Proceedings in
Informatics and Information Technologies

WIKT 2014
9th Workshop on Intelligent
and Knowledge Oriented Technologies

Ladislav Hluchý, Mária Bieliková, Ján Paralič (Eds.)

WIKT 2014

9th Workshop on Intelligent
and Knowledge Oriented Technologies

Proceedings

November 20-21, 2014
Smolenice, Slovakia

INSTITUTE OF INFORMATICS
SLOVAK ACADEMY OF SCIENCES

WIKT 2014 was organized by

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

Editors

- Ladislav Hluchý
  Institute of Informatics, Slovak Academy of Sciences
  Dúbravska cesta 9, 845 07 Bratislava, Slovakia
  ladislav.hluchy@savba.sk
- Mária Bieliková
  Faculty of Informatics and Information Technologies,
  Slovak University of Technology in Bratislava
  Ilkovičkova 2, 842 16 Bratislava, Slovakia
  maria.bielikova@stuba.sk
- Ján Paralič
  Faculty of Electrical Engineering and Informatics,
  Technical University of Košice
  Letná 9, 042 00 Košice, Slovakia
  jan.paralic@tuke.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
- VEGA 2/0185/13 – New methods and approaches on information processing and knowledge bases

All contributions were reviewed by the Programme Committee and printed as delivered by authors without substantial modifications

Published by
Nakladateľstvo STU
Vazovova 5, Bratislava, Slovakia

ISBN 978-80-227-4267-2

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 20th to the 21st of November 2014.

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 eight 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 2014 workshop were:

- Knowledge technologies and their applications
- Big data, technical solutions, application studies
- Knowledge and information modeling, representation of semantics
- Analysis and processing of information sources (documents, electronic communication, databases, knowledge processes)
- Social web and its applications, Social Network Analyses
- Personalized web and its applications, recommendations
- Processing of information sources in Slovak language
- Semantic and service oriented architectures
- 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
- application paper
- special session for PhD-students around dissertation exam

A total of 27 papers were submitted, most of them as work-in-progress. The others were distributed among categories as follows: 5 papers submitted to the doctoral section, 3 as application papers, 3 as visionary contributions, 2 as research contributions and 2 as knowledge practices. Each contribution was reviewed by at least two members of the program committee.

As in the in previous year, the workshop was preceded by meeting of the TraDiCe project (Cognitive travelling in digital space of the Web).

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. We remember with respect and love our dear and excellent colleague Elena Svarinská, an important member of the organizing committee, who passed away suddenly.

November 2014, Smolenice, Slovakia

Ladislav Hluchý, Mária Bieliková, Ján Paralič

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 20. – 21. novembra 2014, kde sa konala tvorivá pracovná dielňa o inteligentných a znalostne orientovaných technológiách WIKT 2014.

Tento ročník nadviazal na tradíciu započatú v roku 2006 na Ústave informatiky Slovenskej akadémie vied v Bratislave. Séria pracovných dielní počas ôsmich 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 Smoleníc.

Hlavné témy dielne WIKT 2014 boli:

- znalostné technológie a ich aplikácie
- big data, možné prístupy, vhodné technológie
- cloud, technické riešenia, aplikačné príklady
- modelovanie informácií a znalostí, reprezentácia sémantiky
- analýza a spracovanie informačných zdrojov (dokumenty, elektronická komunikácia, databázy, znalostné procesy)
- sociálny web a jeho aplikácie, analýza sociálnych sietí
- personalizovaný web a jeho aplikácie, odporúčania
- 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 v rámci nasledujúcich kategórií:

- výskumný príspevok
- prebiehajúci výskum
- vizionársky príspevok
- znalostné praktiky
- aplikačný príspevok
- špeciálna sekcia pre doktorandov okolo dizertačnej skúšky

Celkovo bolo ponúknutých 27 príspevkov, väčšina z nich v kategórii work-in-progress. 5 príspevkov bolo prihlásených do doktorandskej sekcie, 3 do sekcie aplikačný príspevok, 3 do sekcie vizionársky príspevok, 2 do sekcie výskumný príspevok a 2 do sekcie znalostné praktiky. Každý príspevok posúdili minimálne dvaja členovia programového výboru.

Tvorivej dielni WIKT, podobne ako v minulom roku, aj tento rok predchádzalo pracovné stretnutie k projektu TraDiCe (Kognitívne cestovanie po digitálnom svete webu a knižníc s podporou personalizovaných služieb a sociálnych sietí).

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 stalo priestorom pre zanietené vedecké diskusie a pomohlo tak v šírení poznatkov a spolupráci. S úctou a láskou si spomíname na našu drahú a vynikajúcu kolegyňu Elenu Svarinskú, dôležitú členku organizačného výboru, ktorá nás náhle opustila.

November 2014, Smolenice

Ladislav Hluchý, Mária Bieliková, Ján Paralič

Workshop Organization

The 9th Workshop on Intelligent and Knowledge Oriented Technologies (WIKT), held on November 20–21, 2014 in Smolenice, was organised by the Institute of Informatics, Slovak Academy of Sciences in collaboration with Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava and Faculty of Electrical Engineering and Informatics, Technical University Košice.

Programme Committee

Chair

Hluchý, Ladislav – Institute of Informatics, Slovak Academy of Sciences

Members

Bieliková, Mária – Slovak University of Technology in Bratislava
Butka, Peter – Technical University of Košice
Homola, Martin – Comenius University in Bratislava
Horváth, Tomáš – Pavol Jozef Šafárik University in Košice
Chudá, Daniela – Slovak University of Technology in Bratislava
Krajči, Stanislav – Pavol Jozef Šafárik University in Košice
Laclavík, Michal – Institute of Informatics, Slovak Academy of Sciences
Mach, Marián – Technical University of Košice
Machová, Kristína – Technical University of Košice
Matiaško, Karol – University of Žilina
Návrat, Pavol – Slovak University of Technology in Bratislava
Paralič, Ján – Technical University of Košice
Rozinajová, Viera – Slovak University of Technology in Bratislava
Šaloun, Petr – VŠB - Technical University of Ostrava
Vojtáš, Peter – Charles University in Prague
Zendulka, Jaroslav – Brno University of Technology

Organizing Committee

Hluchý, Ladislav
Svarinská, Elena
Rajčániová, Soňa
Nguyen, Giang
Dlugolinský, Štefan
Krajčovič, Tibor
Šimko, Marián

Obsah

Analýza a spracovanie textu

Klasifikácia názorov v konverzačnom obsahu

Martin Mikula, Kristína Machová

Katedra kybernetiky a UI, FEI, Technická Univerzita Košice,
Letná 9, 042 00, Košice,
{Martin.Mikula,Kristina.Machova}@tuke.sk

Abstrakt. S rastúcimi možnosťami internetu v dnešnej dobe rastie aj počet jeho používateľov. Ľudia na webe medzi sebou stále viac a viac komunikujú. Táto komunikácia zohráva významnú úlohu aj v procese rozhodovania. Na základe toho vznikla požiadavka na analýzu obsahu rozsiahlych webových diskusií, takzvaného konverzačného obsahu, pomocou počítačov. Práve problematike analýzy názorov, konkrétne klasifikácii názorov sa venuje aj nasledujúci príspevok. Vytvorili sme preto algoritmus, ktorý umožňuje určiť polaritu príspevku. Pri analýze textu dokážeme spracovať aj intenzifikáciu, negáciu a ich kombinácie. Vytvorili sme 4 klasifikačné slovníky rozdelené podľa typov slov, ktoré obsahujú. Algoritmus sme následne otestovali pričom presnosť sa pohybovala od 33.6% do 95% a návratnosť od 28.7% do 90.4%.

Kľúčové slová: klasifikácia názorov, konverzačný obsah, webová diskusia, slovníkový prístup

1 Úvod

Klasifikácia názorov (opinion classification, sentiment classification) je proces, počas ktorého sa analyzujú názory a postoje používateľa k danej téme. Názor je určený hodnotiacim faktorom (môže byť pozitívny alebo negatívny) a silou. Sila hodnotiaceho faktora je závislá od stupňa intenzity polarity slov, ktoré sa týkajú danej témy a ich počtu. Pod pojmom téma rozumieme napr. hodnotenie produktov, osôb, kníh a podobne. Autorom alebo držiteľom názoru je osoba, ktorá má konkrétny názor na konkrétny objekt. Objekt je definovaný ako téma, na ktorú sa daný názor vzťahuje.

2 Postup pri klasifikácii názorov

V princípe sú známe dva základné odlišné prístupy k riešeniu problému klasifikácie názorov a to [1]:

- *slovníkový prístup*: založený na používaní slovníkov
- *prístup založený na strojovom učení*: najčastejšie používa metódy SVM (Support Vector Machine) a kNN (K-Nearest Neighbours)

Niekedy sa toto delenie uvádza ako delenie na endogénne a exogénne metódy [2].

2.1 Slovníkový prístup

Slovníkový prístup sa zameriava na vyhľadávanie tých slov v texte príspevkov, ktoré sú nositeľmi určitého postoja k téme, pričom sa ignorujú slová, ktoré nemajú subjektivitu. Určenie subjektivity slov predstavuje určitú formu predspracovania textu, keď sa slová roztriedia na tie, ktoré sú použiteľné pre ďalšiu klasifikáciu a ostatné. V ďalšom kroku sa určuje polarita každého subjektívneho slova. V najjednoduchšom prípade môže byť polarita slova pozitívna, negatívna a neutrálna. Polarita slov sa určuje na základe porovnania s vopred pripraveným slovníkom.

Avšak určovanie polarity výrazu s negáciou nie je tak triviálne. Rozlišujeme 2 typy určovania polarity negácie:

- *switch negácia*: hodnota slova sa mení na slovo s rovnakou silou ale opačnej polarity
- *shift negácia*: hodnota slova sa určuje posunom smerom k opačnej polarite o presnú hodnotu (napr. 4)

Ak aj v texte nemáme negáciu, často je potrebné po určení polarity subjektívneho slova určiť ešte silu tejto polarity. Sila polarity slova môže byť odstupňovaná priamo v klasifikačnom slovníku, kde sú slová klasifikované nielen k pozitívnej alebo negatívnej polarite, ale priamo k určitému stupňu tejto polarity. Niekedy silu polarity slova mení (zvyšuje alebo znižuje) iné slovo, takzvaný intenzifikátor, ktoré predchádza spracovávané slovo.

2.2 Prehľad existujúcich aplikácií

Existuje veľké množstvo prác, ktoré sa venujú klasifikácii názorov. Na analýzu sentimentu pomocou slovníkov sa zameriava Taboada a kol.[3]. Vo svojej práci využíva intenzifikáciu, negáciu posunom a porovnáva výsledky medzi jednotlivými slovníkmi a prístupmi ku klasifikácii. Pri testovani riešenia dosiahol presnosti v rozmedzí od 65% do 81%. V anglickom jazyku existuje niekoľko druhov slovníkov, ktoré je možné použiť pri analýze sentimentu. Najznámejšie sú WordNet a SentiWordNet. Práve možnosťou využitia SentiWordNetu na dolovanie názorov sa zaoberá práca Ohama a kol.[4]. Porovnáva manuálne vytvorený slovník s metódou využívajúcou SentiWordNet. Aplikácia, ktorá sa venuje klasifikácii názorov v slovenskom jazyku je KLAN [6] vytvorená na TUKE. KLAN funguje na slovníkovom princípe a na spracovanie intenzifikácie a negácie používa dynamický koeficient. Aplikácia dosiahla 86.2% presnosť pre pozitívne príspevky a 69.2% pre negatívne príspevky. Zaujímavou možnosťou je prepojenie slovníkových metód s metódami strojového učenia. Tento postup je popísaný v práci Zhang a kol.[5]. Najskôr sa aplikuje slovníková metóda, ktorej výsledky sú vstupom pre metódy strojového učenia. Medzi ďalšie práce, ktoré sa venujú tejto téme v českom jazyku patrí *Analýza sentimentu v příspěvcích na sociální síti Twitter* [7]. Práca popisuje aplikáciu, ktorá na analýzu sentimentu využíva n-gramy. Metódy strojového učenia sú aplikované aj v práci Koktan[8], kde využíva na analýzu sentimentu metódy SVM, Naivného Bayesa a Maximálnej entópie.

3 Návrh prístupu ku klasifikácii názorov

Navrhli sme algoritmus na klasifikáciu názorov, ktorý pracuje v troch krokoch. V prvom kroku sa získa text, ktorý chceme analyzovať. V druhom kroku je text rozdelený na vety a slová. Slová sú upravené a porovnávané so slovami v slovníku. V prípade zhody sa slovám v texte priradí zodpovedajúci stupeň polarity zo slovníka. V treťom kroku sa určí výsledná sila polarity celého príspevku ako súčet polarít jednotlivých slov.

Na určenie sily polarity sme zvolili stupnicu, ktorá nadobúda hodnoty od -3 do 3 (od silnej, miernej až po slabú negatívnosť, cez neutralitu po slabú, miernu a silnú pozitívnosť – celkovo 7 stupňov). Použitie záporu v texte sme riešili tzv. switch negáciou (napr. zo silnej pozitívnosti sa stane silná negatívnosť). Pre intenzifikátory sme zvolili hodnoty od 1.00 do 2.00 podľa sily, akou zvyšujú polaritu (napr. mimoriadny - 1.5). Takýmto spôsobom je možné zvyšovať intenzitu v závislosti na sile slova, ktoré chceme intenzifikovať (slová so silnejšou polaritou budú viac zosilnené ako slová so slabšou polaritou).

3.1 Získavanie a spracovanie príspevkov

Texty príspevkov, ktoré majú byť analyzované môžeme zadávať v aplikácii tromi rôznymi spôsobmi. Buď je text zadávaný manuálne z klávesnice, alebo je načítaná skupina príspevkov z textového súboru, alebo je zadaná webová adresa diskusie a z nej sa priamo sťahujú texty komentárov pomocou html *tagov*.

V procese spracovania sa text rozdelí na jednotlivé vety a následne sa v texte odstráni diakritika. Potom sa jednotlivé vety rozdelia na elementárne jednotky - slová. Prídavné mená sú väčšinou hlavným nositeľom polarity. Preto bola použitá upravená verzia Lancasterského stemovacieho algoritmu, dostupného na webovej stránke[1], ktorý ich prevedie do nominatívu množného čísla. Predpony ako kilo-, mega-, mini-, mili- atď., sú zo slov odstránené. Potom sa zistia prípony a tie sú nahradené preddefinovanými znakmi. Keďže v slovenčine môže mať rovnaká pádová prípona v základnom tvare na konci tvrdé *y* aj mäkké *i*, bol zvolený nominatív množného čísla, kde sa na koniec priradí vždy iba mäkké *i*.

- zlej → zlý → zli
- lepšej → lepší → lepsi

Ako už bolo spomenuté, po zistení zhody slova so slovníkom je slovu priradená intenzita polarity na základe údajov zo slovníka. Po každom novom slove sa modifikuje hodnota polarity spracovávanej vety. Ak je spracovávané slovo intenzifikátor, potom algoritmus hľadá nasledujúce slovo s pozitívnou alebo negatívnou polaritou, ku ktorému sa intenzifikátor vzťahuje a hodnota polarity nájdeného slova sa násobí podľa sily intenzifikátora. Ak program narazí na zápor, otočí hodnotu negovaného slova (switch). V prípade, že sa v texte objavia všetky možnosti, výpočet prebieha tak, že sa hodnota aktuálneho slova vynásobí silou intenzifikácie a negáciou (-1). Výsledná hodnota polarity vety je upravená:

[1] http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm

$$log\_hodnota = 1 + log_{10}(hodnota\_vety) \quad (1)$$

3.2 Slovník

Navrhnutá metóda reprezentuje slovníkový prístup. Slovník obsahuje kľúčové slová domény diskusie. Náš slovník bol vytvorený prekladom z anglického jazyka. Následne boli ku všetkým slovám nájdené synonymá. Taktiež sme pridali intenzifikátory a negátory. Ak slovo končí na inú samohlásku ako *o*, je v slovníku zapísané v tvare nominatívu množného čísla. V prípade, že slovo končí spoluhláskou, alebo samohláskou *o*, je uložené bez úprav. Každé slovo v slovníku má priradenú polaritu podľa stupníc opísaných vyššie.

4 Experimenty

Navrhnutý algoritmus sme implementovali a testovali. Testovanie sme realizovali na 4 slovníkoch. Pre slovenčinu v tomto momente neexistuje štandardný dataset určený na testovanie analýzy sentimentu. Preto sme sa rozhodli vybrať na testovanie dáta z diskusií k filmom Zelená míľa a Forrest Gump (www.csfd.cz). Príspevky sme preložili do slovenského jazyka, keďže pôvodne sa jednalo väčšinou o české komentáre a kvôli zachovaniu objektivity v nich boli ponechané gramatické a štylistické chyby. Testovaná vzorka obsahovala 2749 príspevkov. Do budúcnosti pracujeme na vytvorení štandardného datasetu, ktorý bude použitý pre ďalšie testovanie.

Hodnotenie príspevku sme považovali za správne (dobré, valídne), ak sa zhodovalo s hodnotením experta. Na základe zhody, resp. nezhody bola vyčíslená presnosť a návratnosť metódy. Presnosť je podiel správne vyhodnotených pozitívnych príspevkov voči všetkým príspevkom označeným algoritmom ako pozitívne. Návratnosť je podiel správne vyhodnotených pozitívnych príspevkov voči všetkým príspevkom označeným expertom ako pozitívne. Rovnaký spôsob výpočtu sme použili pre vypočítanie presnoti a návratnosti pre neutrálne a negatívne komentáre.

4.1 Testovanie diskusie k filmu Zelená míľa

Diskusia k filmu Zelená míľa obsahuje 1161 pozitívnych, 84 negatívnych a 94 neutrálnych komentárov. Aplikácia dosiahla dosť vysokú presnosť a návratnosť pre pozitívne príspevky (viď tabuľka 1). V priemere sa presnosť pre pozitívne komentáre pohybovala okolo 95% a návratnosť 89.5%. Pri neutrálnych príspevkoch sa návratnosť pohybovala okolo 71%. Dosiahnutá presnosť bola okolo 41%. Najnižšiu hodnotu návratnosti dosiahli príspevky s negatívnym postojom k filmu, kde sa návratnosť pohybovala okolo 49.4%. Presnosť negatívnych príspevkov bola okolo 50.2%.

Table 1. Vyhodnotenie presnosti a návratnosti pre film Zelená míľa.

Miery	Presnosť(%)			Návratnosť(%)		
Slovník	pozit	neutr	negat	pozit	neutr	negat
Slovník_poz+neg	94.9	41.9	53.4	90.4	71.3	46.4
Slovník_intenz	94.9	42.4	54.8	90.6	71.3	47.6
Slovník_zápor	95	41.2	46.2	88.5	72.3	51.2
Slovník_všetko	95	42.2	46.3	88.6	72.3	52.4

4.2 Testovanie diskusie k filmu Forrest Gump

Diskusia k filmu Forrest Gump obsahuje 1169 pozitívnych, 94 negatívnych a 147 neutrálnych komentárov. Tabuľka 2 dokumentuje dosiahnuté vysoké hodnoty presnosti (cca 90.6%) a návratnosti (cca 85%) pri pozitívnych príspevkoch. Oproti predošlému testu výrazne poklesla návratnosť pre neutrálne komentáre (55%) a tiež presnosť (35%). Najnižšia návratnosť bola dosiahnutá pre negatívne príspevky.

Table 2. Vyhodnotenie presnosti a návratnosti pre film Forrest Gump.

Miery	Presnosť(%)			Návratnosť(%)		
Slovník	pozit	neutr	negat	pozit	neutr	negat
Slovník_poz+neg	90.6	36.4	39.7	86.3	56.5	28.7
Slovník_intenz	90.7	36.1	38.9	85.7	57.1	29.8
Slovník_zápor	90.6	34.2	34.6	84.4	55.8	29.8
Slovník_všetko	90.8	33.6	35.3	83.7	56.5	31.9

5 Záver

Webové služby, ktoré sa zaoberajú analýzou webových diskusií sa stávajú čím ďalej, tým populárnejšie. Je to najmä preto, lebo dnešný vyťažený človek už nemá čas čítať celé diskusie k danej téme. V tejto práci je prezentovaný slovníkový prístup ku klasifikácii názoru. Táto metóda dokáže spracovať viacnásobnú intenzifikáciu a negáciu a tiež intenzifikovať negáciu a negovať intenzifikáciu. Prezentovaná aplikácia dosiahla priemernú presnosť 92.8% pre pozitívne komnetáre, čo je lepšie ako aplikácia Klan(86.2%). Avšak presnosť pre negatívne komentára bola 43.4% čo výrazne nižšia hodnota ako predchadzajúca aplikácia (69.2%).

Vyhodnocovanie pozitívnych príspevkov dosiahlo dobré výsledky. Tie sa mierne zlepšili použitím intenzifikátorov. Pridaním slov otáčajúcich polaritu sa výsledky zhoršili, čo bolo pravdepodobne spôsobené použitím *switch* negácie, ktorá nie je veľmi presná. Výsledky testov boli tiež ovplyvnené nízkym počtom neutrálnych

a negatívnych príspevkov v pomere ku kladne hodnotiacim komentárom. Neutrálne príspevky boli väčšinou tie, ktoré hodnotili objektívnu stránku filmu, ako napr. výroky z filmu, komentáre z natáčania alebo nominácie na Oscarov.

Problematické boli hlavne komentáre, v ktorých autor najskôr opisoval filmy ako pozitívne, ale časom zmenil názor. Komentár bol teda vyhodnotený expertom ako negatívny, ale implementácia ho vyhodnotila na základe kladného počiatočného opisu pozitívne. Takisto, keď sa hodnotil film pozitívne, jednalo sa väčšinou o priame hodnotenie, zatiaľ čo negatívne hodnotenie vychádzalo skôr z opisov častí, ktoré sa autorovi komentáru nepáčili. Ďalším typom problematických komentárov boli také, ktoré hodnotili film len bodmi alebo percentami, bez dodatočného opisu. Problémom je aj spracovanie irónie a dvojzmyslov.

Poďakovanie. Tento príspevok vznikol za podpory agentúry VEGA v rámci projektu č. 1/1147/12 „Metódy analýzy kolaboratívnych procesov realizovaných prostredníctvom informačných systémov“.

Referencie

1. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Foundation and Trends in Information Retrieval, Vol.2, No.1-2, 1–135 (2008)
2. Koncz, P.: Aspektovo orientovaná analýza sentimentu. Písomná práca k dizertačnej skúške. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 59 s (2012)
3. Taboada, M., a kol.: Lexicon-based methods for sentiment analysis. Comput. Linguist., vol. 37, no. 2, pp. 267—307 (2011)
4. Ohama, B., Tierney, B.: Opinion Mining with SentiWordNet. In: Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains 266–286 (2010)
5. Zhang, L., a kol.: Combining Lexicon - based and Learning - based Methods for Twitter Sentiment Analysis HP Laboratories, [cit 20.10.2014] (2011) Dostupné z: <http://www.hpl.hp.com/techreports/2011/HPL-2011-89.html>.
6. Machová, K., Krajč, M.: Klasifikácia názorov vo vláknových diskusiách na webe. In: Znalosti 2011 : 10. ročník konference : Stará Lesná, Vysoké Tatry : 1. ledna - 2. února 2011, sborník příspěvků, Ostrava, VŠB-TU, 136–147 (2011)
7. Buryan, J.: Analýza sentimentu v příspěvcích na sociální síti Twitter. Bakalářská práce. Brno: Masarykova univerzita, Fakulta informatiky, 43 s (2013) Dostupné z: <http://is.muni.cz/th/374039/fi_b/>
8. Koktan, M.: Automatické rozpoznávání (analýza) sentimentu. Diplomová práce. Plzeň: Západočeská univerzita v Plzni, 60 s (2012)

Multi-aspect Document Content Analysis using Ontological Modelling

Martin Milicka and Radek Burget

Faculty of Information Technology, IT4Innovations Centre of Excellence
Brno University of Technology, Bozetechova 2, 612 66 Brno, Czech Republic
{imilicka,burgetr}@fit.vutbr.cz

Abstract. Existing methods of information extraction from web documents are usually based on a single aspect of the document or its contents such as the code, textual features or visual features. Due to the great variability of the available online documents, it seems reasonable to combine multiple kinds of analysis in order to use all the available knowledge for identifying a particular information in the document. In this paper, we propose an ontological document model that allows to integrate the results of the analysis of different document aspects. We propose a generic architecture of an information extraction system based on this model and we show its applicability on a practical example.

Keywords: document modeling, information extraction, page segmentation, content classification, ontology, RDF

1 Introduction

Information extraction (IE) from web documents is a difficult task mainly because of very loose and variable structure of the documents and lack of available metadata or annotations. Most common IE approaches analyze mainly the the HTML code (DOM), the text of the document (named entity recognition, statistical analysis of the text, etc.) or the visual presentation (page layout and visual features of the presented contents). Usually, only one of these aspects is used. However, the web is diverse: Depending on the nature of the presented information and the target users, the visual hints may be crucial for some web pages while other pages may be primarily text-oriented and the visual presentation plays a secondary role. Therefore, analyzing multiple aspects together seems to be a promising way of research.

Several models have been introduced for representing documents: DOM [5] is a standard for modelling HTML document code. Similarly, CSS [1] defines a formatting model that describes the contents of a rendered page. In the layout analysis area, the page segmentation algorithms usually use specific models for representing the segmentation results [2, 3]. E.g., VIPS [3] represents the segmented page as a hierarchy of visual blocks and separators. The mentioned models are not intended to be shared by multiple applications; there is usually no explicit representation of the model defined that would allow storing a

created model and sharing it among multiple applications or analysis methods. RDF-based models may be used for storing metadata and annotations in PDF files [4].

In this paper, we propose an ontology-based extensible model of web documents that allows to integrate the results of multiple analysis algorithms that include the visual organization of the page (layout) and other visual features (fonts, colors, etc.), results of the visual area classification based on visual features and the results of text classification including the named entity recognition (NER) algorithms. We propose an architecture of an IE system based on this model and we show how the multiple aspect analysis may be used for improving the results of information extraction in the domain of news articles [6].

2 Ontological Document Model

A document may be described on different levels of abstraction. We define three levels of document description where each level adds a specific knowledge about the document.

