kandidátoch. Existujúce prístupy vychádzajú z kolaboratívneho filtrovania (napr. [5]), pričom sa využívajú (o)hlasy používateľov na určenie popularity (kvality) zdrojových kódov. Napr., programátori preferujú výsledky vyhľadávania z dobre známych „open-source“ projektov pred menej známymi. Pre lepšiu identifikáciu správneho komponentu by bolo vhodné, keby sme zahrnuli okrem štatistík o projekte taktiež renomé autorov.

Hodnotenie zdrojového kódu na základe jeho renomé môže byť vierohodný spôsob na usporadúvanie výsledkov vyhľadávania s využitím sociálnych faktorov. Výskumné štúdie ukazujú, že relevancia aj renomé sú dôležité, avšak nie je úplne zrejmý vzťah medzi nimi. V prípade, že budeme klásť väčší dôraz na renomé zdrojových kódov, môžu nastať situácie, že „znevýhodníme“ relevantné výsledky vyhľadávania z menej populárnych projektov. Preto považujeme za dôležité preskúmať mieru, do ktorej uprednostniť renomé pred popularitou, resp. vzťah týchto dvoch charakteristík.

Existujúce prístupy, ktoré sa zameriavajú na odporúčanie expertov pre zdrojové kódy (napr. [5]), sú založené na autorstve a interakciách programátorov s kódom. Akýmsi „tichým“ predpokladom týchto prístupov je, že práve autorstvo je významným činiteľom na určenie, kto pozná cieľový zdrojový kód. Nedávne štúdie a výsledky ukazujú, že by mali byť zohľadňované ďalšie faktory na vylepšenie existujúcich prístupov, napr. úloha (významnosť) zdrojového kódu v systéme. Avšak, zatiaľ čo autorstvo môže byť ľahko určené z repozitára zdrojových kódov, určiť spoľahlivo významnosť zdrojového kódu je problém.

4 Zhrnutie

Identifikovali sme tri otvorené problémy, a to určenie zámeru programátora, identifikácia uznávaného zdrojového kódu a určenie znalostí programátora. Modelovanie znalostí programátorov môže byť užitočné pri kolaborácii, identifikácii uznávaného zdrojového kódu, alebo personalizovaní výsledkov vyhľadávania [1]. Na strane druhej, na základe určenia zámeru cieľového programátora môžeme vo výsledkoch vyhľadávania uprednostniť (uznávaný) zdrojový kód, ktorý bude zodpovedať jeho skutočným potrebám. Nedávne štúdie ukazujú, že okrem relevancie výsledkov je taktiež potrebné zohľadňovať aj ich reputáciu, čo si vyžaduje kolaboratívne filtrovanie a modelovanie znalostí programátorov. V našej ďalšej práci sa preto sústredíme na identifikáciu uznávaného zdrojového kódu na základe modelovania znalostí programátorov, pričom sme publikovali metódu na automatickú extrakciu skúseností programátorov zo zdrojových kódov [4].

Poďakovanie. Táto práca bola čiastočne podporená projektom VEGA VG1/0675/11, TRADICE 0208-10, PerConIK 26240220039 a vznikla vďaka podpore v rámci OP Výskum a vývoj pre projekt ITMS: 26240220039, spolufinancovaný ERDF.

Literatúra

1. Bielikova, M., et al.: Webification of Software Development: General Outline and the Case of Enterprise Application Development. In *Procedia Technology, 3rd World Conf. on Inf. Tech.*, Vol. 4, to appear.
2. Dit, B., Revelle, et al.: Feature Location in Source Code: A Taxonomy and Survey, *Journal of Software Maintenance and Evolution: Research and Practice*, 2011, to appear.
3. Gay, G., Haiduc, S., Marcus, A., Menzies, T.: On the use of relevance feedback in IR-based concept location. In *ICSM*, IEEE, 2009, pp. 351-360.
4. Holub, M., Kuric, E., Bielikova, M.: Modeling the knowledge of a software developer using light-weight semantics. In *ZNALOSTI 2012*, Mikulov, Czech Rep., 2012, to appear.
5. Ichii, M., Hayase, et al.: Software component recommendation using collaborative filtering. In *Proc. of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation*, Washington, IEEE Computer Society, 2009, pp. 17-20.
6. LaToza, T. D.: Answering common questions about code. In *Companion of the 30th international conference on Software engineering*, ICSE Companion '08, ACM, pp. 983-986.
7. Ye, Y., Fischer, G.: Supporting reuse by delivering task-relevant and personalized information. *In Proc. of the 24th Int. Conf. on Software Eng.*, NY, ACM, 2002, pp. 513-523.

Zlepšovanie nájditeľnosti informácií v diskusných skupinách pomocou nástrojov organizácie poznania

Andrea Hrčková[1]

Filozofická fakulta Univerzity Komenského,
Katedra knižnično-informačnej vedy, Gondova 2, Bratislava
hrckova.andrea@gmail.com

Abstrakt. Diskusné skupiny sú informačne bohatým miestom, obsahujúcim neverejné znalosti, ktoré na iných miestach ťažko nájdeme. V mnohých prípadoch je však problematické ich spätné vyhľadanie. Jedným z riešení je organizácia obsahu diskusných skupín pomocou niektorého z nástrojov organizácie poznania. V príspevku sa snažíme navrhnúť najvhodnejší spôsob organizácie informácií vzhľadom na špecifiká fungovania virtuálnych skupín.

Kľúčové slová: diskusné skupiny, taxonómie, ontológie, fazety, folksonómie, SIOC, Schema

1 Charakter komunikácie v diskusných skupinách

Podľa Kujawski a kol. [1] je dnes typická štruktúra fóra podobná rozvetvenému stromu, kde jedna téma podnecuje komentáre a diskusiu na ďalšie príbuzné obsahy (obr. 1). Štruktúra stromov záleží aj na téme diskusie. Špecializované diskusie majú na rozdiel od všeobecných krátke a málo rozvetvené „konáre“, pretože iba málo používateľov má dostatok vedomostí na to, aby sa k nim vedel vyjadriť.

Z toho dôvodu je centralizovaná organizácia poznania v malých fórach dostatočná. Problém prichádza, keď je fórum všeobecnejšieho charakteru, a teda aj jeho „konáre“ sú dlhé a husto rozvetvené. Správny nástroj organizácie poznania diskusných skupín by mal dokázať zatriediť všetky príspevky. Aj malé fórum sa postupom času môže rozrastať a práve preto by mal byť nástroj, organizujúci jeho poznanie dostatočne flexibilný. Organizácia poznatkov na týchto miestach teda nemôže byť striktná ani centralizovaná.

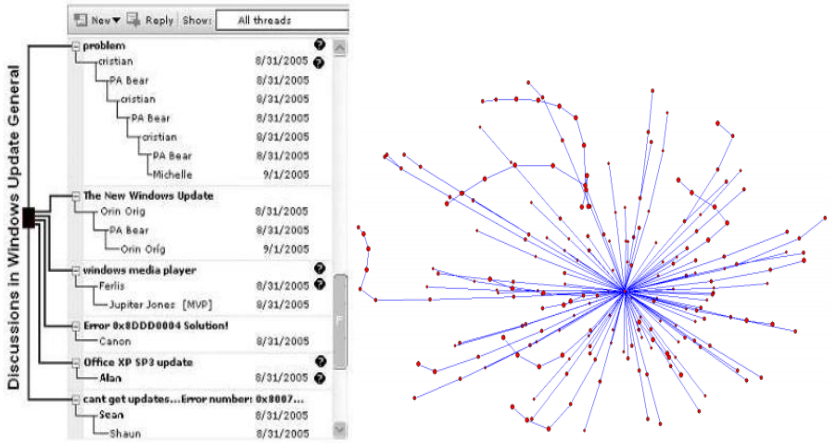
2 Možnosti organizácie poznatkov v diskusných skupinách

Mnohé diskusné fóra sa v súčasnosti organizačne podobajú na divoký západ. Príspevky neorganizujú vôbec, zoradené sú iba podľa času alebo abecedne. Príkladom je

[1] Školiteľ: Prof. PhDr. Jaroslav Šušol, PhD., Univerzita Komenského, Filozofická fakulta, Katedra knižničnej a informačnej vedy

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 123–126.

fórum na zdieľanie kníh Soulwar. Takéto triedenie je však nedostatočné a používatelia sa dostanú ku knihe, ktorú potrebujú, iba náhodne.



Obr. 1. Štruktúra malej diskusnej skupiny Physic. Zdroj: Kujawski a kol. (2007).

Ďalším využívaným spôsobom organizácie príspevkov vo fórach sú *taxonómie*. Taxonómia je prísne hierarchický nástroj organizácie poznania, pozostávajúci v prípade diskusných fór z moderátorom dopredu vytvorených kategórií [2]. Príkladom fóra s prepracovanou tematickou taxonómiou je Ebook. Taxonómie nedokážu zorganizovať všetky informácie podľa požiadaviek každého používateľa. Svedčia o tom mnohé podnety priamo od používateľov fóra, týkajúce sa ďalšej možnej kategorizácie.

Inak je to v prípade *folksonómií*, ktoré umožňujú pridávať značky (tagy) k vlastným príspevkom samotnými používateľmi. Tag je metainformácia o objekte, ktorú pridávajú používatelia individuálne podľa svojho vlastného slovníka [3]. Ostatní účastníci môžu na základe nich príspevky vyhľadať, preto majú aj spoločenskú funkciu. Jedným z príkladov fóra s možnosťou organizácie pomocou tagov je Ebay forum.

Podľa Weinbergera [4] je organizácia poznania pomocou tagov lacným a jednoduchým riešením pre využitie múdrosti davu. Je však možná iba vtedy, ak sú používatelia motivovaní urobiť pre seba a ostatných niečo navyše. Aj Wal [5] uvádza ako hlavný dôvod tagovania osobný prospech, ktorý vyplýva z chýbajúcich metadát a ďalšieho kontextu. Iní tagujú iba kvôli záujmu, či potreby socializácie [3].

Tento typ organizácie využívajú rôzne typy virtuálnych komunít, napr. aj pri tvorbe záložiek na portáli Delicious. Považujeme ho za vhodný aj pre diskusné fóra práve kvôli ich demokratickému a socializačnému charakteru. Folksonómie dokážu zatriediť naozaj všetky informácie vo fóre, pričom autor príspevku sám najlepšie vie, ktoré slová ho najlepšie charakterizujú. Nevýhodou tohto typu organizácie je jej prílišná voľnosť, pričom ak je nekontrolovaná, môžeme sa dostať na chaotickú úroveň neorganizovaného fóra. Preto by bolo výhodnejšie tento spôsob kombinovať s niektorou z prísnejších hierarchií ako napríklad taxonómie alebo fazety.

Fazety sú adaptívne kategórie, ktoré umožňujú nájsť rovnaké údaje viacerými spôsobmi [3]. Jednou z najznámejších služieb sprostredkujúcich diskusie a využívajúcich fazetovú klasifikáciu sú Google Groups. Pri podrobnejšej analýze [6] sme našli ďalšie

dve, ktoré v tejto službe chýbajú (prístupnosť a kontrola skupiny) a nedostatočnú kategorizáciu v prípade fazety „téma diskusnej skupiny“ (iba do 13 kategórií). Navrhli sme ju skombinovať s používateľsky vytvorenými kategóriami (tagmi), ktoré sú v rámci tejto fazety žiadúce. V podobe doplnenej o tagy by sme mohli považovať fazetovú klasifikáciu za najvhodnejšiu alternatívu organizácie príspevkov v diskusných skupinách. Pre menšie fóra však úplne postačí aj prepracovaná taxonómia.

Niektorí autori zachádzajú pri organizácii poznania za hranice jednotlivých fór. Navrhujú prepojenie poznania v diskusných fórach, blogoch a iných virtuálnych komunitách pomocou *ontológií***.** Ontológie poskytujú zdieľaný slovník, ktorý popisuje istú oblasť. Definujú objekty/pojmy, ktoré s touto oblasťou súvisia a súčasne popisujú ich vlastnosti a vzťahy [7]. Vzťahy medzi týmito objektmi môžu byť rôzne na rozdiel od prísne hierarchickej štruktúry taxonómií. Ontológie sú základným kameňom sémantického webu, preto by takto prepojené komunity mohli byť súčasťou jeho myšlienky a používatelia by sa k ich poznatkom dostali oveľa jednoduchšie. Napr. Breslin [8] navrhuje využiť už existujúcu ontológiu *SIOC* (Semantically Interlinked Online Communities) s rámcom RDF, využiteľnú práve pre diskusné fóra. Problémom je, že sémantický web sa stále nedarí plne zrealizovať, pretože tvorcovia webových sídiel nemajú motiváciu komerčného charakteru pre takéto prepojenie. Aj preto väčšinou stále používajú pri ich tvorbe neštruktúrovaný jazyk namiesto štruktúrovaného.

Zmena je však na dosah vďaka spoločnému projektu najväčších vyhľadávacích nástrojov Google, Yahoo a Bing. Projekt sa nazýva *Schema.org* a určuje html tagy pre konkrétne časti obsahu ako napríklad person (osoba), event (udalosť), organization (firma). Slovník projektu Schema by mohol byť úspešný práve kvôli SEO (optimalizácii stránok pre vyhľadávacie nástroje). Tvorcovia webových sídiel ho budú využívať na to, aby sa dostali na čo najvyššiu pozíciu vo výsledkoch vyhľadávania. V prípade virtuálnych komunít by sa dala použiť kategoria schémy „Thing“ (Vec) a jej súčasť „CreativeWork“ (kreatívna práca).

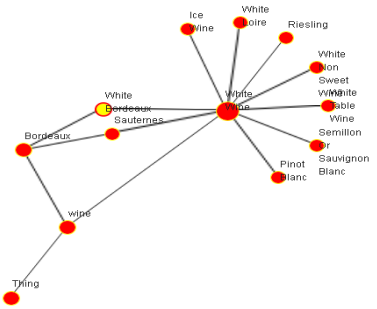
V súčasnosti sa tvorcovia SIOC postupne Scheme prispôsobujú. Vytvorili slovník, prekladajúci metadáta v Scheme do rámca RDF. Nájdeme ho na stránke Schema.rdfs.org. Vyzerá to tak, že myšlienka sémantického webu zabudnutá neostane. Z dôvodu vyhľadateľnosti fóra a informácií v ňom je tento spôsob organizácie poznania viac než vhodný.

Ontológie je možné vizualizovať pomocou grafov či sietí. Takáto vizualizácia používateľom výrazne zjednodušuje prístup k poznatkom a popri tom im prináša zážitok. Jedným z nástrojov, umožňujúcich vizualizáciu ontológií je TouchGraph [9]. Na obr. 2 vidíme test vizualizácie na tému víno. Každá podtéma sa dokáže ďalej rozbaľovať po jej označení myšou. Takáto forma sa najviac podobá stromovej štruktúre diskusných fór, ktorú popísal Kujawski a kol. [1].

3 Záver

Vidíme, že organizácia poznatkov v diskusných fórach by nemala byť striktná ani centralizovaná. Musí rešpektovať charakter fóra, a preto by mala obsiahnuť jeho obsah z rôznych aspektov. Ako nástroje organizácie poznania sú na tento účel vhodné fazety alebo ontológie. Prikláňame sa však k ontológiám, pretože umožňujú prepojiť

rôzne fóra, siete a blogy na internete, a tým ešte efektívnejšie využiť ich poznatky. Činnosti v tejto oblasti by sa mali odvíjať od dohodnutého projektu Schema.org. V niektorých prípadoch, ako napríklad téma fóra, by sa organizácia mohla ponechať na samotných používateľoch, ktorí by ich označovali pomocou tagov (značiek).



Obr. 2. Vizualizácia ontológií pomocou služby TouchGraph (téma víno).

Poďakovanie. Príspevok bol spracovaný v rámci výskumnej úlohy APVV-0208-10 TRA-DICE - Kognitívne cestovanie po digitálnom svete webu a knižníc s podporou personalizovaných služieb a využitím sociálnych sietí

Použitá literatúra

1. Kujawski, B., Holyst, J.A., Rodgers, G.J.: Growing Trees in Internet News Groups and Forums. [online]. In: Physical Review, 2007. [cit. 29. 2. 2012]. Dostupné na: arxiv.org/PS_cache/physics/pdf/0701/0701349v1.pdf
2. Steinerová, J., Grešková, M., Ilavská, J.: Vyhľadávanie informácií a organizácia poznania v elektronickom prostredí. Bratislava: Filozofická fakulta UK, 2010.
3. Makulová, Buzová: Manažment informačných zdrojov a knižnično-informačných služieb. [online]. Bratislava: ELET, 2011. 174 s. [cit. 15. 9. 2011]. Dostupné na: http://www.elet.sk/externe/MIZKIS_ucebnica.pdf
4. Weinberger, D.: Taxonomies and Tags: from Trees to Piles of Leaves. [online]. In: Evident, 2006 [cit. 19. 08. 2011]. Dostupné na: www.hyperorg.com/blogger/misc/ taxonomies_and_tags.html
5. Wal, T.V.: Tagging to Folksonomy. [online]. In: SlideShare, 2008. [cit. 19. 8. 2011]. Dostupné na: www.slideshare.net/vanderwal/tagging-to-folksonomy
6. Hrčková, A.: Analýza súčasného stavu a trendov v oblasti vyhľadávania informácií. (Diplomová práca). Bratislava: Filozofická fakulta UK, 2010.
7. Švihla: Ontológie. Krátky úvod. [online]. [cit. 2. 8. 2011]. Dostupné na: www.svihla.net/research/publications/slides/ontologie/msvihla_ontologie.sxi.pdf
8. Breslin et al.: Towards Semanticly- Interlinked Online Communities. [online]. Galway: Digital Enterprise Research Institute, 2004. [cit. 22. 1. 2012]. Dostupné na: sw.deri.org/2004/12/sioc/index.pdf
9. Decreane, D.: Experimental TouchGraph Visualization. [online]. In: Online Ontology Visualization, 2009. [cit. 30. 1. 2012]. Dostupné na: ontologyonline.blogspot.com/2009/01/experimental-touchgraph-visualization.html

Diskusný klub

Optimalizácia prenosu správ prostredím automobilových ad hoc sietí

Tomáš Bača

Katedra softvérových technológií, FRI,
Žilinská univerzita v Žiline, Univerzitná 1, 010 26, Žilina
tomas.baca@kst.uniza.sk

Abstrakt. Automobilová ad hoc sieť (*VANET*) môže byť chápaná ako zdroj a úložisko dát užitočných pri riadení automobilu. Článok, ktorý zhŕňa prebiehajúci výskum, opisuje komunikačný protokol optimalizujúci prenos správ pre potreby databázového systému v prostredí *VANET*. V článku je tiež opísaný simulačný nástroj *AdHocSim.FRI*, ktorý je aktívne vyvíjaný za účelom experimentálneho overenia tejto a podobných výskumných úloh.

Kľúčové slová: VANET, protokol, distribuovaný databázový systém, simulácia

1 Úvod

Jednou z hlavných výskumných oblastí Fakulty riadenia a informatiky Žilinskej univerzity je oblasť inteligentných dopravných systémov. Do tejto oblasti patrí nesporne aj využívanie automobilovej ad hoc siete (*VANET*) ako veľký zdroj užitočných informácií, ktoré v nej vznikajú, šíria a udržiavajú sa a tým prispievajú k bezpečnosti, plynulosti a pohodliu cestnej premávky.

V roku 2010 navrhol Janech vo svojej dizertačnej práci koncept distribuovaného databázového systému pre prostredie mobilných a automobilových ad hoc sietí. Neskôr aj implementoval funkčný prototyp tohto systému a nazval ho *ad-db .FRI*. Zaujímavou vlastnosťou systému je schopnosť komunikovať a získavať informácie aj v prípade, že sieť nie je celistvá. Používateľ vtedy síce získa len tú časť dát, ktorá je práve dostupná, ale na ich základe sa dokáže rozhodovať určite lepšie, ako keby získal iba správu o chybe [4,5,6].

Túto problematiku ďalej rozširuje aj Lieskovský, ktorý sa zaoberá algoritmami pre šírenie a distribúciu dát prostredím *VANET*. Pokračuje v myšlienke, zvyšovania „spokojnosti“ používateľov sprístupňovaním väčšieho množstva aktuálnejších dát aj v prípade ich neúplnosti [7,11,8].

Iným spôsobom na tieto témy nadväzujem vo vzájomnej spolupráci aj ja. Zaoberám sa komunikačným protokolom, ktorý by mal *ad-db .FRI* využívať na posielanie správ medzi databázovými uzlami. Za účelom možnosti experimentálneho overenia nášho spoločného výskumu taktiež vyvíjam vlastný simulačný nástroj nazvaný *AdHocSim.FRI* [1,3,9,10].

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 129–132.

2 Automobilová ad hoc sieť – *VANET*

Automobilová ad hoc sieť, po anglicky nazývaná *Vehicular Ad hoc Network*, skrátene označovaná ako *VANET* je bezdrôtová sieť, v ktorej uzly komunikujú na báze tzv. *peer-to-peer*, čiže bez pomoci akýchkoľvek špecializovaných sieťových zariadení a centralizovaného riadenia. Väčšina uzlov siete *VANET* je tvorená automobilmi, ktoré sa pohybujú cestnou sieťou, pričom dodržiavajú pravidlá cestnej premávky. Okrem toho môžu byť súčasťou siete aj statické uzly umiestnené napríklad v inteligentnej dopravnej značke, inteligentnej budove a pod.

Hlavnou úlohou siete *VANET* je prispieť k zvýšeniu bezpečnosti cestnej premávky distribúciou správ o bezprostredne hroziacom nebezpečenstve. Varovanie môže automobil automaticky vytvoriť, zaslať a ďalej šíriť, aby v prípade potreby mohol iný automobil toto varovanie zobraziť svojmu šoférovi. Druhoradou úlohou je zvyšovanie plynulosti cestnej premávky, napríklad asistenciou pri zachytávaní zelenej vlny alebo pri výbere optimálnej trasy vzhľadom na aktuálnu dopravnú situáciu. Okrem toho môže *VANET* napomáhať pri zabezpečovaní doplnkových služieb, ktoré môžu využiť pasažieri pre zvýšenie svojho komfortu. [12,13,14]

Komunikáciu vo *VANET* komplikuje predovšetkým rýchlosť, akou sa automobily často krát pohybujú a okrem toho má značný vplyv aj tienenie budov v mestách. Topológia siete sa preto nepretržite mení a zasielanie správy konkrétnemu uzlu na veľkú vzdialenosť je veľmi náročné.

3 Optimalizácia prenosu veľkých správ cez *VANET*

Prenosmi veľkých správ prostredím *VANET* sa väčšina výskumníkov nezaoberá, my ich však potrebujeme na zaslanie databázovej odpovede v systéme *ad-db .FRI*. Ako som preukázal vo svojej dizertačnej práci [2], klasický spôsob, teda kódovanie objektovej štruktúry niektorým zo známych kódovaní[1] a následný prenos protokolom transportnej vrstvy *TCP*, dosahuje pomerne nízku účinnosť. Jednou z príčin je spôsob jeho fungovania, ktoré najskôr vytvára spojenie a následne pokračuje „pomalým štartom“. V približne 10% prípadov pri tom odosielateľ premešká krátku príležitosť na to, aby zaslal aspoň časť správy.

Vytvoril som preto jednoduchý mechanizmus, ktorý na aplikačnej úrovni zabezpečí spätnú väzbu pre protokol *UDP* a preskočí proces vytvorenia spojenia a v porovnaní s *TCP* prenesie v priemere o 25% viac dát. V súčasnosti však mechanizmus nerieši opätovné posielanie chýbajúcich datagramov, iba náhle prerušenie spojenia. Vytvorený komunikačný protokol preto musí zvládnuť, ak sú zo správy prenášanej vo viacerých datagramoch niektoré z datagramov doručené mimo poradie, doručené viacnásobne, alebo nie sú doručené vôbec.

Aby dva uzly mohli komunikovať týmto protokolom, musia mať rovnakú znalosť o prenášanej objektovej štruktúre. Napríklad v prípade *ad-db .FRI* je podmienka splnená, ak obidva uzly pracujú s rovnakou globálnou konceptuálnou

[1] *XML*, *JSON*, *YAML*, *ASN.1 DER*, ...

schémou databázy. Z databázového dopytu preto obaja odvodia rovnakú deklaráciu objektovej štruktúry databázovej odpovede a z nej sú potom jasné typy prenášaných objektov, zoznamy ich atribútov, a teda poradie a veľkosti jednotlivých položiek v kódovanej správe. Na rozdiel od kódovania *ASN.1 DER*[2] preto tieto informácie do správy zapísané nie sú, čo zmenšuje prenášanú správu približne o jednu tretinu[3]. Pri dodržaní pevných pravidiel zápisu jednotlivých položiek objektovej štruktúry do správy a pridaní vhodných identifikátorov do každého datagramu je možné prijaté datagramy spracovať nezávisle na sebe.

4 Simulačný nástroj *AdHocSim.FRI*

Nástroj *AdHocSim.FRI* som začal vyvíjať, pretože mne ani mojim kolegom z rôznych príčin nevyhovovali simulačné nástroje, ktoré sme pred tým testovali[4]. Rozhodol som sa preto vytvoriť nástroj, s ktorým budem môcť experimentálne vyhodnocovať vlastnosti rôznych algoritmov nasadených do prostredia *VANET*.

Nástroj obsahuje viaceré samostatné modely – cestnú sieť v mestskom prostredí, cestnú premávku, šírenie rádiového signálu, protokoly *IEEE 802.11p*, *ARP*, *IPv6*, *UDP*, *TCP*, automobil s palubným počítačom a operačným systémom. Zvlášť model operačného systému je zaujímavý tým, že poskytuje API[5], ktoré umožňuje v simulácii testovať aplikácie naprogramované pre reálny počítač[6], čím sa výrazne uľahčuje testovanie zložitejších aplikácií.

Taktiež sú v nástroji zahrnuté funkcie uľahčujúce vykonávanie parametrických štúdií[7] využitím výpočtového výkonu viacerých počítačov súčasne, napríklad v počítačovej učebni.

5 Ďalšie pokračovanie výskumu

V rámci svojho ďalšieho výskumu by som rád začlenil navrhnutý komunikačný protokol do systému *ad-db .FRI* a s využitím počítačovej simulácie presnejšie vyhodnotil jeho prínos. Okrem toho považujem za zaujímavú myšlienku vyskúšať modifikáciu tohto protokolu, ktorá na úrovni transportnej vrstvy použije protokol *SCTP* namiesto *UDP*.

Okrem toho mám v úmysle pokračovať vo vývoji svojho simulačného nástroja, pričom by som v tejto oblasti chcel nadviazať spolupráci s ďalšími odborníkmi. Vylepšenia budú potrebné pre model šírenia rádiového signálu, model cestnej premávky a taktiež doplnenie smerovacích protokolov a protokolu *SCTP*.

[2] Toto kódovanie pred každú hodnotu zapíše najskôr jej typ a veľkosť.
[3] Presný pomer závisí od konkrétnej situácie.
[4] Nástroje NCTUns6.0, EstiNet7.0, QualNet, Opnet, NS2, NS3.
[5] Application Programming Interface – programovateľné rozhranie.
[6] API obsahuje volania pre prácu so sieťovými rozhraniami, časom, vláknami a zisťovanie geografickej polohy a pohybu automobilu.
[7] Experiment, pri ktorom sa skúma vplyv zmeny hodnoty jedného alebo viacerých vstupných parametrov na výsledky simulácie.