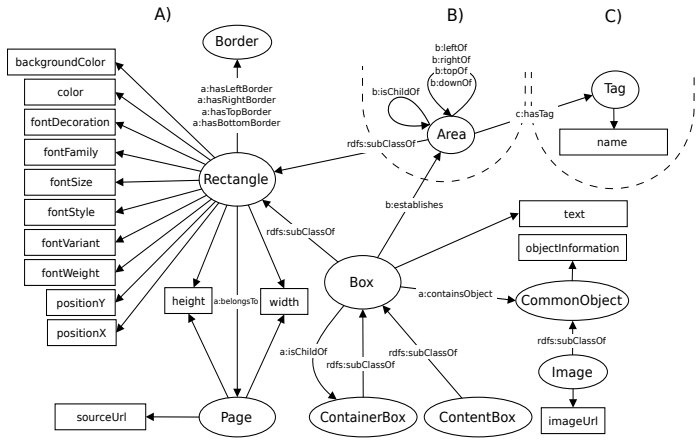


Fig. 1. A) Box model ontology B) Segmentation ontology C) Classification ontology

1. Box model description (*rendered page level description*) represents the output of the page rendering process – visual features of the individual content parts and their positions on the resulting page.
2. *Semantic level* where the box model is extended with an additional semantic information as described below.
3. *Domain description* that represents a connection to the specific domain of the processed documents.

We have designed a set of ontologies that allow representing all the information about a document using RDF. The *Box model ontology* (fig. 1A) represents

the box model description. The document is represented as a set of (possibly nested) rendered content *boxes* together with their size, position and visual features.

The remaining ontologies (fig. 1B and 1C) belong to the semantic level. The *Segmentation ontology* extends the Box model ontology by the possibility of representing larger visual areas. Its basic *Area* class represents the visual areas detected during page segmentation. Finally, the *Classification ontology* allows to add a number of classes (tags) to the individual visual areas. The class assignment may be produced by a classification algorithm based on different features (e.g. text classification or visual classification) or manually, e.g. when creating a training set of documents.

3 Model Application for Information Extraction

The architecture of an IE system based on the proposed model is built around a central RDF repository that stores the information about all the processed documents as shown in the figure 2.

During the *model initialization* the source document is transformed to a format-independent RDF model based on our box model ontology. In the *model building* phase, further analysis steps such as content classification or page segmentation are applied on the model (in any order, possibly in several iterations); the results are represented using our semantic level ontologies. Alternatively, some information such as manually annotated classes may be added by the user using an interactive tool (visual editor). Finally, based on the results of the previous analysis steps, we map certain parts of the created model to a domain ontology which is actually the extraction step.

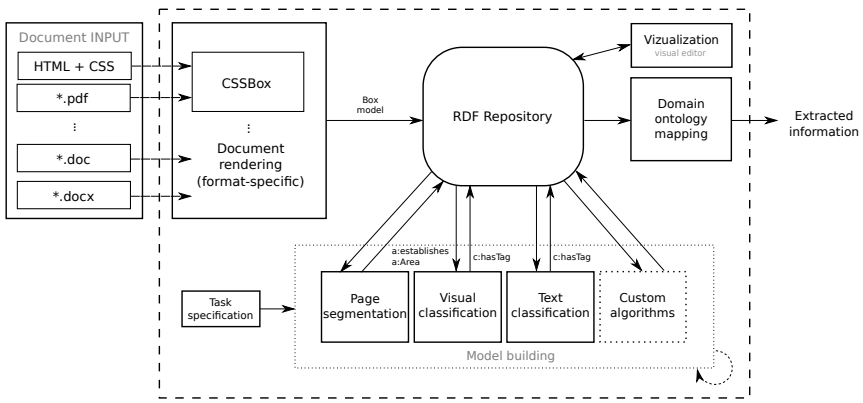


Fig. 2. A generic architecture of an IE system based on the ontological model

For testing the proposed concept, we have chosen the domain of online news articles where the task is to recognize a published article within a larger web

page and to distinguish its individual parts such as heading, date of publication, paragraphs, etc. The whole IE process consists of the following steps:

For the model initialization, we have used our CSSBox[1] rendering engine that produces a box model that is later serialized to the RDF description.

The model building phase includes page segmentation, text classification based on NER (for recognizing names, places, dates, etc.) and visual classification based on the visual features of the content as described in [2]. During these steps, each detected visual area is assigned a set of *tags* that indicate the probability that the given area represents a certain part of the article. Based on the assigned tags, the areas are finally mapped to a simple Article domain ontology that models an article and its its individual parts.

Our preliminary experiments run on the *reuters.com* and *cnn.com* news portals show, that the combination of several classification methods may increase the IE precision in comparison to a single-aspect classification published in [2].

4 Conclusions

We have proposed an ontological document model suitable for the description of different aspects of web documents on several levels of abstraction. The model allows sharing all the knowledge about the document and its contents among multiple analysis methods and combine their results. We have also shown the general architecture of an IE system based on this model and we have shown its applicability in a particular domain. The actual precision of the information extraction depends on the quality of results of the individual analysis methods, the way they are combined and the used method of domain ontology mapping.

This work was supported by the BUT FIT grant FIT-S-14-2299 and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

References

1. Bos, B., Lie, H.W., Lilley, C., Jacobs, I.: Cascading Style Sheets, level 2, CSS2 Specification. The World Wide Web Consortium (1998)
2. Burget, R., Rudolfová, I.: Web page element classification based on visual features. In: 1st Asian Conference on Intelligent Information and Database Systems ACIIDS 2009. pp. 67–72. IEEE Computer Society (2009)
3. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: VIPS: a Vision-based Page Segmentation Algorithm. Microsoft Research (2003)
4. Eriksson, H.: The semantic-document approach to combining documents and ontologies. Int. J. Hum.-Comput. Stud. 65(7), 624–639 (Jul 2007)
5. Hors, A.L., Hgaret, P.L., Wood, L., Nicol, G., Robie, J., Champion, M., Byrne, S.: Document Object Model (DOM) Level 3 Core Specification. The World Wide Web Consortium (2004)
6. Shi, J., Liu, L.: Web information extraction based on news domain ontology theory. In: Web Society (SWS), 2010 IEEE 2nd Symposium on. pp. 416–419 (Aug 2010)

[1] http://cssbox.sourceforge.net/

Modelovanie významu slov vo vektorovom priestore čŕt

Márius Šajgalík, Marián Šimko, Michal Barla, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 2, 842 16 Bratislava, Slovakia
{marius.sajgalik,marian.simko,
michal.barla,maria.bielikova}@stuba.sk

Abstrakt. V súčasnosti zažívame veľký rozmach v oblasti hĺbkového učenia. Hĺbkové učenie zasiahlo aj oblasť spracovania prirodzeného jazyka, kde pomaly začínajú dominovať modely významu slov, ktoré reprezentujú slová ako vektory latentných čŕt. Jednou z najväčších výhod týchto modelov je, že sa trénujú bez učiteľa, čo umožňuje automatizáciu spracovania „surového“ textu, teda bez akýchkoľvek sémantických anotácií, ktoré boli potrebné na trénovanie modelov využívajúcich ontológie slov. To zároveň umožňuje spracovanie oveľa väčšieho objemu dát, ktoré už nepotrebujeme manuálne značkovať, čo otvára ďalšie nové možnosti výskumu.

Kľúčové slová: vektor čŕt, hĺbkové učenie, spracovanie prirodzeného jazyka

1 Úvod

Na automatizované pochopenie významu slova sa ešte nedávno používali ručne vytvorené slovníky, taxonómie, či ontológie. Najmä ontológiám sa pripisoval veľký potenciál, keďže z formálneho hľadiska dokážu najlepšie štruktúrovane opísať vlastnosti slov (entít, konceptov). Postupom času sa však ukazuje, že napriek ich silnej schopnosti formálneho opisu, nám ontológie akoby nepostačovali. Dáta, vrátane obrovského množstva novinových článkov, blogov, konverzácií na sociálnych sieťach, či iného textu v prirodzenom jazyku, sa generujú príliš rýchlo a preto potrebujeme automatizovať pochopenie významu takéhoto „surového“ textu. Hoci bola snaha vytvoriť ontológiu slov (manuálne [13] aj automatizovane [15]), jej použitie nie je jednoduché, keďže na to je potrebné najprv priradiť slovám príslušné významy (koncepty) z ontológie. Už určovanie významu slov v určitej miere obmedzuje použitie takýchto ontológií svojou pomerne nízkou úspešnosťou [15] (najmä pri určovaní významov na nižšej úrovni abstrakcie). Ani správne určenie významu slov nám však ešte nestačí pri riešení ďalších úloh ako napr. meranie podobnosti slov, čo je ďalší zložitý problém [5].

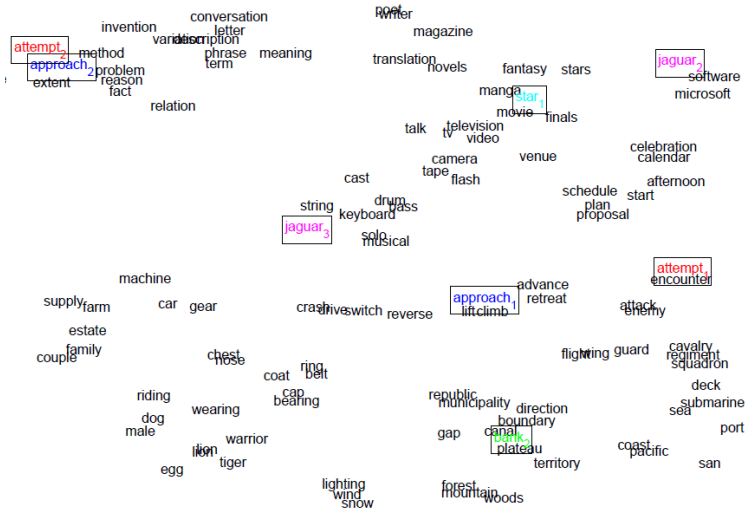
S rozmachom hĺbkového učenia sa v posledných rokoch začali presadzovať nové metódy bez učiteľa, ktoré sú schopné naučiť sa význam slov z čistého textu (bez akýchkoľvek anotácií) [1]. Tieto metódy mapujú slová do viacrozmerného vektorového priestoru. Význam slova je zakódovaný vo vektore latentných čŕt, preto

sa takáto reprezentácia nazýva aj ako distribuovaná reprezentácia slov. Takýto model je na rozdiel od ontológie jednoduchší na použitie, keďže už nepotrebujeme určovať presný význam slova. V latentnom vektore čŕt sú zakódované všetky možné významy slova a presný význam sa upresňuje na základe kontextu, v ktorom sa slovo vyskytuje. Vektorový priestor nám umožňuje jednoducho merať podobnosť slov a taktiež zachováva analogické vzťahy medzi slovami [10].

2 Existujúce modely

Za prelomový možno považovať prístup [1] z roku 2006, kde autori predstavujú model založený na neurónových sieťach, ktorý dokáže lepšie modelovať jazyk ako tradičné n-gramové metódy. V roku 2008 Collobert a Weston [2] vymysleli novú jednotnú architektúru neurónových sietí, ktorá sa dokáže oveľa efektívnejšie naučiť vektor čŕt slov automaticky z neoznačkovaného textu a je možné ju použiť na viaceré úlohy spracovania prirodzeného jazyka. Prístup v [14] predstavuje podobný pravdepodobnostný model, ktorý vytvára hierarchiu slov a tak exponenciálne zrýchľuje výpočtovú zložitosť.

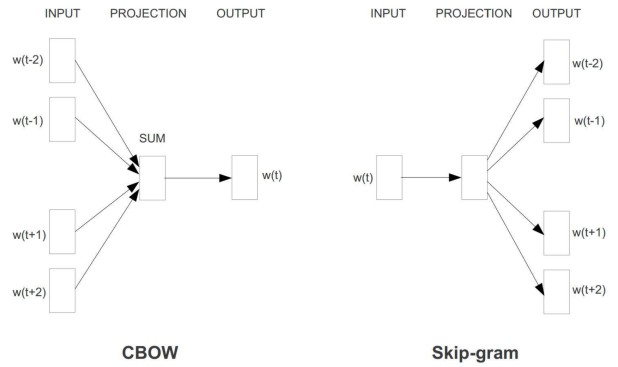
V [4] nachádzame pokus o kombináciu vektorov čŕt a ontológie konceptov. Z lokálneho a globálneho kontextu sa model učí viacero vektorov pre každé slovo na základe homoným a polysém (viď Obrázok 1). Hoci ide o zaujímavý koncept, novšie metódy sa stále držia jednoduchšieho použitia iba jedného vektora na jedno slovo (napr. [9] alebo [16] opísané nižšie).



Obr. 1. Vizualizácia vektorov slov s využitím globálneho kontextu a viacerých prototypov slov.

Po viacročnom výskume použitia rekurentných neurónových sietí v oblasti modelovania jazyka [12], Mikolov vymyslel dve nové architektúry (viď Obrázok 2) na výpočet vektorovej reprezentácie slov [9]. Vďaka ich jednoduchosti je možné

oveľa efektívnejšie trénovanie ako v prípade rekurentných neurónových sietí. Trénovanie je založené na posuvnom kontextovom okne, podobnom ako sa využíva napr. aj pri určovaní slovných druhov [17]. Zatiaľ čo architektúra CBOW sa učí uhádnuť prostredné slovo na základe okolitých kontextových slov, architektúra Skip-gram sa presne naopak učí uhádnuť všetky kontextové slová na základe jediného prostredného slova. Architektúra Skip-gram sa trénuje dlhšie, avšak je lepšia pre menej frekventované slová. Obe tieto architektúry sú implementované v nástroji word2vec[1], ktorý sa vďaka svojej jednoduchosti použitia a zároveň optimalizovanej efektívnosti výpočtu s možnosťou paralelizácie, stal veľmi populárnym a otvoril dvere aj začínajúcim výskumníkom v tejto oblasti.



Obr. 2. Dva nové modely na výpočet vektorovej reprezentácie slov – CBOW a Skip-gram.

GloVe [16] predstavuje jeden z najnovších prístupov na výpočet vektorovej reprezentácie slov. Snaží sa skombinovať výhody modelov založených na posuvnom kontextovom okne a modelov založených na faktorizácii matice ako napr. LSA [3]. Namiesto prechádzania celým korpusom sa učí len na základe globálnej štatistiky výskytov slov v spoločnom kontexte, vďaka čomu je možné ešte efektívnejšie trénovanie tohto modelu ako v prípade ostatných modelov. Tento model sa zameriava na zlepšenie modelovania podobnosti slov. Využíva pozorovanie, že podobnosť slov je ovplyvnená podielom pravdepodobností spoločného výskytu slov, ktorý je buď oveľa väčší, alebo oveľa menší ako 1.

3 Vlastnosti vektorov čŕt a ich aplikácia na viaceré úlohy spracovania prirodzeného jazyka

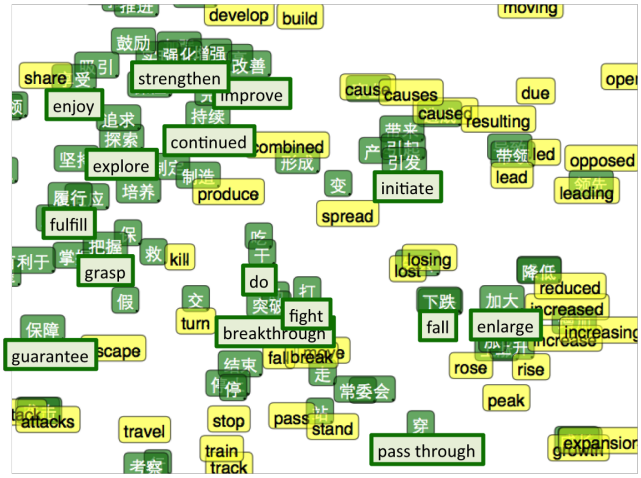
Vektorový priestor čŕt ukrýva viaceré zaujímavé vlastnosti. Umožňuje nám jednoducho merať podobnosť slov meraním podobnosti vektorov. Vektory môžeme navyše jednoducho sčítavať a „skladať“ význam viacslovných fráz, či dopytov. Pomocou odčítavania vektorov môžeme vyjadriť aj vzťahy ako vektory čŕt. Podľa analýzy v [11] zachováva vektorový priestor čŕt viaceré pravidelnosti vo vzťahoch

[1] word2vec - https://code.google.com/p/word2vec/

medzi slovami. Vektor čŕt kóduje aj syntaktické aj sémantické črty [11]. Kombináciou metód s učiteľom a bez učiteľa je možné obohatiť vektor čŕt aj o sentiment [13].

Už v [2] autori navrhli jednotnú architektúru pre spracovanie prirodzeného jazyka, ktorá dokáže úspešne riešiť až niekoľko úloh súčasne – určovanie slovných druhov, syntaktická analýza, určovanie menných entít, sémantických rol, sémanticky podobných slov a vyhodnocovanie, či je veta gramaticky správna, a či vôbec dáva zmysel. Podľa štúdie v [19], vektory čŕt možno natrénovať vopred bez učiteľa, bez toho, aby sme dopredu poznali úlohu, v ktorej chceme tieto vektory čŕt použiť. Všeobecne natrénované vektory čŕt možno jednoducho použiť v existujúcich metódach s učiteľom na spracovanie prirodzeného jazyka, aj keď autori uznávajú, že výsledky nie sú také dobré ako v prípade kombinovaného trénovania s učiteľom a bez učiteľa. Autori [19] úspešne aplikujú vektory čŕt na určovanie menných entít a syntaktickú analýzu.

Vektory čŕt možno aplikovať aj na vylepšenie strojového prekladu. Prístup v [20] sa učí súčasne vektory čŕt pre angličtinu a čínštinu (viď Obrázok 3). Nevýhodou tohto prístupu je, že vyžaduje súbežný dvojjazyčný text zarovnaný na úrovni slov. Existuje však aj jednoduchší spôsob opísaný v [10], kde sa vektory čŕt trénujú pre každý jazyk zvlášť. Autori [10] zistili, že preklad sa dá realizovať nájdením obyčajného lineárneho zobrazenia z vektorového priestoru jedného jazyka do vektorového priestoru druhého.



Obr. 3. Spoločné modelovanie vektorov slov pre angličtinu a čínštinu.

Napriek tomu, že vektorový priestor čŕt predstavuje viacrozmerné dáta, existujú efektívne metódy aj na redukciu dimenzií viacrozmerných dát. Okrem klasických metód ako napr. PCA [6], je na vizualizáciu dát v 2D priestore momentálne najpopulárnejšia metóda t-SNE [8], ktorá dokáže zachovať . Túto metódu možno použiť na prezentáciu výsledkov, alebo aj v skorších etapách výskumu na vizualizáciu a kontrolu medzivýsledkov napr. aj pri úlohách ako je extrakcia kľúčových slov [18].

Poďakovanie. Táto publikácia vznikla vďaka čiastočnej podpore projektov VEGA VG1/0675/11 a APVV 0208-10.

4 Referencie

1. Bengio, Yoshua, et al. "Neural probabilistic language models." Innovations in Machine Learning. Springer Berlin Heidelberg, 2006. 137-186.
2. Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." Proceedings of the 25th international conference on Machine learning. ACM, 2008. 160-167.
3. Dumais, Susan T. "Latent semantic analysis." Annual review of information science and technology 38.1 (2004): 188-230.
4. Huang, Eric H., et al. "Improving word representations via global context and multiple word prototypes." Proceedings of the 50th Annual Meeting of ACL: Long Papers-Volume 1. ACL, 2012. 873-882.
5. Jabeen, Shahida, Xiaoying Gao, and Peter Andreae. "A Hybrid Model for Learning Semantic Relatedness Using Wikipedia-Based Features." Web Information Systems Engineering–WISE 2014. Springer International Publishing, 2014. 523-533.
6. Jolliffe, Ian. Principal component analysis. In: Encyclopedia of Statistics in Behavioral Science, Vol. 3, 1580-1584, Wiley, 2005.
7. Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." Proceedings of the 49th Annual Meeting of ACL: Human Language Technologies-Vol. 1. ACL, 2011. 142-150.
8. Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of Machine Learning Research 9.2579-2605 (2008): 85.
9. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
10. Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation." arXiv preprint arXiv:1309.4168 (2013).
11. Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." HLT-NAACL. 2013. 746-751.
12. Mikolov, Tomáš. Statistical language models based on neural networks. Diss. Ph. D. thesis, Brno University of Technology, 2012.
13. Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.
14. Mnih, Andriy, and Geoffrey E. Hinton. "A scalable hierarchical distributed language model." Advances in neural information processing systems. 2009. 1081-1088.
15. Navigli, Roberto, and Simone Paolo Ponzetto. "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network." Artificial Intelligence 193 (2012): 217-250.
16. Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. "GloVe: Global Vectors for Word Representation."
17. Sánchez-Villamil, Enrique, Mikel L. Forcada, and Rafael C. Carrasco. "Unsupervised training of a finite-state sliding-window part-of-speech tagger." Advances in Natural Language Processing. Springer Berlin Heidelberg, 2004. 454-463.
18. Šajgalík, M., Barla, M., Bieliková, M.: Exploring Multidimensional Continuous Feature Space to Extract Relevant Words. In: Proc. of SLSP 2014, Springer-Verlag, 2014.
19. Turian, Joseph, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning." Proceedings of the 48th Annual Meeting of ACL. ACL, 2010. 384-394.
20. Zou, Will Y., et al. "Bilingual Word Embeddings for Phrase-Based Machine Translation." EMNLP. 2013. 1393-1398.

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 18–21

Towards Understanding Multilingual Search Query Intent

Michal Laclavík[1 2], Marek Ciglan[2], Štefan Dlugolinský[2]
Sam Steingold[1], and Alex Dorman[1]

[1] Magnetic Media Online, New York, USA,
[2] Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

Abstract. In this paper we describe an Entity Search and Query Understanding experiment on Slovak Language. In our previous work we participated in the ERD Challenge focusing on recognizing entities in search queries, here we try to extend this approach to other languages, while experimenting with Slovak Language. Wikipedia is used as Knowledge Base providing entities such as people, places or locations to be recognized in and answered for user search queries.

Keywords: entity search, Slovak language, query understanding, Wikipedia

1 Introduction

In this paper we describe the Slovak language extension of our contribution [1] to Entity Search by participating at *2014 Entity Recognition and Disambiguation Challenge*[3] [2]. We have participated in the *Short Track* of the challenge, which focused on recognizing mentions of entities in a search queries, disambiguating them, and mapping them to the entities in a given knowledge base - subset of Freebase[4] containing of more than 2 million of entities.

For Slovak language, we have taken the Slovak Wikipedia containing more than 190,000 articles (concepts or entities) as a knowledge base for entity recognition and disambiguation and we discuss our result in applying our original ERD approach on this Slovak data.

In the ERD, our system was evaluated as the 4^{th} best with F1 score of 65.57%[5]. We believe that our system has some unique features. In our research [3], we try to address Entity Search, Query Understanding or Question Answering problems by combing efforts from typical information retrieval models, semantic web, information extraction and complex networks.

We are developing Entity Search related applications like Query Categorization or Enterprise Search, where some of these efforts have been developed and tested. Participation in the ERD challenge [2] helped us enhance and introduce new techniques, where we combine approaches from these fields as described in

[3] `http://web-ngram.research.microsoft.com/erd2014/`
[4] `http://www.freebase.com/`
[5] `http://tinyurl.com/ShortTrackERD14`

the ERD paper[1] and extended here to Slovak and, potentially, to any language where sufficient Wikipedia data is available.

Motivation for Magnetic (Magnetic Media Online[6]) and IISAS (Institute of Informatics, Slovak academy of Sciences[7]) to participate in the ERD Challenge comes from our effort to build a scalable Query categorization (QC) system based on Wikipedia corpus instead of the entire web. Magnetic needs to understand query intend to perform well the business of Search Retargeting, a form of targeted online advertising. In the domain of search retargeting, the audiences are modeled based on the search queries users conduct on the websites they visited. Search retargeting focuses on displaying advertisements to users who conducted searches for specific keywords or categories in the past. For this domain, QC is the essential technique for user modeling and better user targeting. Magnetic tries to address foreign languages, thus support for multilingual query categorization is essential. Entity search, which was focus of the ERD challenge [2] and also this paper is the first step in our QC approach.

In addition to QC, IISAS motivation comes also from the VENIS project[8], where we tried to solve Enterprise Search by incorporating both structured data (database items) and unstructured data (emails, documents) to model entities in an enterprise - especially in SMEs. We would like to address also Multilingualism in our Enterprise Search.

The main contributions of this paper in addition to the ERD approach[1] applied on English are the following:

- Selecting and annotating 100 user queries as an evaluation dataset.
- Creating two version of Slovak Wikipedia index with and without special diacritic characters.
- Evaluating search query results on both indexes.

2 Experimenting on Slovak Wikipedia

In the past we have already experimented with Slovak Wikipedia[4], we have parsed the data and tried to use it for search or Named Entity Recognition. Now we used the same approach as for ERD, but we have created an index from Slovak Wikipedia. In our ERD solution we did not use any special language-dependent NLP to disambiguate entities or map them on search queries. So here we tested if this approach can really be applied to such languages as Slovak.

Soon after the first tests we discovered the following problems: not enough alternative names for entities; user queries with and without special characters.

We did not tackle the first problem by any means, but we can add additional alternative names from Wikipage infoboxes for example. Concerning high number of queries without the special diacritic characters, we have created two indexes with and without diacritics characters and experimented with them.

[6] http://www.magnetic.com/
[7] http://ikt.ui.sav.sk/
[8] http://www.venis-project.eu/

We have selected 100 queries from Magnetic search data[9], which are available online for future research. We have filtered out queries containing profanity or sexually explicit content. We have manually annotated these queries with concepts from Slovak Wikipedia. Slovak Wikipedia does not contain all desired information compared to the English one. For example, concepts like some popular TV series or Social Security office wikipage were missing, which somewhat limits the returned search results and can have an impact on lower coverage of entity search or query categorization.

We have also found out that 35% of queries (35/101) were missing special characters. This means that building an index without special characters is important, otherwise 35% of queries would stay unanswered. On the other hand this can bring some decrease of precision, where "dieta" would be same basic form for "diéta" (diet) as well as "dieťa" (child). However, results where special characters were removed were much better as described in next section.

3 Evaluation

In this section we discuss results on a sample of 100 Slovak queries - the data with annotations which we have prepared for this paper.

In the ERD Challenge [2], the evaluation focused solely on F1, because it was easier to identify borderline cases with zero retrieved or annotated entities for a query. However, we wanted to get an idea about Precision and Recall while developing the system, so we have calculated Macro Precision and Macro Recall, where there was no problem with borderline cases. In addition to these, we have also calculated Macro F1 and two types of Micro F1. Micro F1 calculated in the same way as defined by the ERD organizers, which we refer to as *Micro F1 Set* and *Micro F1*, which considered each returned entity as correct or incorrect independently of the defined interpretation sets. We have applied the same technique to the Slovak dataset evaluation. For more details see also ERD paper [1] and ERD guidelines[10].