Poďakovanie. Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt: *Centrum excelentnosti pre systémy a služby inteligentnej dopravy II.,* ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

Literatúra

1. Bača, T.: Adhocsim.fri, http://adhocsim.dotfri.info/
2. Bača, T.: Optimalizácia prenosu správ v ad hoc sieťach. Ph.D. thesis, Žilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Žilina (2012), dizertačná práca
3. Bača, T., Kršák, E.: Adhocsim.fri - simulator of vanet applications. Journal of Information, Control and Management Systems 9(3 Spec. iss.), 153–157 (2011)
4. Janech, J.: Distributed database systems in the dynamic networks environment. International Journal on Informatipn Technologies and Security (1), 43–52 (2010)
5. Janech, J.: Riadenie procesov pri distribúcii databáz. Ph.D. thesis, Žilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Žilina (2010), dizertačná práca
6. Janech, J.: Distributed database systems in the dynamic networks environment. In: TRANSCOM 2011: 9-th European conference of young research and scientific workers. Žilinská univerzita, Žilina (jún 2011)
7. Janech, J., Lieskovsky, A., Kršák, E.: Comparation of strategies for data replication in vanet environment. In: Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on. pp. 575 – 580 (marec 2012)
8. Lieskovský, A.: Replikácia dát v ad-hoc sieťach. Ph.D. thesis, Žilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Žilina (2012), dizertačná práca
9. Lieskovský, A., Bača, T., Janech, J.: Comparison of data distribution in vanet urban environment. In: Digital technologies 2010: 7th international workshop on digital technologies, circuits, systems and signal processing. University of Žilina, EDIS, Žilina, Slovakia (November 2010)
10. Lieskovský, A., Janech, J., Bača, T.: Data replication in distributed database systems in vanet environment. In: ITST 2011 : 11th international conference on ITS Telecommunications. pp. 447–452. Piscataway, NJ: IEEE, St. Petersburg, Russia (august 2011)
11. Lieskovsky, A., Janech, J., Matiaško, K.: PDDA algorithm for DDBS in VANET. In: IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS). Beijin, China (jún 2012)
12. Popescu-Zeletin, R., Radusch, I., Rigani, M.A.: Vehicular-2-X Communication. Springer-Verlag, Berlin Heidelberg (2010)
13. Portál Európskej únie: Deployment of Intelligent Transport Systems in Europe (2010), http://europa.eu/legislation_summaries/transport/intelligent_transport_navigation_by_satellite/tr0010_en.htm
14. Portál Európskej únie: Europa – Glossary – Lisbon Strategy (2010), http://europa.eu/scadplus/glossary/lisbon_strategy_en.htm

Integrované prostredie pre podporu kolaboratívneho procesu modelovania politík

Peter Bednár [1], Peter Butka[1,3], Karol Furdík[2], Marián Mach[1,3], Peter Smatana[2]

[1]Ekonomická fakulta, Technická univerzita v Košiciach, Boženy Němcovej 32, 040 01 Košice, Slovenská republika
peter.bednar@tuke.sk, peter.butka@tuke.sk
[2]Intersoft, a.s., Floriánska 19, 040 01 Košice, Slovenská republika
karol.furdik@intersoft.sk, psmatana@gmail.com
[3]Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Letná 9, 042 00 Košice, Slovenská republika
marian.mach@tuke.sk

Abstrakt. Táto práca sa zaoberá návrhom a integráciou rôznych nástrojov pre podporu procesu kolaboratívneho modelovania politík v rámci projektu OCOPOMO. Tento proces kombinuje tvorbu a analýzu voľne písaných scenárov a agentových simulačných modelov. Model politík pre danú doménu sa vytvára iteratívne použitím kooperácie viacerých zainteresovaných používateľských skupín (riadiaci pracovníci, spoločnosti a firmy, analytici, neziskové organizácie, verejnosť).

Kľúčové slová: agentové systémy, modelovanie politík, simulácia, kolaborácia, architektúra, integrácia nástrojov

1 Úvod

Informačné a komunikačné technológie (IKT) sú intenzívne používané v oblasti elektronickej verejnej správy s cieľom podporiť inováciu a aktívne a kvalifikované zapojenie verejnosti do rozhodovacích procesov. Jednou z hlavných funkčností IKT riešení v tejto oblasti je poskytnúť efektívnu spoluprácu a výmenu informácií medzi zainteresovanými subjektmi (úrady, verejnosť, súkromný sektor, verejný sektor) počas iniciácie politík, ako aj ich vývoja, implementácie, monitorovania a evaluácie. Zahrnutie širšej verejnosti v procese tvorby politík je dôležité a môže znamenať pridanú hodnotu. Preto aj v rámci FP7 vznikla téma zaoberajúca sa modelovaním politík, kde je cieľom poskytnúť metodológiu a nástroje pre uskutočnenie simulácií integrujúcich všetky relevantné parametre, relácie a scenáre, ktoré sú potrené pre predpoveď možných dopadov navrhovaných politík [1][2][4][5].

Návrh softvérovej platformy a metodológie poskytujúcej prostredie pre kolaboratívne modelovanie politík je cieľom európskeho výskumno-vývojového projektu OCOPOMO (Open COllaboration for POlicy MOdelling). OCOPOMO je spolufinancované Európskou komisiou v rámci siedmeho rámcového programu (FP7). Koordi-

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 133–136.

nátorom je University of Koblenz-Landau (Nemecko), projektové konzorcium pozostáva z 10 partnerov z 5 európskych krajín. Tento 3-ročný projekt, ktorý začal v roku 2010, je testovaný na pilotných aplikáciách v Taliansku, Veľkej Británii a na Slovensku.

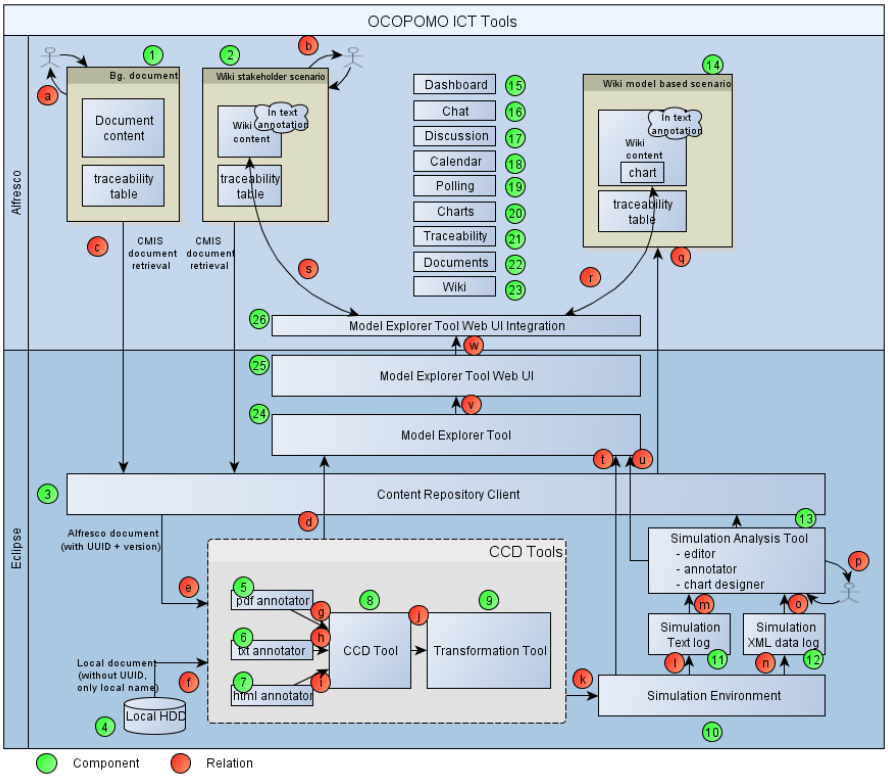
2 Platforma OCOPOMO – Systém pre podporu kolaboratívneho procesu modelovania politík

Proces modelovania politiky je v projekte OCOPOMO založený na kombinácii popisných scenárov a formálnych modelov, ktoré sú vytvárané a modifikované kolaboratívne rôznymi skupinami zainteresovaných osôb používajúcich nástroje pre komunikáciu a výmenu informácií [1][3]. Navrhnutá architektúra je popísaná na nasledujúcom obrázku 1. V uvedenej schéme sú číslami označené softvérové komponenty a písmenami sú označené väzby medzi komponentmi resp. jednotlivé kroky navrhnutého iteratívneho procesu modelovania politík. Proces je možné sumarizovať do nasledujúcich krokov:

1. Vytvorenie počiatočných scenárov
2. Anotovanie scenárov a modelovanie pomocou konceptuálneho konzistent-ného popis (Conceptual Consistent Description)
3. Generovanie štruktúry kódu simulačného modelu z CCD
4. Implementácia simulačného modelu v simulačnom prostredí
5. Spúšťanie a analýza simulácii
6. Vytvorenie simulačného scenára

Počiatočné scenáre sú písané vo voľnom alebo semi-štrukturovanom texte, ktoré popisujú znalosti spolupracujúcich osôb z danej domény. Editovanie a komentovanie počiatočných scenárov (a)(b) prebieha v kolaboratívnom prostredí (komponenty 15-23), ktoré poskytuje podporu pre zdieľanie dokumentov, wiki stránky, hlasovania, online chat a diskusné fóra. Súčasťou kolaboratívneho prostredia je aj zdieľaná plocha (dashboard), ktorá prehľadne zobrazuje všetky kolaboratívne aktivity a umožňuje rýchli prístup k často meneným artefaktom. Výsledné počiatočné scenáre sú vo forme komentovaných wiki stránok (2), ktoré môžu odkazovať na podporné dokumenty rôznych formátov (1).

Pre modelovanie politík na abstraktnej úrovni bol navrhnutý konceptuálny konzistentný popis (CCD), ktorý umožňuje špecifikovať prostredie, triedy zúčastnených agentov a pravidlá ich chovania nezávisle na používanom simulačnom prostredí. CCD plní dve hlavné úlohy a to a) počiatočný a simulačný scenár sú anotované konceptmi z CCD čo umožňuje transparentné mapovanie počiatočných znalostí doménových expertov a znalostí extrahovaných z výsledkov simulácii; a b) podporuje návrh a implementáciu simulačného prostredia, keďže je na základe CCD možné vygenerovať (komponent 9 – Transformation tool) kostru implementačných tried a pravidiel pre zvolené simulačné prostredie. Nástroje pre modelovanie CCD, anotovanie scenárov a implementovanie simulačných modelov boli spolu so simulačným prostredím integrované do jedného prostredia založeného na Eclipse platforme.



Obr. 1. Integračná architektúra platformy OCOPOMO.

Ako zvolené simulačné prostredie bol zvolený systém Repast, ktorý riadi prostredie a interakciu agentov. Samotná báza pravidiel agentov (reprezentujúcich organizácie a ľudí) a prostredia, ktoré definujú ich chovanie je riadená plavidlovým systémom DRAMS (Declarative Rule-based Agent Modelling System) vyvíjaným v projekte.

Pri vykonávaní simulácii sú zapisované logovacie záznamy, ktoré umožňujú spätne sledovať chovanie agentov a zmenu ich stavu, ktorý je reprezentovaný ako množina stavových premenných. Na základe týchto záznamov potom analytik vytvára simulačný scenár, ktorý spätne anotuje na CCD model. Zmenu stavových premenných je možné agregovať do relačnej formy, ktorú je možné priamo vizualizovať v podobe grafov ktoré porovnávajú hodnoty pri rôznych udalostiach, alebo animujú ich vývoj v čase. Grafy sú vložené priamo do simulačného scenára, ktorý je na konci procesu spätne prezentovaný zainteresovaným osobám. Okrem vizualizácie simulačných dát sú súčasťou scenára aj anotácie, ktoré umožňujú zainteresovaným osobám zistiť, ktoré časti počiatočných scenárov sú relevantné k danej časti simulačného scenára. Celý proces je iteratívny, t.j. po publikovaní simulačných scenárov môžu zúčastnené osoby upraviť a rozšíriť počiatočné scenáre, resp. komentovať simulačné scenáre čo vedie ku zmenám v CCD a implementácii simulačného modelu, resp. k dodatočným simuláciám.

3 Záver

Prezentovaný návrh a integrácia IKT nástrojov má za cieľ podporiť proces modelovania politík identifikovaný v rámci projektu OCOPOMO. Pre túto potrebu boli špecifikované a podporené procesy kolaboratívnej tvorby scenárov budúceho vývoja, anotácie scenárov, konštrukcie formálnych agentových modelov politiky, iteratívnych simulácií a ich evaluácie, a to všetko použitím jednej integrovanej OCOPOMO platformy. V tomto príspevku sme predstavili architektúru, jednotlivé nástroje a niektoré integračné detaily tejto platformy, ktoré by mali naplniť pôvodné zámery projektu. Navrhnutá platforma bola implementovaná a je aktuálne testovaná v rámci pilotných aplikácií v priebehu tohto roku (2012). Systém je testovaný na troch rôznych pilotných miestach - Slovensko (Košický samosprávny kraj - politika využívania obnoviteľných zdrojov energie), Taliansko (región Campania - optimálna alokácia štrukturálnych zdrojov EU) a Veľká Británia (Londýn - politika výstavby bytov a domov v oblasti Londýna). Ďalšie informácie o projekte OCOPOMO je možné nájsť na http://www.ocopomo.eu.

Acknowledgment. The OCOPOMO project is co-funded by the European Commission within the 7th Framework Programme, theme ICT-2009.7.3, contract No. 248128.

Referencie

1. Bicking M, Wimmer MA. A Scenario-based Approach Towards Open Collaboration for Policy Modelling. *Electronic Government (EGOV 2011)*, Springer, Berlin, pp. 223-234, 2011.
2. Dignum F, Dignum V, Jonker CM. Towards Agents for Policy Making. *9th International Workshop on Multi-Agent-Based Simulation*, LNAI 5269, Springer, pp.141-153, 2008.
3. Janssen M, Van der Duin P, Wimmer MA. Framework and Methodology: Methodology for scenario building. *Roadmapping eGovernment Research: Visions and Measures towards Innovative Governments in 2020*, pp.23-28, 2007.
4. Moreno-Jimenez JM, Polasek W. E-Cognocracy and the Participation of Immigrants in EGovernance. *TED Conference on e-Government Electronic Democracy: The Challenge Ahead, Schriftenreihe Informatik 13*, pp.18-26, 2005.
5. Nowak A, Szamrej J, Latane B. From private attitude to public opinion: A dynamic theory of social impact. *Psychological Review, vol. 97*, pp. 362-376, 1990.

Metódy spracovania a ich algoritmy pre rozsiahle dáta

Peter Drábik, Petr Šaloun

Fakulta elektrotechniky a informatiky, VŠB-Technická univerzita Ostrava
17. listopadu 15/2172, 708 33 Ostrava - Poruba, Česká republika
peter.drabik.st@vsb.cz, petr.saloun@vsb.cz

Abstrakt. V práci prezentujeme rozhodovacie stromy, ktoré sú obľúbené pre ich jednoduchosť a ich reprezentácia je ľahko pochopiteľná pre človeka. Taktiež ľahko zvládajú veľa dimenzií. Ako preukázali viaceré experimenty náhodné rozhodovacie lesy majú viacero výhod pri použití u viacdimenzionálnych dát oproti urýchleným rozhodovacím stromom. Cieľom práce je popísať rozhodovacie stromy, ich algoritmy, náhodné rozhodovacie lesy, ktoré sa ukazujú ako veľmi perspektívne pri riešení astronomických problémov a urobiť stručný prehľad implementácií v nástrojoch pre strojové učenie a získavanie dát.

Kľúčové slová: rozhodovacie stromy, prediktívne získavanie dát, náhodné rozhodovacie lesy

1 Úvod

Manipulácia, spracovanie a modelovanie veľkého množstva dát predstavuje pre IT veľkú výzvu a tvorí dôležitú časť výpočetných problémov v astronómii. Základné požiadavky zviazané s veľkým množstvom dát môžu byť zhrnuté do dvoch položiek:

1. potreba "federácie" exerimentálnych dát, prostredníctvom ich získania zo svetových archívov a definovaním série noriem pre ich formáty a prístupové protokoly.
2. implementácia inovatívnych nástrojov pre získavanie dát a znalostí, ich príjemné používateľské rozhranie, ich škálovateľnosť a aby boli čo možno najviac založené na asynchrónnych mechanizmoch.

S každou novou technológiou sa zväčšuje parametrový priestor alebo sa umožňuje lepšie vzorkovanie. Z tohto dôvodu vedecké využitie multi-pásma (D pásma), multi-epochy (K epochy) univerza vedie k hľadaniu vzorcov a trendov medzi N bodmi DxK dimenzionálneho priestoru, kde $N > 10^9$, $D >> 100$, $K > 10$ [8]. Preto sú kladené čoraz väčšie nároky na efektívne spracovávanie vysoko dimenzionálnych dát. Získavanie dát používajúce označené údaje sa nazýva učenie s učiteľom. Ak je určený atribút kategorický, úloha sa nazýva klasifikácia, ak je číselný, úlohou je tzv. regresia [2].

Klasifikácia je procesom hľadania modelu (alebo funkcie), ktorá popisuje a rozlišuje datové triedy alebo koncepty. Odvodené modely sú založené na analýze trénova-

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 137–140.

cích dát (tj dátových objektov, ktorých označenie je známe). Tento odvodený model môže byť reprezentovaný *kasifikačnými (IF-THEN) pravidlami, rozhodovacími stromami, matematickými formulami a neurónovými sieťami.* Rozhodovací strom (DT) je stromová štruktúra podobajúca sa vývojovému diagramu, kde každý uzol označuje test nad hodnotou atribútu, každá vetva výsledok testu a listy reprezentujú triedy alebo ich distribúciu. DT sú ľahko previediteľné na klasifikačné pravidlá [1].

2 Konštrukcia prediktívneho modelu

Na učenie DT používame algoritmus nazývaný *stromový induktor*. Tieto induktory konštruujú automaticky DT z daného datasetu. Obvykle je cieľom nájsť optimálny DT, pomocou minimalizácie generalizačnej chyby[3]. Ako príklad môžeme uviesť nasledovné algoritmy: ID3, C4.5, CART, CHAID, RainForest, BOAT a ďalšie.

ID3, C4.5, a CART prijali "*chamtivý*" prístup, podľa ktorého sú DT konštruované rekurzívne zhora - dole spôsobom rozdeľuj a panuj. Začínajú s tréningovou množinou n-tíc, ktorým je priradená trieda. Tá sa postupne delí na podmnožiny [1].

RainForest - rozdeľuje výsledky výberových metód v zmenšenej verzii oproti pôvodnej, bez zmeny výsledku. Podporované sú iba univarietné rozdelenia. Induktor ošetruje aj prípad, keď sa AVC skupina (Attribute-Value, Classlabel) nevtesná do pamäte.

BOAT - používa štatistické techniky známe ako “bootstrapping”, aby vytvoril menšie podmožiny z trénovacích dát, z ktorých sa každá do pamäte vmestí. Každá podmnožina je použitá na vytvorenie stromu, čo vedie k vytvoreniu niekoľkých stromov. Keď sa trénovacie dáta nevmestia do pamäte, použije podmožinu, ktorá sa do nej vmestí. Zväčša potrebuje len dva prechody trénovacou množinou s n-ticami priradených k jednotlivým triedam. Je to výhoda, pretože tradičné induktory potrebujú jeden prechod na každú úroveň stromu.Je 2-3x rýchlejší ako RainForest [1].

3 Experimenty s rozhodovacími stromami

Experimenty v [6] nad datasetmi s počtom atribútov v približnom rozsahu od 750 do 700K preukázali, že náhodné rozhodovacie lesy sa lepšie správali pri veľmi vysokých rozmeroch a že je ľahké paralelizovať váhy efektívne k vysokým rozmerom. Výsledky potvrdili experimenty v [10], kde zrýchlené DT boli efektívnejšie, keď bola dimenzionalita nízka. Táto štúdia odporúča voľbu zrýchlených DT až do 4000 dimenzií. Nad týmto počtom majú RDF najlepší celkový výkon. Táto metóda bola úspešne použitá pri riešení astronomických problémov [4, 5]. V tej istej aplikačnej doméne plánujeme nasadenie aj my.

4 Náhodné rozhodovacie lesy – RDF

RDF boli predstavené Ho (1995) [7]. Metóda RDF je založená na zbierke rozhodovacích stromov, postavenými s niektorými prvkami náhodného výberu. Táto

metóda potrebuje trénovací dataset. V RDF nie je prítomné žiadne orezávanie stromov. Namiesto toho, RDF vytvoria sadu veľmi nekorelovaných stromov, a kombinujú ich výsledky, aby moholi vytvoriť generalizovaný prediktor. RF sú kombináciou stromových prediktorov takých, že každý strom závisí na hodnotách nezávisle vzorkovaného náhodného vektora a majú rovnakú distribúciu pre všetky stromy v lese. Generalizačná chyba pre lesy konverguje tak veľmi k limite, ako sa počet stromov v lese zväčší. Chyba klasifikátorov stromu v lese závisí na sile jednotlivých stromov a na korelácii medzi nimi. Použitím náhodného výberu funkcií k rozdeleniu každého uzlu ustúpi chybovosť, ktorá obstojí aj v porovnaní s AdaBoost[1], ale je viac odolná vzhľadom na šum[9].

5 Software s implementovaným RDF

- Random Forests®[2] - Random Forests(tm) je obchodná značka Lea Breiman a Adele Cutler a je licencovaná exkluzívne pre Salford Systems .
- Waffles[3] - snaží sa byť svetovo najrozsiahlejšia zbierka nástrojov príkazového riadku pre strojové učenie a dolovanie dát. Waffles je implementované v C +.
- R[4] - je prostredie určené pre štatistické výpočty a grafiku. Obsahuje balíček randomForest. Poskytuje R rozhranie k originálnym programom vo Fortrane.
- Rf-ace[5] - je efektívna implementácia robustného algoritmu strojového učenia. Rf-ace je napísané v C++. Obsahuje efektívnu implementáciu RDF.
- Weka 3[6] - Je open-source zbierka algoritmov strojového učenia, vytvorených na University of Waikato na Novom Zélande. Program je napísaný v Jave.

V nasledujúcej tabuľke viackrát testujeme voľne dostupné UCI[7] datasety, ktoré najskôr zamiešame a potom rozdelíme na trénovaciu časť 70 % a testovaciu 30 %.

Tab. 1. Nameraná miera chybovosti nami testovaných nástrojov.

	R [%]	Rf-ace [%]	Waffles [%]
ionosphere	8,6%	10,4%	14,3%
sonar	12,7%	14,3%	14,3%
iris	4,4%	2,2%	2,2%
zoo	9,7%	10,1%	6,4%
segment	2,7%	4,8%	1,4%

[1] AdaBoost nemá žiadne náhodné prvky a súbor stromov rastie pri po sebe idúcich vyvažovaniach z trénovacieho datasetu, kde súčasné váhy závisia na predchádzajúcej formácii súboru.
[2] Komerčný softvér . Dostupný na http://www.salford-systems.com/en/
[3] Voľný softvér . Dostupný na http://sourceforge.net/projects/waffles/files/
[4] Voľný softvér . Dostupný na http://cran.r-project.org/
[5] Voľný softvér . Dostupný na http://code.google.com/p/rf-ace/downloads/list
[6] Voľný softvér . Dostupný na http://www.cs.waikato.ac.nz/ml/weka/
[7] Dostupné na http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html

6 Záver

Ako bolo dokázané v[6], RDF pracujú lepšie s vysoko dimenzionálnymi dátami, ktoré sa pomerne často vyskytujú v astronomickej problematike. Z tohto dôvodu sa dá odporúčiť implementácia RDF do systémov určených pre získavanie dát a ich výskumy ako metódu učenia s učiteľom pre klasifikáciu. Ako príklad môžme uviesť DAME[8](Data Mining & Exploration), ktoré má distribuovanú infraštruktúru získavania dát zameranú na výskumy v obrovských datasetoch s metódami strojového učenia. Naším ďalším cieľom je porovnanie efektívnosti nástrojov pracujúcich s RDF oproti už implementovaným klasifikačným nástrojom v systéme DAME. O to sa pokúsime aj pre konkrétny prípad fotometrických červených posunov[9] dovoľujúcich nám určiť vzdialenosť väčšiny kvazarov[11].

Poďakovanie. Práca bola čiastočne podporená projektom FRVŠ 2012/347 "Adaptivní webové systémy" a centralizvoaným rozvojovým projektom MŠMT "Elektronická podpora tvůrčí výuky v oblasti IT a vyhledávání talentů ".

Literatúra

1. J. Han and M. Kamber. Data mining: concepts and techniques. The Morgan Kaufmann series in data management systems. Elsevier, 2006.
2. M. Bramer. Principles of Data Mining (Undergraduate Topics in Computer Science).Springer-Verlag New York, Inc., 2007.
3. J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest - a framework for fast decision tree construction of large datasets. Data Min. Knowl. Discov., pp. 151-169, 2000.
4. J. Albert. Implementation of the random forest method for the imaging atmospheric cherenkov telescope magic. Technical report, September 2007.
5. L. Breiman, M. Last, and J. Rice. Random forests: finding quasars. Statistical Challenges in Astronomy, pp. 243-254, 2003.
6. R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. ICML, pp. 96-103, 2008.
7. T. K. Ho. Random decision forests. In Document Analysis and Recognition, Proceedings of the Third International Conference on Document Analysis and Recognition, volume 1, strany 278 –282 vol.1, august 1995.
8. S. Cavuoti, M. Brescia, G. Longo. Data mining and Knowledge Discovery Resources for Astronomy in the Web 2.0 Age. Conference 8451 Software and Cyberinfrastructure for Astronomy II, júl 2012.
9. L. Breiman. RANDOM FORESTS-RANDOM FEATURES. Technical Report, september 1999.
10. R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. ICML, pp. 161-168, 2006.
11. P. Škoda. Astroinformatika, Virtuální observatoř a petabytové přehlídky oblohy. Datakon 2010, tutoriály. pp. 49-74. ISBN 978-80-7368-425-9

[8] Stránky projektu DAME - http://dame.dsf.unina.it/
[9] Dáta sú dostupné na http://dame.dsf.unina.it/dame_photoz.html

Konzistencia distribuovaného riešenia

Stanislav Dvorščák[1], Kristína Machová[2]

Fakulta elektrotechniky a informatiky,
Technická univerzita v Košiciach
stanislav-dvorscak@solumiss.eu[1], kristina.machova@tuke.sk[2]

Abstrakt. Článok naznačuje spôsob riešenia transakcií v prostredí distribuovaných systémov sústreďujúc sa na ľahkú integráciu s existujúcimi technológiami. Pri návrhu riešenia bolo prihliadnuté na problémy súvisiace s použitím Web služieb (WS - Web Services), ako aj prihliadnutie na problémy súvisiace z bezstavovými tzv. REST službami. Transakčný kontrolór resp. manažér bol analyzovaný z pohľadu výhod a nevýhod centralizovaného resp. decentralizovaného umiestnenia. Za analýzu a návrh boli stanovené hlavne prostredia založené na Java technológiách s prihliadnutím na možnosť prenosu návrhu aj na iné prostredia.

Kľúčové slová: X/Open XA, Java Transaction API (JTA), Java Transaction Service (JTS)

1 Úvod

Distribuované systémy sa používajú za rôznymi účelmi akými sú napríklad: škálovateľnosť, dostupnosť, zapuzdrenie a iné.

V rámci projektu, v ktorom je tvorená masívna sémantická sieť, vznikla potreba škálovania tejto sémantickej siete naprieč väčšiemu počtu uzlov. Bez tohto škálovania by inferencia masívnej siete nebola možná. Každý uzol reprezentuje oddelenú jednotku, ktorá môže byť distribuovaná naprieč cloudom reprezentovaného vo forme n-virtuálnych počítačov. V danom prípade teda hovoríme o distribuovanom systéme.

2 Analýza

Okrem riešenia hore spomenutých problémov, distribuované riešenia zavádzajú vyššiu komplexitu. Jedným z problémov ktorý je potreba riešiť je zabezpečenie konzistencie dát. Systém počas vykonávania jednotlivých operácií môže dočasne prejsť do nekonzistentného stavu, ale v konečnom dôsledku nový stav musí byť naprieč celým systémom opäť konzistentný. Inak sa celý systém musí vrátiť do stavu, v ktorom bol pred začatím transakcie.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 141–144.

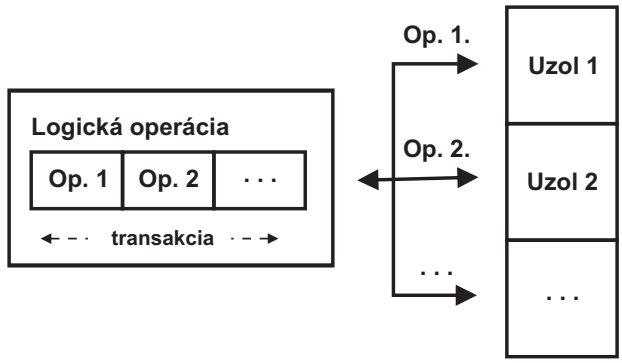
2.1 Uzol

Jednotlivé časti distribuovaných systémov sa niekedy označujú ako uzly. Pod týmto pojmom si je možné predstaviť napríklad fyzický počítač, virtuálny počítač, OSGi kontajner a pod. V skutočnosti sa jedná o elementárnu jednotku poskytujúcu funkcionalitu. Pričom tieto uzly sú v určitom pohľade nezávislé (hardwerová, virtuálna nezávislosť a pod). Každý uzol je definovaný vnútorným stavom viď obrázok 1., pričom ten sa mení pod vplyvom interakcií jednotlivých uzlov. Tento vnútorný stav môže byť perzistentný alebo tranzientný.

Obr. 1. Uzol distribuovaného systému.

2.2 Distribuovaná transakcia

Každý uzol je možné definovať jeho vnútorným stavom. Čo je však dôležité, zmenu tohto stavu spravidla nedefinujeme vzhľadom na jednotkovú interakciu - jednu operáciu. Ale je definovaná v súbehu zmien stavov viacerých uzlov vyvolaných logickou operáciou viď obrázok 2. Táto logická operácia nie je nič iné ako zloženie viacerých operácií. Ak je na túto logickú operáciu kladený dôraz na atomicitu - dvoch a viacerých uzlov, tak vtedy pojednávame o distribuovanej transakcii. Pod pojmom atomicita sa rozumie, že buď zmena stavu prebehne na všetkých uzloch úspešne alebo vôbec.



Obr. 2. Distribuovaná transakcia.

2.3 Existujúce technológie

Vo svete Java technológií existujú dva štandardy (tretí nie je Java špecifický), ktoré sa venujú transakciám a nimi sú: Java Transaction API - JSR907 (JTA) a Java Transaction Service (JTS).

Java Transaction API definuje architektúru a to hlavne to, ako sa jednotlivé časti transakcií navzájom interagujú, hlavne vo forme vysoko-úrovňového popisu realizovaného prostredníctvom rozhraní.

Tie popisujú interakciu aplikácie a transakčného manažéra a ďalej popisujú interakciu aplikačného servera a transakčného manažéra. Rovnako definuje mapovanie vzhľadom na X/Open XA [1].

Java Transaction Service definuje spôsob implementácie transakčného manažéra a mapovanie vzhľadom na OTS (Object Transaction Service). Okrem toho definuje aj IIOP (Internet InterORB Protocol), ktorý umožňuje použitie transakcií prostredníctvom RMI/IIOP (Remote Method Invocation). Dá sa povedať, že nám to umožňuje používať transakčný manažér na diaľku [2].

Non-Java prostredia X/Open XA ale aj OTS sú technológie nezávisle od prostrediach založených na technológiách Java (definované v rámci špecifikácií CORBA). V prípade X/Open XA pojednávame o XA zdrojoch, ktoré sú implementované väčšinou dnešných databáz, ale aj úložiskami dokumentov, finančnými systémami a pod. Na druhej strane OTS technológia je prenesená aj do iných jazykov napr. C resp. C++.

3 Návrh distribuovaných transakcií

Pri práci s transakciami v distribuovanom prostredí máme dve možnosti. Jednou z nich je propagácia transakčného manažéra na jednotlivé uzly. A následné pridanie jednotlivého XA zdroja do takejto transakcie. Alebo zapuzdrenie transakcie uzla do formy XA zdroja.

3.1 Propagácia transakčného manažéra

Pod pojmom propagácie transakčného manažéra, pojednávame o sprístupnení transakčného manažéra ostatným uzlom tak, aby fyzicky existoval len jeden transakčný manažér. Tu je ale potreba uvedomenia si niekoľkých faktov. JTA špecifikácia striktne nevyžaduje, aby transakcia transakčného manažéra mohla byť použitá viacerými vláknami. Iba niektoré z nich majú túto vlastnosť. A po druhé, XA zdroj môže byť použitý v tej istej chvíli v nanajvýš jednom transakčnom kontexte [3].

Z týchto dvoch faktov vyplýva potreba synchronizácie práce s transakčným manažérom. To je čiastočne podporované RMI (Remote Method Invocation) vlastnosťami.

3.2 Zapuzdrenie uzla do formy XA zdroja

Alternatívnym riešením je zapuzdrenie celého uzla tak, aby sa stal XA zdrojom. V takomto prípade môže byť transakčný manažér úplne oddelený od architektúry. A jednotlivé uzly sú pridávané ako XA zdroje do tohto transakčného manažéra. Prípadne každý uzol má vlastného transakčného manažéra. Uzol ktorý štartuje globálnu transakciu použije svoj vlastný transakčný manažér na pridanie jednotlivých XA zdrojov. Ak samotný uzol spolupracuje s ďalšími XA zdrojmi, je potreba zabezpečiť pridanie týchto XA zdrojov takisto do transakcie, alebo je možné schovať tieto XA zdroje v rámci XA zdroja reprezentovaného uzlom. V druhom prípade takto zapuzdrený XA zdroj, nesmie byť propagovaný k ostatným uzlom. To jest tento XA zdroj je iba vnútorným a preto môže byť schovaný za fasádu XA zdroja uzla. Pod pojmom XA zdroj - rozumieme spojenie, ktoré má priradené XA zdroj, tak ako to je definované v [1].

Nespornou výhodou je jednoduchosť rozhrania XA zdroja, ktorý je relatívne jednoducho reprezentovateľný vo forme REST služieb.

4 Záver

V praxi sa softvérový architekti stretávajú pri integrácií s neexistenciou podpori transakcií. To má za následok problematickosť integrácie netransakčných systémov, do celkového transakčného riešenia. Transakčnosť systému môže mať aj negatívne vlastnosti. A však rozhodnutie toho, či systém má, alebo nemá byť transakčný, by malo byť ponechané na rozhodnutí systému do ktorého sa takýto systém vkladá, napr. vo forme transakčného a netransakčného rozhrania.

V rámci nami tvorenej sémantickej siete sú kladené zvýšené požiadavky na distribúciu výpočtov naprieč veľkému počtu uzlov. Pričom je nežiadúce, aby inferencia sémantickej siete naprieč viacerými uzlami viedla k nekonzistencii počas neočakávanej situácii. A to z toho dôvodu, že inferencia je v našom prípade kontinuálna a má vplyv na samotnú sémantickú sieť. Každý uzol spracúva časť sémantickej siete. A každý tento uzol sa tvári ako XA zdroj. To jest volili sme riešenie vo forme zapuzdrenia uzla do formy XA zdroja. Toto riešenie má niekoľko výhod: možnosť ľahšieho použitia systému v heterogénnych (non-Java prostrediach), možnosť reprezentácie takéhoto zdroja vo forme REST služieb. Každý uzol má vlastný transakčný manažér, čiže pád transakčného manažéra nemá vplyv na funkcionalitu celého systému.

Literatúra

1. Susan Cheung, Vlada Matena, Java Transaction API (JTA), Sun Microsystems Inc. **1.1** (November, 2002)
2. Susan Cheung, Java Transaction Service (JTS), Sun Microsystems Inc. **1.0** (December, 1999)
3. X/Open Company Ltd., Distributed Transaction Processing: The XA Specification, X/Open CAE Specification (December, 1991)
4. Fintan Bolton, PURE CORBA, Sams Publishing, (Júl, 2001)

Distribuovaný databázový systém AD-DB .FRI

Ján Janech

Katedra softvérových technológií, FRI, Žilinská univerzita v Žiline
Univerzitná 8215/1, 010 26 Žilina
jan.janech@kst.uniza.sk

Abstrakt. Práca sa venuje problematike zavedenia distribuovaných databázových systémov v prostredí VANET. Popisuje súčasný stav riešenia a identifikuje jeho hlavné nedostatky. Ako riešenie problémov navrhuje využitie vlastného systému AD-DB .FRI.

Kľúčové slová: VANET, distribuovaný databázový systém, AD-DB .FRI

1 Úvod

Požiadavky na distribuované databázové systémy sa časom menia. To, čo bolo pred tridsiatimi rokmi nemožné, sa dnes pomaly stáva skutočnosťou. Sieťové architektúry sa menia, čoraz viac sa používajú bezdrôtové siete. To so sebou prináša komplikácie v podobe pomalšej komunikácie, nižšej bezpečnosti, väčšej pravdepodobnosti výpadku spojenia atď…

Najväčšiu výzvu však predstavujú bezdrôtové siete typu ad-hoc, hlavne MANET (Mobile Ad-hoc NETwork) a VANET (Vehicular Ad-hoc NETwork). Pri tomto type siete nie je jednoduché (niekedy dokonca nie je možné) získať informácie o štruktúre siete, o jej účastníkoch, hierarchii a podobne. Klient požadujúci službu nevie, či mu ju je schopný niekto poskytnúť, alebo nie.

Ako som už spomínal, špeciálnym prípadom sú mobilné ad-hoc siete. MANET je autonómny systém mobilných uzlov [2]. Systém môže operovať izolovane, alebo môže mať rozhranie s pevnou komunikačnou sieťou [2]. Vzhľadom na mobilitu uzlov sa sieť neustále reorganizuje, s čím musia počítať všetky systémy nasadené do tohto prostredia. Špecifikum automobilových ad-hoc sietí oproti mobilným ad-hoc sieťam je hlavne spôsob pohybu uzlov [3, 17]. Kým v MANET sa jedná o pohyb viac-menej náhodný (uzol sa môže v ktoromkoľvek okamihu rozhodnúť o zmene smeru), vo VANET je pohyb uzlov obmedzený cestnou infraštruktúrou. Rozdiel je aj v rýchlosti. V MANET sa predpokladá relatívne pomalý pohyb uzlov. Automobily sa však pohybujú vysokými rýchlosťami.

Cieľom výskumu popísaného v tomto článku je umožniť použitie distribuovaného databázového systému aj v takomto dynamickom systéme. Vďaka tomu by bolo možné distribuovať na jednotlivé uzly siete rôzne informácie o stave premávky, mapách oblasti, voľných parkoviskách a podobne. Zdrojom týchto dát by mohli byť ostatné uzly databázy, senzory auta, alebo rozpoznaný obraz z kamery[18].

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 145–148.

2 Súčasný stav riešenia problematiky

Mimo mnou navrhnutého systému AD-DB .FRI neexistuje žiadne riešenie pre oblasť VANET. V histórii sa však vyskytlo niekoľko pokusov o vytvorenie špecializovaného systému pre oblasť MANET.

Hlavným zástupcom takýchto systémov je protokol TriM [4]. Ten je navrhnutý hlavne s ohľadom na energetickú spotrebu. Snaží sa teda minimalizovať čas strávený komunikáciou. To je aj dôvod, prečo nemá zmysel nasadzovať tento protokol do prostredia VANET. V každom automobile je k dispozícii kvalitný zdroj energie a preto nemá zmysel jej prílišné šetrenie. Naopak kvôli veľkej mobilite uzlov a z toho vyplývajúcim rapídnym zmenám topológie siete je vyžadovaná častejšia komunikácia ako je tomu v prípade siete MANET. Ďalšou veľkou nevýhodou protokolu TriM je, že vyžaduje, aby v systéme existovali uzly (tzv. LMH – Large Mobile Host), ktoré majú k dispozícii celý obsah databázy[5].

Niektoré nedostatky protokolu TriM sa snaží odstrániť protokol HDD3M. Umožňuje napríklad, aby boli dáta fragmentované po celej sieti. Vyžaduje však, aby v sieti boli prístupné uzly (DD – Database Directory), ktoré obsahujú informácie o rozložení všetkých fragmentov v sieti. To je nemožné dosiahnuť v sieti VANET, kvôli jej rapídnym zmenám a jej rozsiahlosti. Okrem toho sa, rovnako ako protokol TriM, zameriava hlavne na energetickú spotrebu.

Ako je vidieť, systémy navrhnuté pre použitie v sieti MANET vychádzajú z odlišných predpokladov a požiadaviek a preto je nemožné použiť ich v sieti VANET.

3 Riešenie

Riešením by bolo sprístupniť v distribuovanom databázovom systéme iba tú časť dát, ktorá je priamo prístupná danému uzlu v danom čase. Možností, ako to dosiahnuť je niekoľko, ich popis by však bol nad rámec tohto článku. Je možné ich nájsť v rôznych publikáciách [6, 7, 8] a mojej dizertačnej práci [9].

Finálny návrh architektúry AD-DB .FRI a jeho algoritmov na spracovanie distribuovaných dotazov vyplýva z vlastností VANET. Vzhľadom na to, že v tejto sieti uzly nemajú vedomosti o uzloch v svojom okolí, komunikácia sa vždy iniciuje pomocou broadcast správy. Správa obsahuje informáciu o dotaze, ktorý treba vykonať. Ak niektorý z uzlov, ktoré zachytia správu, dokáže spracovať aspoň časť tohto dotazu, spracuje ho a pošle výsledok dotazovaciemu uzlu vo forme unicast správy.

4 Vyhodnotenie riešenia

Experimentálne overenie riešenia by bolo komplikované vzhľadom na to, že hardvérové zariadenia určené pre prácu v systéme VANET existujú momentálne len vo forme prototypov s pomerne veľkou obstarávacou cenou. Hlavným nástrojom overovania technológií pre VANET siete je preto simulácia.

Na simuláciu som využil nástroj NCTUns 6.0 schopný simulovať aj sieť VANET [10]. Jedná sa o simulačný nástroj založený na princípoch diskrétnej udalostnej simulácie. Výhodou je aj to, že je možné do simulácie zaradiť reálnu aplikáciu napísanú v jazyku C/C++ [11].

Pri ďalšom výskume som sa však zameral na riešenie AdHocSim.FRI[12, 16] vyvíjané na našej fakulte. Veľkou výhodou bolo urýchlenie simulácie, ktoré mi umožnilo vykonávať veľa pokusov a otestovať tak systém AD-DB .FRI v rôznych situáciach. Výsledky týchto simulácií som spolu s kolegami publikoval na rôznych medzinárodných konferenciách [13, 14, 15].

5 Záver

V článku bol v krátkosti predstavený nástroj AD-DB .FRI, ktorý umožňuje použitie distribuovaného databázového systému v prostredí VANET. Navrhnutý systém bol implementovaný vo forme prototypu[1] a otestovaný pomocou simulácie. Na môj výskum nadväzujú výskumné témy Tomáša Baču (prenos dát vo VANET) a Antona Lieskovského (replikácia dát vo VANET), ktorý tento rok úspešne obhájili dizertačné práce v oblasti.

Poďakovanie. Táto štúdia/publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt: *Centrum excelentnosti pre systémy a služby inteligentnej dopravy II.*, ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja. "Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

1. Oszu, M. Tamer – Valduriez, Patrick: Principles of Distributed Database Systems. Prvé vyd. [s.l.]. Prentice Hall. 1991. ISBN 0-13-691643-0.
2. Corson, S. – Macker, J.: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. [s.l.]. IETF. jan 1999. Request for Comments; no 2501. Dostupné online: <http://www.ietf.org/rfc/rfc2501.txt>.
3. Jerbi, M. – Senouci, S.-M. – Meraihi, R. – Ghamri-Doudane, Y.: An Improved Vehicular Ad Hoc Routing Protocol for City Environments. In: Communications, 2007. ICC '07. IEEE International Conference on. jún 2007. ISBN 1-4244-0353-7 .
4. Fife, Leslie D. – Gruenwald, Le: TriM: Tri-Modal Data Communication in Mobile Ad-Hoc Networks. In: Lecture Notes in Computer Science. [s.l.]. Springer Berlin / Heidelberg. 2004. vol.3180. ISSN 0302-9743.
5. Rahbar, Afsaneh – Mohsenzadeh, Mehran – Rahmani, Amir Masoud: HDD3M: A New Data Communication Protocol for Heterogeneous Distributed Mobile Databases in Mobile Ad Hoc Networks. In: ICETC'09: Proceedings of the 2009 International Conference on Education Technology and Computer. Washington, DC, USA : IEEE Computer Society, 2009. ISBN 978-0-7695-3609-5.

[1] Prototyp je možné využiť na výskumné účely a je ho možné získať po osobnej dohode na email adrese uvedenej na začiatku článku.

6. Janech, Ján – Bača, Tomáš – Tavač, Marek: Distribuovaný databázový systém v prostredí ad-hoc sietí. In: Datakon 2010. Ostrava : Ostravská univerzita, október 2010. ISBN 978-80-7368-424-2.
7. Janech, Ján: Distributed Database Systems in the Dynamic Networks Environment. In: International Journal on Information Technologies and Security. Sofia : Union of Scientists in Bulgaria, február 2010. n° 1. ISSN 1313-8251.
8. Janech, Ján – Bača, Tomáš: Distributed Database Systems in the Dynamic Networks Environment. In: Communications. Žilina : Žilinská Univerzita, november 2011. vol. 13, n° 4. ISSN 1335-4205.
9. Janech, Ján: Riadenie procesov pri distribúcii databáz. Žilina : Žilinská univerzita v Žiline, Fakulta riadenia a informatiky, september 2010. dizertačná práca.
10. Wang, S.-Y., Lin, C.-C.: NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches. IEEE 68th. Vehicular Technology Conference (2008)
11. Wang, S.-Y., Chou, C.-L.: Ad Hoc Networks: New Research, (2009), kapitola NCTUns simulator for wireless vehicular ad hoc network research.
12. Bača, Tomáš: Benefits of coroutines in discrete-event simulation. In: OBJEKTY 2011: Proceedings of the 16th international conference on object-oriented technologies. Žilina : EDIS -- publisher of the University of Žilina, november 2011. ISBN 978-80-554-0432-5.
13. Lieskovský, Anton – Janech, Ján – Bača, Tomáš: Data replication in distributed database systems in VANET environment. In: ICSESS 2011 : proceedings 2011 IEEE 2nd international conference on software engineering and service science. Beijing, China : Piscataway, NJ: IEEE, júl 2011. ISBN 978-1-4244-9696-9.
14. Lieskovský, Anton – Janech, Ján – Bača, Tomáš: Data replication in distributed database systems in VANET environment. In: ITST 2011: 11th international conference on ITS Telecommunications. St. Petersburg, Russia : Piscataway, NJ: IEEE, august 2011. ISBN 978-1-61284-670-5.
15. Janech, Ján – Lieskovský, Anton – Kršák, Emil: Comparation of Strategies for Data Replication in VANET Environment. In: Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on. Fukuoka, Japan : Fukuoka Institute of Technology, march 2012. ISBN 978-0-7695-4652-0.
16. Bača, Tomáš – Kršák, Emil: AdHocSim.FRI - simulator of vanet applications. In: Journal of Information, Control and Management Systems. vol. 9, no. 3. 2011. ISSN 1336-1716.
17. Lieskovský, Anton – Bača, Tomáš: Dissemination of safety data in vehicular Ad hoc network. In: QUAERE 2011: recenzovaný sborník příspěvků interdisciplinární mezinárodní vědecké konference doktorandů. 2011. ISBN 978-80-904877-3-4.
18. Kršák, Emil – Toth, Štefan: Traffic sign recognition and localization for databases of traffic signs. In: Acta Electrotechnica et Informatica. vol. 11, no. 4. 2011. ISSN 1335-8243.

Optimalizácia personalizovaného odporúčania skupinám a jednotlivcom

Michal Kompan, Mária Bieliková

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave, Ilkovičova 3, 842 16 Bratislava
{kompan,bielik}@fiit.stuba.sk

Abstrakt. Prístupy pre personalizované odporúčanie sú zamerané na poskytnutie najlepších prvkov (najvyššie predikované hodnotenie). Často je však z pohľadu spokojnosti používateľov klasické zoradenie prvkov od najlepšieho k horším nepostačujúce. V tomto príspevku prezentujeme možnosti preusporiadania prvkov navrhnutých pre odporúčanie, s cieľom dosiahnuť maximálnu spokojnosť používateľa po sekvencii odporúčaných prvkov. Pri odporúčaní skupinám dostáva úloha preusporiadania ďalší rozmer – je nevyhnutné zohľadniť aj ostatných používateľov. Navrhnuté riešenie je možné využiť tak pri odporúčaní skupinám či jednotlivcom bez nutnosti zmeny existujúcich prístupov pre odporúčanie – pričom dochádza len k preusporiadaniu odporúčaných prvkov.

Kľúčové slová: odporúčanie, preusporiadanie prvkov, skupiny

1 Úvod

Personalizované odporúčanie je dnes neoddeliteľnou súčasťou moderných webových aplikácií. V praxi sa používajú dva základné prístupy – odporúčanie založené na obsahu a kolaboratívne odporúčanie. Kým odporúčanie založené na obsahu sa zameriava na analýzu odporúčaných prvkov (napr. podobnosť medzi obľúbenými článkami), kolaboratívne odporúčanie stavia na analýze podobnosti používateľov a ich preferencií. Tieto dva prístupy sa vzhľadom na ich obmedzenia ako napr. riedkosť hodnotení pri kolaboratívnom odporúčaní alebo množstvo nových prvkov pri obsahovom odporúčaní často kombinujú a tým vzniká takzvané hybridné odporúčanie [1].

S rozvojom webu a sociálnej aktivity používateľov sa ukázalo ako vhodné a potrebné generovať personalizované odporúčania aj pre skupiny používateľov a nielen pre jednotlivcov. V tomto prípade je nevyhnutné v prvom kroku agregovať preferencie jednotlivých členov skupín a následne vygenerovať odporúčanie, resp. vygenerovať odporúčanie pre takto získaný model používateľa [4].

Proces generovania personalizovaných odporúčaní môžeme chápať ako predikčnú úlohu, kedy sa na základe predchádzajúcej aktivity používateľa (či už podobnosti obsahu alebo podobnosti používateľov) snažíme predikovať hodnotenia prvkov, ktoré ešte nevidel. Je zrejmé, že pri takejto paralele sa odporúčanie často zúži len na výber takých prvkov, ktoré sú predpovedané ako najvyššie hodnotené a len tieto sa prezen-

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 149–152.

tujú používateľovi. Vo viacerých prípadoch je však odporúčanie generované ako postupnosť prvkov – sekvencia (najmä pri obsahu, ktorý je automaticky, bez výberu používateľom, zobrazený). V takejto situácii je nevyhnutné zoradiť vygenerované odporúčania tak, aby sme maximalizovali spokojnosť používateľa, ktorý nemôže niektoré prvky preskočiť. Podobná situácia nastáva pri skupine, kedy je samotné rozhodnutie o výbere konkrétneho prvku veľmi obtiažne.

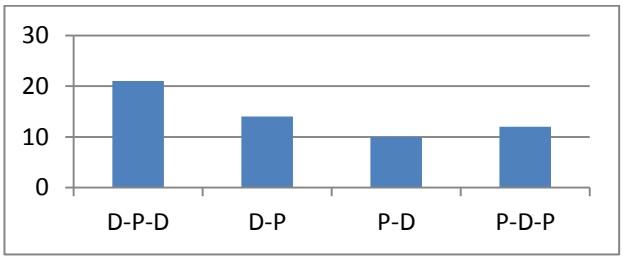
Ďalšou komplikáciou pri generovaní odporúčaní pre skupinu, je jej stálosť [2]. V prípade, že skupina zažíva vygenerované odporúčania (napr. sledovanie televízie) a niektorý jej člen prípadne členovia ju opustia, je otázne ako reagovať na tento stav. Generovanie odporúčania pre túto novú skupinu však nemusí byť vždy optimálnym riešením, vzhľadom na preferencie používateľov, ktoré sme identifikovali v ankete.

2 Preusporiadavanie odporúčaní

S cieľom identifikovať preferencie používateľov sme vykonali experiment na vzorke 35 respondentov (študenti informatiky). Týchto respondentov sme požiadali, aby zoradili 10 ohodnotených prvkov pripravených pre odporúčanie (predikované hodnotenia) podľa ich preferencií tak, aby mali na konci odporúčania čo najlepší pocit, resp. aby maximalizovali usporiadaním prvkov spokojnosť s odporúčaním. Vzory usporiadania prvkov môžeme rozdeliť na štyri základné typy:

- Dobré - priemerné - dobré hodnotenia (∨)
- Dobré - priemerné hodnotenia (>)
- Priemerné - dobré hodnotenia (<)
- Priemerné - dobré - priemerné hodnotenia (∧)

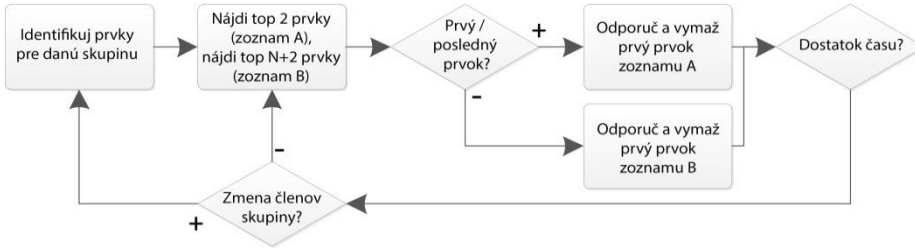
Ako môžeme vidieť (Obr. 1.) respondenti preferujú silný začiatok a silný koniec odporúčania, zatiaľ čo v priebehu odporúčania sa môžu vyskytovať aj horšie hodnotené prvky. Tento fakt nás však privádza k riešeniu ako čo najlepšie prispôsobiť takýto typ odporúčania pre dynamicky sa meniace skupiny.



Obrázok 1. Preferencie hodnotení prvkov (D-dobré, P- priemerné).

Na základe preferencií používateľov, ktoré sme získali v prieskume, navrhujeme hybridný prístup (Obr. 2.) pre preusporiadanie vygenerovaných prvkov pre odporúčanie s cieľom maximalizovať spokojnosť používateľov v rámci skupiny po sekvencii odporúčaných prvkov. Prístup je založený na uchovaní najlepších prvkov (na základe pred-

ikovaného hodnotenia) pre začiatok a koniec odporúčania, zatiaľ čo prvky uprostred sekvencie sú náhodne vyberané z množiny „top N“ prvkov.



Obrázok 2. Navrhovaný prístup pre preusporiadanie prvkov.

Navrhnutý prístup reflektuje pozorované preferované stratégie, pričom zmena zloženia skupiny, neovplyvňuje proces výpočtu odporúčania. Rovnako je navrhnutý prístup možné uplatniť aj pri odporúčaní jednotlivcom, kedy je samotná skupina reprezentovaná jedným členom.

2.1 Modelovanie spokojnosti

Ako je zrejmé, používatelia v skupine sú navzájom ovplyvňovaní na základe viacerých aspektov. V prvom rade používateľov navzájom ovplyvňuje aktuálna spokojnosť s daným odporúčaním, pričom intenzita tohto vplyvu je odvodená od vlastností sociálnych väzieb medzi danými používateľmi [3]. V prípade dvoch používateľov je tento proces ovplyvnenia pomerne zrozumiteľný, avšak v prípade viacerých členov sa počet väzieb a tým aj spôsobov ovplyvnenia dramaticky zvýši [5].

V kontexte preusporiadavania prvkov je teda zrejmé, že samotné predikované hodnotenie sa môže výrazne líšiť na skutočnej „zažívanej“ spokojnosti, pretože táto je založená nie len na spokojnosti so samotným prvkom, ale aj na spokojnosti s predchádzajúcimi prvkami a na ostatných používateľoch s daným odporúčaním. S postupným prechádzaním odporúčanej sekvencie, teda zanechávajú odporúčané prvky v používateľovi dojmy, ktoré ovplyvňujú hodnotenie ďalších prvkov zaradených do sekvencie.

Pre modelovanie takto vzniknutej situácie navrhujeme jednoduchý model, založený na šírení aktivácie, kedy môžeme zložité vnútro-skupinové väzby reprezentovať grafom. Ako sme už naznačili prvky videné v minulosti rovnako ovplyvňujú spokojnosť používateľa a teda je nevyhnutné zohľadniť aj takéto prvky. Samotný výpočet potom môžeme vyjadriť ako:

$$Spok_{i\epsilon I,u\epsilon U} = \frac{2.631}{3}\left(\frac{\sum_{i=1}^{n-1}\left(log_{n-1}\sqrt{i+1}\; spread(n_i,u)\right)}{n-1}\right) + \frac{2}{3}spread(n,u)$$

Ako môžeme vidieť, navrhnutý model zohľadňuje predchádzajúcu históriu videných prvkov, pričom intenzita zabúdania je zohľadnená logaritmickou krivkou a váha predchádzajúcich prvkov a ich spokojnosti je 1/3. Výsledná spokojnosť pre konkrétneho používateľa je následne vypočítaná ako lineárna kombinácia medzi aktuálnou a pred-

chádzajúcou spokojnosťou používateľa. Jednou z hlavných vlastností navrhnutého prístupu je možnosť jeho aplikácie aj pre odporúčanie jednotlivcom. Ak ovplyvnenie používateľmi nahradíme aktuálnym kontextom požívateľa (deň v týždni, počasie a pod.), môžeme rovnakým spôsobom modelovať reálnu spokojnosť používateľa.

3 Zhodnotenie

Aby sme overili navrhnuté prístupy využiteľné pre preusporiadanie odporúčaní jednotlivcom alebo skupinám, uskutočnili sme experiment, v ktorom sme expertom poskytli odporúčania preusporiadané naším prístupom a odporúčania zoradené štandardne od najlepšieho po najhoršie predikované hodnotenie.

V prvom prípade, sme vygenerovali odporúčanie pre 10 náhodne vytvorených skupín (2-6 členov). Odporúčania sme následne preusporiadali navrhnutým prístupom (Obr. 2.) – bez zapojenia šírenia aktivácie. Traja doménoví experti následne ohodnotili ponúknuté odporúčania, pričom im boli známe preferencie používateľov skupiny. Pri využití preusporiadania bolo získané priemerné hodnotenie *3.3* (škála 1-5), zatiaľ čo bez použitia navrhnutého presusporiadania *2.4*. Prvotné výsledky teda potvrdzujú našu hypotézu, že navrhnuté riešenie umožňuje pri využití rovnakých prvkov získať vyššiu spokojnosť používateľa resp. používateľov na konci odporúčania.

Ďalším krokom je rozšírenie predikcie hodnotenia prvku, ktoré je vypočítané na základe odporúčacej metódy o výpočet skutočnej spokojnosti prostredníctvom šírenia aktivácie. Týmto spôsobom je možné modelovať zložité vnútro skupinové procesy, rovnako je prístup využiteľný aj pre zohľadnenie kontextu pri odporúčaní jedincovi.