Additionally, we computed the novel information-theoretic Proficiency metric [5], which measures the share of information content of the annotated dataset captured by the categorization. Its value is 1 for the perfect categorization and 0 for a categorization which is independent (in the sense of Probability Theory) from the annotation.

As one can see, there is always a few percent gap between Micro F1 and Micro F1 set. On our Slovak dataset it is even broader than on English one. The results for all applied measures are summarized in Table 1. In the Table 1, we list evaluations for Slovak query dataset with two different indexes - with special characters "SK" row and without special characters "SK ASCII" row, where all special characters were converted to its ASCII equivalent. We can see that improvement with ASCII index is very significant. Nevertheless, while on SK dataset we have achieved only about 47% F1 set, on the English TREC dataset

[9] http://ikt.ui.sav.sk/research/ERD/

[10] http://web-ngram.research.microsoft.com/erd2014/Docs/Detail%20Rules.pdf

Table 1. Results on Slovak queries dataset compared with beta TREC data[1] and ERD results[1]. New Proficiency metric is also reported.

	Macro Precision	Macro Recall	Macro F1	Micro F1	Micro F1 Set	Proficiency
SK ASCII	0.6067	0.5094	0.5538	0.5871	0.4686	0.4818
SK	0.4646	0.4340	0.4488	0.5005	0.3660	0.4333
EN TREC	0.7222	0.7761	0.7482	0.7968	0.7674	0.7650
EN ERD	-	-	-	-	0.6557	-

(see [1]) we have achieved a higher F1 set of 77% or 66% on ERD evaluation. ERD results (the "ERD" row) are available only for the F1 set since this results were evaluated by ERD organizers and they provided only one measure - F1.

4 Conclusion

In this paper we have shown that our Entity Search approach [1] used for the ERD challenge [2] can be applied also to other languages. The results are still worse than on English, but can be improved. Slovak Wikipedia is also one of the smaller Wikipedias and it is likely that better results can be achieved on Wikipedias with more then 1 million of pages.

In the future we would like to enhance our approach with new sources of alternative names, and also applying this approach on other European languages.

Acknowledgments. This work is supported by Magnetic, and also by project VENIS FP7-284984, VEGA 2/0185/13 and CLAN APVV-0809-11.

References

1. Michal Laclavik, Marek Ciglan, Alex Dorman, Stefan Dlugolinsky, Sam Steingold, and Martin Seleng. 2014. A search based approach to entity recognition: magnetic and IISAS team at ERD challenge. In Proceedings of the first international workshop on Entity recognition & disambiguation (ERD '14). ACM, New York, NY, USA, 63–68. DOI=10.1145/2633211.2634352
2. David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June (Paul) Hsu, Kuansan Wang. 2014. ERD 2014: Entity Recognition and Disambiguation Challenge. SIGIR Forum, 2014 (forthcoming), ACM.
3. Michal Laclavík, Marek Ciglan. 2013. Towards entity search: Research roadmap. In WIKT 2013 proceedings, 2013, p. 161-166. ISBN 978-80-8143-128-9.
4. Michal Laclavík, Štefan Dlugolinský, Michal Blanárik. 2013. Experimenting with Slovak Wikipedia as a Source for Language Technologies. In Proceedings of SLOVKO 2013, pages 160-165, 2013,
5. Sam Steingold, Michal Laclavík. 2014. An Information Theoretic Metric for Multi-Class Categorization. In preparation[11].

[11] `https://github.com/Magnetic/proficiency-metric`

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 22–26

Využitie softvéru BOINC pre podporu realizácie výskumných a projektových úloh dolovania v textoch

Peter Náhori, Peter Butka

Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Letná 9, 042 00 Košice, Slovenská republika
peter.nahori@student.tuke.sk, peter.butka@tuke.sk

Abstrakt. Tento príspevok sa venuje návrhu projektu využitia softvéru BOINC v úlohách dolovania textov pre výskumné a výučbové účely. Cieľom je aplikovanie paradigmy dobrovoľníckeho počítania v distribuovanom prostredí, pre ktorý bol BOINC vyvinutý a je aktuálne aplikovaný v úlohách spracovania dát v mnohých rozsiahlych projektoch vo svete. Našim cieľom je využiť prostriedky BOINC pre prípravu projektu umožňujúceho v rámci portálu a existujúcej infraštruktúry našich laboratórií vytváranie, správu a realizáciu úloh dolovania textov prevádzaných výskumníkmi alebo študentmi. Pre tento účel bude využitá integrácia našej knižnice JBOWL pre dolovanie v textoch a BOINC infraštruktúry. Výsledkom by mal byť relatívne jednoduchý portál umožňujúci väčšiemu počtu používateľov (výskumníkom a študentom) realizovať rozsiahle experimenty podľa potrieb ich výskumných alebo projektových úloh, čo bude umožnené spúšťaním úloh v distribuovanom prostredí BOINC infraštruktúry.

Kľúčové slová: dolovanie textov, BOINC, JBOWL, distribuované počítanie

1 Úvod

Získavanie znalostí a dolovanie dát v textoch má veľmi dôležité miesto vo výskume v 21. storočí. Jedným z problémov je realizácia výpočtov na veľkých dátových množinách a pre rôzne nastavenia parametrov algoritmov a modelov. Navyše je vo výskumnej a výučbovej praxi užitočné, ak je k dispozícii možnosť realizovať experimenty pre väčšie množstvo používateľov v rámci existujúcej výpočtovej infraštruktúry pracoviska. To vedie k využitiu distribúcie výpočtov, ktorej praktická realizácia je často postavená na samostatnom výpočtovom gride. Alternatívou k tomuto je využitie paradigmy tzv. dobrovoľníckeho počítania – zapojenia bežných počítačov s aktuálne voľnými prostriedkami do siete pre distribuovanie výpočtov. Príkladom takéhoto riešenia je softvér BOINC [2], ktorý sa celosvetovo používa na zapojenie dobrovoľníkov (a ich počítačov) do rôznych výpočtových kampaní.
Cieľom prezentovaného systému je využiť výpočtové prostriedky nášho pracoviska a zapojiť ich do internej BOINC infraštruktúry. Následne by bol vytvorený výpočtový projekt dolovania textov pomocou BOINC. Ďalším krokom by mal byť teda návrh a vytvorenie príslušného („front-end“) portálu pre výskumníkov a študentov, ktorý im umožní realizovať rozsiahle experimenty podľa potrieb ich výskumných alebo

projektových úloh, čo bude umožnené spúšťaním úloh v distribuovanom prostredí BOINC infraštruktúry.
Na realizáciu získavania znalostí a dolovania v textoch sme vybrali knižnicu JBOWL (Java Bag-Of-Words Library) [1] vyvinutú na našom pracovisku, pomocou ktorej sa realizujú výskumné úlohy, projekty záverečných prác a výučba zo zameraním na dolovanie textových dokumentov. Ide o softvérovú knižnicu implementovanú v jazyku Java, ktorá poskytuje objektový model a rozhrania (API) pre vytváranie aplikácií spracovania textu, dolovania v textoch a vyhľadávania informácií. Pre potrebu realizácie úloh dolovania z textov v distribuovanom prostredí sme využili už spomenutý softvér BOINC, ktorý bol špeciálne vyvinutý pre distribuované výpočty využívajúce dobrovoľne poskytnuté zdroje počítačov pripojených na internet (v našom prípade budú poskytnuté voľné zdroje v rámci pracoviska). Celá infraštruktúra bude sprístupnená prostredníctvom webového rozhrania, pomocou ktorého budú používatelia môcť realizovať svoje experimenty.

2 Použité technológie

Dôležitými prvkami projektu sú použité technológie. Nakoľko portálová časť je zatiaľ len v príprave, stručne popíšeme iba BOINC a knižnicu JBOWL.

BOINC

Systém BOINC (Berkeley Open Infrastructure for Network Computing) je softvérová platforma špeciálne vyvinutá pre distribuované výpočty (prvou aplikáciou bola analýza dát v rámci programu SETI), využívajúca dobrovoľne poskytnuté zdroje počítačov pripojených na internet. Táto infraštruktúra je vhodná na použitie v rôznych výskumných projektoch analyzujúcich rozsiahle experimentálne dáta alebo realizujúce paralelné výpočty. V rámci BOINC existuje dnes veľké množstvo projektov v oblasti fyziky, chémie, biológie, matematiky, materiálového výskumu, ale aj analýzy dát na úrovni počítačových vied. Všeobecným cieľom BOINC-u je presadzovať paradigmu výpočtov pomocou verejných prostriedkov, t.j., podporiť vytváranie výpočtových projektov a vyzývať veľkú časť majiteľov PC vo svete zúčastniť sa jedného alebo viacerých projektov. Vybrané špecifické ciele:

- Zníženie prekážok pre vstup do infraštruktúry BOINC, t.j., uľahčenie vytvorenia serverovej (projektovej) časti BOINC infraštruktúry.
- Zdieľanie zdrojov medzi autonómnymi projektmi – BOINC projekty sú autonómne, avšak majiteľ PC sa môže bez problémov zúčastniť na viacerých projektoch a môže priradiť ku každému projektu podiel svojich zdrojov.
- Podpora rôznych aplikácií – BOINC podporuje širokú škálu aplikácií, poskytuje flexibilný a škálovateľný mechanizmus pre distribúciu dát a jeho algoritmy plánovania inteligentne porovnáva požiadavky a zdroje.

Nakoľko BOINC primárne nepracuje s Java programami, pre tento účel je použitý BOINC Java Wrapper, ktorý umožňuje zabaliť spúšťanie Java programu v rámci balíka úloh BOINC a realizuje tak klasickú BOINC výpočtovú úlohu.

JBOWL

Systém JBowl začal vznikať od roku 2003 na pôde Katedry kybernetiky a umelej inteligencie FEI TU v Košiciach. Jeho oblasti skúmania sú predovšetkým manažment a reprezentácia znalostí, dolovanie a objavovanie znalostí v textoch, vyhľadávanie a extrakcia informácií, sémantický web a sémantické technológie vo všeobecnosti [5]. Vo všetkých týchto oblastiach je primárnym zdrojom údajov písaný text, organizovaný do potenciálne rozsiahlej štruktúry súborov elektronických textových dokumentov. Z toho vyplynuli všeobecné kritériá pre budovanie systému JBowl, ktorými sú jednoduchá rozšíriteľnosť a modulárna konštrukcia vnútorných modulov na predspracovanie, jazykovú analýzu, indexáciu a ďalšiu analýzu veľkých textových súborov. JBOWL umožňuje predspracovávať rozsiahle kolekcie textových dokumentov pomocou flexibilnej množiny dostupných techník predspracovania, adaptabilných na rôzne typy a formáty textu (napr. čistý text, HTML alebo XML), podporuje indexáciu a vyhľadávanie v rozsiahlych súboroch textových dokumentov s možnosťou využitia na experimenty s rôznymi vyhľadávacími technikami, klasifikačnými a zhlukovacími algoritmami, ako aj znalostnými štruktúrami (ontológie, kontrolované slovníky, atď.).

3 Návrh a realizácia projektu

Základné možnosti využitia BOINC infraštruktúry ako virtuálneho výpočtového gridu v prostredí univerzity už boli analyzované v [4] (popísané stručne aj v [3]). Tu navrhovaný projekt výrazne rozširuje pôvodné využitie a možnosti infraštruktúry, pričom jeho realizáciu je možné rozdeliť do nasledujúcich krokov:

3.1 Vytvorenie BOINC servera

Pre inštaláciu serverovej časti BOINC máme nainštalovaný server na báze OS Linux Debian (i386). Po jeho nastavení sme si vytvorili vlastný BOINC projekt, pre dolovanie v textoch, na ktorý sa účastníci budú môcť pripojiť. Tým pádom bude možné využiť výpočtové prostriedky aj nášho pracoviska a zapojiť ich do internej BOINC infraštruktúry s cieľom zvýšiť výpočtový výkon, využiť nevyužité zdroje a vytvorenie výskumnej stanice na báze BOINC.

3.2 Vytvorenie webového rozhrania – *boinc-tuke.eu*

Webové rozhranie je jedno z kľúčových krokov, ktoré má za úlohu sprístupniť danú infraštruktúru všetkým výskumníkom/používateľom, ktorý sa uchádzajú o realizáciu svojich experimentov. Web stránka je všeobecnou stránkou Technickej univerzity v Košiciach a jeho realizovaných projektoch využitím systému BOINC. Webové rozhranie sa nachádza na doméne: *boinc-tuke.eu* a bolo vytvorené za pomoci CMS systému Joomla, ktorý je základom pre tvorbu dynamického webu. Web stránka je v štádiu vývoja a zatiaľ nie je sprístupnená širokej verejnosti.

Jedným z realizovaných projektov na báze BOINC je aj projekt s názvom JBowl. Po vybratí projektu sa nám zobrazí stručný popis projektu a e-formulár na vytvorenie vlastných experimentov, t. j. vlastných úloh. V e-formulári bude musieť výskumník zadať meno, e-mail a po vybratí želaného prístupu/algoritmu/techniky dolovania sa objavia k nemu prislúchajúce možnosti ako zadávanie hodnôt, či nahratie textových súborov. Po odoslaní sa o uchovanie potrebných údajov postará MySQL databáza so vzdialeným prístupom.

3.3 Vytvorenie aplikácie – JavaDB

Po odoslaní nových úloh do databázy nasleduje fáza načítania, testovania dát, atď., ktoré má zabezpečiť vytvorená aplikácia JavaDB. Aplikácia JavaDB má množstvo funkcií, ktoré majú za úlohu bezproblémový chod celého procesu. Zabezpečuje:

a) sťahovanie dát z úloh a ich nahradenie v príslušnom adresári knižnice JBowl
b) neustále testovanie novej, respektíve ukončenej úlohy a s tým súvisiace odosielanie výstupov zadávateľovi danej úlohy na zadaný email
c) vytvorenie novej BOINC úlohy, nastavenie, nahradenie a vymazanie obsahu BOINC projektu, vytvorenie tzv. workunitov.

3.4 Vytvorenie aplikácie – JbowlBoinc

Pre samotné spúšťanie aplikácie a výber formy techniky dolovania slúži aplikácia JBowlBoinc, ktorá sa spúšťa ako primárna aplikácia na hosťovskom počítači. Zahŕňa v sebe testovanie formy techniky dolovania podľa zadanej úlohy a výber k nemu prislúchajúceho algoritmu/procesu. O všetko ostatné sa postará systém BOINC a nami vytvorená aplikácia JavaDB.

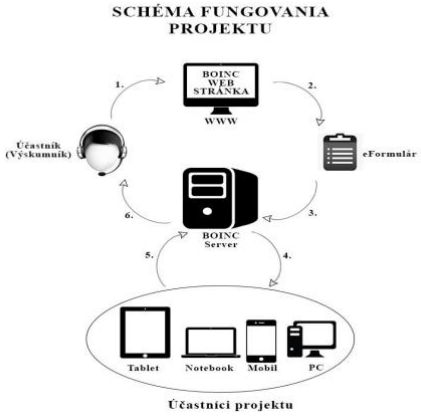
4 Popis fungovania projektu

Pre lepšie pochopenie fungovania celého projektu slúži Obr.1, na ktorom sú zobrazené základné kroky chodu projektu. Popis jednotlivých krokov je nasledujúci:

1. Účastník/výskumník sa pomocou adresy ***boinc-tuke.eu*** dostane na hlavnú stránku web rozhrania. Pri registrácii zadá svoje meno a e-mailovú adresu (pre notifikácie o stave výpočtov a umiestnení výsledkov).
2. Z „Menu“ vyberie možnosť „Projects“ a následne si vyberie realizovaný projekt - JBowl. Po výbere projektu sa zobrazí ***elektronický formulár*** pre výber algoritmu a prislúchajúcich hodnôt a nahratie vstupných súborov. Po odoslaní novej úlohy sa úloha zaradí do fronty úloh na spracovanie systémom BOINC a o uloženie úloh sa postará databáza MySQL.
3. Pomocou vytvorenej aplikácie ***JavaDB*** sa načítajú údaje z databázy zadané účastníkmi. Odstránia sa predošlé nahraté súbory, nastavenia a výsledky a sú nahradené novými podľa danej aktuálnej úlohy.
4. Aplikácia ***JavaDB*** zodpovedá za vytvorenie tzv. workunitov, ktoré sa za pomoci systému BOINC rozpošlú spolu s dátami pripojeným výpočtových

zdrojom daného projektu. Aplikácia ***JBowlBoinc*** slúži na rozpoznanie algoritmu a potrebných vstupov, ako aj pre spustenie samotného algoritmu.

5. Po skončení výpočtov sa všetky ***výsledky výpočtov*** pošlú naspäť na server.
6. BOINC server dáta spracuje a informáciu o výsledkoch pošle zadávateľovi danej úlohy (email notifikácia). Proces sa následne môže opakovať.



Obr. 1 - Schéma fungovania projektu

5 Záver

Projekt využitia softvéru BOINC pre podporu realizácie úloh dolovania v textoch v distribuovanom prostredí je aktuálne rozpracovaný. Je potrebné dopracovať aplikáciu JavaDB a vytvoriť JBowlBoinc aplikáciu (programové rozhranie pre výber a realizáciu algoritmov knižnice JBowl). Následne bude vytvorené webové rozhranie portálovej časti systému a realizované prvotné testovanie. Výsledný projekt bude nasadený a testovaný v prostredí nášho pracoviska.

Poďakovanie. Táto práca bola podporovaná v rámci VEGA grantu č.1/1147/12 a grantu Agentúry pre podporu výskumu a vývoja v rámci projektu č.APVV-0208-10.

Referencie

1. Bednár P, Butka, P, Paralič J. Java library for support of text mining and retrieval. Proc. of *ZNALOSTI 2005*, Stará Lesná, pp. 162-169, 2005.
2. BOINC projekt – http://boinc.berkeley.edu
3. Butka P, Náhori P. Využitie BOINC softvéru ako distribuovaného výpočtového prostredia na zvolenej infraštruktúre. In: *EEI 4*, TU Košice, pp. 203-208, 2013.
4. Náhori P. Návrh využitia projektu BOINC pre vytvorenie virtuálneho univerzitného superpočítačového centra. Bakalárska práca, FEI TU Košice, 2012.
5. Paralič J, Furdík K, Tutoky G, Bednár P, Sarnovský M, Butka P, Babič F. Dolovanie znalostí z textov. Equilibria, Košice, 2010.

Modelovanie domény, dolovanie v dátach, odvodzovanie

Diagnostika metabolického syndrómu ako riadený proces dolovania v dátach

František Babič, Alexandra Lukáčová, Ján Paralič

Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Letná 9/B, 042 00 Košice, Slovenská republika
{Frantisek.Babic, Alexandra.Lukacova, Jan.Paralic}@tuke.sk

Abstrakt. Tento článok predstavuje stručný sumár experimentálnej štúdie, ktorej cieľom je ukázať aplikačný potenciál vhodných metód dolovania v dátach pri analýze medicínskych dát. V tomto prípade išlo o diagnostiku tzv. Metabolického syndrómu, ktorý predstavuje súbor rizikových faktorov kardiovaskulárneho charakteru. Na tento účel sme použili dátovú množinu popisujúcu medicínsku prax spolupracujúceho klinického lekára z Chorvátska, t.j. jednotliví pacienti sú charakterizovaní širokou množinou parametrov, bežne zisťovaných a vyhodnocovaných v ambulanciách praktických lekárov. Kľúčová je v rámci našej metodiky úzka spolupráca s lekárom, ktorý v jednotlivých iteráciách analyzuje získané prediktívne modely a formuluje upresňujúce hypotézy. To si okrem iného vyžaduje použitie takých modelov a techník, ktoré budú pre lekára zrozumiteľné, napr. rozhodovacie stromy alebo asociačné pravidlá. Týmto rozhodnutím sme zabezpečili obojstrannú výmenu informácií, výsledkom ktorej sú znalosti využiteľné v klinickej praxi.

Kľúčové slová: Metabolický syndróm, rozhodovacie stromy, rozhodovacie pravidlá

1 Úvod

Nasadenie vhodných metód dolovania v dátach do rôznych aplikačných oblastí predstavuje v dnešnej dobe veľmi využívanú alternatívu ako identifikovať tzv. skryté znalosti, ako jednoduchým spôsobom tieto dáta zorganizovať alebo ako ich zrozumiteľným spôsobom prezentovať koncovým používateľom. V prípade medicínskych dát je nutná úzka spolupráca a výmena informácií s doménovým expertom, keďže ide o pomerne zložitú a informačne rozsiahlu oblasť. Ideálnym krokom v tomto prípade je zamerať analytické postupy jedným smerom, čiže nesnažiť sa analyzovať všetko, ale venovať pozornosť napr. diagnostike jednej choroby. Bežný spôsob ako klinický lekár diagnostikuje možný výskyt danej choroby je postupný zber všetkých potrebných vstupných faktorov, na základe ktorých si následne vytvorí celkový obraz o zdravotnom stave pacienta a urobí rozhodnutie. Tento postup je však vo väčšine prípadov pomerne časovo náročný a najmä si vyžaduje neustály prehľad a pochopenie stále rastúceho objemu dát.

Práve táto situácia vytvára priestor pre nasadenie vhodných analytických metód, prostredníctvom ktorých bude tento objem dát spracovaný, analyzovaný a dosiahnuté výsledky prezentované používateľovi v zrozumiteľnej forme. V našom prípade bola sledovanou chorobou tzv. Metabolický syndróm (MetSy), ktorý predstavuje súbor rizikových faktorov vedúcich k vzniku kardiovaskulárnych chorôb. Tieto faktory majú spoločného menovateľa a to nedostatočnú odpoveď organizmu na inzulín. Medzi typické faktory teda patria centrálna obezita, porušená tolerancia glukózy, hypertenzia, vysoká hladina tukov, atď. Včasná diagnostika tohto syndrómu umožní pacientovi zmeniť svoj životný štýl, čím zmierni jednotlivé príznaky a zníži riziko kardiovaskulárnych chorôb a cukrovky.

Článok je členený na niekoľko základných častí, kde úlohou prvej je uviesť čitateľa do prezentovanej problematiky; v druhej sú prehľadným spôsobom popísané vybrané kroky analytického procesu; a záver sumarizuje dosiahnuté výsledky a načrtáva ďalšie kroky do budúcej práce.

Tento článok prehľadným spôsobom predstavuje pomerne rozsiahlu skupinu realizovaných experimentov; ich detailnejší popis je možné nájsť v ďalšom článku autorov, ktorý bol nedávno publikovaný v rámci konferencie ITBAM 2014 [1] ako súčasť multikonferencie DEXA 2014.

1.1 Súčasný stav problematiky

Analýza medicínskych dát predstavuje zaujímavú oblasť nasadenia vhodných metód štatistiky, strojového učenia alebo umelej inteligencie. Cieľom je analyzovať príčiny výskytu rôznych chorôb na základe dostupných vstupných faktorov, spomedzi ktorých je možné takto identifikovať vhodné biomarkery. V oblasti diagnostiky MetSy predstavuje zaujímavý príklad štúdia [2], ktorej autori použili Bayesovské siete na analýzu dátovej množiny reprezentujúcej viac ako tisíc pacientov popísaných 18 vstupnými atribútmi, zozbieranú v Yonchon County, Kórea. Podobná práca pochádza tiež z Ďalekého východu [3], v rámci ktorej výskumníci v Thajsku skúmali vzťah medzi hematologickými parametrami a glykemickým statusom za účelom zavedenia kvantitatívneho modelu pre identifikáciu jednotlivcov trpiacich chorobou Diabetes Mellitus (cukrovka). Na určenie glykemického statusu použili SVM alebo neurónové siete, na identifikáciu spoločného výskytu kľúčových parametrov tzv. asociačnú analýzu.

Spoločným znakom nielen týchto dvoch článkov ale celkovo relevantných prípadových štúdií je nutnosť realizovať navrhnuté experimenty a vyhodnotiť dosiahnuté výsledky v úzkej spolupráci s doménovými expertmi.

2 Analýza Dátovej Sady

Popísaný analytický proces bol realizovaný v súlade s metodológiou CRISP-DM, ktorá predstavuje najčastejšie používaný rámec pri riešení úloh analýzy dát a pozostáva zo 6 základných fáz: pochopenie cieľa, pochopenie dát, predspracovanie dát, modelovanie, vyhodnotenie a nasadenie. Samozrejme, tento základný rámec je dostatočne generický a v prípade potreby je ho možné prispôsobiť aktuálnym požiadavkám riešenej úlohy.

Experimenty boli realizované pomocou softvéru R a nástroja SPSS Clementine 10.1.

2.1 Pochopenie cieľa

Dostupná dátová množina predstavuje jednoducho získateľné parametre (faktory), ktoré sú súčasťou zdravotných záznamov pacienta. Na ich základe je možné následne diagnostikovať, či daný pacient trpí MetSy alebo nie. Táto informácia je kódovaná ako binárna premenná, preto z pohľadu analýzy dát pôjde o klasifikačnú úlohu. Okrem samotného výsledku diagnostiky je dôležité poznať aj dôvody, prečo je výsledok pozitívny alebo negatívny. Túto informáciu je možné získať napr. vo forme pravidiel, ktoré budú jednoznačne determinovať aká kombinácia vstupných faktorov a ich príslušných hodnôt vedie k pozitívnemu alebo negatívnemu výsledku diagnostiky, t.j. dolovanie rozhodovacích pravidiel vo forme rozhodovacieho stromu alebo dolovanie asociačných pravidiel. Na vyhodnotenie boli použité typické ukazovatele v oboch prípadoch, čiže miera presnosti klasifikácie, resp. podpora a spoľahlivosť. Tieto ukazovatele však slúžili len ako pomocné veličiny, najdôležitejším krokom vo fáze vyhodnotenia bola spätná väzba od experta, pri ktorej využíval svoje nadobudnuté znalosti, odbornú literatúru a skúsenosti z klinickej praxe. možno preto konštatovať, že šlo nielen o prediktívne, ale najmä o popisné dolovanie v dátach.