Poďakovanie. Táto práca bola čiastočne podporená projektom Získavanie, spracovanie, vizualizácia textových informácií na základe analýzy relácií podobnosti VG1/0971/11 a Kognitívne cestovanie po digitálnom svete webu a knižníc s podporou personalizovaných služieb a sociálnych sietí 0208-10.

Literatúra

1. Boratto, L., Carta, S., Chessa, A., Agelli, M., and Clemente, M. Group Recom. with Automatic Identification of Users Communities. In Web Intellig. and Intellig. Agent Tech., 2009. WI-IAT'09. Int. Joint Conf., vol. 3, pp. 547–550. IEEE. (2009).
2. Jameson, A. and Smyth, B. Recommendation to groups. In The adaptive web, Springer-Verlag, pp. 596–627. (2007).
3. Masthoff, J. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. In User Modeling and User-Adapted Int. 14., vol. 14, Springer, pp. 37–85. (2004)
4. Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. Recommender Systems Handbook, Springer. (2011).
5. Robbins, P. and Aydede, M. The Cambridge Handbook of Situated Cognition. Cambridge University Press, New York, NY, USA, 1st edition. (2008).

Rozpoznávanie pomenovaných entít v úlohách automatickej geografickej anotácie textových zdrojov

Peter Koncz[1], Ján Paralič[1]

[1]Katedra kybernetiky a umelej inteligencie, FEI,
Technická univerzita v Košiciach, Letná 9/B, 042 00 Košice
{peter.koncz, jan.paralic}@tuke.sk

Abstrakt. Automatické rozpoznávanie geografických entít v rámci neštruktúrovaného textu je dôležitou súčasťou vznikajúcich geograficky lokalizovaných služieb. Práca je preto venovaná návrhu riešenia pre rozpoznávanie pomenovaných entít v rámci úloh geografickej anotácie textových zdrojov. Popisované riešenie vychádza z integrácie existujúcich metód, pričom je rozšírené o disambiguáciu pomenovaných entít na základe ich sémantického kontextu, ako aj automatické získavanie počiatočných zoznamov pomenovaných entít. Overovanie sémantického kontextu pomenovaných entít je realizované na základe ontológií a asociačných sietí, kým získavanie počiatočných zoznamov pomenovaných entít je realizované prostredníctvom voľne dostupných rozhraní pre programovanie aplikácií. Popisované riešenie predstavuje návrh aplikácie a malo by byť východiskom pre ďalší výskum zameraný na realizovanie častí navrhovaného riešenia.

Kľúčové slová: Rozpoznávanie pomenovaných entít, automatická anotácia textov, geoparsovanie, geokódovanie

1 Úvod

Pomenované entity (named entity) je termín pôvodne zavedený pre označenie ľudí, lokalít, organizácií, ako aj niektorých číselných výrazov. Identifikácia odkazov na tieto entity v texte je označovaná ako rozpoznávanie pomenovaných entít [1-3]. Zoznam typov entít sa v priebehu rokov rozširoval, no lokality a neskôr aj ich jednotlivé typy predstavujú jeden zo základných typov pomenovaných entít. Aplikácia metód rozpoznávania pomenovaných entít pri určovaní geografického kontextu, označované aj ako georeferencovanie (georeferencing), pozostáva z tzv. geoparsovania (geoparsing) a geokódovania (geocoding). Geoparsovanie zodpovedá identifikácii a extrahovaniu geografických názvov z textu, kým geokódovanie predstavuje priradenie špecifických geografických indikátorov k týmto názvom, zvyčajne vyjadreným v podobe zemepisných súradníc [4] a zodpovedá disambiguácii pomenovaných entít. Význam uvedených metód spočíva v možnosti ich použitia na automatickú geografickú anotáciu neštruktúrovaných textov, využívanú v rámci geograficky lokalizovaných služieb. Mnohé z metód realizácie týchto úloh sú však časovo náročné a cieľom práce je preto návrh aplikácie umožňujúcej automatickú geografickú anotáciu textov

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 153–156.

s minimálnou potrebou ľudského zásahu. Práca zároveň predstavuje základ pre ďalší výskum súvisiaci s realizáciou jednotlivých častí navrhovaného riešenia.

2 Súvisiace práce

Súčasné metódy rozpoznávania pomenovaných entít, podobne ako aj metódy v rámci iných úloh spracovania prirodzeného jazyka, zvyknú byť rozdeľované do dvoch hlavných skupín [1, 5]. Prvá skupina metód vychádza z použitia strojového učenia, pričom na účely rozpoznávania pomenovaných entít je často používaná metóda podporných vektorov, ako aj metódy maximálnej entropie, skrytých markovových modelov či podmienených náhodných polí [2]. Druhá skupina metód vychádza z použitia tzv. lingvistického prístupu, založeného na zvyčajne manuálne vytvorených slovníkoch a pravidlách. Nedostatkom oboch skupín metód je potreba veľkého množstva ľudskej práce pri tvorbe výsledných algoritmov, ako aj ich doménová závislosť. Čoraz populárnejšími sa preto stávajú tzv. polo-kontrolované postupy rozpoznávania pomenovaných entít [6], v rámci ktorých sa môžeme stretnúť s rozličnými variáciami tzv. metódy vzájomného sebazavádzania (mutual bootstrapping). Táto metóda vychádza z cyklického vzájomného rozširovania počiatočnej množiny pomenovaných entít na základe vzorov ich výskytu a naopak [7]. Metóda vzájomného sebazavádzania je použitá aj v rámci nami navrhovaného riešenia, popisovaného v nasledujúcej kapitole.

3 Navrhované riešenie

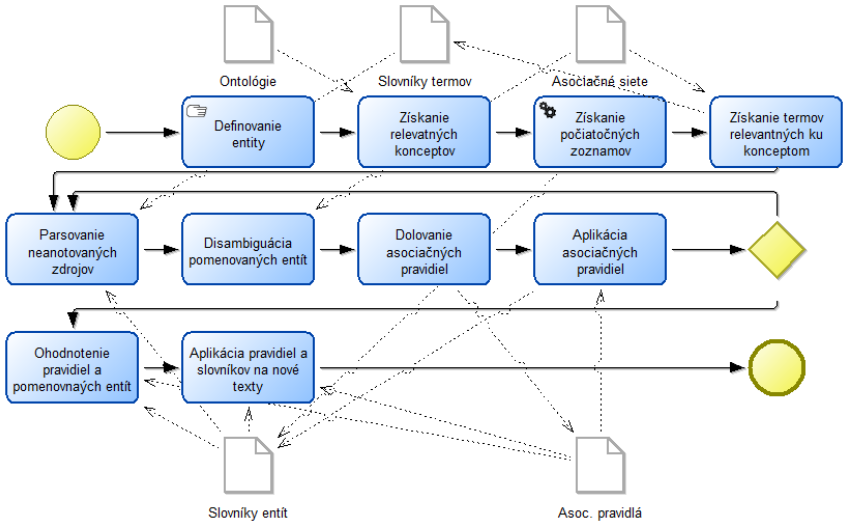
Prvým krokom navrhovaného riešenia, ktorého schéma je znázornená na obrázku 1, je definovanie geografickej entity zadaním jej typu, názvu a zemepisných súradníc. Na základe typu entity sú následne odvodené relevantné koncepty. V súčasnosti existuje viacero voľne dostupných ontológií pre rozličné typy lokalít, ktorých príklady je možné nájsť napr. v práci Ma a kol. [8]. Inštancie týchto konceptov predstavujú tzv. počiatočné zoznamy. Tie boli v rámci nami analyzovaných prác zvyčajne zadávané manuálne. V kontexte geografických entít však existuje veľké množstvo voľne dostupných rozhraní pre programovanie aplikácií (API) predstavujúcich zdroje inštancií konceptov. Tieto entity sú spravidla určené svojim názvom, typom a geografickými identifikátormi. Okrem počiatočných zoznamov pomenovaných entít je potrebné identifikovať aj termy relevantné vzhľadom k získaným konceptom, k čomu sú v rámci navrhovaného riešenia použité tzv. asociačné siete (viď nižšie). Získané termy, ich synonymá a relevantné lexikografické variácie, ako aj pomenované entity sú identifikované v procese parsovania neanotovaných zdrojov. V procese disambiguácie je následne určená pravdepodobnosť s ktorou reťazec zodpovedá hľadanej pomenovanej entite. V rámci navrhovaného riešenia je najskôr overovaný typ pomenovanej entity a následne je identifikovaná konkrétna pomenovaná entita, pričom sú používané nasledovné heuristiky:

- Heuristika spoločného výskytu. Využíva predpoklad, podľa ktorého sa navzájom súvisiace termy v textoch vyskytujú často spoločne. Namiesto zvyčajne používa-

ných mier asociácie termov ako napr. bodová vzájomná informácia (pointwise mutual information), sme sa v rámci navrhovaného riešenia rozhodli použiť kombináciu klasických ontológií a asociačných sietí. V rámci nej sú medziľahlé uzly tvorené konceptmi a listové uzly sú predstavované termami, pričom hrany medzi medziľahlými uzlami reprezentujú vzťahy medzi konceptmi a vážené hrany reprezentujú mieru asociácie termu a konceptu.

- Heuristika spoločného zdroja. Využíva predpoklad, podľa ktorého je pravdepodobnosť výskytu rovnakej pomenovanej entity resp. jej typu v rámci množiny dokumentov z rovnakého zdroja (webová doména, autor a pod.) vyššia než pravdepodobnosť v prípade rovnako veľkej množiny pochádzajúcej z rôznych zdrojov.
- Heuristika významných udalostí. Využíva predpoklad, podľa ktorého je pravdepodobnosť výskytu určitých pomenovaných entít resp. ich typov vyššia v určitých časových obdobiach súvisiacich s významnými udalosťami.
- Heuristika priestorovej blízkosti. Využíva predpoklad, podľa ktorého je pravdepodobnosť výskytu entity závislá na vzdialenosti (geografickej alebo v texte) k iným entitám v rámci rovnakého dokumentu resp. ich skupiny.

Po disambiguácii je možné získať vzory výskytu pomenovaných entít pre extrahovanie nových pomenovaných entít. Tieto vzory sú predstavované sekvenčnými asociačnými pravidlami zohľadňujúcimi termy v okolí pomenovaných entít, ako aj ich poradie. Ich hlavnou výhodou je nízka výpočtová náročnosť pri ich aplikovaní na neanotované texty. Kvalita získaných asociačných pravidiel je ohodnotená vzhľadom k jednotlivým typom pomenovaných entít. Týmto spôsobom je neustále rozširovaná základná množina pomenovaných entít ako aj pravidiel. Po dosiahnutí stanovených ukončovacích podmienok, ktorými môže byť prírastok pomenovaných entít či pravidiel, je možné prejsť k celkovému vyhodnoteniu získaných pravidiel a pomenovaných entít a tieto následne aplikovať na nové texty.



Obr. 1. Schéma navrhovaného riešenia.

4 Záver

V práci bol prezentovaný návrh aplikácie umožňujúcej automatickú geografickú anotáciu zdrojov s minimálnou potrebou ľudského zásahu. Pri návrhu riešenia sme vychádzali z existujúcich overených riešení, hoci spôsob integrácie týchto riešení je špecifický. Špecifická pre navrhované riešenie je aj použitá metóda disambiguácie, využívajúca kombináciu ontológií a asociačných sietí, ako aj automatizovaný spôsob získavania geografických konceptov a ich inštancií. Vzhľadom ku komplexnosti navrhovaného riešenia je potrebné realizovať viaceré experimenty pre potvrdenie jeho vhodnosti, ktoré by mali byť predmetom budúceho výskumu.

Poďakovanie. Táto práca bola podporovaná Vedeckou grantovou agentúrou Ministerstva školstva, vedy, výskumu a športu Slovenskej republiky a Slovenskej akadémie vied na základe zmluvy č. 1/1147/12 (50%) a Agentúrou na podporu výskumu a vývoja na základe zmluvy č. APVV-0208-10 (50%).

Použitá literatúra

1. Kozareva, Z., Ferrández, O., Montoyo, A., Muñoz, R., Suárez, A., Gómez, J.: Combining data-driven systems for improving Named Entity Recognition. Data Knowl. Eng. 61, 449-466 (2007).
2. Saha, S.K., Narayan, S., Sarkar, S., Mitra, P.: A composite kernel for named entity recognition. Pattern Recognition Letters 31, 1591-1597 (2010).
3. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30, 3-26 (2007).
4. Zubizarreta, Á., de la Fuente, P., Cantera, J., Arias, M., Cabrero, J., García, G., Llamas, C., Vegas, J.: Extracting Geographic Context from the Web: GeoReferencing in MyMoSe. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) Advances in Information Retrieval, vol. 5478, pp. 554-561. Springer Berlin / Heidelberg (2009).
5. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 359-367. Association for Computational Linguistics, Portland, Oregon (2011).
6. Nadeau, D.: Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision. pp. 142. University of Ottawa (2007).
7. Riloff, E., Jones, R.: Learning dictionaries for information extraction by multi-level bootstrapping. Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, pp. 474-479. American Association for Artificial Intelligence, Orlando, Florida, United States (1999).
8. Ma, X., Carranza, E.J.M., Wu, C., van der Meer, F.D.: Ontology-aided annotation, visualization, and generalization of geological time-scale information from online geological map services. Computers & Geosciences 40, 107-119 (2012).

Interfacing between Ontologies and Real-Time Systems

M. Kvassay[1], L. Hluchý[1], B. Kryza[2], J. Kitowski[2]

[1]Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9,
845 07 Bratislava, Slovakia
{marcel.kvassay, hluchy.ui}@savba.sk
[2]Academic Computer Centre CYFRONET AGH, ul. Nawojki 11, Kraków, 30-950, Poland
{bkryza, kito}@agh.edu.pl

Abstract. Ontologies represent an attractive way of storing semantic information, yet ontology reasoners are often too slow for systems that need to perform in real time. This article shows how to mitigate the problem in the context of human behaviour simulations by designing efficient in-memory structures into which the results of complex ontological queries can be pre-loaded in the simulation initialization stage.

Keywords: ontology, human behaviour simulation, real-time system, interface.

1 Ontology Support to Human Behaviour Simulations

Human Behaviour simulations undoubtedly qualify as semantically rich applications, which can benefit from the "shared explicit conceptualization" provided by ontologies. As we explained in [1], our agents are composed from simpler components (motives and behaviours), so a suitable ontology can guide users in the definition of new agent types and guarantee the consistency of the definitions. Later, during the simulation, the same information can be used to determine which motives trigger which behaviours under which conditions. Here, quick access to information is crucial, especially when the simulation has to perform in real-time, as happened in our recent research project EUSAS ("European Urban Simulation of Asymmetric Scenarios") financed by 20 nations under the *Joint Investment Program Force Protection* of the European Defence Agency. In this article we focus on the second aspect – how to provide ontological information to real-time applications as fast as possible.

For our present purpose we have simplified the structure of the ontology as compared to that presented in [1]. Fig. 1 below shows an example of *Withdraw* (instance of *Behaviour Pattern* class), which is triggered when the motive *Fear* is action-leading (property *relatedMotive*) and its intensity (normalized to the percentage scale) falls in the interval <0%, 50%) as defined by the lower and the upper bounds *hasThreshold* and *hasLimit*. In the agent modelling framework described in [1] the same structure would fit the other behaviours as well. During the simulation, each agent determines its action-leading motive and the percentage intensity of the behaviour choice variable (typically the motive itself), and then uses this information to activate the appropriate behaviour. In the next section we evaluate the difference in

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 157–160.

performance when this task is performed by the ontology directly as compared to fast in-memory structures that are pre-loaded with the necessary ontological information in the simulation initialization stage.

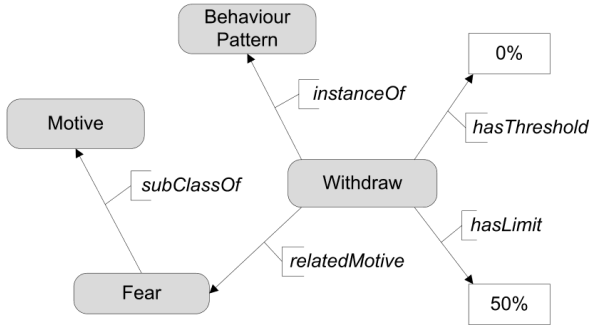


Fig. 1. Simplified ontological description of *Withdraw* behaviour triggered by the motive *Fear*.

2 Interface Data Structures and Experimental Validation

To extract the proper behaviour (***bp***) from the ontology for a given action-leading motive (<***motive***>) and the percentage intensity of the choice variable (<***intensity%***>), we used the following SPARQL query (leaving out prefix definitions for simplicity):

```
SELECT ?bp WHERE {
  ?bp rdf:type Behaviour:BehaviourPattern .
  ?bp Behaviour:hasThreshold  ?lower .
  ?bp Behaviour:hasLimit  ?upper .
  ?bp Behaviour:relatedMotive <motive> .
FILTER (?lower <= <intensity%> && ?upper > <intensity%>)}
```

To store and extract the same information from an in-memory structure, we opted for a fast two-level nested *TreeMap* shown in the following Java code snippet:

```
TreeMap<String, TreeMap<Integer, String>> mbMap;
…
TreeMap<Integer, String> t1 = mbMap.get(<motive>);
Map.Entry<Integer, String> t2 =
                        t1.floorEntry(<intensity%>);
String bp = t2.getValue();
```

For each ***motive*** represented by an URI String, the two-level TreeMap ***mbMap*** stores a nested TreeMap ***t1***, which holds appropriate behaviours (as URI strings) indexed by their lower bounds (***hasThreshold*** values). So, retrieving the behaviour takes two steps, including the call to ***floorEntry()*** method of the Java *NavigableMap* interface.

In the experiments we have used two sizes of the ontology and the in-memory map. The "small" ones contained about ten elements – two motives (*Anger* and *Fear*),

eight behaviours and a few auxiliary entities. The "large" ones were populated with 1000 random motives and 34 random behaviours per motive, so they held about 34000 elements each. The tests were performed on Intel Core i7 860 @ 2.8 GHz machine, using Java 1.6, Jena version 2.6.3 and Pellet version 2.2.2 libraries. All values were averaged over ten independent runs of the same test.

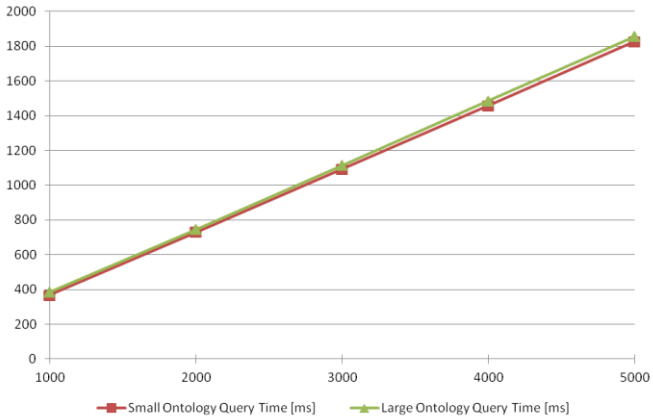


Fig. 2. Direct ontology access times in milliseconds.

Fig. 2 shows direct ontology access times for 1000 to 5000 queries, while Fig. 3 below shows in-memory map access times for 500,000 to 1,000,000 queries. The experiment confirms that direct ontology access is indeed considerably slower than in-memory map access but, interestingly, it does not seem to depend on the number of elements in the ontology, only on the number of queries (which is a function of the number of simulated agents and how often they query the ontology). This indicates

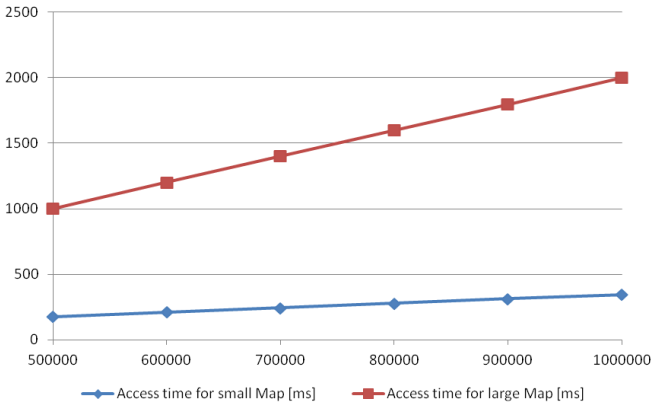


Fig. 3. In-memory access times (in milliseconds) for two-level TreeMap are directly proportional to the number of queries, with a logarithmic dependence on the number of map entries.

that Pellet is optimized for searching complex knowledge bases but takes time to translate the queries into the optimal form. The results should not be construed as a reason for eliminating ontologies from real-time systems, rather the opposite. The main benefit of ontologies is not their real time performance but the flexibility of the system and minimization of preprocessing and configuration effort whenever there is a change (e.g. when new types of objects or actions are added to the system). Most importantly, by applying the separation of concerns paradigm to the design of the ontologies, the system is able to find, using solely Description Logics reasoning rules, the space of all possible actions the agent can take and allow the agent implementation to select the most appropriate one with respect to agents' current motive and context (available objects, position, energy). Our experiments show that it is possible to combine the ontology with real-time systems through efficient in-memory structures and so enjoy the best of both worlds: speed as well as flexibility and intelligence. We believe the approach presented here – an offline ontology reasoner through which the system is configured in the initialization stage in combination with fast in-memory structures for storing pre-loaded ontology information – can be used almost universally in various scenarios, so long as the purpose of the intended SPARQL queries can be fulfilled by accessing a piece of information indexed and stored in a fast multi-level key-value structure, such as a nested *TreeMap*.

In our future work, we plan to implement the functionality of automatic generation of the environment and object ontology hierarchies, using the X2R tool supporting the conversion process from legacy relational, XML or LDAP repositories to Web Ontology Language form [2] and evaluate the approach with much larger data sets using ontological knowledge bases for ontology management such as GOM [3].

Acknowledgments. This work is supported by the EDA project A-0938-RT-GC EUSAS and by Slovak Research and Development Agency under the contract No. APVV-0233-10.

References

1. Hluchý, L, M. Kvassay, Š. Dlugolinský, B. Schneider, H. Bracker, B. Kryza and J. Kitowski. 2012. "Towards More Realistic Human Behaviour Simulation: Modelling Concept, Deriving Ontology and Semantic Framework." In *Applied Computational Intelligence in Engineering and Information Technology, Studies in Computational Intelligence*, Edited by R.-E. Precup, Sz. Kovacs, S. Preitl and E. M. Petriu. Berlin: Springer.
2. A. Mylka, A. Mylka, B. Kryza, J. Kitowski, Integration of Heterogenous Data Sources in an Ontological Knowledge Base, Computing and Informatics. volume 31, number 1, 2012, p.189-223.
3. B. Kryza, R. Slota, M. Majewska, J. Pieczykolan, J. Kitowski, Grid organizational memory-provision of a high-level Grid abstraction layer supported by ontology alignment, Future Generation Computer Systems, 23 (3) (2007) 348-358.

Improving Entity and Relation Discovery by User Interaction with Semantic Graphs

Michal Laclavík

Institute of Informatics, Slovak Academy of Sciences,
Dúbravská cesta 9, 845 07 Bratislava
michal.laclavik@savba.sk

Abstract. In this paper we discuss how to improve information extraction (IE) results with user interaction when searching relations in semantic networks. We discuss if the user interaction is feasible approach for IE improvement.

Keywords: graph, information extraction, user interaction

1 Introduction

In our previous work we described the prototype, which extract semantic networks from texts and it is able to search for relations among entities in these networks [3, 4, 6]. The challenge we would like to address in this paper is appropriate entity extraction, which have big impact on the relation search results. The wrongly detected entities or entities which are irrelevant for the problem domain have significant effect on the search results. Thus we try to implement user interaction features, which can manipulate graph/entities and have further impact on better IE (entity discovery).

In [5] we have described the Rule based Named Entity extraction approach. The approach used is quite simple, but also open to plug-in any state of the art named entity recognition (NER) system. The main contribution lies in: *simplicity* - regular expression patterns and gazetteers; configurability - possibility to plug-in existing NER systems; *trees and graph* structures formation from text

In this paper we describe how user interaction can help to improve IE and underlying semantic graph. The user input has been previously considered in form of tags or LinkedData such as DBPedia [2] or direct user interaction for improving IE [1]. In our work we use a bit different approach. We describe our approach on DSK[2] use case, where we have crawled about 100 web pages relevant for DSK case and tested the approach. In the experiment provided in chapter 3 we just work with small graph extracted from 3 Wikipedia pages related to DSK, IMF[3] and PS[4] for simplicity.

[2] http://en.wikipedia.org/wiki/Dominique_Strauss-Kahn
[3] http://en.wikipedia.org/wiki/International_Monetary_Fund
[4] http://en.wikipedia.org/wiki/Socialist_Party_(France)

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 161–164.

2 User Interaction with Semantic Graph

Patterns and gazetteers can be tuned quite well for the application, where the data are quite known as we have showed in [5]. However, it is hard to extract valuable entities from text in open domains such as web. We have extracted text graphs from web (DSK use case) using simple approach of discovering candidates for named entities. We have simply extracted all words starting with capital letters, which were considered as type-less named entities (marked as NE in Fig. 1 and in experiment). In addition to type-less entities, we were extracting people names, cities, countries or dates using simple gazetteer and rule based approach. You can imagine that type-less entities brought also lot of nonsense entities. Here we discuss how they can be cleaned up with help of user, when doing analytical task of relations discovery.

After extracting text graphs from the web pages we got many false entities related to *IMF* or *Straus-Kahn* as you can see in Fig. 1.

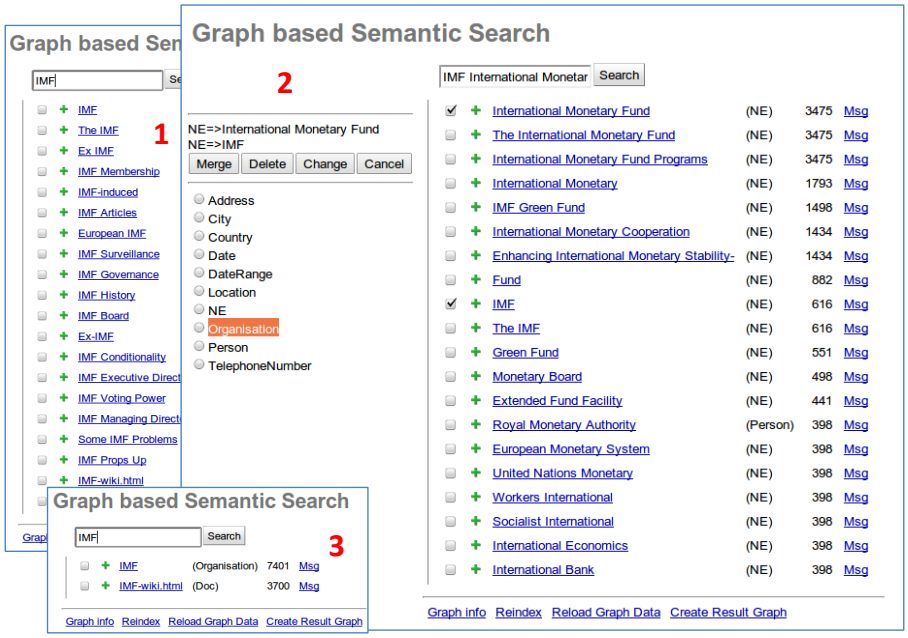


Fig. 1. Full-text search and user interaction with IMF related entities.

Fig. 1 shows us full text results for *IMF* on screen 1. On the screen 2, we have selected type-less entities *IMF* and *International Monetary Fund* and change their type to *Organization*. This had impact on creating positive gazetteers, where both organization names were added to list of organizations. When the collection was reparsed, *IMF* related wrongly identified entities such as *Ex-IMF*, *The IMF* or *IMF Voting Power* (Screen 1) disappeared from the graph (see Screen 3) based on defined rule, where type-less entity cannot overlap with confirmed or type-defined entity.

Similarly, *Strauss-Kahn* was identified as type-less entity, where other type-less entities such as *Strauss-Kahn Case or Strauss-Kahn Released After* were wrongly detected. After modifying entity type to person and reparsing, only relevant *Person* entities are present in the graph data.

Important feature for user interaction is also deleting wrongly identified entities. In our case we get rid of many such entities by redefining the entity type. However simple approach can detect many one word entities such as days of the week, the common words with capital letters in article titles, menus or at the beginning of sentences. Such entities can then interconnect the data which does not belong together. Even if we identify some entities correctly, sometimes is better to delete them to fulfill intended analytical task. For example if *BBC* is detected in each web page from *BBC* crawl application, it makes sense to remove *BBC* entity since it connects all the data and adds no information. When deleting the node, entity node with all related links to this node is deleted. Thus any user interaction has immediate impact on the quality of search results. When user deletes the node, negative gazetteers are being created. After reparsing the data all such nodes should disappear in all documents where it was detected before.

Name of the cities are detected by list of major cities in the world. This include also *Mobile* city, which is located in USA. Anyhow, when parsing technical news, *Mobile* will get wrongly detected on many places. It makes sense to delete it and thus it will get to negative gazetteer. Similar case is people name recognition. People names are detected based on first name lists and regular expressions. This simple approach works quite well, but it detects also nonsense people like *Ivory Coast*, *Roman Catholic Church* or *Royal Monetary Authority* (see Fig. 1, screen 2), because *Ivory*, *Roman* or *Royal* are first names. It is easier to delete wrongly identified peoples or change entity type then discard useful detection rules.

3 Experiments

As we mentioned earlier, we have experimented with this approach on the web data from BBC, Wikipedia or the web pages found by Google API on concrete topic. Same way we have gathered web pages on DSK use case. However due to simplicity we provide evaluation only on 3 pages from Wikipedia.

Table 1. Entity and Graph data; Left: data statistics, Right: changes statistics.

	Raw Occurrence	Clean Occurrence	Diff	Diff %	Raw Count	Clean Count	Diff	Diff %
Entities	4682	4776	94	2.01%	2881	2857	-24	-0.83%
Edges	4678	4772	94	2.01%	4383	4377	-6	-0.14%
Person	704	841	137	19.46%	409	411	2	0.49%
Organization	0	360	360		0	4	4	
NE	1893	1490	-403	-21.29%	918	860	-58	-6.32%
Total	2597	2691	94	3.62%	1327	1275	-52	-3.92%

Changes Occurrence	900
Changes Graph	64
Deleted Entities	21
Change Type Entities	6
Operations together	**27**
Improvement occ. count	873
Improvement occ.	32.33x
Improvement count	37
Improvement	**1.76x**

The Table 1, on the left, shows the statistic of the extracted entities of the *raw* data before the user interaction with the graph and the *clean* data show the data after some

cleaning operation done by user. The difference between the *occurrence* and *count* columns is that in *occurrence* we count multiple occurrence of the entity in the text as detected, and in *count* column every key-value pair (e.g. Organization=>IMF) occurs only once.

After a user interaction involving 27 user operations (taking about 2-3 minutes of browsing and interacting with the data), there is high number of changes on the further extraction process (32-times on occurrence in text), which goes far behind the consumed user time. User will immediately get improved results. After reparsing the data based on new positive and negative gazetteers the number of changes (*clean* columns) increases as well.

Table 1, on the right, compares the number of changes done by user to the actual number of changes made on the graph data. We see that in final graph this makes roughly 1.8 times of changes increase most of them leading to better search results.

4 Conclusion

User interaction seems to be quite time consuming effort, however if users see the immediate impact on better search results and better identification of entities in text, they may be willing to do it. We would like to further extend the approach to automatically create training data set for IE machine learning approach.

Acknowledgment. This work is supported by projects: CLAN APVV-0809-11, VENIS FP7-284984 and VEGA 2/0184/10.

References

1. Alexiei Dingli, Fabio Ciravegna, and Yorick Wilks: Automatic Semantic Annotation using Unsupervised Information Extraction and Integration. in Proceedings of the K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation, 2003.
2. Milne, D. and Witten, I.H. (2008) Learning to link with Wikipedia. In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM'2008), Napa Valley, California.
3. Michal Laclavík, Martin Šeleng, Marek Ciglan, Štefan Dlugolinský, Ladislav Hluchý: gSemSearch: Objavovanie relácií v kolekciách textových a grafových dát; In 6th Workshop on Intelligent and Knowledge Oriented Technologies : WIKT 2011 proceedings, Košice, 2011, p. 1-5. ISBN 978-80-89284-99-3.
4. Michal Laclavík, Marek Ciglan, Štefan Dlugolinský, Martin Šeleng, Ladislav Hluchý: Emails as Graph: Relation Discovery in Email Archive. In Email2012 workshop, WWW 2012, April 16–20, 2012, Lyon, France, pages 841-846, 2012.
5. M. Laclavik, S. Dlugolinsky, M. Seleng, M. Kvassay, E. Gatial, Z. Balogh, L. Hluchy: Email Analysis and Information Extraction for Enterprise Benefit. In Computing and informatics, 2011, vol. 30, no. 1, p. 57-87. ISSN 1335-9150, Special Issue on Business Collaboration Support for micro, small, and medium-sized Enterprises.
6. Michal Laclavík, Ladislav Hluchý: Využitie sociálnych sietí pri vyhľadávaní v emailoch. In WIKT 2010 - Bratislava : ÚI SAV, 2010, p. 68-71. ISBN 978-80-970145-2-0.

Scalable Framework for Semantic Web Crawling

Ivo Lašek[1,2], Ondřej Klimpera[1], Peter Vojtáš[2]

[1] Czech Technical University in Prague, Faculty of Information Technology, Prague, Czech Republic
{lasekivo,klimpond}@fit.cvut.cz
[2] Charles University in Prague, Faculty of Mathematics and Physics, Prague, Czech Republic
vojtas@ksi.mff.cuni.cz

Abstract. In this paper, we introduce a new approach to semantic Web crawling. We propose a scalable framework for semantic data crawling that works as a service. Users can post their crawling requests via a RESTful API, the requests are processed on a scalable Hadoop cluster and aggregated results are returned in the desired format.

Keywords: Linked Data, Crawler, MapReduce

1 Introduction and Related Work

In this paper, we introduce our approach to semantic data crawling. We proposed and developed a semantic web crawling framework that supports all major ways to obtain data from semantic web (crawling RDF resources, RDF embedded in web pages, querying SPARQL endpoints and crawling semantic sitemaps). The framework is designed as a Web service. In order to support the scalability needed for crawling Web resources the whole service runs on the Hadoop computer cluster using MapReduce algorithm [2] to distribute the workload.

Probably one of the first adopters of crawling technologies for Semantic Web were authors of semantic Web search engines. When Watson [10] and Swoogle [3] were developed, one of the biggest crawling problems was, how to actually discover semantic data resources. The authors proposed several heuristics including exploring well-known ontology repositories and querying Google with a special type of queries. The crawling of the discovered content by Watson relies on Heritrix crawler[3].

The piplined crawling architecture was proposed for MultiCrawler [6] employed in SWSE semantic search engine [5, 4]. MultiCrawler deals also with performance scaling by distributing processed pages to individual computers based on a hashing function. However, the authors do not deal with a fail over scenario, where some computers in the cluster might break down.

The multithreaded crawler was used to obtain data for Falcons search engine [1]. A more sophisticated crawling infrastructure is employed in Sindice

[3] http://crawler.archive.org

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 165-168.

project [9]. We adapted the format of semantic sitemap that the authors propose. However, our project focuses rather at processing individual crawling requests than indexing all semantic documents on the Web.

LDSpider [7] is a Java project that enables performing custom crawling tasks. The spider performs concurrent crawling by starting multiple threads. However, all the threads still use shared CPU, memory and storage. We aim at providing a distributed crawling framework that can run on multiple computers.

Ordinary web crawlers like Apache Nutch [8] do not deal with semantic web data and are designed to follow and index only ordinary web links.

2 Crawling Semantic Web Resources

While crawling Web resources in general (and RDF resources in our case) it is necessary to avoid overloading of crawled servers. The *Robots Exclusion Protocol* published inside robots.txt files is a valuable source of information for crawlers.

A valuable source of crawling information are *semantic sitemaps*. Our crawler is able to extract three types of information from them: URL resources, sitemap index files and semantic data dump locations.

Another way how to make semantic data accessible is to provide a *SPARQL endpoint*. Data sources provide an endpoint API consuming SPARQL queries. We designed the crawler so that it is able to send HTTP GET requests to given endpoints, which is the most common way of their access. Results are then retrieved and used for further processing, i.e. further crawling of contained resources, if requested by the user - according to a requested crawling depth.

3 Parallel Processing with MapReduce

We have chosen the MapReduce programming model for the crawling. It is the proven concept for this type of activity used by Google [2], Apache Nutch [8] or other crawlers. Our crawler uses MapReduce model implementation provided by Apache Hadoop framework[4]. Crawling of Web resources is done by launching series of dependent MapReduce jobs, where each one has a designated task to be performed.

The Crawling process starts with user's request with a list of various type of resources (URLs, Sitemaps and SPARQL queries to selected SPARQL endpoints). The crawler creates two job feeders. One is responsible for creating an initial seed-list *SeedListJobFeeder*, second one deals with crawling Web resources to depth *CrawlingJobFeeder*.

To create a seed list, it has to be checked if there are some robots.txt and sitemap resources. If so, MapReduce jobs are created to crawl them and discover new locations, which are later added to the seed-list. The disallowed patterns are stored. Sitemap locations are collected and processed. Crawling sitemaps is a little bit more difficult, as each sitemap can contain references to other sitemaps

[4] Apache Hadoop Project homepage: `http://hadoop.apache.org/`

on the server. The crawler has to crawl them in depth and each level requires a new MapReduce job.

By crawling to the depth, each level of crawling requires a new MapReduce job.

Before the final data merge is triggered, we extract found `owl:sameAs` statements. The final MapReduce job reads crawled triples from all depths and checks subject, predicate and object nodes against the `owl:sameAs` mapping index. If a match is found, the appropriate triple's part is replaced with its reference value. These results are written to Distributed File System and copied to crawler's local storage. Such data are then accessible via REST API to the user.

When a user submits a request for crawling to the API, it is added to the queue of waiting jobs. These requests are processed in separate threads managed by a thread pool of fixed size. There is a MapReduceJobRunner which creates mentioned job feeders and handles their submission to the cluster. This class also handles job status and progress monitoring and provides its information to the user.

We tested our crawler on a small scale cluster with 7 nodes. Each node had its own local 20GB storage. First Node has an additional NFS mounted storage with 1TB available disk space, which was used to store job results, log files, and the database file structure.

We ran a test downloading data from `http://dbpedia.org` and `http://data.nytimes.com` semantic dump files. The crawling results contained over 30 million triples in more than 4.2GB of data.

Figure 1 shows percentage share of processes used for extracting triples from one URL. Average size of a resource was 270kB containing approximately 1600 triples. In average, there were about 10 triples per millisecond extracted on one node.

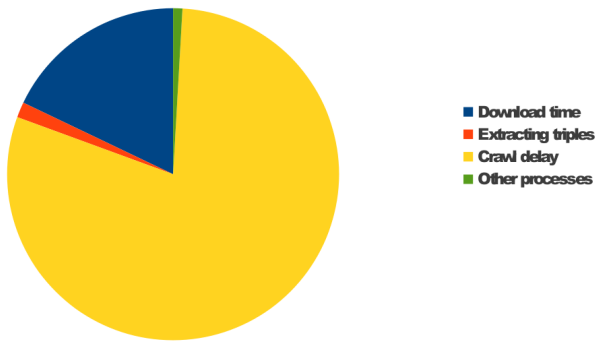


Fig. 1. Time consumption of URL processing tasks on small data.

4 Conclusion

We proposed an architecture for distributed crawling of semantic data. The introduced framework enables consumption of data from various types of resources including data dumps listed in semantic sitemaps, individual semantic Web documents, SPARQL endpoints and ordinary Web pages containing a structured data mark-up.

Currently the project is in beta version and can be downloaded from our Web page[5].

Acknowledgments. This work has been partially supported the grant of Czech Technical University in Prague (SGS12/093/OHK3/1T/18) and by the grant of The Czech Science Foundation (GAČR) P202/10/0761.

References

1. Gong Cheng, Yuzhong Qu, and Yuzhong Qu. Searching linked objects with falcons: Approach, implementation and evaluation. pages 49–70 (2009)
2. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113 (January 2008)
3. Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 652–659, New York, NY, USA, (2004)
4. Andreas Harth, Aidan Hogan, Renaud Delbru, Jürgen Umbrich, Stefan Decker, and Stefan Decker. Swse: answers before links. (2007)
5. Andreas Harth, Aidan Hogan, Jürgen Umbrich, and Stefan Decker. Swse: Objects before documents! (2007)
6. Andreas Harth, Jürgen Umbrich, and Stefan Decker. Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 258–271. Springer Berlin / Heidelberg (2006)
7. Robert Isele, Andreas Harth, Jrgen Umbrich, Christian Bizer, and Christian Bizer. Ldspider: An open-source crawling framework for the web of linked data. In *Poster at the International Semantic Web Conference (ISWC2010), Shanghai* (2010)
8. Rohit Khare and Doug Cutting. Nutch: A flexible and scalable open-source web search engine. Technical report (2004)
9. Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, and Giovanni Tummarello. Sindice.com: A document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies* (2008)
10. Marta Sabou, Martin Dzbor, Claudio Baldassarre, Sofia Angeletou, and Enrico Motta. Watson: A gateway for the semantic web. In *Poster session of the European Semantic Web Conference, ESWC* (2007)

[5] `http://research.i-lasek.cz/projects/semantic-web-crawler`

Replikácia dát v Ad-Hoc sieti typu VANET

Anton Lieskovský

Katedra informatiky, Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Univerzitná 1, 010 26, Žilina
anton.lieskovsky@fri.uniza.sk

Abstrakt. Článok zahŕňa aktuálny výskum, ktorý prebieha na FRI na Žilinskej univerzite, v oblasti replikácií dát v Ad-Hoc sieti typu VANET[1], a teda v oblasti informačno-komunikačných technológií a inteligentných dopravných systémov (ITS[2]). Nakoľko vedeckých úloh v sieti VANET je niekoľko, tento článok predstavuje konkrétne tému replikácií dát. Ide predovšetkým o replikačné techniky a replikačné algoritmy používané vo VANET za účelom zabezpečiť efektívne šírenie dát. Efektivita šírenia dát je pritom závislá od počtu uzlov, ktoré dáta prijali a od zaťaženia siete.

Kľúčové slová: OBU, RSU, VANET, dáta, replika

1 Význam šírenia dát v Ad-Hoc

Šírenie dát vo VANET sieti je dôležitou súčasťou rozvoja ITS pri znižovaní dopravnej nehodovosti, zvyšovaní informovanosti účastníkov dopravne infraštruktúry a podpory rozhodovania v rôznych dopravných situáciách. Jednoduchý scenár odohrávajúci sa vo VANET je nasledovný: V dopravnej sieti sa nachádzajú autá, ktoré svojim pohybom neustále menia svoju pozíciu. Okrem častej zmeny topológie siete to z pohľadu mobilného uzla (auta, OBU[3]) znamená neustály dopyt po informáciách nielen dopravného charakteru. Informácie, ktoré sa môžu vyskytnúť sú: údaje zaznamenané senzormi iných áut, varovania pred kolíziou, informácie o stave ciest, regionálna predpoveď počasia, informácie o voľných parkovacích miestach, možnosť rezervácie parkovacieho miesta, informácie o cenách pohonných hmôt, turistické informácie o geograficky blízkych kultúrnych pamiatkach a udalostiach… Tieto dáta sú vo VANET sieti šírené a následne spracované na jednotlivých uzloch.

Šírenie rôznorodých dát vo VANET je obmedzené predovšetkým prenosovým kanálom. Rozhodnutia o konkrétne šírených dátach musia byť preto vykonané tak, aby nedochádzalo k znemožneniu komunikácie kvôli preťaženiu komunikačného kanála. Preto všetky uzly patriace do VANET musia "zvážiť" aké dáta, kedy a komu pošlú.

[1] VANET – Vehicle Ad-Hoc Networks, *(automobilová Ad-Hoc sieť)*
[2] ITS – Inteligent Transportation Systems, *(inteligentné dopravné systémy)*
[3] OBU – On Board Unit, *(komunikačné zariadenie umiestnené v palubnej jednotke automobilu)*

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 169–172.

Zároveň je ale potrebné dodávať do siete aktuálne dáta. O tieto rozhodnutia sa starajú replikačné algoritmy, ktorých predstavenie je cieľom tohto článku.

2 Šírenie dát v Ad-Hoc

Vzhľadom na existenciu rôznych typov Ad-Hoc sieti (MANET[4], VANET, WMN[5], WSN[6], PAN[7]), ktoré sa skladajú z mobilných uzlov a statických uzlov (OBU, RSU[8]) existuje niekoľko prístupov šírenia dát. Základné delenia šírenia dát sú:

Z pohľadu počtu koncových uzlov:

- *one –to–all* model šírenia: uzol vysiela dáta všetkým dostupným uzlom,
- *one –to–one* model šírenia: uzol vysiela dáta konkrétnemu uzlu.

Z pohľadu uzla, ktorý dáta šíri [1]:

- *režim šírenia*: uzol šíri dáta do prostredia v periodických intervaloch,
- *režim na požiadanie*: umožňuje klientskemu uzlu požiadať priamo o údaje od zdrojového uzla,
- *hybridný režim*: ide o kombináciu dvoch predošlých.

Z pohľadu použitej technológie [2,3]:

- jednoduchá záplavová technológia (angl. Simple flooding),
- technológia založená na pravdepodobnosti (angl. Probability based),
- technológia založená na lokalizácii oblasti (angl. Area based),
- technológia založená na susedných uzloch (angl. Neighborhood based).

Všetky spomenuté postupy a technológie sú založené na šírení dát cez Pull alebo Push metódu resp. ich kombináciu. Ide o štandardný spôsob šírenia dát v bezdrôtovej sieti.

3 Požiadavky na replikačné algoritmy vo VANET

Špecifické vlastnosti VANET siete spôsobujú, že algoritmy určené na replikáciu dát v MANET sieti sú neefektívne. Preto sa od algoritmov, ktoré budú replikovať dáta vo VANET požaduje predovšetkým:

- schopnosť uzlov vytvárať (alokovať) zdroje dát,
- schopnosť uzlov zabezpečiť replikovanie aj neaktuálnych dát,
- vedieť pracovať vo VANET ako v multi-databázovom prostredí,
- považovať RSU za autoritu z pohľadu držiteľa originálnych dát,

[4] MANET – Mobile Ad-Hoc Networks
[5] WMN – Wireless Mesh Network
[6] WSN - Wireless Sensor Networks
[7] PAN – Personal Area Network
[8] RSU – Road Side Unit *(WIFI zariadenie umiestnené ako pevný bod cestnej siete vo VANET)*

- zabezpečiť funkčnosť systému napriek absencii informácii o vzťahoch medzi uzlami .

4 Replikačné algoritmy

Za účelom replikácie dát vo VANET som vytvoril šesť algoritmov, ktoré spĺňajú požiadavky z kapitoly 3, kladené na replikačný algoritmus. Pre replikácie dát bol použitý DDBS [9] AD-DB.FRI [11], ktorý je určený pre VANET a je vyvíjaný na Fakulte Riadenia a Informatiky na Žilinskej univerzite (FRI, ŽU). Algoritmy boli následne testované v simulačnom nástroji *adhocsim.fri* [7, 8, 9], ktorý je taktiež vyvíjaný na FRI, ŽU.

Ide o nasledovné algoritmy:

- **AORPlD**: *(Active OBU Replica Pull Dissemination)* Pre replikáciu používa predovšetkým PULL metódu šírenia dát. [5, 10]
- **AORPsD**: *(Active OBU Replica Pshl Dissemination)* Pre replikáciu používa predovšetkým PUSH metódu šírenia dát. [5, 10]
- **SPA**: *(Simple Pulling Algorithm)* OBU uzly vo VANET sa snažia iba o aktualizáciu replík, o ktoré majú v aktuálnom čase záujem. [4, 10]
- **IRA**: *(Independent Replication Algorithm)* Replikačný algoritmus, ktorý sa snaží vytvárať aj zdroje dát, ktoré nie sú na danom uzle požadované. [4, 10]
- **DRA**: *(Dependent Replication Algorithm)* Algoritmus podobný ako IRA, ale vykonávanie tela algoritmu je závislé od vzniku dopytu na OBU. [4, 10]
- **PDDA**: *(Push Different Data Algoritmus)* Algoritmus, ktorý sa snaží šíriť informácie, ktoré iné uzly nemajú. Zabezpečuje vyváženú distribúciu všetkých typov dát medzi uzlami. [6, 10]

Počas simulácií, ktoré boli zamerané na distribúciu dát vo VANET, bola zisťovaná predovšetkým spokojnosť uzlov s distribuovanými dátami, resp., či dáta prijaté mobilným uzlom neboli staré z hľadiska požiadavky uzla. Ako najefektívnejšie algoritmy boli vyhodnotené DRA a PDDA. Okrem relatívne dobre dosahovanej spokojnosti z pohľadu mobilných uzlov nevznikalo pri nich nadmerné zaťaženie komunikačného kanála.

5 Ďalšie pokračovanie výskumu

Úspešnosť replikačných algoritmov vo VANET je vyhodnocovaná z pohľadu zaťaženia komunikačného kanála a z pohľadu spokojnosti [10] mobilných uzlov s prijatými dátami. Zvyšovanie spokojnosti OBU uzlov s prijatými dátami preto naďalej ostáva

[9] DDBS – distribuovaný databázový systém

[10] Spokojnosť – hodnota, ktorú je možné vyčísliť v intervale od <0..1>, a vypočítať ju zvolenou funkciou príslušnosti. (*Napr. 1 – aktuálne dáta, 0- staré neaktuálne dáta.)*

cieľom tejto práce. Ďalším krokom v oblasti replikácií dát bude prenášanie väčších dát vo VANET a vyriešenie otázky bezpečnosti prenášaných dát.

Poďakovanie. Táto štúdia/publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt: *Centrum excelentnosti pre systémy a služby inteligentnej dopravy II*., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja. "Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

1. V., Kumar,: Mobile database systems. [s.l.]. Wiley-Interscience. 2006. Wiley Series on Parallel and Distributed Computing. ISBN 9780471467922.
2. Brad, Williams, – Tracy, Camp,: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing. New York, NY, USA : Lausanne, Switzerland : ACM, 2002. MobiHoc '02. ISBN 1-58113-501-7.
3. I.S.Committee: Wireless LAN Medium ACCESS CONTROL (MAC) and Physical Layer Specifications. 1997.
4. J., Janech, – A., Lieskovsky, – E., Krsak,: Comparation of strategies for data replication in VANET environment. In: 26th IEEE International Conference On Advanced Information Networking And Applications Workshop. Fukuoka Institute of Technology, Japan, March 2012.
5. A., Lieskovsky, – J., Janech, – T., Baca,: Data replication in distributed database systems in VANET environment. In: IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS). july 2011. ISBN 978-1-4244-9696-9.
6. Lieskovsky, A., Janech, J., Matiaško, K.: PDDA algorithm for DDBS in VANET. In: IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS). Beijin, China (jún 2012), ISBN: 978-1-4673-2007-8
7. Bača, T.: Adhocsim.fri, http://adhocsim.dotfri.info/
8. Bača, T., Kršák, E.: Adhocsim.fri - simulator of vanet applications. Journal of Information, Control and Management Systems 9(3 Spec. iss.), 153-157(2011)
9. Lieskovský, A., Janech, J., Ba£a, T.: Data replication in distributed database systems in vanet environment. In: ITST 2011 : 11th international conference on ITS Telecommunications. pp. 447-452. Piscataway, NJ: IEEE, St. Petersburg, Russia (august 2011)
10. Lieskovský, A.: Replikácia dát v ad-hoc sieťach. Ph.D. thesis, Ž□ilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Žilina (2012), dizertačná práca
11. Janech, J.: Distributed database systems in the dynamic networks environment. International Journal on Information Technologies and Security (1), 43-52 (2010)

The Smart Office Application Supported by a Living Lab Environment

Gabriel Lukáč[1,2], Karol Furdík[2]

[2] Technical University of Košice, Faculty of Electrical Engineering and Informatics, Letná 9, 042 00 Košice, Slovakia
gabriel.lukac@tuke.sk
[1] InterSoft, a.s., Floriánska 19, 040 01 Košice, Slovakia
{gabriel.lukac, karol.furdik}@intersoft.sk

Abstract. The paper presents the Smart Office application, designed and implemented within the FP7 project ELLIOT. The solution is based on the LinkSmart semantic middleware that provides an Internet of Things platform for interoperable functional connection of devices, sensors, services, and software systems. Co-creative development and adaptation of the application is supported by the Living Lab environment that enables monitoring and evaluation of user interactions in knowledge, social, and business aspects. The prototype implementation of the Smart Office application is presented together with the results obtained from experiments on occupancy sensing in an office.

Keywords: Internet of Things, semantic middleware, sensor data fusion, Living Lab, Smart Office application.

1 Introduction

The Internet of Things (IoT) is nowadays a progressive and widely applied research domain that includes the features of device networking, sensitivity, adaptability, transparency, ubiquity and intelligence embedded in various devices employed in everyday life activities of people [1], [3]. One of key issues that needs to be solved when designing an innovative IoT application is the maintenance and adjustment of networked devices in accordance with needs, preferences, and requirements of end users. This research challenge is addressed in the European FP7 project ELLIOT, http://www.elliot-project.eu, by employing so-called Living Lab approach [5] to facilitate a people-centric design. Namely, ELLIOT targets the development of an experimentation and co-creation platform for IoT applications, where the user interaction and experience is measured and evaluated by means of knowledge, social, and business characteristics [2]. The ELLIOT platform is built on the LinkSmart middleware [6], which is an IoT-enabling technology originally developed in the FP6 project Hydra, http://www.hydramiddleware.eu, and its further enhancements are ongoing

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 173–176.

within the FP7 project EBBITS, http://www.ebbits-project.eu. Host institutions of the authors were/are participating as development partners in all three mentioned projects and authors are actively involved in the development of respective solutions.

In the following sections, the Smart Office pilot of ELLIOT will be presented as adaptive IoT application that targets the energy efficiency, optimization of work conditions, and alignment of user preferences with business processes in an office.

2 The Living Lab approach and semantic IoT middleware

The concept of Smart Office is still rather new and, in comparison to the well-established Smart House, can be considered as mostly a research topic [4]. In ELLIOT, however, a co-creative environment of Living Labs is employed to evaluate the user interactions and adjust the Smart Office to the user needs that are quantified according to the KSB (Knowledge-Social-Business) experience model [2]. The KSB model is represented as an OWL ontology of concepts describing the knowledge, social, and business aspects of user's perception. Evaluation of these aspects is based on a semantic mapping of ontology concepts to key performance indicators (KPIs) extracted from the sensor data.

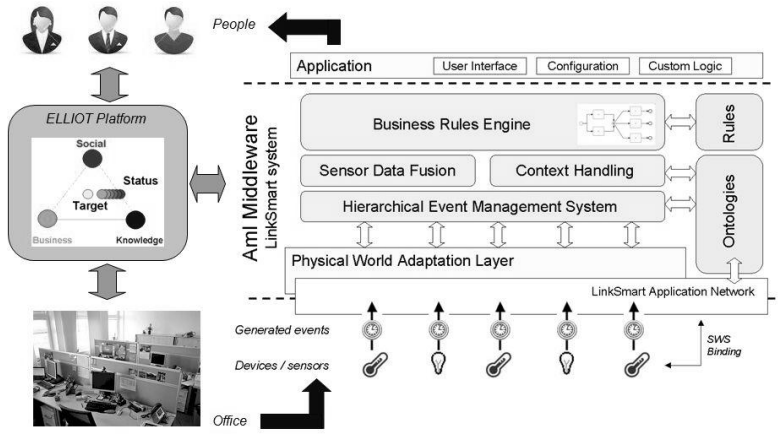


Fig. 1. The LinkSmart IoT middleware connected to the ELLIOT platform.

The high-level architecture of the Smart Office pilot application in ELLIOT is presented in Fig. 1. Devices installed in the office room are wrapped by a semantic web service interface (SWS Binding) provided by the LinkSmart middleware. Events generated by sensors and devices propagate through the Physical World Adaptation Layer to inner LinkSmart modules that provide capabilities for enriching events with proper context (including, e.g., historical data) and fusing events into more complex structures that are dynamically handled by executable business processes and rules. All these mechanisms are supported by system and domain ontologies implemented by means of OWLIM framework, http://www.ontotext.com/owlim. Inner modules of

LinkSmart are implemented in Java, device interfaces in C++, and the Business Rules Engine is built on JBoss Drools, `http://www.jboss.org/drools/`. Fused and semantically enriched sensor data are exposed to users by means of proper interfaces - for example, as a Smart Phone or web application. In parallel, interactions of employees in the office are monitored by a set of dedicated sensors and devices, processed by the LinkSmart middleware in accordance with specified business processes, and transformed to KPIs compatible with the KSB experience model of ELLIOT. According to the evaluation and comparison of monitored and desired KSB values (cf. the KSB triangle on the left hand side in Fig. 1), the Smart Office application can be adjusted and adapted to actual preferences of employees.

3 Implementation of the Smart Office application

The Smart Office pilot application of ELLIOT was installed in Košice, in premises of RWE IT Slovakia, s.r.o., and is maintained by the InterSoft, a.s., an application user partner of the ELLIOT project. The pilot application is located in the office of open space room type, occupied by eight employees. The schema of sensors and devices distributed in the office room is depicted in Fig. 2. Sensors for measuring energy consumption, outdoor/indoor temperature, and air humidity are built on Plugwise devices, while Arduino-based devices such as passive infrared motion sensors (PIR) and RFID readers were utilized for monitoring a presence of employees in the office.