2.2 Pochopenie dát

Dátová množina obsahuje informácie o 93 pacientoch z klinickej praxe, ktorú vykonáva spolupracujúci expert v Chorvátsku. Medzi týmito pacientmi sa nachádza 35 mužov a 58 žien vo vekovom intervale 50 až 89 rokov, u ktorých je pomer pozitívna vs. negatívna diagnostika MetSy 60 ku 33. Každý pacient je zároveň charakterizovaný hodnotami 59 faktorov, ktoré predstavujú kľúčové vstupy pre následnú diagnostiku či už vo forme rozhodovacích stromov alebo ďalších experimentov. Z dôvodu limitu na rozsah článku nie je možné prezentovať celú množinu týchto faktorov, ako príklad budú použité faktory tvoriace definíciu IDF.

IDF (International Diabetes Federation) definícia [4] predstavuje jednu z tradičných metód na diagnostiku choroby MetSy, nazývanej tiež syndróm X alebo syndróm inzulínovej rezistencie:

- Kritéria stanovené pre ženy: (pomer obvodu pása a bokov > 0,85 *OR* BMI[1] > 30 kg/m^2) *AND* najmenej 2 splnené zo 4 nasledujúcich podmienok: Vysoký krvný tlak = áno *OR* hladina triglyceridov >1,7 mmol/L *OR* hladina HDL cholesterolu <1,3 mmol/L *OR* hladina krvného cukru ≥ 5,6 mmol/L *OR* Diabetes mellitus = yes.
- Kritéria stanovené pre mužov: (pomer obvodu pása a bokov > 0,9 *OR* BMI > 30 kg/m^2) *AND* najmenej 2 splnené zo 4 nasledujúcich podmienok: Vysoký krvný tlak = áno *OR* hladina triglyceridov >1,7 mmol/L *OR* hladina HDL cholesterolu <1,0 mmol/L *OR* hladina krvného cukru ≥ 5,6 mmol/L *OR* Diabetes mellitus = yes.

[1] Body mass index = mass [kg] / height^2 [m]

Zároveň je možné predpokladať, že sa v rámci experimentov objavia aj iné faktory, prípadne iné rozdelenia hodnôt, ktoré budú mať signifikantný vplyv na diagnostiku MetSy a budú môcť slúžiť ako účinné a lacné biomarkery.

2.3 Analýza a vyhodnotenie

V rámci spomínanej experimentálnej štúdie bolo realizovaných viacero experimentov využívajúcich rôzne metódy dolovania v dátach. V prvom prípade išlo o generovanie rozhodovacích stromov prostredníctvom tradičných algoritmov C4.5 a C5.0 [5]. Z dôvodu pomerne malej vzorky dát sme namiesto tradičného rozdelenia na trénovaciu a testovaciu množinu použili 10-násobnú krížovú validáciu. Výsledné rozhodovacie stromy (na celej vstupnej množine dát, na dátach reprezentujúcich ženy zvlášť a mužských pacientov zvlášť) do veľkej miery potvrdili tzv. IDF definíciu, ale zároveň ukázali aj viacero nových zaujímavých zistení, napr.:

- Porovnanie pravidiel relevantných pre mužov a ženy ukázalo dôležité odlišnosti pri diagnostike MetSy, ktoré je možné potvrdiť aj príslušnou literatúrou [6, 7]: ženy sú náchylnejšie na diabetes a príslušné faktory, muži na druhej strane skôr na faktory sprevádzajúce abdominálnu obezitu. Ako ďalší dôležitý rozlišovací biomarker bol označený tzv. *FOLNA* (koncentrácia kyseliny listovej), ktorý vystupuje v rozhodovacích stromoch len pre mužských pacientov. Dôvodom môže byť práve fakt, že gastroduodenálne poruchy sú častejšie u mužov ako u žien, čo vedie k malabsorpcii a nedostatku kyseliny listovej.
- Zo súboru nových zistení vyberáme biomarker *HbA1c* (glykovaný hemoglobín, parameter odrážajúci priemernú hladina glukózy v krvi počas posledných troch mesiacov), ktorého vplyv už bol potvrdený inou štúdiou [8], ale v rámci našich experimentov bol súčasťou pravidiel pre pozitívnu diagnostiku MetSy spolu s ďalšími faktormi ako kardiovaskulárne ochorenia alebo hladina kortizolu v dopoludňajších hodinách.

Ďalší súbor experimentov bol venovaný experimentálnej identifikácii optimálnej hraničnej hodnoty c, ktorá najlepšie rozdeľuje chorých a zdravých pacientov. Na tento účel sme použili Youdenov index (J) [9], ktorý je definovaný ako

$$J = Senzitivita + Špecifickosť - 1 \quad (1)$$

Jeho výhoda spočíva v ponúknutí najlepšieho výsledku s rešpektom k celkovej správnej klasifikácii maximalizovaním sumy senzitivity (pomer správne klasifikovaných pozitívnych prípadov voči všetkým pozitívnym prípadom) a špecifickosti (pomer správne klasifikovaných negatívnych prípadov voči negatívnym prípadom). Rozsah J je $<0,1>$, kde hodnota 1 znamená, že všetci chorí i zdraví pacienti boli správne klasifikovaní a hodnota 0 naopak značí, že zvolená hraničná hodnota je úplne neefektívna [10]. Úroveň spoľahlivosti bola nastavená na 0,95. O hraničných hodnotách sme uvažovali iba v prípade atribútov, keď bola potvrdená štatistická významnosť nepárovým Studentovým t-testom (t.j. $p<0,05$).

Na základe odporúčania spolupracujúceho klinického lekára sme sa zamerali na nasledujúce rizikové faktory zahrnuté v atribútoch o zápale, veku, renálnej dysfunkcii,

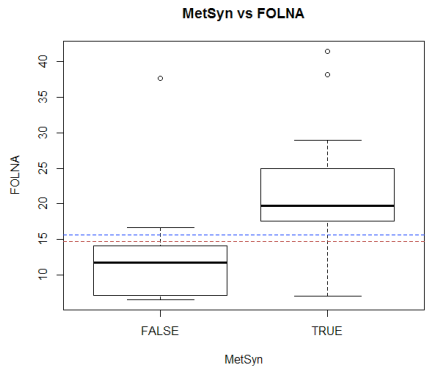
podvýžive, dysfunkcii štítnej žľazy, o hormónoch, anémii/krvnej viskozity, antropometrických hodnôt a glykovaného hemoglobínu. Z dôvodu, že MetSy má rôzne charakteristiky pre mužov a ženy, vykonali sme tento typ experimentov na dvoch dátových vzorkách (35M/58Ž).

Výsledky Studentovho nepároveho testu indikovali ako štatisticky významné len atribúty *FOLNA* a *HbA1c* pre mužov a *MO* (monocyty v bielych krvinkách) a *TSH* (tyreotropný hormón) pre ženy. Optimálne hraničné hodnoty sú prezentované v Tabuľka 1.

Tabuľka 1 Optimálne hraničné hodnoty pre identifikované premenné (PPV- pozitívne predikované hodnoty, NPV- negatívne predikované hodnoty)

Premenná	Hraničná hodnota	Senzitivita (%)	Špecifickosť (%)	PPV (%)	NPV (%)
FOLNA (M)	15,6	95,65	83,33	91,67	90,91
HbA1c (M)	4,5	39,13	100	100	46,15
MO (Ž)	5,5	86,5	14,3	64	37,5
TSH (Ž)	2,69	22,22	100	100	41,67

Z Tabuľka 1 vyplýva, že iba parameter *FOLNA* vykazuje vynikajúce výsledky všetkých štatistických mier. Vďaka týmto vlastnostiam môžeme tento atribút považovať za nový biomarker choroby MetSy, zvlášť vhodný pre skríning mužskej populácie. Premenné *MO* a *TSH* určené pre ženskú populáciu môžu byť tiež užitočné, napriek tomu, že v ich prípadoch vzťah medzi PPV a NPV nie je uspokojivý. Vzhľadom k tomu, že premenná *MO* vykazuje lepšie výsledky citlivosti a premenná *TSH* špecifickosti, ich kombinácia v modeli by mohla byť prínosná. Otázkou ale ostáva, či táto kombinácia je zaujímavá v porovnaní s klasickou metódou vyhodnocovania, založenou na použití konvenčnej definície Metsy. Na Obr.1 je možné vidieť porovnanie nájdenej hraničnej hodnoty pre atribút *FOLNA* pomocou štatistickej analýzy a metódou rozhodovacích stromov, pričom hodnoty sú veľmi podobné.



Obr.1 Rozloženie hodnôt atribútu FOLNA na vzorke dát u mužov s naznačením hraničných hodnôt (červená čiara značí hraničný bod nájdený rozhodovacím stromom, modrá hraničný bod nájdený štatistickou analýzou)

3 Záver

Včasná diagnostika Metabolického syndrómu predstavuje pre pacientov príležitosť znížiť riziko výskytu zdravotných komplikácií typu ateroskleróza, infarkt myokardu alebo mozgová príhoda. Na tento účel sa používajú rôzne vstupné faktory, ktorých vplyv na výslednú hodnotu diagnostiky (pozitívna alebo negatívna) bol predmetom realizovanej experimentálnej štúdie. Dosiahnuté výsledky potvrdili vstupné hypotézy založené na tzv. IDF definícii MetSy a zároveň priniesli viacero nových zistení, ktoré boli následne overené spolupracujúcim expertom prostredníctvom relevantnej odbornej literatúry alebo na základe jeho skúseností z klinickej praxe.

Poďakovanie. Táto publikácia vznikla vďaka podpore Vedeckej grantovej agentúry MŠVVaŠ SR a SAV projekt č. 1/1147/12 (50%) a podpore v rámci operačného programu Výskum a vývoj, pre projekt: Univerzitný vedecký park TECHNICOM pre inovačné aplikácie s podporou znalostných technológií, kód ITMS: 26220220182, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja (50%).

Referencie

1. Babič F, Majnarić L, Lukáčová A, Paralič J, Holzinger A (2014) On Patient's Characteristics Extraction for Metabolic Syndrome Diagnosis: Predictive Modelling Based on Machine Learning. In: Bursa M, Khuri S, Renda ME (eds) Information Technology in Bio- and Medical Informatics. Springer International Publishing, pp 118–132
2. Park H-S, Cho S-B (2012) Evolutionary attribute ordering in Bayesian networks for predicting the metabolic syndrome. Expert Systems with Applications 39:4240–4249
3. Worachartcheewan A, Nantasenamat C, Prasertsrithong P, Amranan J, Monnor T, Chaisatit T, Nuchpramool W, Prachayasittikul V (2013) Machine Learning Approaches for discerning intercorrelation of Hematological Parameters and Glucose Level for identification of diabetes mellitus. EXCLI Journal 12:885–893
4. International Diabetes Federation (2006) The IDF consensus worldwide definition of the Metabolic Syndrome.
5. Holzinger A, Zupan M (2013) KNODWAT: A scientific framework application for testing knowledge discovery methods for the biomedical domain. BMC Bioinformatics 14:1–10
6. Festa A, D'Agostino R, Howard G, Mykkänen L, Tracy RP, Haffner SM (2000) Chronic Subclinical Inflammation as Part of the Insulin Resistance Syndrome The Insulin Resistance Atherosclerosis Study (IRAS). Circulation 102:42–47
7. Onat A, Hergenç G, Keleş I, Doğan Y, Türkmen S, Sansoy V (2005) Sex difference in development of diabetes and cardiovascular disease on the way from obesity and metabolic syndrome. Metab Clin Exp 54:800–808
8. Sluik D, Boeing H, Montonen J, et al (2012) HbA1c Measured in Stored Erythrocytes Is Positively Linearly Associated with Mortality in Individuals with Diabetes Mellitus. PLoS ONE 7:e38877
9. Youden WJ (1950) Index for rating diagnostic tests. Cancer 3:32–35
10. Lai C-Y, Tian L, Schisterman EF (2012) Exact confidence interval estimation for the Youden index and its corresponding optimal cut-point. Computational Statistics & Data Analysis 56:1103–1114

Modelovanie témy v prúde dát z mikroblogov

Miroslav Smatana, Peter Koncz, Ján Paralič, Peter Bednár

Katedra kybernetiky a umelej inteligencie, FEI TU v Košiciach, Slovenská Republika
miroslav.smatana@student.tuke.sk
{jan.paralic, peter.koncz, peter.bednar}@tuke.sk

Abstrakt. Článok pojednáva o výskume ktorého cieľom je vyvinúť systém pre modelovanie témy v prúde dát z mikroblogov a následne ho implementovať vo forme webovej aplikácie. Zdrojom dát sú krátke textové správy zo služby Twitter. V článku je prezentovaný postup na modelovanie tém z krátkych správ (do 200 znakov). Tieto správy navrhovaný systém spracuje, identifikuje preberané témy a vráti témy s ich podrobnejším popisom a prislúchajúcimi správami. Článok popisuje architektúru navrhovaného systému a experimenty s týmto systémom. Experimenty sa sústreďujú na kvalitu zhlukovania správ, ktoré pojednávajú o rovnakej téme.

Kľúčové slová: modelovanie témy, zhlukovanie, spracovanie prirodzeného jazyka

1 Úvod

V posledných rokoch sú sociálne siete vnímané ako jeden z najsilnejších komunikačných nástrojov súčasnosti. Každým dňom je na nich publikované nespočetné množstvo názorov, postojov k rôznym témam, obrázkov, videí a pod.

Tieto príspevky odrážajú názory ľudí na aktuálny vývoj rôznych tém vo svete. Preto majú význam z pohľadu informatívnosti ako pre používateľov tak aj pre spoločnosti. Toto tvrdenie podporuje aj fakt, že vyhľadávače ako Google[1] a Bing[2] zahŕňajú príspevky zo sociálnych sietí do svojho vyhľadávania.

Kvôli množstvu dostupných príspevkov nastáva potreba ich automatického spracovania.

V tomto článku sa zameriavame na automatické spracovanie príspevkov (krátkych správ) z mikroblokov, ktoré tvoria podtriedu sociálnych sietí. Aj napriek tomu, že sa v oblasti spracovania krátkych správ vedú výskumy už niekoľko rokov, stále nepatrí medzi triviálne úlohy. Hlavné problémy v tejto oblasti možno zhrnúť do nasledovných bodov:

[1] https://www.google.sk/
[2] http://www.bing.com/

- Obsahujú správy o maximálnej dĺžke 200 znakov. V tak krátkom texte je pre stroj obtiažne nájsť dostatočné množstvo informácií na ich ďalšie spracovanie.
- V textoch je často využívaný internetový slang.
- Mnoho zo správ obsahuje len odkazy na externé webové stránky.

V nasledujúcich častiach bude predstavený systém na automatické spracovanie krátkych správ zo siete Twitter[3], ktorý vyhodnotí krátke správy z pohľadu sentimentu a témy o ktorej pojednávajú. Navrhovaný systém má za úlohu taktiež popis jednotlivých tém a to priradením ich názvu, kľúčových slov, kľúčových hashtagov a sumarizáciu zahrnutých správ.

2 Modelovanie témy

Hlavnou úlohou, ktorú sme sa v rámci navrhovaného systému podujali riešiť zapadá do rámca modelovania témy z textových dokumentov. Modelovanie témy možno charakterizovať ako zoskupenie podobných dokumentov do zhlukov s ich následným popisom. Úlohu modelovania témy možno rozdeliť na dve samostatné podúlohy: zhlukovanie dokumentov a popis zhlukov.

Existuje množstvo metód zhlukovania dokumentov, mnohé z nich vyžadujú zadať ako parameter počet zhlukov, napr. k-means, k-medoids. Z ďalších prístupov spomenieme aglomeratívne zhlukovanie a Latentné Dirichletovho rozdelenie (LDA) [1][2]. V prezentovanom systéme sme sa rozhodli použiť metódu, ktorá nepatrí medzi štandardne používané metódy v tejto oblasti a je popísaná v nasledujúcej kapitole.

Základné metódy používané na popis zhlukov sú napr. metóda tf-idf a metódy založené na spoločnom výskyte slov v dokumente, ktoré používame aj v našom systéme [3].

3 Návrh systému

Systém je navrhnutý ako knižnica v jazyku Java s využitím niekoľkých podporných knižníc (napr. Gate[4], Gephi[5]). Je schopný automaticky spracovať vstupný súbor krátkych správ zo siete Twitter (množinu tweetov) a na výstupe poskytnúť najčastejšie rozoberané témy spolu s ich popisom a priradenými správami. Taktiež ponúka možnosť podpory viacerých jazykov a dávkového spracovania.

Architektúra systému je znázornená na Obr. 1. V prvej fáze je vykonané predspracovanie vstupných tweetov, pričom sú tieto tweety rozdelené na vety, unigramy a bigramy. Taktiež sa z nich získajú údaje o emotikonoch a metadáta ako čas, jazyk a pod. Následne sa vykonáva zhlukovanie tweetov pomocou hashtagov. Používaná metóda patrí medzi grafové metódy na zisťovanie komunít, popísaná je v práci [4],

[3]https://twitter.com/
[4]https://gate.ac.uk/
[5]https://gephi.github.io/

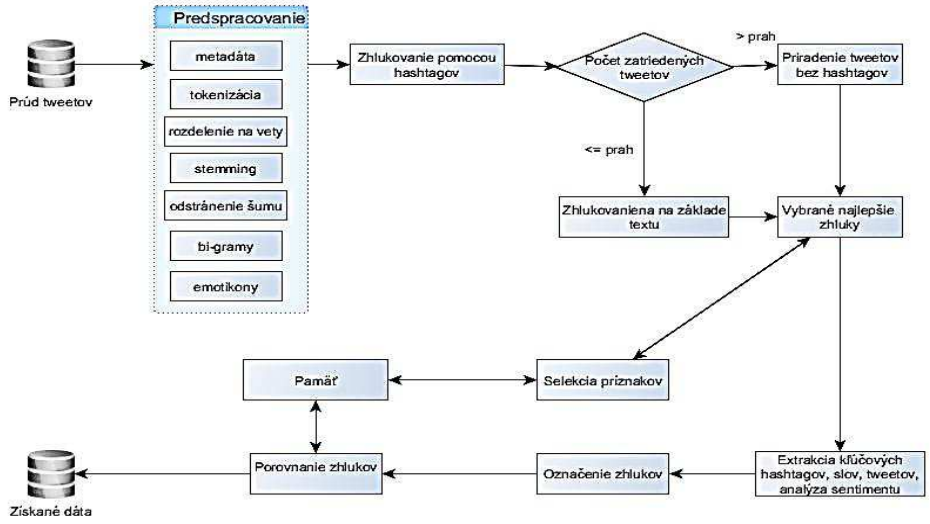
kde uzly grafu predstavujú hashtagy zo všetkých tweetov a hrany predstavujú ich spoločný výskyt v jednotlivých tweetoch.

Ak sa pomocou tejto metódy podarilo vytvoriť dostatočné množstvo tém, ktoré pokrývajú viac tweetov ako je určený prah, tak sa k týmto témam priradzujú aj ostatné tweety, ktoré neobsahujú hashtagy. Keď je tento počet menší ako stanovený prah, použije sa už spomínaná metóda zhlukovania avšak uzly v tomto prípade predstavujú jednotlivé tweety a hrany ich textovú podobnosť.

Po nájdení tém z prúdu vstupných dát sa vykonáva výber najviac frekventovaných tém, ktoré sú dané nadpriemerným počtom tweetov.

Je nevyhnutné tieto nájdené témy nejakým spôsobom popísať. V navrhovanom systéme sme sa rozhodli pre každú tému 1) extrahovať kľúčové slová, 2) kľúčové hashtagy, ďalej robíme 3) sumarizáciu najzaujímavejších tweetov pre danú tému a 4) identifikujeme sentiment pre každý tweet. Sentiment pojednáva o tom či je daný tweet pozitívny alebo negatívny. Pre zlepšenie výsledkov extrakcie uvedených informácií sme sa rozhodli implementovať modul selekcie príznakov, ktorý ako selekčnú metódu využíva informačný zisk [5].

Systém taktiež ponúka možnosť dávkového učenia, kde témy z aktuálneho prúdu dát porovnáva s témami v pamäti a hľadá medzi nimi podobnosť.



Obr. 1 Architektúra navrhovaného systému

4 Experimenty

Navrhovaný systém sme overovali z pohľadu kvality modelovania tém nad dátovou množinou, ktorá pozostávala z 2945 tweetov a 4 tém. Kvalitu sme hodnotili pomocou 3 faktorov: pokrytie (percentuálne vyjadrenie pomeru počtu tweetov, ktoré systém priradil k nejakej z tém k počtu všetkých tweetov, ktoré vstupovali do systému), čistota a normalized mutual information (NMI) popísaných v práci [6]. Výsledky

prezentuje Tabuľka 1. V nej možno vidieť, že metódy použité v navrhovanom systéme dosahujú nad testovacími dátami lepšie výsledky ako štandardné zhlukovacie metódy. Štandardné metódy však dosahujú lepšie hodnoty pokrytia, čo je zapríčinené tým, že nami použité metódy nemusia priradiť všetky tweety k nejakej téme ale môžu ich označiť ako odpad a nezahrnúť ich do svojich výsledkov.

Tabuľka 1 Porovnanie metód zhlukovania

Metóda	Počet zhlukov	Pokrytie	Čistota	NMI
k-means	4	1,0	0,766	0,728
LDA	4	1,0	0,699	0,438
Zhlukovanie pomocou hash-tagov	5	0,925	0,892	0,768
Zhlukovanie na základe textu	4	0,991	0,956	0,848

5 Záver

I keď modelovanie témy z krátkych textov sa javí ako pomerne zložitý problém, tak na základe prezentovaných výsledkov je možné vidieť, že nami navrhnutý systém dosahuje pomerne kvalitné výsledky. Keďže prezentovaný systém je navrhnutý ako knižnica, je ho možné použiť na podporu rôznych iných systémov. V budúcnosti plánujeme upraviť metódy extrakcie informácií a znížiť časovú náročnosť výpočtov.

Poďakovanie

Táto práca bola podporovaná Agentúrou na podporu výskumu a vývoja na základe zmluvy č. SK-CZ-2013-0062 (50%) a vďaka podpore v rámci operačného programu Výskum a vývoj, pre projekt: Univerzitný vedecký park TECHNICOM pre inovačné aplikácie s podporou znalostných technológií, kód ITMS: 26220220182, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja (50%).

Literatúra

1. Steinbach, M., et al.: A Comparsion of Document Clustering Techniques. In: KDD Workshop on Text Mining, Minesota (2000).
2. Gales, M.: Unsupervised Clustering and Latent Dirichlet Allocation. In: MPhil in Advanced Computer Science, Lent (2011).
3. Lott, B.: Survey of Keyword Extraction Techniques. (2012).
4. Bondel, V., et al.: Fast unfolding of communities in large networks. In: J. Stat. Mech., (2008).
5. Forman, G.: An extensive empirical study of feature selection metrics for text classification. In: J. Mach. Learn. Res., vol. 3, pp. 1289-1305, (2003).
6. Manning, Ch., et al.: Introduction to Information Retrieval. Cambridge University Press (2008). ISBN: 0521865719

Paralelné a po častiach hľadajúce riešenie využitia optimalizačných algoritmov

Tomáš Cádrik[1], Marián Mach[1]

[1]Katedra Kybernetiky a umelej inteligencie, Technická univerzita v Košiciach
{tomas.cadrik, marian.mach}@tuke.sk

Abstrakt. Je niekoľko oblastí, kde sa používajú algoritmy ako sú napríklad evolučné klasifikačné systémy, kde sa naučené znalosti ukladajú vo forme modelu daného algoritmu. Ak sa má použiť agent s naučenými znalosťami, je potrebné, aby mal v sebe model, ktorý by vedel interpretovať naučené znalosti. Keďže v dnešnej dobe sa rozmáha cloud a cloud robotika, je nemysliteľné, aby agent v podobe robota mal v sebe uložený model algoritmu, pretože hlavný účel cloud robotiky je odbremeniť ho od zložitých výpočtov. Tento článok predstavuje možnosť, ako extrahovať pravidla z interakcie agenta používajúceho ZCS klasifikačný systém a prostredia. Daný algoritmus využíva paralelne niekoľko evolučných algoritmov a hľadá riešenie po častiach.

Kľúčové slová: Animat problém, cloud robotika, evolučný algoritmus, extrakcia pravidiel, ZCS

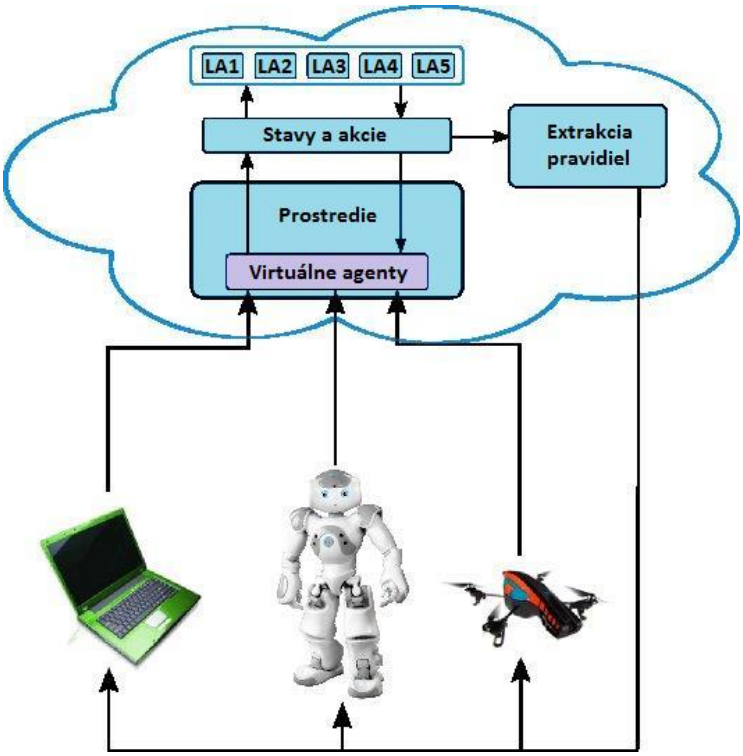
1 Úvod

Hlavným cieľom nášho výskumu je vytvoriť učiaci systém, kde by jednotlivé učiace procesy prebiehali na cloude. Rovnako by sa na základe naučených znalostí tvorila databáza, ktorá by sa taktiež nachádzala na cloude. Hlavná definícia cloudu sa nachádza v [1] a prehľad výskumu v oblasti cloudu sa nachádza v [2]. Naučené znalosti by používali zariadenia (agenti, roboty). Používanie kombinácie cloudu a robotiky sa nazýva cloud robotika [3].