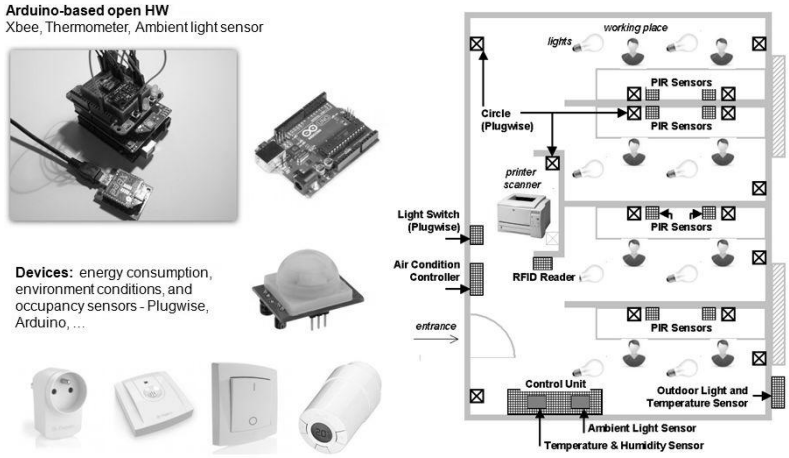


Fig. 2. Sensors and devices distributed in the office room.

Installation of the Smart Office prototype was accomplished in June 2012. First round of experiments was focused on the monitoring of a presence of employees on their working places, which is important for further adjustment of working environment settings to personal preferences of individual employees. It is noteworthy to say that a simple motion sensor is not suitable for monitoring the presence in offices, because it

evaluates a movement of subjects. According to our experiments with PIR, the precision of a simple motion sensor is only 45-50%. To increase the precision, we have used a combination of PIR, RFID cards (which, however, require an additional, potentially disturbing, action from employees), and energy consumption measured on particular working places. Thanks to the events fusion mechanism of LinkSmart, it was possible to combine these three inputs and obtain occupancy sensing indications with the precision of 94%.

4 Conclusions and Future Work

As a part of the prototype implementation, we have developed a web-based Smart Office portal that serves as a system control and monitoring console for office employees. It enables to set up personal preferences on temperature, air conditioning, or light intensity, to view historical data on energy consumption, to plan business trips or vacations, etc. This web portal, together with installed sensors, will be employed in the second round of experiments focused on two scenarios, namely a competition on the best energy savers in the office, and an adaptation of office environment to a defined business process (e.g., a plan of projects, meetings, business trips, or vacations). Experiments on these scenarios are planned for autumn and winter of 2012. Based on the obtained results, the Smart Office application and the ELLIOT platform as whole will be updated regularly. Final release is expected to be available in February 2013.

Acknowledgments. The work presented in the paper was supported by the project "Experiential Living Lab for the Internet Of Things" (ELLIOT, `http://www.elliot-project.eu`), co-founded by the European Commission within the 7th Framework Programme under the contract No. 287560.

5 References

1. Ashton, K.: That 'Internet of Things'Thing. In: RFID Journal, June 22, New York (2009).
2. Bifulco, A., Santoro, R.: A Conceptual Framework for Professional Virtual Communities. In: International Federation for Information Processing, Vol. 186, pp. 417-424 (2005).
3. Cook, D. J., Augusto, J. C., Jakkula, V. R.: Ambient Intelligence: Technologies, Applications, and Opportunities. In: Pervasive and Mobile Computing, 5 (4), Elsevier B.V., pp. 277-298 (2007).
4. Röcker, C.: Services and Applications for Smart Office Environments - A Survey of State-of-the-Art Usage Scenarios. In: Proceedings of ICCIT'10, January 27-29, Cape Town, South Africa, pp. 385-401 (2010).
5. Schaffers, H., Budweg, S., Kristensen, K., Ruland, R.: A Living Lab Approach for Enhancing Collaboration in Professional Communities. In: Proceedings of the International Conference on Concurrent Enterprising, June 22-24, Leiden, Netherlands (2009)
6. SourceForge.net: LinkSmart Middleware. [online] Available at http://sourceforge.net/projects/linksmart/. Accessed: September 5, 2012.

Cloud Computing as a Platform for Distributed Data Analysis

Martin Sarnovský, Peter Butka

Faculty of Electrical Engineering and Informatics,
Department of Cybernetics and Artificial Intelligence,
Technical University Košice, Letná 9, 04200 Košice, Slovakia
Martin.Sarnovsky@tuke.sk, Peter.Butka@tuke.sk

Abstract. In this paper we describe the proposal of cloud computing platform for support of distributed data analysis tasks. Computational complexity of data mining tasks from large data tables is considerable. Therefore we propose cloud infrastructure used for increase of computational effectiveness, because particular models are built in parallel/distributed way. These cloud modules will be a part of more complex data analytical system, which design progress is presented at the within the paper.

Keywords: Cloud computing, Data analysis, Data/Text mining

1 Introduction

Cloud computing paradigm is not a new one. Main idea behind using of the term "cloud" in various contexts can be traced to 1990s, when this term was used to describe networks. Cloud computing is also often compared to both grid and utility computing [1]. There are four main service models of the cloud:

1. Infrastructure as a Service (IaaS) means on-demand providing of infrastructural resources. Examples of IaaS providers include Amazon EC2, GoGrid and Flexiscale.
2. Platform as a Service (PaaS) refers to provisioning of the platform resources, such as software development frameworks. In general it allows to customer deploy its own application on the cloud using the specific programming languages or development kits. Examples include Google App Engine, Microsoft Windows Azure.
3. Software as a Service (SaaS) represents the on-demand applications over the Internet. The applications are accessible via browsers and customer is not allowed to control the infrastructure. Examples of this concept are Salesforce.com, SAP Business ByDesign, or Google Docs.

There are various different types of clouds from the privacy perspective [4]:

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 177–180.

- Public cloud is a cloud on general public level. Service provider offers their resources as services on the Internet. Main disadvantage of using public clouds is data controlling and as well as security issues.
- Private or internal clouds are infrastructures that are designed to be used by specific organization. A private cloud may be built and managed by the organization isself or by external provider. Cloud of that type offers the highest level of security and data control.
- Community cloud can be viewed as a type of private cloud. The main difference is, that infrastructure is shared by group of organizations.
- Hybrid clouds combine the one or more aforementioned principles in order to remove the limitations of the particular models. Typical is a combination of public and private cloud models. Hybrid clouds offer more flexibility than both public and private cloud models. They are able to provide greater control and security over the data and still facilitate on-demand service expansion and contraction.

2 On-line Data Analysis Cloud-based Tool

Main benefit of proposed method is that it is planned to be used as a part of complex information system for on-line data analysis. Our main motivation is to provide coherent system leveraging of analytical cloud services and providing simple user interface for users as well as administration and monitoring interface, built on open source cloud platform such as Gridgain [2]. Proposed architecture is depicted on Fig. 1.

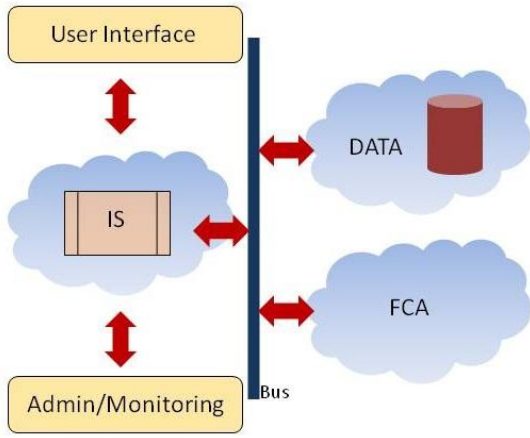


Fig. 1. Proposed architecture of data analysis tool.

Presented architecture consists of:

1. Data – cloud-based module that covers storage of the data, data access and various pre-processing methods.

2. Algorithms – cloud-based services providing particular analytical methods. Each analytical method or group of methods can be represented as one service (in Fig. 1 Formal Concept Analysis (FCA) analytical module as an example). Bus is used as an interconnection between other architecture elements.
3. IS – Information System that manages workflow of data analysis tasks and provides necessary interfaces for users as well as administrators.
4. Bus – element responsible for communication between the cloud-based architecture elements.
5. User Interface – web-based set of tools providing functionality of the system for standard users (data analyst, researchers, etc.)
6. Admin/Monitoring – system administration and monitoring interface, related to management of the infrastructure, resources and modules.

This architecture can be applied in various workflows. Data analyst will be able to open dataset in the user interface. Then he will be able to perform needed data pre-processing tasks which is handled by communicating with data-cloud, e.g. user can prepare his own object-attribute model from selected subset of data for analysis purposes. Various analytical modules (e.g. FCA, Clustering, Classification, etc.) can be executed on top of this data in order to retrieve the results [3]. The results in this case are in form of browsable output structure. Monitoring module can provide access to the cloud infrastructure and provide the task execution report, which includes statistics about resources usage, logs, availability of results, etc.

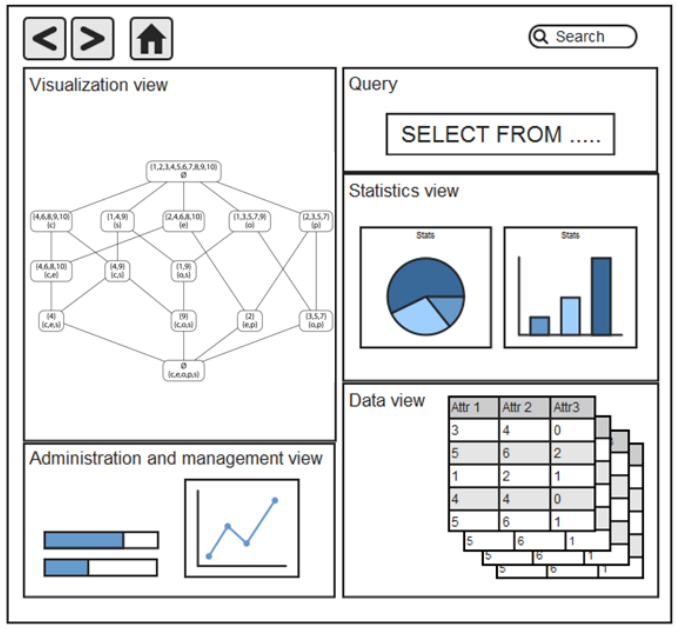


Fig. 2. Mock-up of on-line data analysis tool design.

Fig. 2 depicts mock-up design of proposed on-line data analytical tool. The main vision is to provide a portal-like structure consisting of these views:

- Data view provides browsing and selection of data tables from available data sets. User should be able to browse the data sets available in the data cloud.
- Query view provides the interface for selection of concrete data from one or more tables (including different subsets of objects and/or attributes) and combines them into user's own logical data table for further analysis.
- Statistics view provides simple visualization of main data characteristics from current selection.
- Visualization view provides visualization of resulted output model (according to selected data) which will be interactively browsable by user.
- Administration and management view serves for monitoring of cloud services and infrastructure, view will be accessible for users responsible for infrastructure testing and management.

3 Conclusion

We proposed the architecture of online data analysis tool based on cloud infrastructure. Such approach increases the computational effectiveness of particular methods by performing them in parallel fashion on the cloud. On the other hand, infrastructure is hidden from the data analyst user view, which can use the system through the user interface that provides tool functionalities. A main benefit of cloud infrastructure usage is in ability to provide computing and data platform for proposed on-line data analysis tool. Cloud provided performance is crucial as the complexity of data analysis algorithms increases with growing numbers of users and their respective selections of data subsets. This tool is expected to be used by students and teachers for educational purposes, as well as by data analysts and researchers.

Acknowledgements. This work was supported by the Slovak VEGA Grant No. 1/1147/12, and by the Slovak Research and Development Agency under contract APVV-0208-10.

References

1. I. Foster, C. Kesselman, Computional Grids, The Grid – Blueprint for a new Computing Infrastructure. Morgan Kaufmann, 1999.
2. GridGain 3.0 - High Performance Cloud Computing Whitepaper, available online, http://www.gridgain.com/media/gridgain_white_ paper.pdf, 2011
3. M. Sarnovský, P. Butka, J. Paralič, "Grid-based Support for Different Text Mining Tasks", In: Acta Polytechnica Hungarica, vol. 6, no. 4, pp. 5-27, 2009.
4. NIST Definition of Cloud Computing v15, available online, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-defv15.doc, 2011

Hodnotenie vlastnosti produktov na základe recenzií používateľov

Miroslav Smatana, Peter Smatana, Ján Paralič, Peter Koncz

Katedra kybernetiky a umelej inteligencie, FEI TU v Košiciach, Slovenská republika
miroslav.smatana@student.tuke.sk
{jan.paralic, peter.koncz}@tuke.sk
psmatana@gmail.com

Abstrakt. V tomto článku popisujeme návrh systému, ktorý bude slúžiť na hodnotenie vlastností objektov, ako sú tovary, služby, osoby, miesta, na základe recenzií používateľov. V článku je prezentované riešenie, ktoré na základe manuálne pripraveného korpusu recenzií bude vedieť vyhodnotiť ostatné recenzie o danom produkte bez potreby externých lexikónov. Ako produkt na ktorý sme toto riešenie aplikovali v pilotnej fáze sme zvolili mobilný telefón. Taktiež v článku nájdete model daného riešenia s popisom jeho jednotlivých blokov a ostatné možné spôsoby riešenia a využitie aspektovo orientovanej (analýza zameraná na hodnotenie vlastnosti produktov) analýzy sentimentu v praxi.

Kľúčové slová: analýza sentimentu, klasifikácia, spracovanie prirodzeného jazyka, anotácia, recenzie, produkty

1 Úvod

V posledných rokoch s narastajúcim počtom sociálnych sietí a internetových obchodov sa stáva trendom, že čoraz viac ľudí si tam chodí medzi sebou vymieňať svoje názory, rady, skúsenosti. Môžeme tam nájsť odpovede na mnoho našich otázok: aký mobil je pre mňa najlepší?, aké skúsenosti majú ľudia s danou cestovnou kanceláriou? a pod.

Pre získanie odpovedí na tieto otázky nám môžu pomôcť metódy spracovania prirodzeného jazyka, presnejšie analýza sentimentu. Najjednoduchšou a zároveň najčastejšou úlohou, ktorú analýza sentimentu rieši je klasifikácia informácie v textovej podobe do dvoch tried t.j. či informácia má pozitívny alebo negatívny charakter. Zložitejšími úlohami, ktoré táto disciplína môže riešiť je klasifikácia do viacerých tried. Konkrétne v našom prípade pri mobilných telefónoch to je zisťovanie či príspevok pojednáva o batérii, fotoaparáte, displeji atď. a ako danú vlastnosť ohodnocuje: veľmi dobre, dobre, neutrálne, slabo, veľmi slabo.

Metódy ktoré riešia túto problematiku môžeme rozdeliť na 2 skupiny [1] a to endogénne a exogénne. Tieto metódy sa odlišujú v údajoch použitých pri tvorbe modelu, ktorý je následne aplikovaný na klasifikáciu nových hodnotení.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 181–184.

Pri endogénnych metódach analýzy sentimentu je odhad sentimentu funkciou algoritmu a vzorky údajov, anotovanej vzhľadom k cieľovému atribútu. Nevyužíva žiadne externé znalosti.

V prípade exogénnych postupov je odhad sentimentu funkciou algoritmu, vzorky údajov, ako aj externých znalostí.

Systém AROPP označujeme za endogénnu metódu, pretože nevyužíva žiadne externé znalosti okrem znalostí, ktoré sú súčasťou algoritmu alebo použitej trénovacej vzorky.

2 Návrh architektúry systému AROPP

Systém AROPP je postavený na platforme Java s použitím externých knižníc. Obr.1. prezentuje architektúru systému, ktorého základné komponenty sú modul na získavanie dát, modul predspracovania a moduly vytvárania a aplikácie klasifikačných modelov. Ich bližší popis nájdete v nasledujúcich sekciách.

Získavanie dát. Jedným z najdôležitejších krokov pri práci s dátami je získanie kvalitných dát. My sme zvolili recenzie mobilných telefónov z internetového portálu Amazon[1]. Dáta získavame systémom Heritrix[2], ktorý nám slúži ako platforma pre crawlovanie dát z internetu.

Predspracovanie. Na predspracovanie dát používame existujúce moduly zo softvérového balíka RapidMiner[3], ktoré používame vďaka sprístupnenému API. Na kvalite predspracovania je závislá kvalita výslednej klasifikácie. HTML extraktor je použitý na získanie recenzií zo stránky produktu. Stemmer v našom prípade slúži na zmenšenie priestoru prehľadávania vo fáze klasifikácie, týmto spôsobom sa snažíme znížiť množstvo manuálne anotovaných dát. Ako stemmer používame Porterov stemmer [4]. Posledným krokom predspracovania je separácia viet, kde následne ďalšie moduly tieto vety budú klasifikovať na základe kategórie, do ktorej patria alebo na základe ich sentimentu.

Vytváranie klasifikačných modelov. Pre dosiahnutie dobrých výsledkov klasifikátora je dôležitým krokom zostaviť si vhodný doménový model. My sme si do našej pilotnej domény zvolili 6 aspektov, ktoré budeme vo vetách objavovať: rozmery a hmotnosť, displej, ovládanie, fotoaparát, batéria, pomer cena/výkon.

Ďalším krokom je manuálne anotovanie vzorky dát používanej vo fáze učenia. Na to použijeme vlastný anotačný nástroj navrhnutý špeciálne pre účely analýzy sentimentu viď Obr. 2.

Pri klasifikácii vzhľadom na doménový model ako aj pri analýze sentimentu viet sme sa rozhodli použiť na zmenšenie priestoru príznakov (príznakmi je n-gramový model viet) informačný zisk a ako klasifikátor sme zvolili Naive Bayes [5, 3].

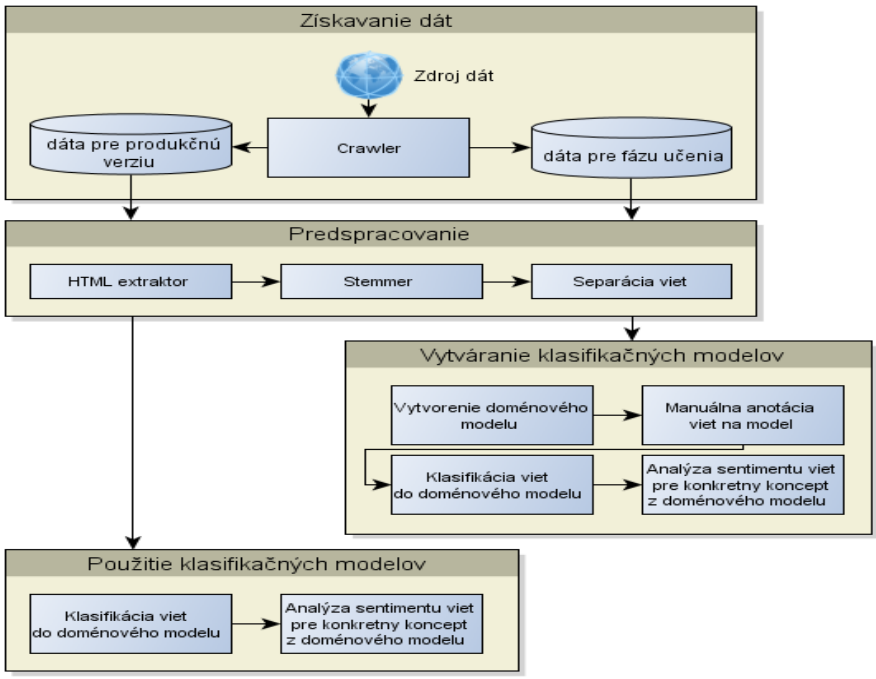
Použitie klasifikačných modelov. Vytvorené klasifikátory a otestované pri 10 násobnej krížovej validácii následne nasadíme do reálnej aplikácie. Kvalita

[1] http://www.amazon.com

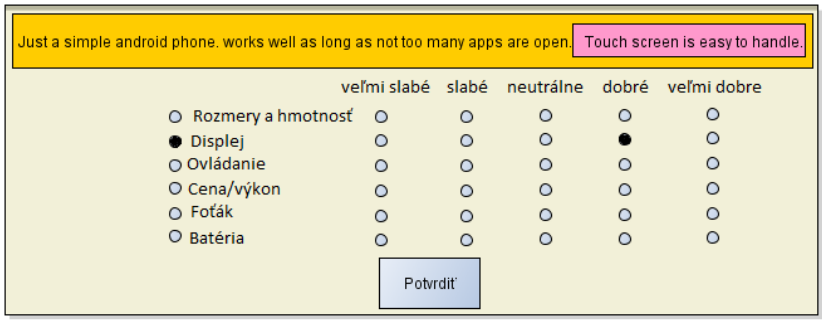
[2] https://webarchive.jira.com/wiki/display/Heritrix/Heritrix

[3] http://rapid-i.com/content/view/181/190/lang,en/

klasifikátorov záleží hlavne na tom ako bol vo fáze učenia pokrytý priestor danej domény príkladmi, na ktorých sme ich trénovali.



Obr. 1. Architektúra systém AROPP.



Obr. 2. Návrh anotačného nástroja.

3 Experimenty

Na overenie navrhovaného konceptu sme manuálne anotovali 112 viet (z toho 56 s pozitívnym hodnotením a 56 s negatívnym hodnotením danej vlastnosti) recenzií pojednávajúcich o displejoch mobilných telefónov.

Tabuľka 1. prezentuje vzťah medzi výskytom určitého n-gramu a jeho vplyvom na pozitívne či negatívne hodnotenie vety. Už bez použitia metód strojového učenia jednoduché štatistiky ukazujú, že by sme mohli byť schopný analyzovať sentiment viet a po dokončení vývoja systému by mohlo byť zaujímavé porovnať náš prístup s existujúcimi prístupmi.

Tab. 1. Frekvencia n-gramov v dvoch skupinách hodnotení z pohľadu ich sentimentu.

N-gramy	Pozitívne hodnotenia	Negatívne hodnotenia
good	21	1
great	14	0
Nice	11	0
awesome	5	0
Screen is very beautiful	6	0
terrible	0	8
bad	1	8
Screen is small	0	4
touchscreen is awful	0	2
touch screen sucks	0	4

4 Záver

Systém AROPP je vo fáze vývoja. V budúcnosti plánujeme znížiť množstvo manuálne anotovaných dokumentov pomocou metód kombinujúcich učenie bez učiteľa a učenie s učiteľom (resp. semisupervised lerning) [2]. Veľkým prínosom je modul samostatného anotačného nástroja, ktorý je špeciálne navrhnutý pre potreby aspektovo orientovanej analýzy sentimentu. Vďaka jeho jednoduchému používateľskému rozhraniu a anotácii už predspracovaného textu je manuálna anotácia rýchla a zjednodušená do maximálnej možnej miery.

Poďakovanie. Táto práca bola podporovaná Agentúrou na podporu výskumu a vývoja na základe zmluvy č. APVV-0208-10 (50%) a Vedeckou grantovou agentúrou Ministerstva školstva, vedy, výskumu a športu Slovenskej republiky a Slovenskej akadémie vied na základe zmluvy č. 1/1147/12 (50%).

Literatúra

1. Koncz, P. (2012). Aspektovo orientovaná analýza sentimentu, Písomná práca k dizertačnej skúške, Košice.
2. Nadeau, D. (2007). Semi-Supervised Named Entity Recognition, PhD thesis, University of Ottawa.
3. G. Forman, "An extensive empirical study of feature selection metrics for text classification," J. Mach. Learn. Res., vol. 3, pp. 1289-1305, 2003.
4. Porter, M.F. (1980), “An algorithm for suffix stripping”, Program, Vol. 14 No.3, pp. 130-137.
5. Mitchell, T. M. (1997). Machine Learning, McGraw-Hill, New York.

Systém na extrakciu informácií z homogénnych textových dokumentov

Peter Smatana

Katedra kybernetiky a umelej inteligencie,
FEI TU v Košiciach, Slovensko
psmatana@gmail.com

Abstrakt. Článok pojednáva o aplikovanom výskume v rámci projektu ITLIS, ktorého cieľom je vyvinúť vyhľadávací a vizualizačný nástroj nad dátami integrovanými z rozličných zdrojov. Zdrojom dát sú verejne prístupné registre obsahujúce informácie hlavne o ľudoch, firmách a iných organizáciach. V článku je prezentovaný nekonvenčný prístup k extrahovaniu dát z textových dokumentov. Tento prístup je založený na predpoklade, že dokumenty nie sú štrukturované, no popisujú rôzne subjekty zo špecifického uhla pohľadu. Tento popis býva na vysokej odbornej úrovni (napr. záznamy o pacientoch, rozhodnutia sudcov vo veciach obchodných spoločností), ktorý je laikom ťažko zrozumiteľný, no vyznačuje sa množstvom stabilných formulácií. Tieto formulácie systém Luwak identifikuje a následne transformuje do štrukturovanej formy. V rámci vývoja systému bol navrhnutý algoritmus na určovanie vzdialenosti dvoch reťazcov, viacvrstvová Levenshteinova vzdialenosť, ktorá je základom pre objavovanie nových vzorov.

Kľúčové slová: extrakcia informácií, Levenshteinova vzdialenosť, spracovanie prirodzeného jazyka

1 Úvod

Popisovaný systém je časťou projektu ITLIS[1], ktorého cieľom je vyvinúť vyhľadávací a vizualizačný nástroj nad dátami integrovanými z rozličných zdrojov. Zdrojom dát sú verejne prístupné registre obsahujúce informácie hlavne o ľudoch, firmách a iných organizáciach. Mnohé registre obsahujú homogénne[2] neštrukturované dáta v textovej podobe, ktoré je potrebné transformovať do tabuľkovej formy. Túto úlohu na seba preberá semi-automaticky anotačný systém Luwak[3]

[1] http://itlis.eu

[2] Pod homogénnou textovou množinou budeme rozumieť dokumenty, ktoré popisujú rôzne subjekty z rovnakého uhla pohľadu, čiže obsahom a formuláciami viet sú si veľmi podobné, no líšia sa v identifikátoroch, hodnotách, ktoré daný subjekt pre určitú vlastnosť nadobúda.

[3] Slovo luwak označuje v indonézčine zviera - cibetku, ktorá se živí hmyzom, tropickými plodmi a taktiež plodmi kávovníka, z ktorých sa vyrába najdrahšia káva na

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 185–188.

vyvíjaný v rámci tohto projektu. V nasledujúcich odsekoch bude predstavená architektúra systému ako aj algoritmus na určenie vzdialenosti dvoch reťazcov, viacvrstvová Levenshteinova vzdialenosť (Multilayer Levenshtein distance - MLD).

2 Extrakcia informácii

Úloha, ktorú sme sa podujali riešiť zapadá do rámca extrahovania informácii (IE) z textových dokumentov[5]. Bolo vynájdených mnoho metód pre IE, v počiatkoch to boli hlavne manuálne tvorené pravidlové systémy[2], neskôr začali dominovať systémy postavené na algoritmoch strojového učenia[1][4], tie však stále vyžadujú kvalitne anotované veľké trénovacie množiny. Z týchto dôvodov sa tieto metódy kombinujú s prvkami aktívneho a semi-kontrolovaného učenia, ktoré pomáhajú zmenšiť veľkosť trénovacej množiny. Tieto prvky používame aj v našom systéme, ktorého návrh je popísaný v nasledujúcej kapitole.

3 Návrh systému

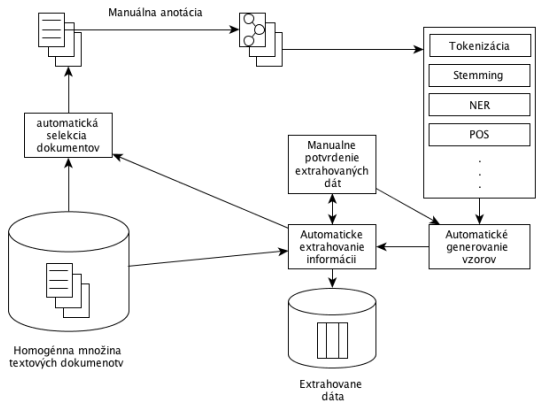
Systém Luwak je navrhnutý ako interaktívny systém, ktorý je v spolupráci s doménovým a znalostným expertom schopný transformovať neštrukturované textové dokumenty do štrukturovanej formy. Extrakčný proces môžme rozdeliť do dvoch fáz:

- Prípravná (oboznámenie sa s homogénnou množinou dokumentov; definovanie cieľa doménovým expertom; príprava množiny dokumentov; prvotný návrh anotačného modelu; výber a konfigurácia modulov predspracovania textov)
- Samotná extrakcia - iteratívny proces (viď Obr. 1)

V prvej fáze je nevyhnutná komunikácia doménového experta (ako používateľa extrahovaných dát) a znalostného inžiniera, ktorý vyhodnotí technické možnosti a riadi celý proces spracovania dokumentov. Samotná extrakcia je viac menej manuálna práca, v ktorej systém riadi tvorenie trénovacej množiny plus presnosť a návratnosť objavených extrakčných vzorov. Iteračný proces končí dosiahnutím úplného pokrytia vstupnej množiny dokumentov alebo jej definovanej časti.