Plánovaný systém by mal fungovať nasledovne: Užívateľ zadá úlohu zariadeniam (agentom). Tento agent sa pripojí na cloudovú službu. Najprv sa skontroluje, či databáza obsahuje pravidla, ktorými by bolo možné vyriešiť zadanú úlohu. Ak sa v databáze pravidiel nenachádzajú potrebné pravidla, začne učenie. Učenie bude prebiehať na virtuálnom prostredí s virtuálnymi agentami. Keď sa učenie skončí, naučené znalosti sa pošlú zariadeniu, ktoré začne riešiť zadanú úlohu.

Problémom však ostáva, že v prípade použitia napríklad neurónových sietí, resp. evolučných klasifikačných systémov [4], je potrebné mať v zariadení model, ktorý by vedel zaobchádzať s naučeným modelom. Keďže tejto možnosti sa chceme vyhnúť, je potrebné z naučeného modelu vyextrahovať pravidlá, ktoré by zariadenia boli schopné používať. Na tieto účely bol vytvorený algoritmus, ktorý je schopný vytvoriť pravidlá z interakcie učiaceho algoritmu riadiaceho agenta a prostredia, v ktorom sa ten

agent pohybuje. Ako učiaci algoritmus na riadenie agenta sa použil ZCS klasifikačný systém [5]. Na extrakciu bola vytvorená metóda, používajúca paralelne viac evolučných algoritmov [6] a tie evolvujú riešenie po častiach. Schéma plánovaného cloudového systému sa nachádza na Obr. 1. Plánované je použiť cloudové riešenie od firmy Microsoft – Azure.



Obr. 1. Cloudová služba na učenie pravidiel a vytváranie databázy pravidiel. LA1 až LA5 sú učiace algoritmy. Obrázok vychádza z [7].

Článok je organizovaný nasledovne. Sekcia 2 obsahuje popis ZCS a animat problému, ktorý je použitý ako prostredie pre agenta. Sekcia 3 obsahuje popis novovytvoreného algoritmu na extrakciu pravidiel. Sekcia 4 obsahuje experimenty dokazujúce funkčnosť vytvorenej metódy. Sekcia 5 obsahuje záver.

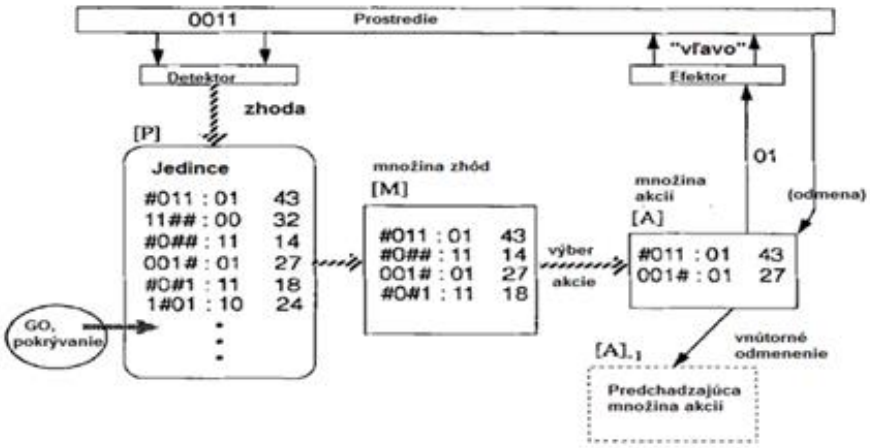
2 ZCS klasifikačný systém a animat problém

Klasifikačné systémy boli predstavené Johnom Hollandom. Tieto metódy používajú jedincov kódujúcich jednotlivé pravidlá, rovnako aj genetické operátory. Delia sa na dva základné štýly: Pittsburghský a Michiganský. Pri Pittsburghskom štýle jeden jedinec znázorňuje celú sadu pravidiel. V Michiganskom štýle jeden jedinec znázorňuje jedno pravidlo. Existujú dva základné klasifikačné systémy Michigenského štý-

lu: ZCS [5] (Klasifikačný systém nultej úrovne) a XCS [8] (Klasifikačný systém založený na presnosti). Tieto systémy boli použité na riešenie jednokrokových úloh ako k-multiplexer problém [8] resp. úloh sekvenčného typu ako animat problém [5][8], alebo corridor problém [9].

Animat problém je prostredie, obsahujúceho agenta, prekážky, prázdne cesty a jedlo. Cieľom je dostať sa z počiatočnej pozície k jedlu pri použití čo najmenšieho počtu krokov. Agent sa môže pohybovať ôsmimi smermi a v prípade, že sa pohne smerom ku prekážke, stratí ťah a ostane stáť na mieste.

Schéma ZCS systému sa nachádza na Obr. 2.



Obr. 2. Schéma ZCS klasifikačného systému z [5]

Jedinec v ZCS sa skladá z predpokladovej časti a záverovej časti. Predpokladová časť sa skladá zo znakov {0, 1, #}, kde znak # znamená, že na danej pozícií nezáleží. Záverová časť môže obsahovať ľubovoľné znaky. Na začiatku sa vygeneruje náhodná populácia. Následne začína interakcia ZCS a prostredia. Prostredie vráti svoj stav (stav agenta). Následne sa vyberú tie jedince kde všetky pozície neobsahujúce # sa zhodujú so vstupom z prostredia. Tieto jedince sa skopírujú do [M] (Množiny zhôd). Následne sa vyberie jedinec z [M] pomocou ruletovej metódy na základe hodnôt vhodností. Tento jedinec ako aj všetci ostatný jedinci z [M] s akciou ako vybraný jedinec sa skopírujú do množiny akcií [A]. Táto akcia sa pošle prostrediu a tam sa vykoná. Prostredie následne vráti odmenu. Na základe tejto odmeny sa použitím metód strojového učenia [10] aktualizuje [A] aj [A] z predchádzajúceho cyklu.

ZCS taktiež obsahuje operátor pokrytia, ktorý vytvorí nového jedinca korešpondujúceho so vstupom z prostredia, a genetické operátory vytvárajúce nových jedincov pomocou kríženia a mutácie.

Jedinec v ZCS kóduje stav prostredia ako osem susedstvo agenta. Prvé dve hodnoty kódujú pozíciu na sever od agenta. Ostatné dvojice v jedincovi kódujú postupne pozície od tej vrchnej v smere hodinových ručičiek. Prostredie vracia odmenu 1000 ak agent došiel na políčko s potravou, v opačnom prípade je odmena 0. Vždy keď agent dôjde k potrave, v následnom cykle začína na náhodnom prázdnom políčku.

3 Paralelné a po častiach hľadajúce riešenie využitie evolučného algoritmu

V tomto článku sa spomínajú evolučné algoritmy pri kombinácií s predstavenou metódou. Je však možné použiť ľubovoľný optimalizačný algoritmus používajúci jedincov a rozkladanie riešenia na zložky.

Na začiatku sa vytvorí *n* (parameter metódy) evolučných algoritmov. Každý algoritmus začne hľadať riešenie nezávisle na ostatných. Každý evolučný algoritmus na počiatku hľadá iba časť riešenia. Po skončení procesu hľadania v každom jedincovi zamrzne nájdené riešenie, čo znamená, že s touto časťou riešenia sa už nebude manipulovať. Následne sa opäť spustia evolučné algoritmy. Každý jedinec hľadá druhú časť riešenia, prvá ostáva nezmenená. Po skončení hľadania sa v každom evolučnom algoritme porovná vhodnosť najlepšieho jedinca v danom a predchádzajúcom cykle. Ak v aktuálnom cykle je nižšia, alebo rovná, ako v tom predchádzajúcom, algoritmus zamrzne a už je ďalej nečinný. Keď sa do tohto stavu dostanú všetky evolučné algoritmy, hľadanie končí a riešením je najlepší jedinec spomedzi všetkých jedincoch zo všetkých evolučných algoritmov.

Ak chceme extrahovať pomocou vytvorenej metódy pravidla zo ZCS, je potrebné najprv ZCS naučiť. Následne sa znova spustí naučené ZCS a sa budú ukladať podmienkové a k nim prislúchajúce záverové časti použité pri interakcií s prostredím. Následne sa použijú paralelne a po častiach hľadajúce riešenie evolučné algoritmy na vytváranie vhodnej kombinácie pravidiel, zahŕňajúcej čo najviac pravidiel z interakcie naučeného ZCS s prostredím. Nájdené pravidla môžu obsahovať aj znak #. Vhodnosť jedinca sa vypočíta na základe pravidiel ktoré má v sebe zakódované.

4 Experimenty

Pri experimentoch boli hodnoty parametrov ZCS nastavené podľa [7]. Naša nová vytvorená metóda mala 10 evolučných algoritmov a každý evolučný algoritmus obsahoval 100 jedincov. Bolo použité jednobodové kríženie a mutácia, meniaca pozície jedinca na iný možný znak, ktorý sa môže nachádzať na danej pozícií. Ako selekčná metóda bol použitý 3-árny turnaj. Pri náhrade sa polovica najhorších jedincov v populácií nahradí polovicou najlepších jedincov spomedzi potomkov.

Klasifikačný systém bol učený počas 10000 cyklov. Následne 1000 cyklov bolo použitých na testovanie ZCS. Pre zníženie výpočtovej náročnosti na hľadanie pravidiel pomocou našej metódy bolo použitých iba prvých 200 uložených pravidiel z interakcie ZCS a prostredia.

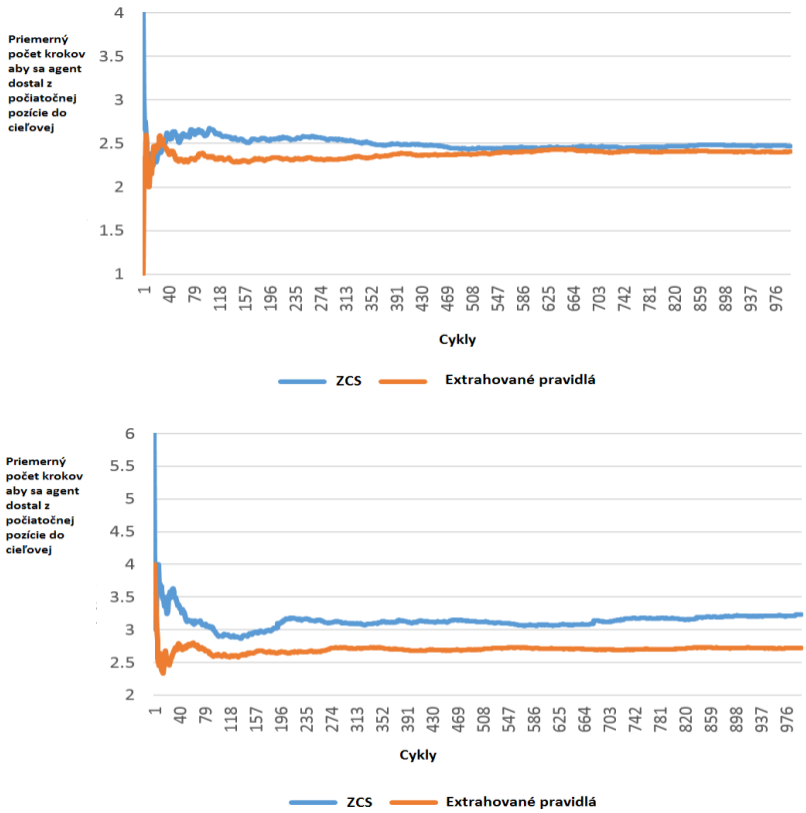
Prostredia na ktorých bola metóda testovaná sa nachádzajú na Obr. 3.

T	T	T	T
T		F	T
T		T	T
T			T
T		T	T
T	T	T	T

T	T	T	T	T
T			F	T
T		T	T	T
T			T	T
T		T	T	T
T	T	T	T	T

Obr. 3. Prostredia MazeF1 resp. MazeF2. T sú prekážky a F je potrava.

Priemerný počet krokov pre agenta aby sa dostal od počiatočnej pozície k potrave v použitých prostrediach sa nachádza na Obr. 4.



Obr. 4. Porovnanie ZCS s vytvoreným extraktorom

Z grafov vyplýva, že pri extrakcií pravidiel pomocou predstavenej metódy je priemerný počet krokov k potrave lepší ako pri použití ZCS. Aj napriek tomu, že v ZCS stále prebieha učenie, je po poslednom cykle výsledok horší. Ak sa v extrahovaných pravidlách nenachádzalo pravidlo s podmienkovou časťou ako bol vstup z prostredia, použila sa náhodná akcia. Z grafov teda vyplýva tiež, že väčšina dôležitých pravidiel bola obsiahnutá.

5 Záver

V predošlých sekciách sa nachádza popis ZCS, animat problému a paralelného po častiach hľadajúceho riešenie využitie evolučného algoritmu. Experimenty ukázali, že túto metódu je možné použiť na extrakciu pravidiel z interakcie medzi ZCS a prostredím, v tomto prípade prostredím animat problému. V oboch prostrediach bola naša nová metóda lepšia ako učiaci sa ZCS.

Ďalším postupom pri použití tejto metódy bude testovanie pri inej konfigurácií evolučného algoritmu (iný typ selekcie a podobne). Rovnako môže byť použitá pri inom učiacom algoritme ako sú napríklad neurónové siete. V tomto prípade sa môže použiť iný optimalizačný algoritmus, ako je napríklad PSO [11], ktorý je pri reprezentácií s reálnymi číslami vhodnejší ako evolučný algoritmus.

Poďakovanie

Výskum podporovaný Národným projektovým grantom pre výskum a vývoj 1/0667/12 „Inkrementálne metódy učenia pre inteligentné systémy“ 2012-2015.

Referencie

1. P. Mell and T. Grace, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” NIST Spec. Publ., vol. 145, p. 7, 2011.
2. D. Lorencik and P. Sincak, “Towards Cloud Robotics Age,” in SCYR 2013 : Proceedings from conference :13th Scientific Conference of Young Researches, 2013, pp. 43–46.
3. G. Hu, W. Tay, and Y. Wen, “Cloud robotics: architecture, challenges and applications,” Network, IEEE, no. June, pp. 21–28, 2012.
4. T. Cádrik and M. Mach, “Evolutionary classifier systems,” in Electrical Engineering and Informatics IV: Proceedings of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice: FEI TUKE, 2013, pp. 168–172.
5. S. W. Wilson, “ZCS: A Zeroth Level Classifier System,” Evolutionary Computation, vol. 2. pp. 1–18, 1994.
6. M. Mach, Evolutionary algorithms: elements and principles. Košice: Elfa, 2009, p. 250.
7. T. Cádrik and M. Mach, “Extracting rules from ZCS evolutionary classifier system for the purpose of future usage in robotic systems,” in Proceedings of the 20th International Conference on Soft Computing MENDEL 2014, Brno, 2014, pp. 393–396.
8. S. W. Wilson, “Classifier Fitness Based on Accuracy,” Evolutionary Computation, vol. 3. pp. 149–175, 1995.
9. K. W. Tang and R. A. Jarvis, “Is XCS Suitable For Problems with Temporal Rewards?,” Int. Conf. Comput. Intell. Model. Control Autom. Int. Conf. Intell. Agents, Web Technol. Internet Commer., vol. 2, 2005.
10. K. Machová, Machine learning: principles and algorithms. Košice: Elfa, 2002, p. 117.
11. R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization,” Swarm Intelligence, vol. 1. pp. 33–57, 2007.

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 45–50

Transformačná regresná technika pre dolovanie v údajoch

Peter Krammer, Ladislav Hluchý

Ústav informatiky, Slovenská akadémia vied
Dúbravská cesta 9, 845 07 Bratislava, Slovenská republika
{peter.krammer,ladislav.hluchy}@savba.sk

Abstrakt. Jedným z problémov v oblasti dolovania je nedostatočný počet dostupných údajov. To spôsobuje nižšiu reprezentatívnosť trénovacej množiny, čo sa prejavuje primárne efektom výrazného zníženia presnosti natrénovaného modelu. Prezentovaný článok predkladá techniku transformujúcu klasickú regresnú úlohu na ekvivalentnú s cieľom zvýšiť počet dostupných záznamov a tak dosiahnuť zvýšenie presnosti modelu. Článok tiež obsahuje experimentálne overenie vplyvu aplikovania techniky na presnosť natrénovaného modelu, ako aj poukazuje na niektoré vybraté vlastnosti a dôsledky využívanej transformácie.

Kľúčové slová: dolovanie, regresia, transformácia údajov, generovanie dát

1 Úvod

Jedným z problémov v oblasti dolovania v údajoch je nedostatok relevantných údajov. Pri aplikovaní techník dolovania sa uvažuje, že máme k dispozícii dostatočne rozsiahle údaje, zvyčajne historické, v ktorých je obsiahnutý aj hľadaný vzťah medzi veličinami. Avšak v praxi sa často stáva, že získanie väčšieho počtu záznamov je značne problematické, resp. finančne nákladné. Relatívne nízky počet záznamov v trénovacej množine môže zapríčiniť nízku reprezentatívnosť množiny, čo má značný vplyv na presnosť natrénovaného modelu. Medzi spôsoby, ako zvýšiť presnosť trénovaného modelu patria rozličné metódy ako sú Bagging [1], Additive regression [2], Boosting [3], Stacking a podobne. Prezentovaná technika do určitej miery aj pripomína niektoré z týchto metód, nakoľko pri predpovedaní cieľovej hodnoty jedného záznamu, musí byť natrénovaný model aplikovaný viacnásobne. Avšak rozdielom je, že technika využíva iba jedinú zvolenú štruktúru modelu, z ktorej má jedinú inštanciu; čo je podrobnejšie uvedené nižšie. Vo všeobecnosti sú však metódy zloženého učenia vhodnejšie najmä na účely spresňovania modelu.

Otázkou zostáva, aké sú možnosti v prípade, ak dostupných údajov je k dispozícii len malý počet. Jednou z možností je analyzovanie dostupných údajov, určenie ich štatistických parametrov a následné vygenerovanie ďalších údajov s rovnakými štatistickými parametrami. Takýto prístup však nie je úplne korektný, nakoľko cieľový atribút nedokážeme striktne zadefinovať vo vzťahu k vstupným atribútom.

Oproti tomu, prezentovaný algoritmus prináša jeden zo spôsobov, ako transformovať regresnú úlohu na ekvivalentnú a vygenerovať pri tom väčší počet záznamov bez toho, aby sme nejaké hodnoty odhadovali. Symbolické znázornenie štruktúry dát pôvodnej úlohy je zobrazené v tabuľke 1.

Table 1. Pôvodná dostupná dátová množina k regresnej úlohe

záznam	vstupný atribút X	vstupný atribút Y	cieľový atribút O
(1)	x_1	y_1	o_1
(2)	x_2	y_2	o_2
(3)	x_3	y_3	o_3
(4)	x_4	y_4	o_4

Dátová Transformácia

Prezentovaná transformácia je vhodná len pre numerické typy atribútov. Pre N dostupných záznamov (v demonštračnom prípade znázornenom v tab. 1 je počet záznamov N = 4 samozrejme len symbolický), bude transformovaná tabuľka obsahovať N^2 - N záznamov, pričom každý záznam transformovanej dátovej množiny je definovaný na základe jednej dvojice pôvodných záznamov. Z pôvodných údajov sa preberajú jednotlivé atribúty a tiež diferencie atribútov. Celkový počet vstupných atribútov je tak vždy dvojnásobne vyšší oproti pôvodnému počtu.

Table 2. Transformovaná dátová množina regresnej úlohy

použité záznamy	X	Y	ΔX	ΔY	ΔO
(1) a (2)	x_1	y_1	x_1-x_2	y_1-y_2	o_1-o_2
(1) a (3)	x_1	y_1	x_1-x_3	y_1-y_3	o_1-o_3
(1) a (4)	x_1	y_1	x_1-x_4	y_1-y_4	o_1-o_4
(2) a (1)	x_2	y_2	x_2-x_1	y_2-y_1	o_2-o_1
....	...	...	...	...	...
(4) a (3)	x_4	y_4	x_4-x_3	y_4-y_3	o_4-o_3

Štruktúra dát po transformácií je znázornená v Tab. 2, pričom na mieste cieľového atribútu je už zmena atribútu O, označená ako ΔO. Počet vstupných atribútov sa zdvojnásobujú, nakoľko sa v trénovacej množine vyskytujú aj pôvodné atribúty aj ich diferencie. Nad takto vytvorenou tabuľkou údajov realizujeme trénovanie regresného modelu. Model označíme ako funkciu f().

$$p = f(X, Y, \Delta X, \Delta Y) \qquad \text{(Eq. 1)}$$

Takto natrénovaný model nám podá odhad diferencie cieľového atribútu voči jednotlivým záznamov trénovacej množiny.

2 Predikcia

Predikcia hodnoty cieľového atribútu O k stanovenému záznamu A (popísaného pomocou vektora (x_A, y_A)) sa realizuje nasledovne. Záznam A, pre ktorý máme vypočítať cieľový atribút O, najskôr transformujeme do tvaru, údajov, ktoré sme použili na trénovanie modelu f(). Keďže máme k dispozícii až N záznamov, preto môžeme vytvoriť až 2.N záznamov o diferenciách jednotlivých záznamov trénovacej množiny voči záznamu A. Takto transformované záznamy sú znázornené v Table. 3.

Table 3. Transformovaný záznam určený k predikcií.

použité záznamy	X	Y	ΔX	ΔY	ΔO
(1) a (A)	x_1	y_1	$x_1 - x_A$	$y_1 - y_A$	p_{1A}
(2) a (A)	x_2	y_2	$x_2 - x_A$	$y_2 - y_A$	p_{2A}
(3) a (A)	x_3	y_3	$x_3 - x_A$	$y_3 - y_A$	p_{3A}
(4) a (A)	x_4	y_4	$x_4 - x_A$	$y_4 - y_A$	p_{4A}
(A) a (1)	x_A	y_A	$x_A - x_1$	$y_A - y_1$	p_{A1}
(A) a (2)	x_A	y_A	$x_A - x_2$	$y_A - y_2$	p_{A2}
(A) a (3)	x_A	y_A	$x_A - x_3$	$y_A - y_3$	p_{A3}
(A) a (4)	x_A	y_A	$x_A - x_4$	$y_A - y_4$	p_{A4}

Na takto transformované dáta sme schopný aplikovať natrénovaný predikčný model f(), ktorý nám tak predpovie odhady veličiny ΔO. Výstupom predikčného modelu f() sú teda hodnoty p, ktoré sú aproximáciou hodnôt Δo. Pre i = 1, 2, 3, 4 platí:

$$p_{iA} \cong \Delta o_{iA} \quad \text{(Eq. 2)}$$

resp.

$$p_{iA} = \Delta o_{iA} \pm E_r \quad \text{(Eq. 3)}$$

pričom E_r predstavuje chybu regresného modelu f(). Pritom platí že

$$\Delta o_{iA} = o_i - o_A \quad \text{(Eq. 4)}$$

a tiež

$$\Delta o_{Ai} = o_A - o_i \quad \text{(Eq. 5)}$$

Je podstatné si uvedomiť, že síce pre hodnoty Δo_{Ai} a Δo_{iA} platí že $\Delta o_{Ai} = -\Delta o_{iA}$; avšak pre ich odhady p_{iA} a p_{Ai} to platiť nemusí nakoľko model f() nemusí byť lineárny. Preto je vhodné použiť všetkých 2N záznamov. Vzhľadom na to, že $p_{iA} \cong \Delta o_{iA}$ (Eq. 2) a známe hodnoty o_i, je možné vyčísliť odhad hodnoty o_A z (Eq. 4) a (Eq. 5), a to z každého jedného riadku tabuľky 3, pri použití vzťahov:

$$o_A = o_i - \Delta o_{iA} \cong o_i - p_{iA} \quad \text{(Eq. 6)}$$

$$o_A = o_i + \Delta o_{Ai} \cong o_i + p_{Ai} \quad \text{(Eq. 7)}$$

Takýmto spôsobom získame 2N odhadov veličiny O_A, ktoré zodpovedajú hodnote cieľového atribútu. Pre určenie výsledného odhadu hodnoty cieľového atribútu sa ponúka viacero možností.

- použitie tradičného aritmetického priemeru z jednotlivých odhadov.
- vylúčenie extrémnych hodnôt (napríklad 1 maximálnej a 1 minimálnej hodnoty), a výpočet aritmetického priemeru zo zostávajúcich hodnôt.

- váhovaný priemer, pričom váhy by boli určené z validácie modelu.
- váhovaný priemer, pričom váhy by boli definované na základe prevrátenej hodnoty vzdialenosti jednotlivých záznamov od predikovaného záznamu.

Nakoľko máme k dispozícii väčší počet odhadov cieľovej hodnoty, naskytuje sa tiež možnosť určiť štatisticky intervalový odhad pre túto hodnotu a ohraničiť tak hodnotu cieľového atribútu. Pre tento účel je však vhodné využiť iba N hodnôt (vzhľadom na riziko výraznej závislosti) pripadajúcich k N záznamov, teda buď prvú alebo druhú polovicu záznamov tabuľky 3.

Trénovanie bolo realizované použitím knižnice Weka [4].

2.1 Vlastnosti

Takýto prístup využívajúci transformované dáta má v dolovaní viacero výhod:

- namiesto N záznamov trénovacej množiny máme k dispozícií N^2 - N záznamov.
- namiesto cieľovej veličiny O predpovedáme zmenu veličiny O, čo môže viesť k vyššej senzitivite modelu.
- model sa aplikuje až 2N krát, teda zohľadnia sa diferencie voči všetkým dostupným záznamom; výsledok je priemerom (možnosť použiť rozličných priemerov) čo zvyšuje stabilitu modelu ako celku.
- pri danom natrénovanom modely, ak získame nové záznamy, je možné ich použiť pri procese predikcie k vygenerovaniu vyššieho počtu transformovaných záznamov pre spresnenie.

Samozrejme použitá technika prináša aj určité nevýhody:

- vyššia časová a pamäťová náročnosť (procesu trénovania aj predikcie) súvisiaca s vyšším počtom záznamov a vyšším počtom aplikovaní modelu.
- ťažko predvídateľný vplyv závislosti záznamov v transformovanej trénovacej množine na stabilitu modelu (v závislosti od typu a štruktúry modelu).

3 Testovanie výkonnosti

Prezentovaná technika bola otestovaná na vygenerovaných dátach, ktoré obsahovali 3 vstupné atribúty označené ako Attr1, Attr2 a Attr3. Cieľový atribút O bol pre testovaciu aj trénovaciu množinu definovaný ako:

$$O = Attr1.\ln(Attr2 + 1).\sqrt{Attr3} \qquad \text{(Eq. 8)}$$

Testovacia množina pozostávala z 1000 záznamov, pričom každý vstupný atribút nadobúdal rovnomerne hodnoty 1, 2, ... 10 (množina obsahovala ich všetky kombinácie). Trénovacia množina pozostávala z náhodne vygenerovaných vstupných atribútov z intervalu <1, 10>. Počet záznamov v trénovacej množine bol 60, pričom v priebehu experimentu bol postupne redukovaný. Pri tejto redukcií trénovacej množiny bola pozornosť smerovaná na výkonnosť modelu. Ako model bola použitá neurónová sieť perceptronov s jednou skrytou vrstvou; aktivačnou funkciou bol

sigmoid. Koeficient učenia bol 0.3 a počet epoch trénovania bol nastavený na 500. Neurónové siete boli najskôr viacnásobne trénované tradičným spôsobom nad originálnymi dátami, ktoré svojou štruktúrou zodpovedajú tabuľke 1. Každý experiment bol realizovaný 4-krát, s rozličnými hodnotami seedu - pre inicializáciu neurónovej siete, pričom z dosiahnutých výsledkov bol vypočítaný priemer.
Celý proces trénovania bol realizovaný s trénovacou množinou obsahujúcou 60 záznamov, po jeho skončení bolo trénovanie zopakované s nižšími počtami záznamov v trénovacej množine. Za rovnakých podmienok bolo vykonané trénovanie neurónových sietí s využitím prezentovanej transformačnej techniky, pričom jediným rozdielom bola štruktúra trénovacích dát, a teda aj spôsob predikcie. Porovnanie výkonnosti pre rozličné rozsahy dátových množín, ako aj rozdielne prístupy sú znázornené v grafoch. Na grafoch Fig.1 a Fig.2 je znázornené porovnanie výkonnosti (vyčíslenej pomocou korelačného koeficientu a strednej kvadratickej odchýlky) natrénovaných modelov. Modré kosoštvorce v grafe reprezentujú výkonnosť modelu natrénovaného tradičným spôsobom, teda ako trénovacia množina bola použitá originálna dátová množina, ktorej štruktúra zodpovedá Table. 1. Červené štvorce prislúchajú prezentovanej metóde využívajúcej transformáciu originálnej regresnej úlohy.

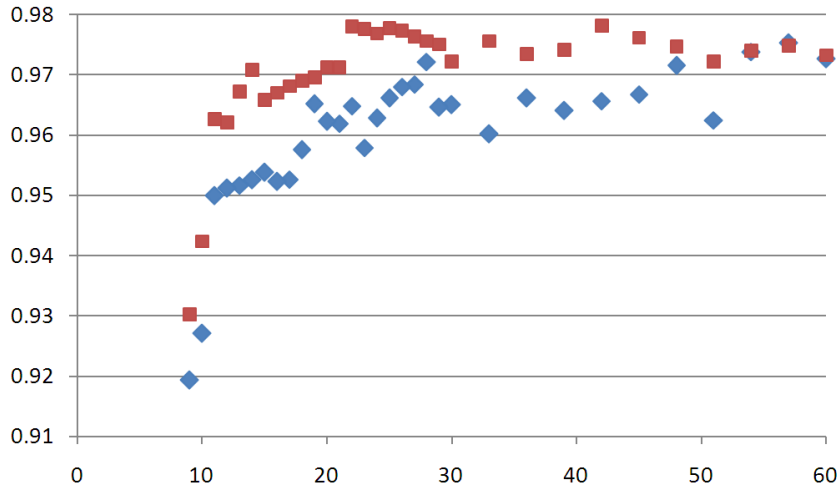


Fig. 1. Závislosť kritéria - korelačného koeficientu od počtu záznamov v trénovacej množine.

Z grafu na Fig. 1 je vidieť že pri vyššom počte záznamov (v našom prípade vyššom než 50), sa jednotlivé značky prekrývajú a výkonnostne sú modely takmer ekvivalentné. V tomto intervale teda transformácia neprináša žiadne zlepšenie. V prípade veľmi malého počtu záznamov (v našom prípade 10 a menej) je zrejmý výrazný prepad výkonnosti modelov spôsobený práve veľmi nízkou reprezentatívnosťou trénovacej množiny. Pre takto enormne nízky počet záznamov teda modelovanie stráca význam. Pri zameraní sa na interval (10, 50) vidíme, že prezentovaná technika dosahuje vo všetkých prípadoch lepšiu presnosť modelu. Tento

aspekt je zrejmý aj z Fig. 2. Samozrejme, že pre odlišné dátové množiny sa jednotlivé počty, pri ktorých dochádza k zmene výkonnosti modelu líšia taktiež.

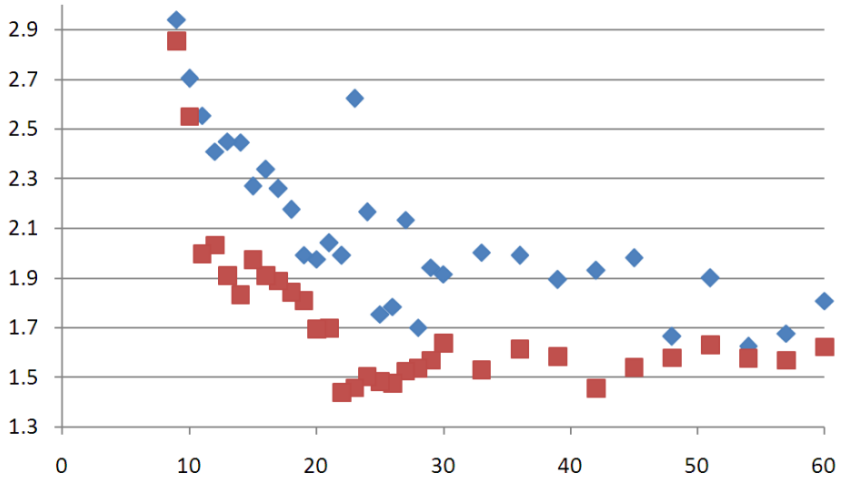


Fig. 2. Závislosť strednej kvadratickej odchýlky od počtu záznamov v trénovacej množine.

4 Zhodnotenie

Predložený článok prezentuje techniku transformácie numerických dát, vhodnú pre trénovanie regresných modelov. Technika je určená primárne pre prípady veľmi nízkych počtov záznamov. Prezentovanú techniku je potrebné otestovať na viacerých rozličných dátových množinách, čo ukáže jej plný potenciál. Taktiež sa naskytujú viaceré verzie, resp. modifikácie prezentovanej techniky, obzvlášť pri určovaní výslednej hodnoty z množstva odhadov.

Ako bolo možné vidieť z grafov, prezentovaná technika dokáže do určitej miery kompenzovať malý počet záznamov miernym zvýšením presnosti modelu. Avšak to, do akej miery je schopná zvýšiť presnosť, resp. v akých prípadoch prináša zlepšenie kvality modelu je nutné ešte podrobne preskúmať a otestovať.

Poďakovanie. Článok je podporovaný z projektov CVR ITMS 26240220082, KC-INTELINSYS ITMS 26240220072, RIOT APVV-0233-10 a VEGA 2/0054/12.

Referencie

[1] Leo Breiman: Bagging predictors. Machine Learning 24(2), 1996, p. 123-140.
[2] J.H. Friedman: Additive Regression - Stochastic Gradient Boosting, 1999.
[3] Yoav Freund, Robert E. Schapire: Experiments with a new boosting algorithm. 13. Inter. Conf. on Machine Learning, San Francisco, 1996, p. 148-156.
[4] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); WEKA Data Mining Software; SIGKDD Explorations, Volume 11, Issue 1.

Objavovanie vzťahov v grafe s využitím pravidiel

Ján Mojžiš, Michal Laclavík

Ústav Informatiky, Slovenská Akadémia Vied,
Dúbravská cesta 9, 845 07 Bratislava, Slovenská republika
{upsyjamo,michal.laclavik}@savba.sk

Abstrakt. Výpočtový model Pregel, sa čoraz viac začína používať pre počítanie distribuovaných výpočtov na grafoch. Pregel nachádza využitie pre výpočty na grafoch sociálnych sietí ale vhodný je aj pre iné oblasti (medicína, grafové algoritmy). V tejto práci navrhujeme algoritmus Pregel Relationship Discovery (PRED) pre výpočtový model Pregel. Objavujeme vzťahy vo veľkom RDF grafe databázy Freebase. Naše pravidlá nám umožňujú zvoliť si množinu vrcholov, medzi ktorými chceme objavovať vzťahy (vrcholy). Veľká výhoda nášho prístupu je v možnosti navigovať sa aj proti smeru hrán, pričom nie je potrebná dodatočná indexácia a pôvodná štruktúra grafu je zachovaná.

Kľúčové slová: Pregel, distribuované počítanie, objavovanie vzťahov, veľké dáta.

1 Úvod

Pregel je distribuovaný výpočtový model pre počítanie na grafoch. Jadrom modelu je vrchol grafu. Výpočet prebieha v metóde *compute()* pre každý vrchol zvlášť. Koľko je vrcholov, toľko je inštancií vrcholu s metódou *compute()*. Pregel je používaný pre výpočty na grafoch sociálnych sietí [1], algoritmy pre riešenie grafovej súvislosti [2], algoritmy pre klastrovanie grafov [3], model overovania LTL logiky [4] a ako možnosť využitia sa objavuje aj oblasť biológie, napríklad pre výpočty na DNA databázach [5].

Podstatný koncept Pregelu je počítanie zamerané na vrcholy grafu a počítanie zložené zo superkrokov. Medzi hlavné výhody Pregelu patrí zníženie záťaže na sieťovú komunikáciu. Autori Pregelu v [6] ďalej dodávajú, že model založený na posielaní správ by mal postačovať pre všetky grafové algoritmy.

Orientované grafy sú modelované v pamäťových štruktúrach ako mapy [7] alebo ako rôzne optimalizácie záznamov vrcholov a hrán, ktoré podporujú rýchle prístupy. V niektorých implementáciách [7] je graf v pamäti uchovávaný v zozname susednosti, čo predstavuje zoznam výstupných hrán a vrcholov. Ak nie je poznačený aj zoznam vstupujúcich vrcholov, nie sme schopní navigovať sa po vstupujúcich hranách. Bez dodatočnej indexácie (informácie o hrane) alebo zmeny štruktúry grafu, nie je možná navigácia proti smeru hrany. Pre hľadanie vzťahov v sociálnych sieťach, sme toho názoru, že nezáleží na orientácii hrán (*Bob pozná Alicu* = jedna orientácia, *Alica pozná Boba* = opačná orientácia). V tejto práci navrhujeme nový škálovateľný algo-

ritmus Pregel Relationship Discovery (PRED) pre výpočtový model Pregel. Algoritmus je schopný objavovať vzťahy bez ohľadu na orientáciu hrán v grafe. Počas výpočtu a po jeho ukončení ostáva pôvodná grafová štruktúra nezmenená, nie je vytváraný žiadny dodatočný index vstupných alebo výstupných hrán.

2 Prehľad riešení

Pre hľadanie vzťahov vo veľkých grafoch, obsahujúcich milióny vrcholov a hrán, alebo grafoch tak veľkých, že nemôžu byť načítane do pamäte jedného stroja, je treba zvážiť škálovateľné riešenia a distribuované modely. V tejto časti porovnávame dva známe distribuované paradigmy, menovite MapReduce a Pregel.

MapReduce je koncept paralelného spracovania problémov nad veľkými dátami s použitím veľkého počtu strojov (uzly), spoločne nazvaných klaster (uzly v tej istej sieti) alebo grid (uzly mimo siete). MapReduce môže ťažiť z výhody lokálnosti dát, spracovaním týchto dát v alebo blízko ich úložiska, za účelom redukcie vzdialenosti, na ktorú musia byť transportované. MapReduce udržuje spolahlivosť vyčlenením niekoľkých operácií nad dátami pre každý uzol v sieti. Každý uzol má odpovedať pravidelne informáciu o ukončenej práci a zmene stavu.

Aj napriek popularite konceptu MapReduce pre paralelné počítanie, Pregel je vhodnejší pre krokové grafové počítanie [8, 1] a v odvolaní sa na [6], použitie MapReduce môže viesť k neoptimálnemu výkonu. Problémy s výkonom pri paralelnom spracovaní grafov sú preberané v [9]. Výpočtový model Pregel sa pokúša riešiť tieto problémy redukciou komunikačného zahltenia (ako možno badať pri MapReduce) s využitím výpočtov založených na sekvencii iterácií, tzv. superkrokov. Tento koncept je viac vhodný pre krokové počítanie a, tiež pre náš cieľ objavovania vzťahov, jeho forma vrcholovo-orientovaného počítania (vertex-centric paradigm). Bližší popis v [6].

3 Návrh riešenia

Pre objavovanie vzťahov navrhujeme riešenie založené na uzatvorených cestách. V základe, rozoznávame 2 typy vrcholov; *zaujímavé* a *obyčajné*. Počas objavovania vzťahov hľadáme vzťah medzi 2 zaujímavými vrcholmi a aspoň 1 obyčajným vrcholom.

Pre formálnu definíciu, majme graf $G = \{V,E\}$. Z množiny všetkých vrcholov V definujeme novú podmnožinu $S \subseteq V$, ktorá predstavuje množinu zaujímavých vrcholov. Potom množina všetkých obyčajných vrcholov bude definovaná vzťahom $O = V - S$, pričom platí $S \wedge O = \emptyset$. Nech je známa funkcia pre získanie všetkých príbuzných vrcholu h, $h \in V$, taká, že $N(h) = \{u | \{h, u\} \in E\}$. Pre získanie príbuzných ktorýchkoľvek vrcholu h, $h \in V$, patriacich do množiny S použijeme nasledovné pravidlá:

$$H = \{u | \exists v_1, v_2, \in S : u \in I \wedge u \in N(v_1) \wedge u \in N(v_2)\} \quad (1)$$

$$h \in H : M = \{u \neq h | u \in N(h) \wedge u \in S\} \quad (2)$$

Kde je:

H množina príbuzných,
S množina všetkých zaujímavých vrcholov,
O množina všetkých obyčajných vrcholov,
M množina zaujímavých príbuzných vrcholu *h*.

4 Algoritmus

Uvedené pravidlá implementujeme v našom algoritme PRED, ktorý je navrhnutý pre výpočtový model Pregel. K modelu Pregel patria synchronizované výpočtové iterácie založené na superkrokoch a výmena informácií prostredníctvom posielania správ.

Vrcholy v PRED môžu byť 2 typov; *zaujímavé* a *obyčajné*. Používateľ zadáva na vstup algoritmu graf *G* = {*V*,*E*} spolu s množinou *S*.

Jedným zo základných ukazovateľov v PRED je *maximálna dĺžka cesty*. Vrcholy posielajú správy pokiaľ sú aktívne, čo znamená pokiaľ dostávajú správy alebo pokiaľ vracajú hodnotu *true* z ich výpočtu *compute()*.

V správach, ktoré sa medzi vrcholmi šíria sú tieto informácie; *dĺžka prekonanej cesty*, *pôvodca správy*, *cieľový vrchol*, *posledný správu šíriaci vrchol* a *superkrok*.

V prvom superkroku, vrcholy z množiny S posielajú správy typu INTERESTING a v 2. superkroku vrcholy z množiny O posielajú PING.

Správy, ktoré vznikajú v tomto čase, majú prvý a posledný krát nastavené hodnoty *pôvodca správy*. Hodnota je nastavená na odosielajúci vrchol.

Od 2. superkroku vrcholy šíria zachytené správy ďalej, na svojich príbuzných pokiaľ nie je dosiahnutá stanovená dĺžka cesty. Pri každom prijatí správy je *dĺžka prekonanej cesty*, ktorá sa v správe uvádza, zvýšená o 1. Ak sa dosiahne stanovená hodnota, určená v *maximálna dĺžka cesty*, správa sa už ďalej nešíri, zaniká. Dĺžka cesty predstavuje určité obmedzenie na hĺbku prehľadávania.

Počas šírenia správy, v prípade typu PING a INTERESTING, odosielajúci vrchol nastavuje v správe hodnotu položke *posledný správu šíriaci vrchol*, ktorú nastavuje na seba.

Ak správu PING zachytí vrchol z množiny S, tento na správu odpovedá poslaním správy INTERESTING_REPLY priamo pôvodcovi správy PING, pričom hodnota v *dĺžka cesty* sa už ďalej v správe nezvyšuje.

Vrchol, ktorý patrí do množiny *O* a zachytí správu INTERESTING je aktivovaný (pridaný do množiny *S*) a *pôvodca správy* je poznačený do pamäte daného vrcholu pre neskoršiu identifikáciu vzťahu. Aktivovaný vrchol správu preposiela svojim príbuzným a následne už na ďalšie správy typu PING odpovedá správami s typom INTERESTING_REPLY.

Výstup z algoritmu tvorí zoznam aktivovaných vrcholov, pričom aby sa vrchol na tento zoznam dostal, musí byť aktivovaný minimálne 2x, zakaždým od iného vrcholu z množiny *S*.

Podmienkou pri posielaní správ, ktorá má zabrániť zacykleniu sa, je, okrem maxima pre dĺžku cesty, aby vrchol, ktorý správu zachytí, neposielal túto správu (typu PING alebo INTERESTING) spätne ani na pôvodcu ani na posledného šíriteľa.

Náš algoritmus je navrhnutý pre hľadanie vzťahov bez ohľadu na orientáciu hrán v grafe. Posielanie správ umožňuje tento návrh zrealizovať. Pre hľadanie vzťahov v orientovaných grafoch však, podľa Obr.1. musíme zaručiť, že vrcholy $a, b \in O$, budú aktivované pre všetky orientácie hrán (Obr.1a. až 1d.).

Pre aktiváciu vrcholov *a, b* v prípadoch grafov na Obr.1b. a 1d., nie je potrebná modifikácia algoritmu. V prípade grafu na Obr.1b. sú vrcholy a, b aktivované vrcholom *c*, ktorý posiela odpoveď INTERESTING_REPLY na správy PING, ktoré odoslali vrcholy *a, b*. V ďalšom prípade, Obr.1d. sú vrcholy *a, b* aktivované správou typu INTERESTING, ktorú poslal vrchol *c*. Vrchol b následne túto správu preposiela ešte vrcholu *a*.

Pre prípad, Obr.1c., opäť algoritmus nepotrebuje modifikáciu, aby vrcholy *a, b* boli aktivované. Vrcholy *a, b* budú aktivované nasledovne. Vrchol *b* bude aktivovaný prvý, pretože vrchol c posiela vrcholu *b* odpoveď na PING s typom správy INTERESTING_REPLY. Vrchol *b* je aktivovaný a správu INTERESTING_REPLY posiela ďalej svojmu príbuznému, vrcholu *a*, ktorý je následne aktivovaný, dĺžka cesty je 2.

Z praktického hľadiska, výpočet môže prebehnúť do 4 superkrokov (pre dĺžku 1; 1. superkrok = rozoslanie INTERESTING, 2. superkrok = odoslanie PING a zachytenie správ, 3. superkrok = zachytenie PING správ, 4. superkrok = koniec, dĺžka bola dosiahnutá). Naše experimenty však ukazujú, že, pre husté grafy (milióny vrcholov a hrán, priemerný stupeň vrcholu 1000), tento prípad generuje príliš mnoho správ v príliš krátkom čase, čo má za následok preplnenie pamäte a následný pád programu. Našim odporúčaním preto je, aby sa zaviedla kvóta na počet poslaných správ, aby sa neželanému pádu, plynúceho z preťaženia pamäte, predišlo.

Pre zaradenie vrcholu *v* do množiny aktivovaných vrcholov *S*, sú potrebné aspoň 2 správy INTERESTING, alebo INTERESTING_REPLY od 2 rôznych vrcholov. Nedokončené posielanie správ je realizované v nasledujúcom superkroku v bode 2a nášho algoritmu.

5 Experiment

V našom experimente je PRED implementovaný v jazyku Java rovnako, ako Sedge, základ výpočtového modelu Pregel.

Používame 6 strojov, 5 typu worker a 1 master v nasledovnej konfigurácii; každý stroj 21 CPU Intel Xeon, 2 GHz, 32 GB RAM and OS Ubuntu 12.04.5 LTS.

V našom experimente hľadáme vzťahy v RDF grafe databázy Freebase (https://www.freebase.com/), obsahujúcej 113×10^6 subjektov and $2{,}7 \times 10^8$ hrán. Pre náš cieľ sme vynechali subjekty, pozostávajúce výhradne z literálov.

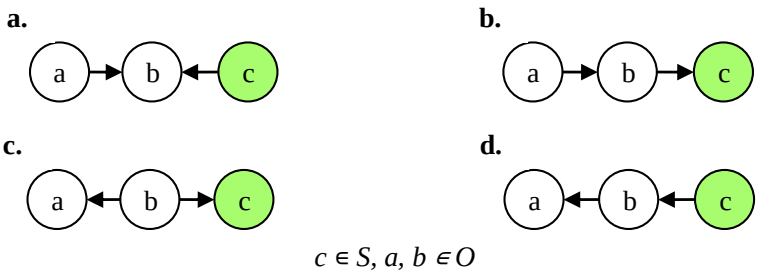
Množinu zaujímavých vrcholov tvoria známi herci alebo speváci, kompletný zoznam je v Tab.1. Aktivované vrcholy, ktoré boli pridané do množiny *S* sú v Tab.2.

Tab. 1. Zoznam zaujímavých vrcholov (množina S).

Freebase MID	Meno
m.0hqly	Steven Seagal
m.0451j	Jet Li
m.05v r84	Jackie Chan
m.0147dk	Will Smith
m.01qg7c	Barry Sonnenfeld

Tab. 2. Aktivované vrcholy, ktoré boli pridané do množiny S.

Freebase MID	Mená	Pridané vrcholy
m.0hqly m.0451j m.05v r84	Steven Seagal Jet Li Jackie Chan	Martial Artist
m.01qg7c m.0147dk	Barry Sonnenfeld Will Smith	Film Producer
m.0147dk m.0451j	Will Smith Jet Li	Chinese Martial Arts The Karate Kid



Obr. 1. Rôzne variácie grafov podľa orientácie hrán pre dĺžku cesty 2, pre ktoré algoritmus PRED funguje bez zmeny. Vrchol $c \in S$, $a, b \in O$.

6 Záver

V práci navrhujeme nový algoritmus PRED, pre implementáciu vo výpočtovom modeli Pregel. Algoritmus je schopný objavovať vzťahy (Tab.2.) v určenej množine obyčajných vrcholov, O, pričom hľadá prepojenia na vrcholy z množiny zaujímavých vrcholov, S (Tab.1.) a to bez ohľadu na orientáciu hrán v grafe. Algoritmus nepotrebuje pre svoje fungovanie dodatočnú indexáciu pre pokrytie obojsmernosti, štruktúra

grafu ostáva v pamäti bez zmeny. Pri jednom načítaní je teda možné realizovať viac rozličných výpočtov.

Poďakovanie. Táto práca je podporovaná projektmi TraDiCe APVV-0208-10 a VEGA 2/0185/13.

7 Referencie

1 Quick, L., Wilkinson, P., & Hardcastle, D. (2012, August). Using pregel-like large scale graph processing frameworks for social network analysis. In Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012) (pp. 457-463). IEEE Computer Society.

2 Yan, D., Cheng, J., Xing, K., Lu, Y., Ng, W., & Bu, Y. Pregel Algorithms for Graph Connectivity Problems with Performance Guarantees. Online: http://www.cse.cuhk.edu.hk/pregelplus/papers/ppa.pdf, 19.sept.2014

3 Perozzi, B., McCubbin, C., Beecher, S., & Halbert, J. T. (2013). Scalable Graph Clustering with Pregel. In Complex Networks IV (pp. 133-144). Springer Berlin Heidelberg.

4 Xie, M., Yang, Q., Zhai, J., & Wang, Q. (2014). A vertex centric parallel algorithm for linear temporal logic model checking in Pregel. Journal of Parallel and Distributed Computing.

5 Schatz, M. C., Langmead, B., & Salzberg, S. L. (2010). Cloud computing and the DNA data race. Nature biotechnology, 28(7), 691.

6 Malewicz, G., Austern, M. H., Bik, A. J., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010, June). Pregel: a system for large-scale graph processing. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (pp. 135-146). ACM.

7 Ciglan, M., & Nørvåg, K. (2010, January). Sgdb–simple graph database optimized for activation spreading computation. In Database Systems for Advanced Applications (pp. 45-56). Springer Berlin Heidelberg.

8 M. Dayarathna and T. Suzumura. A first view of exedra: a domain-specific language for large graph analytics workows. In WWW (Companion Volume), pages 509516, 2013.

9 Lumsdaine, A., et al.: Challenges in Parallel Graph Processing. Parallel Processing Letters, 17(1), pp. 5-20 (2007).

Perspektívy modelovania a predikovania veľkých dát v energetike

Gabriela Kosková, Anna Bou Ezzeddine, Mária Lucká, Viera Rozinajová, Peter Laurinec

Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií, Ilkovičova 2, 842 16 Bratislava
{gabriela.koskova, anna.bou.ezzeddine, maria.lucka, viera.rozinajova, peter.laurinec}@stuba.sk

Abstrakt. V článku predstavujeme problematiku modelovania a predikcie v energetike s akcentom na charakteristiky dát po implementovaní inteligentných meračov. Tieto dáta spĺňajú charakteristiky veľkých dát prúdového typu. Zamerali sme sa na inkrementálne modely, z ktorých algoritmy založené na podporných vektoroch sa javia ako vhodné metódy pri využití paradigmy distribuovaného spracovania.

Kľúčové slová: veľké dáta, energetika, inteligentné meranie, predikcie

1 Úvod

Predikovanie spotreby je extrémne dôležité pre dodávateľov energie a všetkých účastníkov výroby elektrickej energie, jej prenosu, distribúcie a trhu. Presné modely sú podstatné pre plánovanie a riadenie celej siete. Toto je dôležité najmä vzhľadom na veľmi obmedzené možnosti elektrinu skladovať. Klasickými prístupmi na modelovanie a predikciu odberov elektrickej energie je regresná analýza a modely na analýzu časových radov. Tieto prístupy však nebude možné využívať už v blízkom období, vzhľadom na snahu Európskej únie o zavedenie inteligentnej siete v rámci celej Európskej únie. Podľa vyhlášky MH SR č.358/2013 účinnej od 15. novembra 2013 má byť v Slovenskej republike inteligentnými meračmi spotreby elektrickej energie vybavených 80 % zo všetkých odberných miest, ktorých je rádovo dva milióny. Cieľom zavedenia inteligentných meračov a vytvorenia inteligentnej siete je efektívnym a ekonomickým spôsobom organizovať priamu nepretržitú interakciu a komunikáciu medzi spotrebiteľmi, ďalšími používateľmi sietí a dodávateľmi energie. Spotrebiteľom majú umožniť priamo kontrolovať a riadiť spotrebné návyky, motivovať ich k účinnej a výhodnej spotrebe vo väzbe na ceny elektriny v závislosti od doby spotreby. Vďaka cielenejšiemu riadeniu spotreby energie sa očakáva úspora nákladov na elektrickú energiu.