Nie je možné sa vyhnúť manuálnej tvorbe trénovacej množiny ale pomocou znalosti z teórie učenia môžeme minimalizovať množstvo ručne anotovaných dokumentov. To je aj cieľom bloku automatickej selekcie dokumentov v kooperácii s blokom automatického extrahovania informácií.

svete. Cibetke prechádzajú zažívacím traktom len tie najchutnejšie plody kávovníka a týmto spôsobom ich v podstate "anotuje". Tu je možné vidieť paralelu, kde popisovaný anotačný systém sa snaží označiť časti textu s vysokou informačnou hodnotou.



Obr. 1. Schéma extrakčného procesu systému Luwak.

4 Viacvrstvová Levenshteinova vzdialenosť

V teórií informácii je Levenshteinova vzdialenosť[3] metrika nad reťazcami na meranie počtu rozdielov medzi dvomi sekvenciami. Levenshteinova vzdialenosť medzi dvomi reťazcami je definovaná ako minimálny počet zmien potrebných na transformovanie jedného reťazca na druhý s povolenými operáciami: vloženie, zmazanie a substitúcia jedného znaku. My sme ju modifikovali na určenie podobnosti dvoch sekvencií tokenov s tým, že pri porovnávaní sekvencií sa porovnávajú nielen samotné tokeny ale aj ich anotácie, ktoré môžu charakterizovať token na rôznych úrovniach: rozpoznané pomenované entity, syntaktická úroveň, sémantická úroveň, atď.

Viacvrstvovú Levenstheinovu vzdialenosť môžeme definovať ako minimálny počet zmien potrebných na transformovanie jednej postupnosti tokenov na druhú s povolenými operáciami: vloženie, zmazanie a substitúcia. Dva tokeny sa pokladajú za zhodné ak sú rovnaké alebo sa zhodujú v niektorej z anotácií, ktorá bola pridelená daným tokenom v štádiu ich predspracovania.

Ak by sme sa rozhodli porovnať nasledujúce dve vety: *Spoločnosť prenajíma Slovenskej sporiteľni predmetné BMW.* a vetu *Spoločnosť prenajíma Prvej správcovskej auto.* Zároveň zoberieme do úvahy, že vo fáze predspracovania boli rozpoznané pomenované entity: Slovenská sporiteľňa, BMW a Prvá správcovská. POS tagger nám okrem iných anotácii označil BMW a auto ako podstatne meno (v tomto prípade by bola vhodnejšia a potrebná syntaktická úroveň, čiže anotácia, že dané tokeny sú predmetmi viet). Vzdialenosť pomocou Levenshteinovej vzdialenosti by bola 4 no MLD pri porovnaní reťazcov by vrátila hodnotu 1 (slovo *predmetné* sa v druhej vete nevyskytuje), čo vystihuje podobnosť viet.

Nasledujúci pseudokód ukazuje ako je možne implementovať MLD. Zameriame sa na postupnosť tokenov s dĺžky n a postupnosť tokenov t dĺžky m, kde kazdý token postupnosti obsahuje p anotačných vrstiev.

```
int MLDistance(TokenAnnotation s[1..n, 1..p],
TokenAnnotation t[1..m, 1..p]) {
    declare int MLD[0..n, 0..m, 1..p], i, j, k cost
    for i := 1 to m  MLD[0, i] := infinity
    for i := 1 to n   MLD[i, 0] := infinity
    MLD[0, 0] := 0
    for i := 1 to n
        for j := 1 to m
            for k := 1 to p
            cost:= d(s[i, k], t[j, k])
            MLD[i, j, k] := cost + minimum(MLD[i-1, j   , 1..p],
                                           MLD[i   , j-1, 1..p],
                                           MLD[i-1, j-1, 1..p])
    return minimum(MLD[n, m, 1..p])}
```

5 Záver

I keď analýza neštrukturovaných dokumentov v prirodzenom jazyku je veľmi závislá na kvalite predspracovania a veľkosti trénovacej množiny, tak na základe obmedzenia množín dokumentov na homogénne sady očakávame aj so základnými modulmi na predspracovanie textových dokumentov a použitím techník aktívneho a semi-kontrolovaného učenia dosiahnutie vysokej kvality extrahovaných dát s minimálnym množstvom manuálnej anotácie.

Rýchle a efektívne porovnávanie jednotlivých kontextov anotácií je možné aj vďaka navrhnutému MLD algoritmu, ktorý umožňuje porovnávať reťazce cez viacero anotačných vrstiev, čo je nevyhnutné pre správnu funkčnosť extrakčných algoritmov.

Poďakovanie. Rád by som vyjadril svoje poďakovanie prof. Ing. Jánovi Paraličovi, PhD a svojím kolegom Ing. Petrovi Bednárovi, PhD. a Ing. Petrovi Kostelníkovi, PhD. za cenné rady a pripomienky pri vývoji systému Luwak.

Literatúra

1. *University of Massachusetts : Description of the CIRCUS System as Used for MUC-4*, 4 MESSAGE UNDERSTANDING CONFERENCE (MUC-4), Virginia, (1992)
2. G. Dejong. An overview of the FRUMP system. In *Strategies for natural language processing*. Hillsdale, NJ: Lawrence Erlbaum, (1982)
3. VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, (1966)
4. Scott Miller, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, Ralph Weischedel, and The Annotation Group. Algorithms that learn to extract information bbn: Description of the sift system as used for muc-7. In *In Proceedings of MUC-7*, (1998)
5. Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, (2006)

Vyhľadávanie zduplikovaného kódu

Ján Súkeník, Peter Lacko

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislava,
Ilkovičova 3, 842 16 Bratislava
sukenik08@student.fiit.stuba.sk, lacko@fiit.stuba.sk

Abstrakt. Kopírovanie častí zdrojových kódov na viacero miest, ako jeden zo spôsobov znovupoužitia kódu, je v programátorskej praxi bežne zaužívané. A to aj napriek tomu, že každý programátor nie len vie, prečo by nemal kódy kopírovať, ale určite už aj doplatil na následky zduplikovaného kódu. Medzi ne patrí napríklad viac úsilia, ktoré je potrebné vynaložiť na údržbu softvéru, ako aj zvýšený počet chýb, ktoré vznikajú upravením zduplikovaného kódu na niektorých miestach a inde nie. Existuje však len hŕstka nástrojov, ktoré vyhľadávanie zduplikovaného kódu umožňujú a ani tie sa v praxi nepoužívajú. Domnievame sa, že dôvodmi môžu byť ich vysoké výpočtové alebo pamäťové nároky alebo nedostačujúce výsledky. Navrhujeme preto spôsob vyhľadávania zduplikovaného kódu, ktorý využíva abstraktné syntaktické stromy a kombináciu viacerých existujúcich metód z dostupnej literatúry. Našim cieľom je vytvoriť prakticky použiteľný nástroj na inkrementálnu detekciu zduplikovaného kódu.

Kľúčové slová: porovnávanie zdrojových kódov, zduplikovaný kód, abstraktné syntaktické stromy

1 Úvod

Bežnou súčasťou vývoja softvérových systémov je kopírovanie jednej časti kódu na viacero miest. Doterajší výskum v oblasti údržby softvéru naznačuje, že programy obsahujú 5-10% zduplikovaného kódu [1, 3], čo určite nie je zanedbateľné množstvo. Zduplikovaný kód môže mať nepríjemné následky. Nekonzistentnosť zmien medzi jednotlivými duplikátmi môže spôsobiť chyby, čím sa v budúcnosti zvýši cena údržby softvéru. Napriek tomu nepoužívame nástroje, ktoré na duplikáty v kóde upozorňujú.

V literatúre bolo navrhnutých mnoho metód, pomocou ktorých je možné zduplikovaný kód odhaľovať [2, 3]. Medzi najčastejšie používané techniky patria: (1) *porovnávanie textu* – citlivé aj na malé zmeny formátovania, (2) *rôzne metriky kódu* – tiež citlivé na malé zmeny kódu, málo používané [2], (3) *tokenizáciu kódu* – tokenizáciou sa abstrahuje od konkrétneho textového zápisu zdrojového kódu, duplikáty sú odhaľované porovnávaním postupností tokenov, (4) *grafy závislostí* – porovnávajú sa grafy volania funkcií, hľadajú sa izomorfné podgrafy (vo všeobecnosti je to NP úplný problém), (5) *abstraktné syntaktické stromy* – tiež abstrahujú od textového zápisu kódu, porovnávajú sa stromy, ktoré reprezentujú zdrojový kód.

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 189–192.

Existujúce metódy sa líšia aj v spôsobe, ako prebieha hľadanie duplikátov: (1) *jednorazové* – vyhľadávanie prebehne naraz, v budúcnosti ho treba znovu celé opakovať, (2) *inkrementálne* – prebieha postupne, súčasne s tým ako sa zdrojový kód vyvíja.

2 Identifikácia zduplikovaného kódu

V našom riešení sme sa rozhodli pri porovnávaní zdrojových kódov používať abstraktné syntaktické stromy, pretože abstrahujú od textového zápisu kódu pričom informácia o jeho štruktúre ostáva zachovaná. Abstraktné syntaktické stromy sa na statickú analýzu používajú napríklad aj v mnohých vývojových prostrediach (IDE).

Ďalej sme sa rozhodli pre inkrementálnu detekciu duplikátov, ktorá je podľa nás dôležitá pre praktickú použiteľnosť nástrojov na hľadanie zduplikovaného kódu. Pri zmene zdrojových kódov stačí analyzovať iba zmenené časti (čo je menej výpočtovo náročné). Pri inkrementálnej detekcii duplikátov je však potrebné udržiavať index zdrojových kódov (jednorazové metódy index nepotrebujú).

Pri vývoji softvéru sú systémy na správu verzií (Git, SVN, atď.) nevyhnutnosťou. Myslíme, že by mohli zjednodušiť vytváranie a používanie nástrojov na odhaľovanie duplikátov, pretože udržujú informácie o tom, ktoré časti zdrojových kódov sa zmenili. Práve tieto informácie sú potrebné pri inkrementálnom odhaľovaní duplikátov.

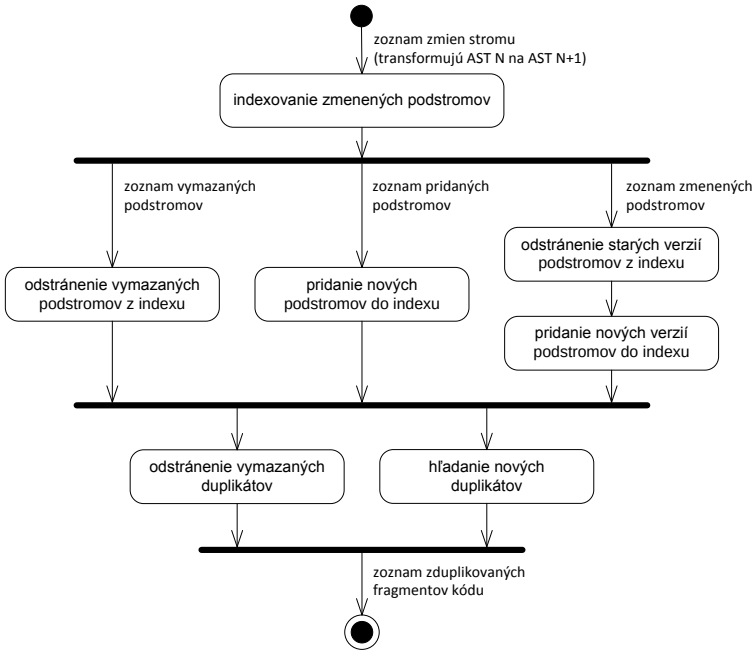
Na obrázku 1 je diagram, ktorý obsahuje návrh procesu jednej iterácie inkrementálneho odhaľovania duplikátov. Vstupom do procesu jej zoznam zmien abstraktných syntaktických stromov, ktoré nastali medzi dvoma verziami zdrojového kódu. Výstupom je zoznam skupín zduplikovaného kódu. Celý proces delíme na tri hlavné časti.

2.1 Indexovanie zmenených podstromov

Všetky podstromy, ktoré boli ovplyvnené zmenami v zdrojovom kóde, je potrebné opätovne zaindexovať. Na to by sme chceli využívať metódu *Exas* [5, 4]. Prezentovaná metóda pracuje s abstraktnými syntaktickými stromami zdrojových kódov a dokáže pracovať inkrementálne. Jej najväčšou výhodou je fakt, že umožňuje indexovať stromy takým spôsobom, aby ostávali zachované informácie o ich štruktúre, ale aby bolo zároveň možné odhaľovať aj čiastočné modifikované duplikáty.

2.2 Aktualizácia indexu

Po tom, ako sa pre všetky modifikované podstromy vypočítajú ich charakteristické vektory, musí byt aktualizovaný aj samotný index. Pri tom môžu nastať nasledujúce situácie: (1) starý kód bol v novej verzii vymazaný a príslušné podstromy treba z indexu odstrániť, (2) v novej verzii bol pridaný nový kód, ktorý sa musí do indexu vložiť a (3) treba odstrániť staré verzie zmeneného kódu a vložiť do indexu nové. Krok 3 je kombináciou prvých dvoch, no aj napriek tomu ho plánujeme spracovávať samostatne. Dôvodom je, že samotná informácia o tom, že nejaký zduplikovaný kód je zmenený, môže pomôcť presnejšie sledovať vývoj duplikátov.



Obr. 1. Všeobecný návrh jednej iterácie inkrementálneho odhaľovania zduplikovaného kódu.

2.3 Detekcia zduplikovaného kódu

Záverečným krokom je samotné vyhodnocovanie podobnosti zindexovaných fragmentov zdrojových kódov. Tento krok môže byť v niektorých prípadoch vykonávaný takmer súbežne s predchádzajúcim (t.j. porovnávanie jednotlivých fragmentov kódu už počas aktualizácie indexu [2]). Môže však vyžadovať aj ďalšie nezávislé spracovanie (napr. potvrdzovanie podozrení na duplikáty [5], alebo vyhľadávanie sekvencií zduplikovaných podstromov [2]). Pri detekcii duplikátov sme sa rozhodli použiť algoritmus, ktorý predstavili Nguyen et al. [5]. Má však nasledujúce nevýhody, ktoré by sme chceli v našej implementácii odstrániť:

1. hľadanie najväčších duplikátov autori navrhujú vykonávať až na záver, omnoho efektívnejším sa však zdá byť použitie takého indexu kódov, ktorý je usporiadaný podľa veľkosti jednotlivých fragmentov, ako používajú Chilowicz et al. [2],
2. prezentovaný algoritmus nespája odkopírované postupnosti podstromov do jedného duplikátu a preskakuje malé podstromy, čo znamená, že nevie odhaliť napr. zduplikované postupnosti príkazov, riešenia uvádzajú napr. [2] alebo [3].

Ďalšie vylepšenie, ktoré v súvislosti s detekciou duplikátov navrhujeme predpokladá, že máme k dispozícii aj zoznam zmien medzi dvoma verziami kódov. Často sa totiž stáva, že zduplikovaný kód je na jednom mieste upravený a na ostatné jeho výskyty sa zabudne [1]. Ak je zmena dostatočné veľká, upravený kód už nemusí byt ďalej algo-

ritmom považovaný za duplikát. Omnoho vhodnejším riešením by bolo upozorniť programátorov na to, že len jeden z duplikátov bol zmenený. Taktiež by bolo zaujímavé vyhodnotiť, ako často sa v reálnych softvérových projektoch stáva, že dva duplikáty sa časom vyvinú na odlišné kódy.

3 Zhodnotenie a ďalšia práca

Vytvorili sme testovaciu implementáciu algoritmu *Exas* pre jazyk Ruby, ktorá pre zadaný strom vytvorí jeho charakteristický vektor. Ten obsahuje počty výskytov tzv. n-ciest (postupnosť vrcholov dĺžky n smerujúca od koreňa k listom, pričom nemusí začínať v koreni ani končiť v listoch) [5, 4]. Z toho vyplýva, že čím zložitejší je kód, resp. jeho zodpovedajúci strom, tým viac rôznych n-ciest bude obsahovať a tým bude príslušný *Exas* vektor dlhší. Preto navrhujeme nasledujúce opatrenia:

1. *zjednodušenie abstraktného syntaktického stromu* – odstránenie listov alebo vrcholov, ktoré nie sú dôležité alebo nie sú vôbec potrebné a pod.,
2. *indexovať len niektoré časti stromu* – napr. len kód, ktorý je vo funkciách alebo metódach, vyššie úrovne stromu (definície tried a pod.) nemá zmysel analyzovať,
3. *obmedziť veľkosť indexovaných podstromov* – s využitím pohyblivého okna postupne indexovať len menšie podstromy (podobne ako v [3] s tokenmi).

Následne implementujeme samotný index zdrojových kódov a porovnávanie *Exas* vektorov s využitím lokálne senzitívneho hashovania [5]. Dôraz pri tom chceme klásť na rýchlosť každého z vyššie uvedených krokov procesu (sekcie 2.1, 2.2 a 2.3).

Poďakovanie. Táto práca bola čiastočne podporená projektom VG1/0675/11, APVV 0233-10 a vznikla vďaka podpore v rámci OP Výskum a vývoj pre projekt ITMS: 26240220039, spolufinancovaný ERDF.

Literatúra

1. Baxter, I., Yahin, A., Moura, L., Sant'Anna, M., and Bier, L. Clone detection using abstract syntax trees. In Proceedings. International Conference on Software Maintenance (1998), IEEE Comput. Soc, pp. 368-377.
2. Chilowicz, M., Duris, E., and Roussel, G. Syntax tree fingerprinting for source code similarity detection. In 2009 IEEE 17th International Conference on Program Comprehension (May 2009), IEEE, pp. 243-247.
3. Hummel, B., Juergens, E., Heinemann, L., and Conradt, M. Index-based code clone detection: incremental, distributed, scalable. 2010 IEEE International Conference on Software Maintenance (Sept. 2010), 1-9.
4. Nguyen, H., Nguyen, T., Pham, N., and Al-Kofahi, J. Accurate and efficient structural characteristic feature extraction for clone detection. Approaches to Software, 1 (2009)
5. Nguyen, T. T., Nguyen, H. A., Al-Kofahi, J. M., Pham, N. H., and Nguyen, T. N. Scalable and incremental clone detection for evolving software. 2009 IEEE International Conference on Software Maintenance (Sept. 2009), 491-494.

Usporiadanie dokumentov podľa relevancie k dopytom na základe analýzy prepojení medzi nimi

Martin Šeleng

Ústav Informatiky, Slovenská Akadémia Vied, Bratislava, Slovensko
Dúbravská cesta 9, 845 07 Bratislava
martin.seleng@savba.sk

Abstrakt. V príspevku priblížime jednotlivé algoritmy na usporiadanie dokumentov podľa relevancie na základe dopytu používateľa a na základe analýzy prepojení (grafu) medzi dokumentmi relevantnými k tomuto dopytu. Rozoberieme najznámejšie algoritmy používané na analýzu prepojení ako sú: PageRank a jeho modifikáciu TrustRank, Hits a Salsa. Následne medzi sebou tieto algoritmy porovnáme a uvedieme výhody a nevýhody jednotlivých algoritmov.

Kľúčové slová: usporiadanie dokumentov, vyhľadávanie dokumentov, PageRank, TrustRank, Hits, Salsa

1 Úvod

V článku predstavíme rôzne spôsoby/algoritmy usporiadania dokumentov[1] na základe analýzy prepojení medzi nimi. V druhej kapitole si predstavíme najznámejšie algoritmy používané veľkými vyhľadávačmi ako sú: Google[2] (PageRank [1] a TrustRank[3], Ask[4] (HITS [2]), a algoritmus SALSA [3][5]. V tretej kapitole zhrnieme výhody a nevýhody jednotlivých algoritmov. V poslednej kapitole priblížime aké problémy sa dajú pomocou opísaných algoritmov riešiť.

2 Usporiadanie dokumentov

Pri dnešnom rozsahu internetu nie je možné používateľovi vrátiť množstvo výsledkov, ale ich aj správne zoradiť. Na ich utriedenie je teda potrebné použiť ďalšie prístupy. Jedným z nich je aj analýza grafu prepojení medzi jednotlivými

[1] V článku budeme uvažovať o dokumentoch ako o webových stránkach/dokumentoch ale, algoritmy sú použiteľné v ľubovoľnej doméne
[2] https://www.google.com/
[3] http://en.wikipedia.org/wiki/TrustRank
[4] http://search.ask.com/
[5] Autorom nie je zatiaľ známy žiaden vyhľadávací stroj pracujúci na základe algoritmu SALSA

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 193–196.

dokumentmi. Modelovanie aktivity náhodného používateľa webu sa dá reprezentovať ako orientovaný graf prepojení vychádzajúcich a vchádzajúcich na webové stránky. Všetky algoritmy vychádzajúce z hodnotenia „dôležitosti“ stránok na základe grafu prepojení vychádzajú z toho, že ak stránka *A* ukazuje na stránku *B*, tak stránka *A* hovorí, že stránka *B* je „asi“ dôležitá. Ak na stránku ukazujú dôležité stránky, tak aj odkazy tejto stránky na iné stránky sa stávajú dôležitými.

2.1 Algoritmus PageRank

PageRank je metóda spoločnosti Google na meranie „dôležitosti“ webových stránok a dokumentov [1]. Algoritmus PageRank je dostatočne známy, a preto sa mu nebudeme v opise venovať (viac o algoritme PageRank môže čitateľ nájsť v [4]).

2.2 Algoritmus HITS

Algoritmus HITS (Hyperlink-Induced Topic Search) niekedy označovaný aj ako: „hubs and authorities“, je podobne ako PageRank algoritmus založený na hodnotení webových stránok na základe ich vzájomných prepojení (liniek). Algoritmus bol vyvinutý a predstavený Jonom Kleinbergom v práci [2] a je to v podstate predchodca algoritmu PageRank. Tento algoritmus vznikol v časoch, keď web vyzeral ešte veľmi jednoducho: existovalo veľa stránok, ktoré neobsahovali priamo hľadanú informáciu („hubs“), ale slúžili len ako akési katalógy, ktoré obsahovali stránky kde túto informáciu možno nájsť („authorities“). Inými slovami dobrý „hub“ obsahuje veľa liniek na dobré „authority“ stránky a na dobrú „authority“ stránku ukazuje veľa dobrých „hub“ stránok.

Definícia 1. Nech $a(i)$ označuje „authority“ hodnotu a $h(i)$ „hub“ hodnotu dokumentu i (pododobne pre dokumenty x a y), potom „authority“ a „hub“ dokumentu i určíme nasledovne: $a(i) = \sum_x h(x)$ („authority” dokumentu i určíme ako súčet všetkých „hub“ hodnôt dokumentov, ktoré ukazujú na stránku i) a $h(i) = \sum_y a(y)$ („hub” dokumentu i určíme ako súčet všetkých „ authority“ hodnôt dokumentov, ktoré ukazujú na stránku i). Po každej iterácii hodnoty „hub“ a „authority“ znormalizujeme: $a(i) := \frac{a(i)}{c_a}$, $h(i) := \frac{h(i)}{c_h}$, kde $c_a = \sqrt{\sum_{i \in G} a(i)^2}$, $c_h = \sqrt{\sum_{i \in G} h(i)^2}$.

Opäť je tento proces iteračný a iterujeme až kým “authority” a “hub” nezačnú konvergovať. Celý tento výpočet sa dá zrýchliť pomocou maticového počtu, o čom hovorí nasledujúca definícia.

Definícia 2. Maticu $L^T L$ označujeme ako „authority“ maticu a zapisujeme: $A = L^T L$ a maticu LL^T označujeme ako „hub“ maticu a zapisujeme: $H = LL^T$, kde L je hyperlinková matica a hľadanie “hub” a “authority” vektorov je vyjadrené nasledujúcimi rovnicami: $a_k(i) = Aa_{k-1}(i)$ a $h_k(i) = Hh_{k-1}(i)$.

2.3 Algoritmus SALSA

Posledným algoritmom na určovanie relavntnosti stránok, ktorý si predstavíme je algoritmus SALSA (SALSA: the stochastic approach for link-structure analysis) pred-

stavený v článku [3]. Algoritmus SALSA vychádza z algoritmu HITS. Rozdiel oproti algoritmu HITS je v tom, ako sa tvorí graf z ktorého počíta charakteristiky. SALSA vytvára bipartitný graf obsahujúci vrcholy „hub“ a „authority“ stránok na základe grafu vráteného dopytom (root/base množina) a priľahlími vrcholmi. Jedna množina obsahuje „authority“ a druhý „hub“ stránky, pričom stránka môže byť zaradená do oboch množín súčasne. Nasledujúca definícia popisuje výpočet relevantnosti stránky podľa algoritmu SALSA.

Definícia 4. Prvky matice *H* definujeme nasledovne: $h_{ij} = \sum_{\{k|[i_h,k_a][j_h,k_a]\in G\}} \frac{1}{deg(i_h)} \frac{1}{deg(k_a)}$, kde $deg(x)$ je počet prepojení vychádzajúci, resp. vchádzajíci do vrcholu *x*. Podobne prvky matice *A* definujeme nasledovne: $a_{ij} = \sum_{\{k|[k_h,i_a][k_h,j_a]\in G\}} \frac{1}{deg(i_a)} \frac{1}{deg(k_h)}$.

Definícia 5. „Authority“ webových stránkov $\overrightarrow{a_{\iota+1}} = \overrightarrow{a_\iota} \cdot A$, kde A je „authority“ matica a $\overrightarrow{a_{\iota+1}}$ je riadkový vektor *i*+1 iterácie „authority“ webových stránok, $\overrightarrow{a_\iota}$ je *i*-ta iterácia „authority“ webových stránok. Podobne aj pre „hub“ webových stránok platí: $\overrightarrow{h_{\iota+1}} = \overrightarrow{h_\iota} \cdot H$, kde H je „hub“ matica a $\overrightarrow{h_{\iota+1}}$ je riadkový vektor *i*+1 iterácie „hub“ webových stránok, $\overrightarrow{h_\iota}$ je *i*-ta iterácia „hub“ webových stránok. „Hub“ matica H je definovaná ako: $H = L_R \cdot L_S^T$ bez nulových riadkov a stĺpcov. „Authority“ matica *A* definovaná nasledovne: $A = L_S^T \cdot L_R$, bez nulových riadkov a stĺpcov. L_R a L_S sú odvodené z matice L a majú v každom nenulovom riadku, resp. stĺpci všetky prvky vydelené sumou riadku, resp. stĺpca.

3 Porovnanie

Jednotlivé algoritmy sme testovali na rôznych grafových štruktúrach kde sme skúmali rýchlosť výpočtu, konvergenciu, výhody a nevýhody jednotlivých algoritmov. V tejto kapitole si zhrnieme výhody a nevýhody jednotlivých algoritmov:

- Algoritmus PageRank
  - hlavnou výhodou je, že sa počíta off-line,
  - nekonvergovanie v prípade neexistujúcich prepojení z koncového uzla grafu,
  - „Link spamming“ – Stránky, ktoré silným spôsobom (prakticky dookola) ukazujú jedna na druhú čím si umelo zvyšujú page rank ktorý následne predávajú zákazníkom, aby sa ich stránky vo vyhľadávačoch založených na hodnotení webových prepojení zobrazovali čo najvyššie. Riešením je algoritmus TrustRank,
  - PageRank počita iba hodnoty „authorities“.
- Algoritmus HITS
  - výhodou je, že sa počíta iba nad dokumentmi vrátenými na dopyt,
  - nevýhodou je zasa časová náročnosť na spracovanie dopytu, nakoľko výpočet relevancie je prevádzaný v čase dopytu,
  - dokument môže obsahovať mnoho „identických“ liniek na ten istý dokument na rôznych hostiteľoch (riešiteľné pomocou váhovania, priradením váhy identickým hranám, napr. váhami, ktoré sú nepriamo úmerné ich počtu),

  - prepojenia sú generované automaticky (napr. správy zverejnené na newsgroups),
  - ak nie je graf súvislý, algoritmus HITS nemusí konvergovať ku konkrétnemu riešeniu,
  - vie sa vysporiadať s link spammingom (nie až tak dobre ako PageRank),
  - algoritmus HITS nie je odolný voči tzv. Tightly Knit Community (TKC) efektu, ktorý vlastne hovorí, že veľmi silne prepojená množina webových stránok môže byť vrátená ako relevantná k dopytu aj keď s ním nič nemá, resp. spĺňa iba časť dopytu.
- Algoritmus SALSA
  - podobné výhody resp., nevýhody ako algoritmus HITS,
  - s TKC efektom sa s ním dokáže lepšie vysporiadať,
  - HITS v podstate závisí iba od množiny „hub“ avšak, algoritmus SALSA od oboch množín „authority“ aj „hub“,
  - lepšie filtruje webový spam ako HITS, ale nie tak dobre ako PageRank.