Inteligentné merače odosielajú informácie o odbere elektrickej energie v 15-minútových intervaloch do centrálneho informačného systému. Pri cieľovej

konfigurácii prenášané a spracovávané dáta spĺňajú charakteristiky veľkých dát, ako je objem a rýchlosť pribúdania dát. Ďalšou dôležitou charakteristikou týchto dát je to, že odbery pre jednotlivé odberné miesta, či distribučné skupiny je možné modelovať pomocou časových radov.

Výzvou témy veľkých dát vo všeobecnosti je spracovanie, analýza, predikcia a vizualizácia veľkých objemov dát v reálnom čase s cieľom podporiť ďalšie rozhodovanie.

Riešenie modelovania, analýzy a predikcie na veľkých energetických dátach si vyžaduje využiť moderné technológie pre distribuované a paralelné spracovanie dát, zvoliť vhodné reprezentácie a modely, ktoré je možné použiť na toky prichádzajúcich dát.

Tento príspevok je organizovaný nasledovne. Časť 2 sa venuje metódam na modelovanie a predikcie odberov elektrickej energie. V časti 3 predstavujeme pojem *veľké dáta* a ich charakteristiky. Časť 4 je venovaná výsledkom analýzy a naznačuje vhodné smery výskumu v oblasti predikcie v energetike pre veľké toky dát. Časť 5 poskytuje krátke zhrnutie.

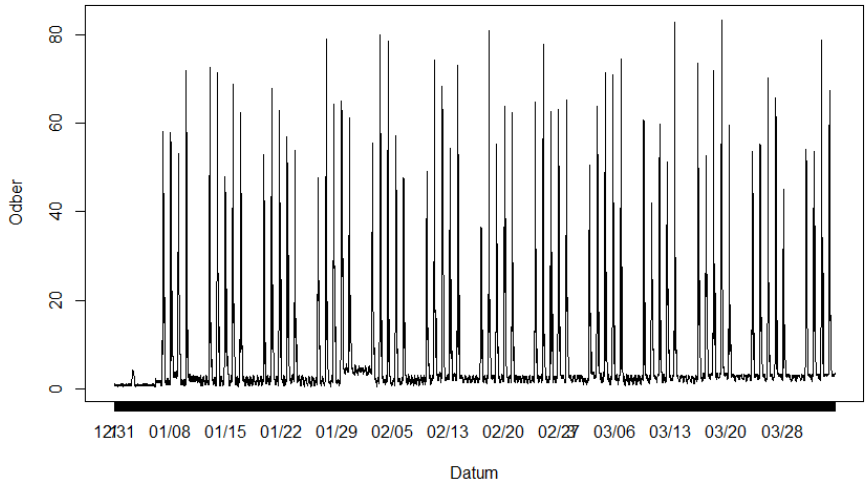


Fig. 1. Ukážka odberu elektrickej energie pre jedno odberné miesto za 3 mesiace.

2 Predikcia odberov elektrickej energie

Dáta z odberov elektrickej energie vykazujú silné sezónne zložky na úrovni hodiny dňa a dňa v týždni, čo je vidieť z dát na Fig. 1. Z priebehu je vidieť tiež vplyv sviatkov (začiatok roka).

Na predpovedanie spotreby sa dajú použiť klasické metódy, ako je regresia, viacnásobná regresia, exponenciálne vyhladzovanie, iteratívne reváhované techniky.

Ďalšími prístupmi sú inkrementálne modely využívajúce neurónové siete, algoritmy podporných vektorov a diskrétna waveletová transformácia.

2.1 Klasické prístupy

Jednou z najpoužívanejších metód na odhad vzťahov medzi premennými je *regresná analýza*. Ideou metódy je pomocou nezávislých premenných modelovať závislú premennú 6. V riešenom probléme je závislou premennou spotreba elektrickej energie, nezávislých premenných sa ponúka viacero a môžu byť v rôznych vzťahoch. Nezávislé premenné môžu byť kvantitatívneho (napr. nadmorská výška) alebo kvalitatívneho (napr. nejaká extrémna situácia, ako výpadok vody apod.) typu. Do modelu je možné zakomponovať aj časovú zložku v rôznych formách. V navrhnutom modeli je prvoradé určiť typ závislosti (napr. polynomiálny) a metódu odhadu parametrov (metóda najmenších štvorcov, metóda maximálnej vierohodnosti) 11.

Ďalšou možnosťou, ako predikovať spotrebu elektrickej energie, je *analýza časového radu*, napr. Box-Jenkinsova metodológia 15. Časový rad je tvorený hodnotami nejakej premennej meranej v čase. Jej ideou je modelovať správanie sa časového radu pomocou rezíduí (variability hodnôt). Je možné použiť niektorý zo štyroch základných modelov tejto metodológie: autoregresný model (AR), model kĺzavých priemerov (MA), zmiešaný model predchádzajúcich dvoch (ARMA) a autoregresný interaktívny model kĺzavých priemerov (ARIMA). Ktorý typ modelu je potrebné použiť závisí od charakteru časového radu.

2.2 Inkrementálne modely

Regresná analýza a analýza časových radov sa dá len ťažko použiť pre veľké objemy dát, ktoré pravidelne pribúdajú. Vhodnejšími sa javia adaptívne, alebo inak povedané inkrementálne metódy a modely. Takéto modely nie je potrebné pri každom novom meraní trénovať od začiatku so zväčšenou trénovacou množinou, ale je možné priebežne ich upravovať vo svetle nových dát. V literatúre je publikovaných niekoľko rôznych adaptívnych prístupov k predikcii elektrickej energie.

Veľkou množinou adaptívnych modelov sú umelé neurónové siete. Príkladom použitia neurónových sietí je práca 8, kde autori použili na predikciu odberov elektrickej energie neurónovú sieť s jednou skrytou vrstvou pozostávajúcou z 10 neurónov. Výsledky boli porovnané so SARIMAX metódou (rozšírenie ARIMA metódy). Neurónová sieť dosiahla mierne horšie výsledky, no bolo možné ju inkrementálne dotrénovať.

Prístupy založené na neurónových sieťach trpia nevýhodami ako je uviaznutie v lokánom minime a náchylnosť na preučenie 4.

Ďalším adaptívnym prístupom je *regresia založená na podporných vektoroch* (SVR – Support Vector Regression) 13. Jej cieľom je nájsť funkciu, ktorej odchýlka od všetkých meraní z trénovacej množiny je najviac ε a súčasne je táto funkcia čo najplochejšia. Pôvodná regresia založená na podporných vektoroch nie je navhrnutá ako inkrementálny model, ktorý by sa pri získaní nových dát dal aktualizovať, no

v literatúre bolo navrhnutých niekoľko obmien pôvodného prístupu s cieľom vytvoriť adaptívnu verziu SVR - AOSVR 9, SOG-SVR 3, 17 a 18.

V článku 19 autor používa SVM na krátkodovú predpoveď spotreby elektrickej energie. Tvrdí, že väčšina lineárnych modelov, ako napr. Kalmanov filter, AR a ARMA modely, nie sú vo všeobecnosti vhodné na modelovanie nelinearít spojených s predikciou spotreby. Použitie SVR (Support Vector Regression) na energetické dáta a s nimi spojené časové rady vyjadrujúce počasie prekonávajú iné.

Efektívnou metódou predikcie pre časové rady prúdových dát, ktoré vyžadujú rýchle spracovanie veľkého množstva dát, je použitie *waveletovej transformácie* a metódy Support Vector Machine (LS-SVM) založenej na metóde najmenších štvorcov. Táto metóda 5 vykazuje vyššiu presnosť ako iné porovnávané metódy a dá sa efektívne implementovať. Reprezentácia prúdových časových radov pomocou waveletovej transformácie bola využitá aj v ďalších publikáciách. V práci 1 predstavujú autori hybridný model aplikujúci Haarovu waveletovú transformáciu a dvojnásobné exponenciálne vyhládzanie na krátkodobé predpovedanie spotreby elektrickej energie.

Metóda AWSOM 10 je adaptívna metóda na spracovanie prúdových dát a na odhalenie vzorov v takýchto dátach. Využíva inkrementálnu diskrétnu waveletovú transformáciu s využitím Haarových waveletov. Využitie waveletovej transformácie umožní odstrániť nadbytočný šum a sústrediť sa tak na podstatné zložky časového radu. V porovnaní s ARIMA modelom dokázal AWSOM lepšie zachytiť dlhodobé správanie sa časového radu, navyše má výhody inkrementálneho modelu.

3 Veľké dáta

Kontinuálny nárast výpočtového výkonu v posledných desaťročiach umožnil produkovať ohromné toky dát, čo bolo príčinou zmeny paradigmy výpočtovej architektúry a mechanizmu spracovania veľkých objemov dát. Pojem *veľké dáta* (z angl. big data) označuje veľké množiny dát, ktoré kvôli ich veľkosti nie je možné ukladať, skladovať, riadiť a analyzovať tradičnými databázovými technológiami 14. Laney Douglas 7 definoval veľké dáta pomocou štyroch dimenzií: *veľkosť* (volume), *rýchlosť* (velocity), *rozmanitosť* (variety) a *variabilita* (variability). Pod veľkosťou dát rozumieme fyzické nároky na uskladnenie dát v pamäti (terabajty, petabajty). Dimenzia rýchlosť naznačuje nutnosť paralelného a distribuovaného spracovania dát, keďže časový faktor je často pri spracovaní veľkých dát kľúčový. Dôležitou špecifikáciou je rozmanitosť dát. Spracovávané sú rôzne typy dát: štrukturované, neštrukturované, text, audio, video. Dimenzia variabilita definuje možnosti interpretovania veľkých dát v čase.

Vzhľadom na potrebu spracovávať veľké dáta, vznikla potreba hľadať riešenia s cieľom zvýšiť výpočtový výkon pri spracovaní veľkých objemov dát pri čo najnižších nákladoch. Riešením tohto problému bol vývoj distribuovaných systémov. Jednou z najznámejších techník distribuovaných systémov je model MapReduce 2. Jeho voľne dostupnou implementáciou je Hadoop 12, 16, ktorý poskytuje súborový

systém a framework určený pre analýzu a transformáciu veľkých dátových množín s použitím MapReduce paradigmy.

4 Predikcie vo veľkých dátach

V oblasti energetiky vzhľadom na charakter dát sú typickými úlohami: predikcia spotreby elektrickej energie, analýza vzťahu spotreby elektrickej energie a externých faktorov, napr. meteorologických či demografických údajov, dolovanie typických motívov a tvarov kriviek odberu elektrickej energie, klastrovanie odberných miest podľa charakteristík odberu, klasifikácia odberateľov elektrickej energie a detekcia anomálií.

Z analýzy stavu poznatkov v tejto oblasti vyplýva, že klasické, neinkrementálne modely nie je možné využívať na modelovanie veľkých dát prúdového typu.

Na predikciu odberov elektrickej energie statických vzoriek sa úspešne využívajú metódy založené na podporných vektoroch. Výhodou týchto metód je to, že rozhodnutia (klasifikácia alebo predikcia) sú vytvárané len na základe vzdialenosti predikovanej inštancie od podporných vektorov, čo je len podmnožina vybraných dát z trénovacej množiny. Tým sa výrazne znižuje časová zložitosť. Prirodzeným posunom od statických množín k prúdovým dátam by boli inkrementálne metódy založené na podporných vektoroch. Pri zložitých úlohách na dosiahnutie výsledkov s vysokou presnosťou však môže byť potrebné veľké množstvo podporných vektorov. V takom prípade sa výhoda nízkej časovej zložitosti stráca. Problémom pri modelovaní priebehov odberov elektrickej energie s výraznými sezónnymi vplyvmi môže byť práve potreba modelovať dáta pomocou veľkého počtu podporných vektorov, čo je cieľom ďalšieho skúmania.

5 Záver

V príspevku sme analyzovali problematiku modelovania a predikovania odberov elektrickej energie s perspektívou potreby modelovania veľkých dát. Vo viacerých prístupoch k predikcii časových radov a odberov elektrickej energie bola na odstránenie šumu použitá waveletová transformácia.

Identifikovali sme potrebu sústrediť sa na inkrementálne modely, kde pre neustále pribúdajúce dáta nie je potrebné model opätovne trénovať, ale len aktualizovať vo svetle nových dát. Z literatúry sa vhodnými javia inkrementálne verzie algoritmov založené na podporných vektoroch. Úskalím pri modelovaní zložitejších dát však môže byť potreba veľkého počtu podporných vektorov, čo môže znemožniť rýchle vytváranie predikcií.

Poďakovanie. Táto publikácia vznikla vďaka podpore projektu v rámci OP Výskum a vývoj pre projekt: „Medzinárodné centrum excelentnosti pre výskum inteligentných a bezpečných informačno-komunikačných technológií a systémov“, ITMS: 2624012003, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

Referencie

1. Annamareddi, S., Gopinathan, S., Dora, B.: A simple Hybrid Model for Short-Term Load Forecasting. Hindawi Publishing Corporation, Journal od Engineering, Volume 2013, Article ID 760860 (2013).
2. Dean, J., Ghemawat, S.: MapReduce: Simplifed Data Processing on Large Clusters. USENIX Association OSDI '04, 6th Symposium on Operating Systems Design and Implementation (2004).
3. Engel, Y., Mannor, S., Meir, R.: Sparce online greedy support vector regression, 13th European Conference on Machine Learning (2002).
4. Guo, Y., Niu, D., Chen, Y.: Support Vector Machine Model in Electricity Load Forecasting, Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 (2006).
5. Kong, Z., Shi, Z., Zuan, J.: Prediction Method of Time Series Data Stream Based on Wavelet Transform and Least Square Support Vector Machine, IEEE, DOI 10.109-ICNC.2008.255 (2008).
6. Lamoš, F., Potocký, R.: Pravdepodobnosť a matematická štatistika. Bratislava, UK (1998).
7. Laney, Douglas: The Importance of Big Data: A Definition, Gartner, Retrieved 21 June 2012. http://www.gartner.com/document/2057415
8. Liu, N., Babushkin, V., Afshari, A.: Short-Term Forecasting of Temperature Driven Electricity Load Using Time Series and Neural Network Model, Journal of Clean Energy Technologies 2 (4) (2014).
9. Ma, J., Theliler, J., Perkins, S.: Accurate On-line Support Vector Regression, Neural Computation., 15, pp. 2683–2703, (2003).
10. Papadimitriou, S., Brockwell, A., Faloutsos, C.: AWSOM: Adaptive, Hands-Off Stream Mining, VLDB (2003).
11. Pázman, A., Lacko, V.: Prednášky z regresných modelov. Bratislava, UK (2012).
12. Schvachko, K., Kuang, H., Radia, S., Chansler, R.: The Hadoop Distributed File System. In Proceedings of IEEE 26th symposium on Mass Storage Systems and Technologies (2010).
13. Smola, A., J., Schölkopf, B.: A tutoral on support vector regression. Journal Statistics and Computing Volume 14 Issue 3, (2004).
14. Snijders, C., Matzat, U., Reips, U.-D.: 'Big Data': Big gaps of knowledge in the field of Internet. International Journal of Internet Science, 7, 1-5. (2012). http://www.ijis.net/ijis7_1/ijis7_1_editorial.html
15. Štulajter ,F.: Predictions in Time Series Using Regression Models, Springer (2002).
16. Venner, J.: Pro Hadoop, Apress, 2009, ISBN13: 978-1-4302-1942-2 (2009).
17. Wu, Q., Liu, W., Yang, Y., Time series online prediction algorithm based on least squares support vector machine. Central South University of Technology, Vol. 14, pp.442-446 (2007).
18. Zhang, H., Wang, X.. Incremental and Online Learning Algorithm for Regression Least Squares Support Vector Machine. Chinese Journal of Computers, 29(3): 400-406 (2006).
19. Zhang, M.-G.:. Short-term load forecasting based on support vector machine regression. Proc. 4th Int. Conf. on Machine Learning and Cybernetics, vol. 7, pp.4310 -4314 (2005).

Budovanie slovenskej bázy poznatkov s využitím prepojených dát

Michal Holub, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova, 842 16 Bratislava 4, Slovensko
{michal.holub, maria.bielikova}@stuba.sk

Abstrakt. V tomto príspevku sa venujeme vízii vybudovania strojovo spracovateľnej bázy poznatkov extrahovaných špeciálne zo slovenských webových zdrojov. Takisto prezentujeme prvotné pokusy s jej realizovaním za účelom využitia pri úlohách personalizácie webu. Existujúce bázy štruktúrovaných dát, akými sú napr. DBpedia alebo YAGO, tvoriace jadro oblaku prepojených dát (angl. Linked Data Cloud), sa zameriavajú na extrakciu údajov zo zdrojov v anglickom jazyku. Obsahujú veľké množstvo entít a faktov o nich, no sú orientované globálne. My diskutujeme zámer a víziu vybudovania slovenskej verzie obdobnej bázy poznatkov, ktorá sa špecificky zameriava na slovenské webové zdroje. Jej cieľom je byť doplnkom ku globálnym bázam a slúžiť špecifickým potrebám metód personalizácie v kontexte slovenského webu.

Kľúčové slová: sémantický web, prepojené dáta, extrakcia entít, DBpedia

1 Úvod

Na webe môžeme nájsť množstvo poznatkov z rôznych oblastí ľudského života, ktoré by sa dali použiť v rôznych inteligentných aplikáciách. Mnohokrát však informáciám chýba štruktúra, označenie významu, sú publikované vo forme zrozumiteľnej ľuďom. Cieľom sémantického webu je transformovať sieť dokumentov na sieť informácií tak, aby tieto boli štruktúrované a zrozumiteľné aj strojom [6].

K naplneniu vízie sémantického webu prispieva iniciatíva prepojených dát (angl. Linked Data) na webe [2]. Vzniká veľa datasetov obsahujúcich fakty o entitách z rôznych domén v strojovo spracovateľnej podobe. Tieto datasety využívajú spoločné kontrolované slovníky, ontológie a identifikátory pre opis rovnakých entít s použitím rámca RDF.

Každý takýto fakt má podobu trojice subjekt – vzťah – predmet. Vzťahy môžu prepájať aj entity naprieč datasetmi, čím vzniká oblak vzájomne prepojených dát, ktorý môžeme reprezentovať grafom. V jadre tohto oblaku sú datasety, ktoré definujú najznámejšie entity reálneho sveta a základné informácie o nich (geografické pojmy, filmy, herci, hudobníci, atď.). Sú nimi datasety DBpedia [1] a YAGO [4]. Na tieto

datasety nadväzujú ďalšie, ktoré sa špecializujú na konkrétnu oblasť (napr. medicína, médiá, knižnice) alebo obsahujú doplňujúce informácie k už definovaným entitám.

Oba spomenuté datasety vznikli v rámci výskumu metód na automatickú transformáciu polo-štruktúrovaných, používateľmi vytvorených dát na štruktúrované, strojovo spracovateľné informácie. Hlavným zdrojom dát pre oba datasety je webová encyklopédia Wikipedia[1].

Štruktúrované dáta z DBpedia alebo YAGO sa dajú využiť pri rôznych úlohách personalizovaného webu. Pri odporúčaní dokumentov na základe obsahu vieme tieto obohatiť o dodatočné informácie o entitách, ktoré sú v dokumentoch zmienené [7]. Dodatočné informácie vieme tiež využiť pri rozlišovaní entít (angl. entity disambiguation) v rámci extrakcie informácií [3].

Štruktúru grafu môžeme využiť na prieskumné vyhľadávanie v neznámej doméne vďaka prítomnosti prepojení medzi entitami. Analýzou týchto prepojení sa môžeme dostať k súvisiacim entitám, o ktoré sa môže používateľ zaujímať, pričom samotným porovnaním ich textovej reprezentácie by sme takúto podobnosť nemuseli objaviť [5].

Grafová štruktúra je tiež vhodná pre realizáciu zložitejších dopytov, ktoré by sme bez prítomnosti prepojení nemohli realizovať. Ak chceme napr. vedieť, ktorí fyzici sa narodili v rovnakom meste ako Albert Einstein, je to s využitím jazyka SPARQL a dát z DBpedie jednoduchá úloha. Avšak tradičným spôsobom vyhľadávania založeným na kľúčových slovách by sme sa k výsledku len ťažko dopátrali.

V oblasti vyhľadávania informácií môžeme zlepšiť výsledky, keď použijeme dodatočnú znalosť o vyhľadávaných entitách z týchto datasetov. Dodatočné informácie môžu tiež pomôcť spresniť model používateľa, príp. model domény v adaptívnom webovom systéme.

Keďže chceme vyššie uvedené funkcie a metódy realizovať aj v rámci slovenského webu, musíme brať do úvahy odlišný jazyk. Samotné metódy odporúčania, vyhľadávania, podpory navigácie alebo zodpovedania dopytov sú často jazykovo nezávislé. Avšak dáta, ktoré používajú, musia odrážať odlišný kontext, t.j. v našom prípade obsahovať informácie o slovenských entitách.

Tu narážame na prvý problém. Anglická Wikipedia obsahuje prevažne entity globálneho významu, ktoré sa tak dostanú aj do oblaku prepojených dát. Avšak z globálneho pohľadu menej významná slovenská entita (napr. film, spevák, športovec) v ňom bude chýbať, príp. nemusí o nej byť dostupných toľko údajov ako v slovenských zdrojoch. Ak však chceme napr. odporúčať články z lokálnych novín, informácie o týchto entitách potrebujeme.

Naším zámerom je preto vybudovať slovenskú verziu bázy poznatkov, akými sú napr. DBpedia a YAGO, ktorá bude obsahovať štruktúrované informácie o entitách reálneho sveta s dôrazom na kontext slovenského webu. Táto báza poznatkov má byť doplnkom uvedených datasetov. Naším cieľom je, aby bola dostupná pre výskumníkov realizujúcich metódy na personalizáciu a adaptáciu webu a webových systémov, ktorí sa špecificky zameriavajú na slovenský obsah.

[1] http://en.wikipedia.org

2 Extrakcia štruktúrovaných informácií z webu

DBpedia aj YAGO používajú ako základný zdroj informácií Wikipediu. Nejedná sa o úplne neštruktúrovaný text, články vo Wikipedii majú určitú šablónu – každý má svoj nadpis, ktorý zodpovedá opisovanej entite, hlavné informácie sú zobrazené v tabuľkách, každý článok je zaradený do kategórií, články z rovnakej kategórie obsahujú rovnaký súbor základných informácií v podobe dvojíc kľúč-hodnota (tzv. infobox).

Z jedného článku sa spravidla extrahuje jedna entita, pričom sa postupuje podľa týchto základných krokov:

- nadpis článku je použitý ako názov extrahovanej entity
- infobox je pomocou sady manuálne definovaných pravidiel transformovaný na atribúty entity a ich hodnoty
- typ entity sa určí podľa kategórií, do ktorých je článok zaradený

Hierarchia kategórií vo Wikipedii slúži na získanie typov entít (tried), ktoré sú prepojené vzťahmi podtrieda-nadtrieda. V tomto kroku sa názvy tried prevádzajú do základného tvaru. YAGO navyše mapuje triedy na koncepty z databázy WordNet[2], čím získava vzťahy súvislosti medzi nimi.

Pri budovaní slovenskej verzie datasetu navrhujeme podobný postup. Jazykovo špecifickým krokom bude prevedenie názvov kategórií na základný tvar, aby sme ich mohli použiť ako typy entít.

Navyše však navrhujeme rozšírenie spracovanie informácií o kategóriách. Vieme z nich totiž získať aj hodnoty niektorých atribútov. Napr. článok s nadpisom *Arthur Schopenhauer* je zaradený v kategórii *Osobnosti z Gdanska*. Pomocou pravidlového systému vieme v tomto prípade z názvu kategórie odvodiť nie len typ entity (osobnosti), ale aj geografickú entitu viažucu sa k pôvodnej entite (Gdansk).

Ďalším problémom datasetov YAGO a DBpedia, ako aj ďalších datasetov v rámci oblaku prepojených dát, je presnosť faktov a dôveryhodnosť informácií. Keďže tieto sú vytvárané automatizovane, navyše zo vstupných dát, ktoré tvorí dav používateľov na webe, obsahujú aj chyby a nepresné informácie. I keď si myslíme, že všetky chyby sa určite eliminovať nedajú, chceme tomuto problému čeliť dvomi krokmi:

1. využitím kontrolovaných slovníkov ako prvotného zdroja dát
2. zavedením skóre pre každý fakt a ich overovaním z viacerých zdrojov

Prvý krok má za cieľ naplnenie bázy poznatkov potenciálne kvalitnými údajmi z kontrolovaných slovníkov, akými sú slovníky používané v múzeách, knižniciach, a iných inštitúciách, ktoré sú voľne dostupné. Navrhujeme najprv získať zoznamy rôznych typov entít z overených zdrojov, až následne ich dopĺňať o atribúty a ich hodnoty extrahované zo slovenskej Wikipedie. Tento krok má zaručiť menej chýb v názvoch entít a ich typoch. Na druhej strane potenciálnym problémom môže byť

[2] http://wordnet.princeton.edu

hľadanie takto získaných entít vo Wikipedii, kedy porovnanie s nadpisom článku nemusí stačiť na úplné rozlíšenie významu.

V druhom kroku navrhujeme zaviesť skóre dôveryhodnosti pre každý fakt, ktorý extrahujeme. Skóre sa bude počítať na základe počtu výskytov inštancií tohto faktu v zdrojových údajoch, ktoré budeme automatizovane vyhľadávať. Takisto budeme hľadať výskyty faktov s vymeneným subjektom a predmetom, a to tak, že ku každému vzťahu, ktorý bude v datasete prípustný, manuálne definujeme jeho symetrickú verziu (ak napr. extrahujeme fakt *Bratislava je hlavné mesto Slovenska*, budeme hľadať aj fakt *Slovensko má hlavné mesto Bratislavu*). Naopak, ak nájdeme protirečiaci fakt s rovnakým subjektom, ale s iným objektom (napr. *Martin je hlavné mesto Slovenska*), skóre faktu znížime. Na základe skóre viacerých faktov môžeme ohodnotiť aj celý zdroj dát, z ktorého tieto fakty extrahujeme, a toto skóre použiť na ohodnotenie novo extrahovaných faktov.