4 Záver

Predstavené algoritmy majú nemalý význam v prípade usporiadania výsledkov podľa akejsi relevantnosti a sú hojne používané vo vyhľadávačoch. Najčastejším prípadom je kombinácia „dôležitosti“ stránky na základe analýzy prepojení s relevantnosťou na základe podobnosti medzi dopytom a dokumentom. Na druhej strane sa dajú tieto algoritmy použiť na analýzu ľubovoľných grafov reprezentujúcich rôzne sociálne siete.

Poďakovanie. Táto práca je vyvíjaná s podporou projektov: TRA-DICE APVV-0208-10, VEGA 2/0184/10.

Referencie

1. Page, L., Brin, S.: The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems 33, 107-117 (1998).
2. Kleinberg, J.: Authoritative sources in a hyperlinked environment. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, (1998).
3. Lempel, R., Moran, S.: SALSA: the stochastic approach for link-structure analysis, ACM Transactions on Information Systems (TOIS), April 2001, vol. 19, issue 2, p. 131-160. ISSN:1046-8188 (2001).
4. Bieliková, M., Návrat, P., Barla, M., Bartalos, P., Ciglan, M., Hamar, J., Kiselkov, M., Laclavík, M., Mažgut, J., Máté, J., Suchal, J., Šeleng, M., Tvarožek, M., Vojtek, P.: Štúdie vybraných tém softvérového inžinierstva (3) ISBN 978-80-227-2701-3, 2007.

Zaznamenávanie aktivity výskumníka v digitálnej knižnici vedeckých zdrojov obohatené o poznámky

Jakub Ševcech, Mária Bieliková, Roman Burger, Michal Barla

Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava, Slovensko
sevcech08@student.fiit.stuba.sk, {bielik, barla}@fiit.stuba.sk,
roman.arnold.burger@gmail.com

Abstrakt. Pre efektívne poskytovanie informácií potrebujeme dobre poznať správanie sa konzumentov informácií. Zber informácií je však náročný aj preto, že základným predpokladom získania vhodnej sady údajov je dobrá motivácia používateľa použiť nástroj, ktorý zaznamenáva aktivitu. V tomto príspevku opisujeme prístup k zaznamenávaniu aktivity výskumníkov na webe, ktorý realizuje zbieranie informácií o navštívených webových stránkach a dopĺňa získané údaje o poznámky, ktoré predstavujú základnú motiváciu pre použitie nášho monitorovacieho nástroja. Pod poznámkami rozumieme záložky, tagy, zvýraznenia v texte a komentáre, ktoré k webovým stránkam (dokumentom) pridávajú samotní návštevníci webových stránok. Na zaznamenávanie aktivity sme vytvorili nástroj Annota, ktorý je realizovaný ako rozšírenie pre internetový prehliadač. Umožňuje pridávať poznámky do zobrazených webových stránok, ale aj PDF dokumentov zobrazených v prehliadači. Vytvorené záložky a poznámky je možné zdieľať so skupinami používateľov. Nástroj je realizovaný všeobecne pre ľubovoľné stránky, pričom však jeho motivačná časť je prispôsobená pre digitálnu knižnicu a navigáciu v nej začínajúcim výskumníkom.

Kľúčové slová: poznámky, web, tagy

1 Úvod

Na sledovanie aktivity používateľov na Webe sa používa niekoľko rôznych prístupov, ktoré sa odlišujú v tom, ktoré aspekty používateľovej aktivity sledujú a v tom, aké prostriedky využívajú na zachytávanie tejto aktivity. Prístupy na strane servera ako napr. analýza logov [9] dokážu odhaliť podrobné informácie o správaní používateľa v jednej doméne, nedokážu však zachytávať aktivitu používateľa, keď túto doménu opustí. Pre sledovanie aktivity používateľa v prostredí Webu je vhodné použiť riešenia na strane klienta [10] alebo proxy server [6], prípadne kombinácie týchto dvoch prístupov. Pomocou týchto riešení je možné sledovať tak navštívené dokumenty ako aj akcie používateľa pri prezeraní dokumentu, pričom tieto nástroje nie sú obmedzené len na stránky poskytované jedným severom.

Základným predpokladom pre zaznamenávanie aktivity používateľa je jeho identifikácia pre potreby previazania dokumentov, ktoré daný používateľ navštívil. Dvojice používateľ - dokument môžeme ďalej obohacovať o časové informácie, informácie

M. Bieliková, M. Šimko (Eds.): WIKT 2012, pp. 197–200.

o obsahu dokumentu a o podrobné informácie o aktivite používateľa pri práci s dokumentom. Príkladom informácií z obsahu dokumentu môžu byť histogram výskytov slov použitý pri indexovaní obsahu alebo kľúčové slová z textu dokumentu, ktoré sa dajú použiť napríklad na modelovanie záujmov používateľa [3]. Na získavanie detailných informácií o aktivite používateľa treba použiť nástroje na strane klienta, ktoré môžu sledovať napríklad trvanie zobrazenia dokumentu, pohyb myšou na obrazovke, zvýrazňovanie a kopírovanie textu ale napríklad aj nástroje, ktoré sledujú pohľad používateľa po obrazovke pomocou webovej kamery [7].

Zbieranie dát o aktivite používateľa má aj dimenziu ochrany súkromia. Ak uvažujeme tento aspekt, tak zbieranie dát na strane klienta dáva lepšie možnosti ochrany, pretože máme kontrolu nad zozbieranými dátami a tretím stranám poskytneme len tie informácie, ktoré používateľ povolí [10].

Zo samotných informácií o obsahu dokumentu len veľmi ťažko odvodíme vzťah používateľa k dokumentu. Na určenie, či používateľ dokument skutočne prečítal a na zistenie jeho názoru na dokument je potrebné získať explicitnú spätnú väzbu alebo odvodiť tieto informácie z implicitnej spätnej väzby. Samotné informácie o navštívení dokumentu [4] spolu z ďalšími typmi implicitnej spätnej väzby [7] sa dajú použiť ako indikátor záujmu o dokument.

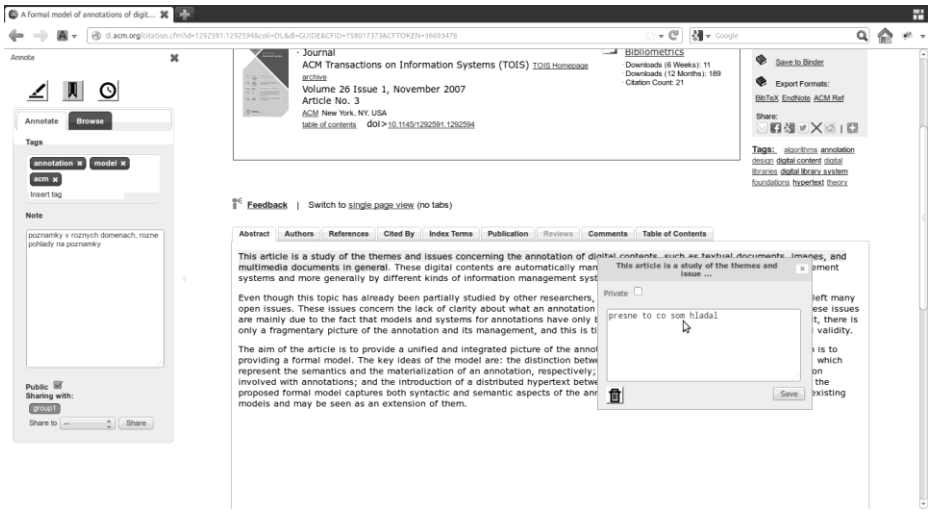
V našom prístupe sme k uvedeným indikátorom pridali aj poznámky priradené používateľom k dokumentom. Tieto poskytujú ďalšiu veľmi cennú spätnú väzbu, na základe ktorej sa dá vyhodnotiť záujem používateľa o dokument. Vďaka zvýrazneniam v texte vieme určiť aj presné časti dokumentu, ktoré používateľa najviac zaujali. Už samotný počet poznámok priradených k dokumentu vyjadruje záujem návštevníkov o dokument [1] a analýzou ich obsahu sa dajú získať presnejšie informácie o názore návštevníkov na obsah dokumentu.

Pri práci s poznámkami sa na ne dá pozerať dvoma spôsobmi [2]: (1) ako na metadáta, ktoré opisujú dokument z pohľadu jeho návštevníkov a (2) ako na ďalší obsah, ktorým obohacujú dokument jeho návštevníci. Získané poznámky sa dajú použiť ako implicitná spätná väzba, ako dodatočný obsah, ktorý obohacuje stránku a v prípade tagov aj ako kategórie, do ktorých si používateľ zatriedil navštívené stránky [5].

2 Zaznamenávanie aktivity a zbieranie poznámok

Navrhli a realizovali sme nástroj Annota, ktorý umožňuje návštevníkom webových stránok vytvárať záložky a pridávať poznámky k textu ľubovoľnej webovej stránky. Podporované poznámky tvoria tagy, komentár k záložke, zvýraznenia textu a k nim priradené komentáre. Nástroj sme realizovali pre účely experimentu ako rozšírenie pre internetový prehliadač Mozilla Firefox, ktoré umožňuje pridávať poznámky do zobrazených webových stránok, ale aj PDF dokumentov zobrazených v prehliadači.

Na obr. 1 je znázornené rozhranie na pridávanie poznámok do webových stránok. Vyhľadávanie a filtrovanie záložiek podľa tagov je možné prostredníctvom webového rozhrania. Vytvorené záložky a poznámky možno zdieľať so skupinou používateľov a tak spolupracovať na obohacovaní obsahu dokumentov. Počas práce s navrhnutým nástrojom sa vytvárajú záznamy o všetkých navštívených stránkach. Všetky údaje otýchto stránkach sa ukladajú na server spolu s poznámkami priradenými k záložkám.



Obr. 1. Rozhranie na pridávanie poznámok do webovej stránky.

Nástroj sme navrhli všeobecne a funguje ako služba na vytváranie záložiek do bežných webových stránok. Pre potreby experimentov v rámci projektu Tradice [8] sme ho prispôsobili pre použitie v digitálnych knižniciach. Sústredili sme sa na podporu začínajúceho výskumníka (napr. diplomanta alebo doktoranda) a jeho vedúceho, resp. vedúceho a jeho študentov pri spoločnom cestovaní v priestore digitálnej knižnice. Výskumníci môžu vytvárať skupiny a v rámci nich zdieľať dokumenty a poznámky, a tak môžu spolupracovať pri obohacovaní dokumentu, či hľadaní zaujímavých zdrojov, resp. priamom odporúčaní zdrojov. Vedúci diplomovej práce má takto prehľad o článkoch, ktorým sa študent venuje a zároveň mu môže odporučiť tie zdroje, ktorými by sa mal zaoberať. Vďaka podpore pridávania záložiek a poznámok do PDF dokumentov a zdieľania záložiek je možná spolupráca vedúceho a študenta a aj študentov navzájom pri vyhľadávaní zdrojov.

Nazbierané poznámky slúžia nielen na zapísanie si myšlienok, ale aj na podporu navigácie napríklad prostredníctvom filtrovania dokumentov na základe priradených tagov alebo pomocou vyhľadávania v texte poznámok.

Služby na vytváranie záložiek v navštívených stránkach sa v poslednej dobe tešia veľkému záujmu. Viaceré takéto služby ponúkajú možnosť pridávania poznámok do webových stránok alebo aj do PDF dokumentov (napr. Mendeley). Mnohé zverejňujú časť získaných údajov ako dátové vzorky určené pre výskum. Jednou z najznámejších takýchto služieb je služba Delicious, ktorá zverejňuje dátovú sadu v podobe zoznamu adries webových stránok a k nim priradených tagov [11]. Žiadna z nich však v súčasnosti neposkytuje verejne dostupnú dátovú sadu, ktorá by obsahovala stránky spolu s priradenými poznámkami.

Nami zozbierané údaje obsahujú okrem adries webových stránok a tagov aj časti textu, ktoré si používatelia zvýraznili a k nim pridané komentáre ako aj informácie o navštívení stránok, ktoré si používatelia nepridali medzi záložky alebo k nim nepridali žiadne poznámky spolu s časovou pečiatkou pre jednotlivé návštevy.

3 Zhrnutie a ďalšia práca

Našim cieľom je skúmať správanie výskumníkov pri cestovaní informačným priestorom za vedeckými zdrojmi. Nástroj Annota, ktorý sme pre tento účel vytvorili, umožňuje pridávať poznámky do bežných webových stránok a do PDF dokumentov zobrazených v internetovom prehliadači a zdieľať vytvorené záložky a poznámky so skupinami ďalších používateľov. Počas práce s nástrojom sa zaznamenávajú informácie o navštívených stránkach spolu s priradenými poznámkami a textom okolo nich. Nástroj je navrhnutý všeobecne pre prácu s bežnými webovými stránkami, avšak v dôležitej motivačnej časti pre jeho používanie sme sa sústredili na scenár začiatočníka výskumníka pri cestovaní v priestore vedeckých zdrojov v digitálnej knižnici.

Získaná dátová sada poslúži najmä na podporu navigácie, organizovania zbierky dokumentov a pri vyhľadávaní, veľký potenciál však má napríklad aj pri odporúčaní.

Poďakovanie. Táto práca bola čiastočne podporená projektami VG1/0675/11 a APVV 0208-10.

Literatúra

1. Agichtein, E., Brill, E., and Dumais, S. Improving web search ranking by incorporating user behavior information. In Proc. of the 29th int. ACM SIGIR conf. on Research and development in information retrieval pp. 19-26, ACM (2006).
2. Agosti, M., & Ferro, N. A formal model of annotations of digital content. In ACM Trans. Inf. Syst., 26 (2007).
3. Barla, M., and Bieliková, M. Ordinary Web Pages as a Source for Metadata Acquisition for Open Corpus User Modeling. In IADIS Int. Conf. WWWInternet, p. 227-233 (2010).
4. Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. Accurately interpreting clickthrough data as implicit feedback. In Proc. of the 28th int. ACM SIGIR conf. on Research and development in information retrieval, pp. 154-161, ACM (2005).
5. Körner, C., Kern, R., Grahsl, H.-P., and Strohmaier, M. Of categorizers and describers. In Proc. of the 21st ACM conf. on Hypertext and hypermedia, p. 157, ACM (2010).
6. Kramár, T., Barla, M., Bieliková, M. PeWeProxy: A Platform for Ubiquitous Personalization of the "Wild" Web. In UMAP 2011: Adjunct Proc. of the 19th Int. Conf. on User Modeling, Adaptation and Personalization. Demo. pp.7-9 (2011).
7. Labaj, M. Implicit Feedback Based Recommendation and Collaboration. In Information Sciences and Technologies. Bulletin of the ACM Slovakia Vol. 3 No. 4. pp. 41-42 (2011)
8. Návrat, P. et al. Od surfovania k personalizovanému cestovaniu v digitálnom informačnom priestore. In Proc. of WIKT 2011, Workshop on Intelligent and Knowledge Oriented Technologies, pp. 25-30 (2011).
9. Srivastava, J., Cooley, R., Deshpande, M., and Tan, P. N. Web usage mining: discovery and applications of usage patterns from web data. In *SIGKDD Explor. Newsl.*, Vol. 1, No. 2, pp. 12-23, (2000).
10. Šajgalík, M. Decentralised User Modelling and Personalisation. In Proc. of Informatics and Information Technologies Student Research Conf. – IIT.SRC'12, pp. 175-180 (2012).
11. Wetzker, R., Zimmermann, C., and Bauckhage, C. Analyzing Social Bookmarking Systems: A del.icio.us Cookbook. In Mining Social Data Workshop Proc., pp. 26-30 (2008).

Konverzačný obsah v kontexte bezpečnosti sociálnych sietí

Ján Štofa, Kristína Machová

Katedra kybernetiky a umelej inteligencie,
Fakulta elektrotechniky a informatiky,
Technická univerzita v Košiciach, Letná 9, 042 00, Košice
jan.stofa@tuke.sk, kristina.machova@tuke.sk

Abstrakt. Predkladaný príspevok sa sústreďuje na konverzačný obsah v komunikácii na sociálnej sieti. Definuje konverzačný obsah, jeho spracovanie a analýzu. Popisuje súčasný stav dolovania komunikačného obsahu. Zaoberá sa uplatnením komunikačného obsahu pri identifikácii jeho autora. V rámci príspevku je rozvíjaná aj problematika voľby vhodných aplikácií a techník dolovania komunikačného obsahu. Bližšie popisuje princíp fungovania aplikácie Bayesovho pravidla v rámci uvedenej problematiky. Zvolené aplikácie prepája do súvislosti s modelovaním tém.

Kľúčové slová: konverzačný obsah, techniky dolovania v texte, bezpečnosť, sociálna sieť

1 Úvod

So stále rastúcim využívaním internetu sa počítačovo sprostredkovaná komunikácia cez textové správy stala populárnou. Tento typ elektronickej diskusie je pozorovaný v „point-to-point“ alebo „multicast“ textovo založených on-line službách ako sú serverové chaty, diskusné fóra, e-mail a textové služby, diskusné skupiny, IRC (Internet Relay Chat), „blogy“ a "mikro-blogové" platformy (napr. „Twitter“ a „Facebook“). Tieto služby vytvárajú veľké množstvo textových dát, ktoré poskytujú zaujímavé možnosti výskumu a dolovania týchto údajov. Predložený príspevok sa sústreďuje na konverzácie, ktoré sa zaoberajú niektorými špecifickými témami.

Objavenie konverzačného obsahu (ďalej KO) predstavuje zaujímavé príležitosti pre analýzu autorstva. Čím viac sa elektronická diskusia stáva populárnou formou komunikácie, tým viac je dôležité dolovanie a odhaľovanie nelegálnych činností týchto elektronických diskusií. Objavovanie významu, hlavne anomálnych alebo podozrivých vzorov, je dôležitou úlohou v mnohých aplikáciách monitorovania chatu. Rozpoznanie významných vzorov môže byť uskutočnené dvoma spôsobmi - objavovaním základnej štruktúry alebo závislosti na dátových tokoch a objavovaním neobvyklého správania, ktoré vyšle signálny alarm.

V našom príspevku sa snažíme zaostriť pozornosť na druhý spomínaný spôsob. V rámci tohto prístupu sa nám zdá byť vhodná metóda sledovania KO v on-line

komunikácii, a na jej základe realizovaná identifikácia autora pomocou vopred definovaných vzorov.

2 Konverzačný obsah – spracovanie a analýza

KO súvisí s kriticky dôležitou otázkou, a to identifikáciou autora. KO nie je ani písanie, ani reč, ale skôr písaný prejav alebo hovorené písanie, či skôr niečo jedinečné. Vzhľadom k svojmu prevažne neformálnemu charakteru, sa KO líši od štandardných textov hlavne syntakticky (frekvencia slov, použitie interpunkčných znamienok, slovosled, úmyselné preklepy). Neformálny charakter KO presnejšie odráža autorovu osobnosť. Tak môžu analýzy KO poskytnúť kľúče o oboch atribútoch - autorovi diskusie a atribútoch samotnej diskusie.

Techniky spracovania KO by mali byť vyvinuté tak, aby boli schopné tieto informácie sprístupniť pre ďalšie podrobnejšie analýzy, ktoré fungujú na princípe modelovania tém a identifikácie autora. Prvý krok, pravdepodobne veľmi potrebný bez ohľadu na dátový súbor, je jednoznačná identifikácia aktérov konverzácie, aby sa predišlo možným problémom s identifikáciou autora a profiláciou modelov. Významným krokom predspracovania je segmentácia konverzácií na kratšie sekcie, v ktorých predmet konverzácie je relatívne samostatne uzavretý a konzistentný. To umožní lepšie využitie KO pri identifikácii jeho autora. Problém, ktorý predstavuje používanie „chatroom“ dát, je použitie veľmi neštandardnej slovenčiny (alebo angličtiny, či taliančiny). Tento problém je najvypuklejší pri spracovaní textu hovorového pravopisu pochádzajúceho zo správ mobilného telefónu. Štandardný prístup v rámci IR (Information Retrieval) pre dokumenty napísané v publikovanej slovenčine (napr. novinové články, výskumné štúdie,...) je odstránenie stopových slov (stopwords), ktoré pôsobia ako skrutky a matice gramatiky a samotné nenesú žiaden sémantický obsah. Odstránenie koncoviek umožňuje vyťažiť koreň (stem) slova.

3 Aplikácie a techniky dolovania konverzačného obsahu

V poslednom desaťročí bola vyvinutá technika Language Modelling (LM) [1] v Information Retrieval (IR), ktorá je založená na matematickej teórii pravdepodobnosti a zásade, že dokumenty a dopyt možno považovať za vygenerované stochastickým procesom. V rámci LM pre IR, sú dokumenty zoradené podľa vierohodnosti (likelihood) daného dopytu. Pravdepodobnosť, že dokument D vyhovuje danému dopytu Q možno vypočítať pomocou známeho Bayesovho pravidla takto: $P(d|q) \approx P(q|d)P(d) \approx P(d)\prod_{t \in q} P(t|d)$.

Na druhej strane tieto pravdepodobnosti termu môžu byť vypočítané pomocou rôznych metód odhadu. Základná relatívna početnosť nefunguje správne kvôli konečným dĺžkam väčšiny dokumentov. Ide o problém, ktorý je spôsobený druhmi extrémne krátkych dokumentov (menej ako 10 slov), ktoré sú prítomné v KO. Tento problém je zmiernený použitím rôznych foriem vyhladzovania (Jelinek-Mercer alebo Dirichletove vyhladzovanie) [2]. V rámci KO, termová distribúcia jednotlivých dokumentov (jeden komentár v rozhovore), môže byť dokument vyhladený buď na základe obsahu zvyšku konverzácie $P(t|c)$, alebo na základe všeobecnej frekvencie

termov vo všetkých konverzáciách $P(t|\cup_i c_i)$, takže $P(t|d) \approx \lambda_1 P_{ML}(t|d) + \lambda_2 P_{ML}(t|c) + \lambda_3 P_{ML}(t|\cup_i c_i)$, čo je Jelinek-Mercer vyhladzovanie predbežného výpočtu váženým súčtom základnej relatívnej početnosti, kedy vyhladzovacie parametre zodpovedajú $\sum_i \lambda_i = 1$ [3]. Topic modelling (TM) je technika, ktorá rozširuje LM tým, že prijíma zložitejšie štatistické procesy na generovanie dokumentov. Každý dokument je zobrazený ako kombinácia tém, kde každá téma je definovaná distribúciou prostredníctvom slov $P(t|z)$. Tieto distribúcie sa možno naučiť priamo z korpusu pomocou štandardných štatistických metód, ako je Gibbsovo vzorkovanie [4].

Dokumenty, ktoré sú predmetom autorských štúdií sú rozdelené do troch kategórií - identifikácia autorstva, podobnosť detekcie, charakterizácia autorstva [5]. Príchod modernej výpočtovej techniky umožnil použitie sofistikovanejších štatistických metód strojového učenia na analýzu autorstva - Skryté Markovovské modely, Regresné modely, Diskriminačné analýzy, Princíp analýzy komponentov, Klasifikačné algoritmy (napr. K-najbližších susedov (kNN), Naivný Bayes, Podporný vektorový mechanizmus (Support Vector Machines - SVM), Genetické algoritmy a Neurónové siete). Techniky dolovania textov boli použité v mnohých rôznych oblastiach a na rôznych typoch dokumentov [6], ktoré však boli len nedávno aplikované na konverzačný obsah a iba niekoľkými výskumnými pracovníkmi. V publikácii od S.C. Huiho [7] bol systém, nazvaný IMAnalysis, navrhnutý tak, aby podporoval analýzu inteligentnej správy pomocou techník dolovania textov. Tento systém poskytuje funkcie na vyhľadávanie správ chatu, analýzu sociálnych sietí a analýzu tém. Vyhľadávanie chatových správ poskytuje všeobecné prehliadanie a podporu vyhľadávania. Analýza sociálnej siete objavuje sociálne interakcie IM chatových používateľov s ich kontaktmi a analýzami tém, pričom automaticky detekuje témy, do ktorých je IM používateľ zapojený. Identifikácia tém sa vykonáva pomocou klasifikátora založeného na Naivnom Bayesovom klasifikátore, Asociatívnych klasifikáciách a modeloch SVM klasifikácie.

Existujú dva súperiace modely, ktoré sa používajú na definovanie problému dolovania v chate - *termovo založený* a *štýlovo založený prístup*. Podľa prof. F. Crestani [8] sa žiadny z týchto prístupov nesnaží riešiť súčasne identifikáciu obsahu a autorstvo KO. Aj napriek tomu, že sú tieto dve výzvy často považované za odlišné, existuje silný vzťah a závislosť medzi témou konverzácie a autorom tejto konverzácie. Je tiež zrejmé, že tak ako rôzni autori môžu diskutovať na rovnakú tému rôznymi spôsobmi, môžu tí istí autori diskutovať o rôznych témach pomocou rôznych štýlov. Preto zdieľame názor, ktorý vyslovil autor publikácie [8], že je možné využiť túto závislosť v kombinácii identifikácie tém a identifikácie autorov KO.

4 Záver

Štandardné IR a techniky dolovania textov boli navrhnuté tak, aby dokázali spracovať štandardné dokumenty (napr. novinové články a obsah webových stránok). Nerátalo sa však s krátkym a neformálnym KO. K dispozícii, ale máme štatistické modelovanie textu pomocou LM a TM techník, ktoré sa javí byť vhodné pre rozšírenie riešenia s extrémne krátkym a gramaticky chybným textom pochádzajúcim z on-line konverzácií. Takýto postup bol použitý aj v oblasti počítačového súdnictva. V rámci neho bola uskutočnená analýza chatovacích miestností na identifikáciu podozrivého

správania sa. Keď polícia identifikuje buď podozrivú internetovú pedofíliu alebo jednu z možných obetí, skopírujú sa obsahy všetkých pevných diskov získaných počas vyšetrovania [9]. Tie často obsahujú záznamy o činnostiach chatovacích miestností. Tieto záznamy umožňujú polícii podložiť ich tvrdenia a identifikovať ďalšie obete či páchateľov. Avšak množstvo údajov obsiahnutých v týchto záznamoch je tak veľké, že si to vyžaduje veľké množstvo ľudského úsilia, aby si pracovník prečítal všetky zaznamenané konverzácie. Mnoho z týchto rozhovorov môže byť v skutočnosti benígneho charakteru, pretože páchatelia využívajú internet aj na iné účely. Je zrejmé, že vhodný softvér môže zúžiť toto hľadanie a znížiť množstvo textu, ktorý policajti musia čítať. Ako tvrdí [8], podobných pokusov už bolo viacero, no požadovaný výsledok sa nedostavil. Je potrebné vyvinúť viac sofistikované modely, ktoré by sa vzťahovali na sledovanie KO kvôli bezpečnosti. Obavy verejnosti z internetovej pedofílie a terorizmu nikdy neboli väčšie ako dnes, a tak technológie, ktoré pomáhajú polícii zatknúť páchateľa, či identifikovať obete, sú potrebné.

Poďakovanie. Tento príspevok vznikol s podporou KEGA grantu MŠVVaŠ SR č. 065TUKE-4/2011 (50%), VEGA grantu MŠ SR č. 1/0042/10 „Metódy identifikácie, anotovania, vyhľadávania, sprístupňovania a kompozície služieb s využitím sémantických meta-dát pre podporu vybraných typov procesov“ (50%).

Referencie

1. W.B Croft and J.D Lafferty, editors. Language Modelling for Information Retrieval. Kluwer Academic Publisher, Dodrecht, The Netherlands, 2003.
2. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst., 22(2):179-214, 2004.
3. Language modeling [on-line] [cit 2012.9.3]. Dostupné na internete: <http:// janstas.weebly.com/uploads/9/1/7/1/9171281/stas_itss_modelovanie_jazyka.pdf>.
4. T. L. Griffiths and M. Steyvers. Finding scientific topics. Proceedings of the National Academy of Science, 101:5228-5235, 2004.
5. T. Kucukyilmaz, B.B Cambazoglu, C. Aykanat, and F Can. Chat mining: Predicting user and message attributes in computer-mediated communication. Information Processing and Management, 44(4):1448-1466, 2008.
6. M. Berry. Survey of Text Mining : Clustering, Classification, and Retrieval. Springer, September 2003.
7. S.C. Hui, Yulan H., and Haichao D. Text mining for chat message analysis. In IEEE Conference on Cybernetics and Intell.Systems, pages 411-416, Sept. 2008.]
8. R. Bache, F. Crestani, D. Canter, and D. Youngs. A language modelling approach to linking crimes styles with offender charateristics. In Proceedings of NLDB 2008, pages 257-268, London, UK, June 2008.)
9. R. Bache, F. Crestani, D. Canter, and D. Youngs. Mining police digital archives to link criminal styles with offender characteristics. In Proceedings of ICADL, pages 493-494, Hanoi, Vietnam, 2007.).

Register autorov

Mária Bieliková, Marián Šimko (Eds.)

WIKT 2012: 7th Workshop on Intelligent
and Knowledge Oriented Technologies

Zborník príspevkov

1.vydanie
222 strán, 100 výtlačkov
Tlač Nakladateľstvo STU v Bratislave
Rok vydania 2012

ISBN 978-80-227-3812-5

ISBN 978-80-227-3812-5