Predpokladáme viaceré spôsoby sprístupnenia bázy poznatkov a informácií v nej obsiahnutých:

1. Cez webové rozhranie – tento spôsob bude určený pre používateľov, ktorí si budú chcieť pomocou webového prehliadača prezrieť základné údaje o vybranej entite.
2. Pomocou webovej služby – týmto spôsobom bude môcť používateľ získať štruktúrovaný záznam s údajmi o vybranej entite, ktorú špecifikuje pomocou URI ako vstupný parameter volanej služby.
3. Pomocou SPARQL koncového bodu – tento spôsob bude umožňovať vyhľadávanie v báze poznatkov pomocou jazyka SPARQL, čo je štandard v tejto oblasti.

Predpokladáme, že postupne budú vnikať ďalšie služby na sprístupňovanie informácií z bázy poznatkov, napr. nadstavba umožňujúca dopytovanie pomocou prirodzeného jazyka (dopyt bude následne prevedený na SPARQL výraz) alebo fazetový prehliadač údajov, príp. nástroj umožňujúci údaje prehliadať exploratívne.

3 Zhrnutie

V tomto príspevku sme predstavili našu víziu vytvorenia slovenskej verzie štruktúrovanej bázy poznatkov, ktorá by dopĺňala podobné datasety existujúce v rámci oblaku prepojených dát na webe. Takáto báza poznatkov bude obsahovať údaje extrahované zo slovenských webových zdrojov, pričom bude obsahovať údaje o entitách reálneho sveta významných v slovenskom kontexte.

Myslíme si, že takáto báza poznatkov je potrebné a má široké využitie v rámci výskumu personalizácie a adaptácie webu, ako aj v rámci vývoja inteligentných webových aplikácií. Bude sa dať využiť ako zdroj dodatočných údajov pri vyhľadávaní a odporúčaní informácií, pri podpore v navigácii a prieskumnom prehľadávaní neznámej domény, ako aj pri zodpovedaní na dopyty zadané používateľmi. Takisto bude použiteľná pre metódy a algoritmy na extrakciu a rozlišovanie pomenovaných entít, extrakciu kľúčových slov z textu, spracovanie slovenského textu, a iné.

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

Literatúra

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. The Semantic Web, LNCS 4825, pp. 722-735. Springer Berlin Heidelberg (2007)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. Int. Journal on Semantic Web and Inf. Systems, Vol. 5, No. 3, pp. 1-22 (2009)
3. Hakimov, S., Oto, S.A., Dogdu, E.: Named Entity Recognition and Disambiguation Using Linked Data and Graph-based Centrality Scoring. SWIM '12: Proc. of 4th Int. Workshop on Semantic Web Information Management, Article No. 4. ACM Press, New York (2012)
4. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. Artificial Intelligence, Vol. 194, pp. 28-61 (2013)
5. Marie, N., Gandon, F., Giboin, A., Palagi, É.: Exploratory Search on Topics Through Different Perspectives with DBpedia. SEM '14: Proc. of 10th Int. Conf. on Semantic Systems, pp. 45-52. ACM Press, New York (2014)
6. Shadbolt, N., Hall, W., Berners-Lee, T.: The Semantic Web Revisited. Intelligent Systems, Vol. 21, No. 3, pp. 96-101 (2006)
7. Stankovic, M., Breitfuss, W., Laublet, P.: Discovering Relevant Topics Using DBPedia: Providing Non-obvious Recommendations. WI-IAT '11: Proc. of 2011 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology, pp. 219-222. IEEE Computer Society, Washington, DC, USA (2011)

Personalizovaná správa multimédií

Michal Kompan, Jakub Šimko, Ondrej Kaššák, Mária Bieliková

Ústav informatiky a softvérového inžinierstva,
Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave
[meno].[priezvisko]@stuba.sk

Abstrakt. Idea odporúčaním personalizovať ponuku TV divákovi, je motivovaná množstvom obsahu a čoraz väčším podielom nelineárnej TV na trhu. Klasická TV sa približuje k stavu, charakteristickému skôr pre webové portály s multimediálnym obsahom (napr. služba Youtube). Tieto portály už odporúčanie realizujú. Na to však využívajú špecifické spôsoby zberu informácií o používateľoch, pričom tieto nie sú priamočiaro aplikovateľné na prípad televízneho diváka. V našej práci, nadväzujúc na víziu inteligentného prehliadača ponuky TV obsahu, sa sústreďujeme na spôsoby zberu informácií o (tradičnom) TV divákovi, jeho modelovania a odporúčania obsahu.

Kľúčové slová: sociálny web, personalizované odporúčanie, obohacovanie metadát

1 Úvod

V tejto práci je našim cieľom prispieť k zníženiu problému zahltenia konzumenta – diváka TV informáciami a nerelevantným obsahom [2]. V tomto článku sa zameriavame na odporúčanie, ako jeden z prostriedkov dosiahnutia tohto cieľa. Našim ohraničením je doména televízneho multimediálneho obsahu, konzumovaného predovšetkým prostredníctvom tradičných TV prijímačov. Problém veľkého množstva obsahu, z ktorého len malá časť je pre konkrétneho diváka relevantná je tu rovnaký, ako pri „novodobých“ webových portáloch poskytujúcich multimediálny obsah. Bez riadnych nástrojov vyhľadávania, abstraktných vizualizácií, prehľadávania a odporúčania nie je možné napĺňanie potrieb divákov v ani jednom z týchto prostredí. Avšak, zatiaľ čo vo webovom prostredí multimédií je do praxe prakticky aplikovaný stav poznania (vyhľadávania, vizualizácií, atď.), divák tradičnej TV sa pri vyhľadávaní vhodného obsahu stále musí spoliehať na sekvenčné prehľadávanie ponuky statických TV programov, čo je pri stále stúpajúcej ponuke TV staníc a pribúdajúcich možnostiach využívať archívy televízií neefektívne.

Uplatneniu postupov z webového prostredia (ktorých účinnosť ukázalo mnoho štúdií [3]), bráni odlišnosť tradičnej TV (od Webu). Rozdiely je badať napríklad v *odlišnom spôsobe navigácie* v samotnom obsahu („prepínanie kanálov“) a jeho oddelení od vyhľadávacích mechanizmov pracujúcich s metadátami (napr. tlačený TV program, statický TV program na webe). Prepájanie týchto dvoch priestorov musí

divák vykonávať manuálne, čo znižuje efektívnosť takéhoto používania. Čo sa týka odporúčania, jeho uplatnenie komplikujú najmä:

- *Odlišné možnosti získavania implicitnej spätnej väzby* od divákov (potrebnej na budovanie modelov záujmov divákov). Namiesto vyhľadávacích dopytov na webe, nesúcich aspoň čiastočne explicitne vyjadrený záujem o určité témy, máme pri tradičnej TV často iba záznamy o prepínaní kanálov „očistené“ od motivácií ktoré sa za nimi skrývali.
- Pri odporúčaní TV obsahu je potrebné brať do úvahy *časové aspekty* (časy vysielania programov).
- *Nedostatok metadát*. Televízny obsah je vysielaný s minimom obsahových metadát, podobne aj komerčne distribuované TV programy. Absentuje napríklad prístup ku kolaboratívne a sociálne tvoreným bázam metadát, známych z webových portálov.

Naša predchádzajúca práca sa zaoberala návrhom inteligentného používateľského rozhrania pre prieskumné vyhľadávanie artefaktov s časovou platnosťou, použiteľnom pre TV programy [4]. V tomto článku na ňu nadväzujeme rozpracovaním princípov odporúčania a zberu metadát, ktoré tvoria pilier pre akúkoľvek prezentačnú vrstvu. Zameriavame sa pritom práve na prekonávanie vyššie uvedených špecifík tradičnej TV.

2 Obohacovanie metadát

Kvalita dát je z pohľadu odporúčacieho systému a samotných odporúčaných prvkov kľúčová. Bez ohľadu na typ odporúčača, ktorý je využitý – obsahový alebo kolaboratívny [5] – kvalita metadát opisujúcich samotné multimediálne prvky významne ovplyvňuje kvalitu samotného odporúčania a modelov používateľov, ktoré vychádzajú z vlastností odporúčaných prvkov.

V doméne inteligentných televízií ale najmä lineárneho obsahu sa často stretávame s problémom kedy sú dostupné opisné dáta v nedostatočnej kvalite. V takomto prípade je nevyhnutné tieto metadáta obohatiť a zabezpečiť tak dostatočnú rozlišovaciu informáciu využiteľnú rôznymi odporúčacími stratégiami.

Základným problémom pri snahe obohatiť dáta dostupné z televízneho programu, je nevyhnutné jednoznačne identifikovať program, nakoľko neexistuje jednoznačný identifikátor v TV programe, pod ktorým by ho bolo možné vyhľadávať naprieč filmovými databázami. Možnosťou ako takýto identifikátor získať, je prepojenie televízneho programu na niektorú z existujúcich databáz (v našom prípade DBpedia[1]).

Navrhnutý postup obohatenia metadát môžeme zhrnúť do nasledujúcich krokov:

1. Získanie originálneho anglického názvu obohacovaného programu

[1] http://www.dbpedia.org/

2. Vyhľadanie programu (pomocou anglického názvu) v niektorej z filmových databáz
3. Priradenie jednoznačného identifikátora danému programu

Vzhľadom na špecifickosť slovenského jazyka je nevyhnutné v prvom kroku získať originálny anglický názov daného programu. Pre tento účel využívame lokalizovanú databázu CSFD[2]. Tu sa snažíme na základe kombinácie lokálneho názvu a niektorého z ďalších – prevažne jazykovo nezávislých údajov zväčša dostupných v TV programe – herci, rok vydania a pod.

Po úspešnom získaní pôvodného názvu daného programu, sa pomocou neho resp. kombinácie s ďalšími informáciami ako rok vydania a pod. môžeme dopytovať do niektorej zo svetových filmových databáz (DBpedia). Takýmto spôsobom vieme s vysokou pravdepodobnosťou (cca. 85% úspech – nerozlišujeme prípady kedy sme identifikovali program zle, alebo sa v danej databáze vôbec nenachádzal) získať univerzálny identifikátor a teda následne obohatiť dostupné metadáta širokou škálou dostupných metadát.

3 Personalizované návrhy

Pri personalizovanom odporúčaní sa všeobecne stretávame s dvoma základnými problémami na ktoré sa snažíme reflektovať – problém informačného zahltenia a problém neviditeľnosti informačného priestoru [1]. Typickým príkladom pre problém informačného zahltenia je doména spravodajstva, kedy za pomerne krátky čas (v kontexte možností jedného používateľa) vzniká obrovské množstvo nových správ, ktoré je v konečnom dôsledku nutné používateľovi filtrovať (resp. koná tak sám). Na strane druhej máme problém neviditeľnosti informačného priestoru, kde je typický príklad práve doména multimédií kedy používatelia často nevedia o existencii programu, ktorý by im za daných okolností vyhovoval.

Pre potreby preskúmania možností generovania personalizovaných návrhov (či už vo forme odporúčania alebo personalizovaného vyhľadávania) sme implementovali prototyp kolaboratívneho odporúčača.

Prakticky každá metóda personalizovaného odporúčania vychádza z potreby hľadania podobnosti – či už podobnosti medzi záujmami rôznych používateľov, prípadne podobnosti medzi samotnými odporúčanými prvkami. Zamerali sme sa preto na rôzne možnosti zhlukovania resp. hľadania podobných používateľov (predpokladáme že zhluk obsahuje n najpodobnejších používateľov).

Každý používateľ bol v našom experimente reprezentovaný vektorom pozostávajúcim z údajov o jeho aktivite (v závislosti od času kedy bola táto vykonaná). Úlohou bolo teda nájsť takých používateľov ktorí sa správajú v podobnom čase približne rovnako – preferujú podobné programy.

Vzhľadom na vysokú dimenzionalitu takýchto vektorov, bolo nevyhnutné počet dimenzií redukovať, na čo sme využili štandardný prístup SVD (angl. Single Value

[2] http://www.csfd.cz

Decomposition). Redukciou sme sa snažili urýchliť výpočtový proces a zároveň zachovať dôležitú rozlišovaciu informáciu. Ako samotný algoritmus zhlukovania sme využili K-means, ktorý rozdelí vstupnú množinu do k zhlukov na základe zvolenej stratégie hľadania podobnosti. Následne sme takéto zhluky využili pri generovaní kolaboratívneho top-N odporúčania.

Dosiahnuté výsledky (presnosť nad 85%) potvrdili našu hypotézu. Používatelia v našom (syntetickom) experimente preferovali len malý okruh prvkov dostupných v danej doméne. Čelili teda problému neviditeľnosti informačného priestoru a teda bolo veľmi jednoduché dosiahnuť vysokú presnosť – odporúčanie najobľúbenejších prvkov v celej doméne. To nám naznačuje, že je nevyhnuté experimentovať na vzorke reálnych používateľov a neskúmať len kvantitatívne ale aj kvalitatívne aspekty navrhnutých metód.

4 Záver

V doteraz vykonaných experimentoch sme ukázali, že je možné prostredníctvom lokalizovanej databázy a podobnosti jestvujúcich metadát prepojiť podstatnú časť TV obsahu na externé zdroje metadát. Ďalej sme ukázali, že kolaboratívnym odporúčaním so zahrnutím podobnosti používateľov, počítanej priamo z vektorov ich aktivít možno dosiahnuť pomerne dobré základné odporúčanie.

V našej ďalšej práci sa budeme zaoberať živým overením takéhoto odporúčania jeho integrovaním do aplikácie prieskumného vyhľadávania v TV programoch [4]. Zároveň bude našou snahou spresniť toto vyhľadávanie práve dodatočnou spätnou väzbou, ktorej zdrojom bude používateľská aplikácia.

Poďakovanie. Táto práca vznikla vďaka podpore projektu Kontextové vyhľadávanie a prehliadanie informácií v sociálnom prostredí webu VG1/0675/11.

Literatúra

1. Adomavicius, G., Tuzhilin, A.: 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,* vol.17, no.6, 734,749, (2005).
2. Marchionini, G.: Exploratory search: from finding to understanding. Commun. ACM 49, 4, 41-46, (2006).
3. Song Q., Menezes, R., Silaghi, M.: A Recommender System for Youtube Based on its Network of Reviewers, In: *Second International Conference on Social Computing (SocialCom)*, IEEE, 323-328, (2010).
4. Šimko, J., Kompan, M. Zeleník, D., Bieliková, M.: Nástroj pre sociálne fazetové prehľadávanie multimédií. In: WIKT 2013 : 8th Workshop on Intelligent and Knowledge oriented Technologies. Centre for Information Technologies, Košice, 7-11, (2013).
5. Tintarev, N.: Explanations of recommendations. In Proc. of the 2007 ACM conf. on Recommender systems (RecSys '07). ACM, New York, NY, USA, 203-206, (2007).

A Building as a Context for Multidomain Information Service – Case Study Virtual FIIT

Alena Kovárová

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova 2, 842 16 Bratislava
alena.kovarova@stuba.sk

Abstract. Due to the ever-growing number of information resources, there is a need for smart and sensitive filtering, searching. Moreover, there is the demand for solutions that combine information sources of different domains, based on their common context for the user. Their right combination and integration would allow users to quickly obtain all essential information and thereby saved them a lot of time. One example of such a context is a building in which the user is working, studying, buying or performs other activity related this place - building. In our case study, we chose the building of our Faculty. The key users are our freshmen, but there are also older students and even teachers and staff. During the study we used known quartet (who, where, when, what). We have identified students' domains of interest, which needs to be combined and we have implemented some of them. Finally, we obtained feedback from the students through questionnaires.

Keywords: Information Macro-Service, Building, Freshmen, University

1 Introduction

Every student spends a considerable time looking for information within various information systems that have presumably been devised to make student's life easier. The most basic one is the University information system (UIS) in which the student's data is kept. It includes an electronic transcript of records, an email box, and possibly also electronic materials connected to lectures and seminars. The UIS itself often does not contain everything a student wants or needs to know when studying at some department - for example, "What time does the next bus leave the bus stop?" or "Where are the university canteens that are closest to my department?" etc. The greatest need for a wider range of information is for the first year students (freshmen) who are new at the Faculty. Clearly something providing the desired information, most likely a mobile application, that would offer information connected to Faculty (building), which students consider as the most important, is expected. However, in order to create such an application, there are several research issues to be tackled. In particular, perhaps the greatest challenge is how to put together in one place information re-

trieved from different information systems to make their retrieval as quick and easy as possible.

There have been various works published on the subject or on related issues. Recently, a recommendation for designing mobile pedestrian navigation system in university campuses was discussed in [1]. Applying the concept of information services in development of a smart campus is presented in [2]. The problem of accessibility in such solutions is tackled in [3]. Some solutions put more stress on educational needs [5] while other on comprehensive digitalization [6] of the campus. Our focus is on usability which is improved by information interconnections and supported by user interface design.

The remainder of the paper is structured as follows. First, we present results of a survey aiming to identify the most important information needs as felt by the freshmen represented by a sample of our 2013 intake. Next, we present several types of information providing services that would match those information needs. These services stand as kinds of specifications for our subsequent designing of a comprehensive information providing solution.

2 Survey

To find out what is the most important for students, we conducted a short survey. Freshmen were the natural candidates for participation. We were able to attract 40 participants who amount to roughly 10% of the Faculty's 2013 intake. The survey included only two simple questions:

1. Do you own a phone with Android? If yes, what type it is and what OS version it has?
2. If there would exist a mobile application for students of this Faculty, what functionality would you appreciate?

Out of 40 respondents there were 29 students (i.e. 73 %) who owned a mobile phone with Android OS. This is no surprise since it corresponds to worldwide statistical distribution where Android OS also leads with 81 %. Distribution among different brands of mobile phones is: Samsung 52 %, HTC 28 %, Sony 10 %, LG 7%, Huawei 3 %.. The first question served for us as a kind of assurance that the student population is consistent in this respect with current global situation. Hence the Android platform became our first design choice with implementation focused to support chiefly versions 2.3 and 4.1.

Next crucial question concerned type of information the students seek (Fig. 1). We sorted them from the most desired one to the most rarely requested. Those depicted solid colour are the ones that we have already implemented. Those hatched ones are either covered by other providers or their implementation has been postponed for various reasons. The respondents were not limited how many information types they can indicate. The total number of information type indications was 116. The individual indications were not prioritized. As seen in Fig. 2, the most desired information type is the building map despite the fact they are placed in printed form at all key locations within the building. The second most desired one is a personal schedule

with the exact times and locations of their classes. Currently, schedules can be retrieved from the UIS in a quite complicated way and moreover, there is a risk that any subsequent change in them may stay unnoticed. The third most desired information type is UIS data.

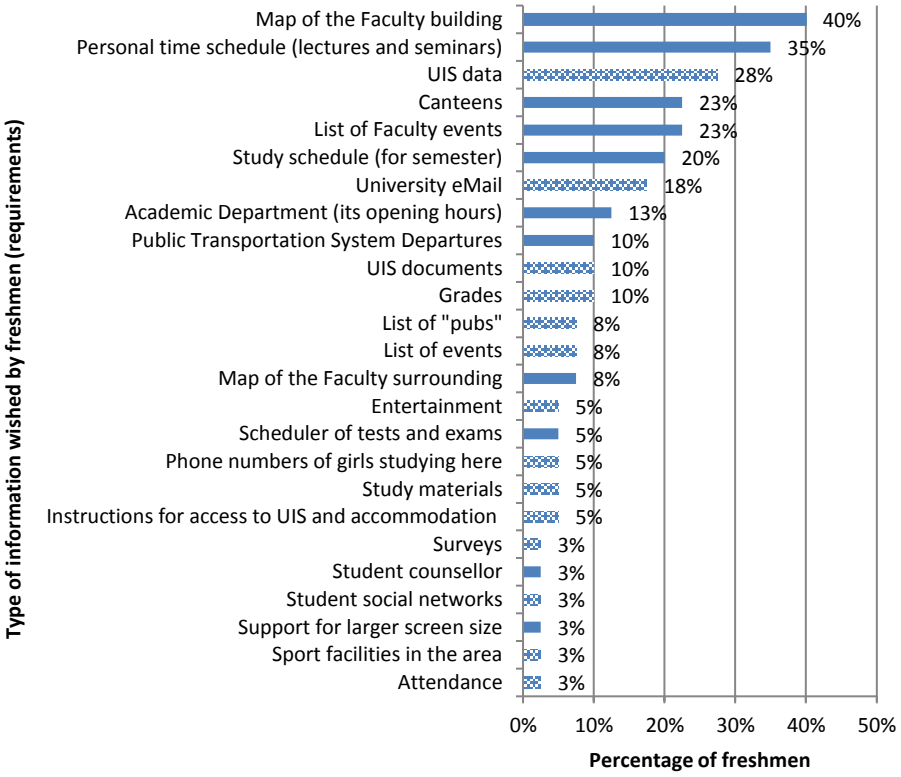


Fig. 1. Wish-list of our freshmen – the type of information (functionality) they seek.

The most time-saving feature of our design are interactive interconnections within information of the four basic types: Person + Time + Place + Event. E.g. for a student in domain of time schedule it can be Teacher + From-To + Room# + Lecture/Seminar. To let him retrieve all the basic information quickly, the overall design must be kept simple. It should offer at least those most desired information in one click or as quickly as possible.

3 Services

We identified eight domains of services, which can cover almost all the students' requirements. For each service we indicate whether it is based on external or internal source of information.

1. Personal services (external – from UIS)
   (a) Time schedule with room + lecture/seminar + teacher (Fig. 2)
   (b) University email, Grades, Points, Attendance, Documents
2. Time schedule services (external, different sources)
   (a) Time schedule of rooms + lectures and seminars + teachers
   (b) Tests and exams time schedule during semester (+ lecture + room)
   (c) Semester time schedule – with holidays and other special days
   (d) Opening hours of academic department office
   (e) Faculty events
3. Transportation service (external – from iTransit)
   (a) Bus stops with bus lines and their departures – from all nearby bus stops
4. Food service (external – from HladnyStudent.sk)
   (a) List of surrounding university canteens with their menu and prices (Fig. 3)
5. Interactive map (internal)
   (a) of building with locations and names of rooms (Fig. 2)
   (b) of surrounding area with locations and names of chosen places, parking lots, pavements
   (c) navigation within space (Fig. 3)
6. Searching (internal)
   (a) Automatic - among list of objects (lectures, rooms, …)
   (b) Automatic - among existing questions and answers or documents and guides
   (c) Manual – there is a list of all information sources (web sites) that a student could need, some of them are already wrapped by application
7. Asking
   (a) Question Answering System (soon external – from Askalot)
   (b) Connections to social networks
8. Free time services - other events, entertainment, pubs, sport activities, etc.

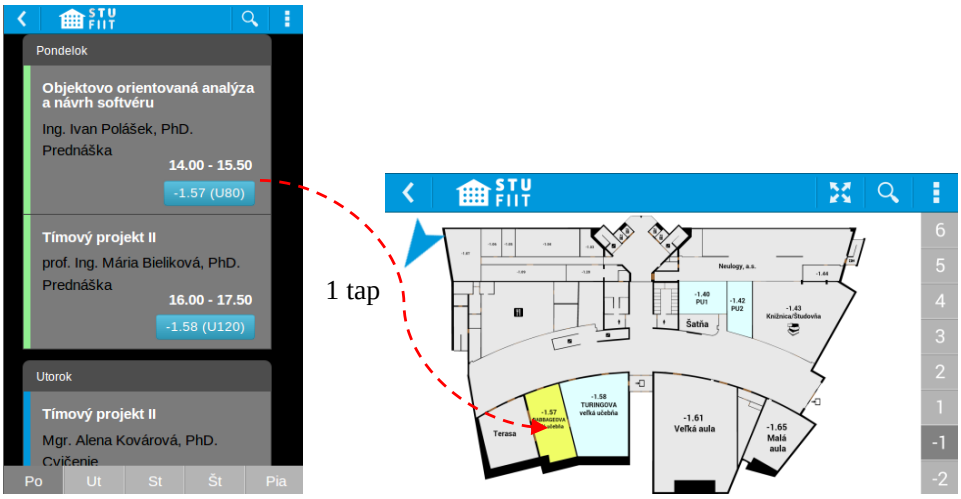


Fig. 2. Screenshots from Virtual FIIT mobile application:
personal time schedule (left) and zoomable map of the building (right)

Each type of service can be augmented by other services of the given type. Their design could also be extended by retrieving and providing more data. In implementing our prototype, however, we concentrated on a critical minimum of services and data provided by them. Our emphasis was on daily automatic data update as well as on provision of services in offline mode.

As was already mentioned, the key value is in interconnections among information sources, which makes the information retrieval easier and quicker. E.g. from a time schedule to the indoor map (Fig. 2) or from the outdoor map to the canteens (Fig. 3).

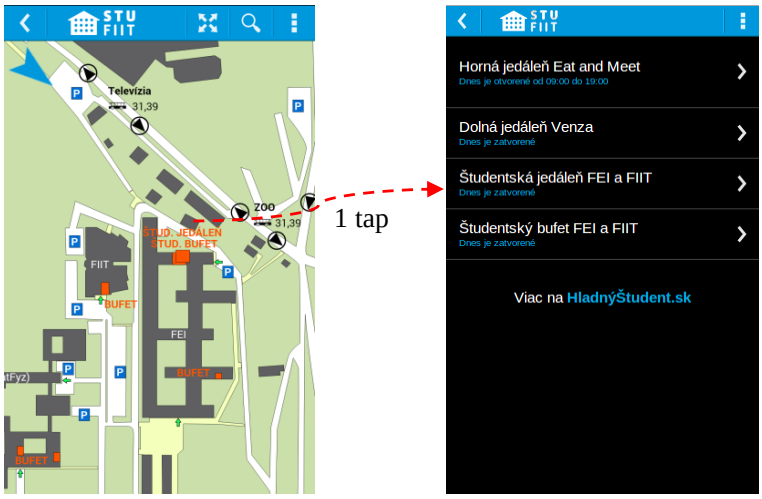


Fig. 3. Screenshots from Virtual FIIT mobile application:
zoomable map of surrounding area (left) and list of nearby university canteens (right)

3.1 Results

Having completed the overall design of the information providing functionalities and of the common user interface, we implemented a working prototype. It is a mobile application running under OS Android, but we also provide our services through the website.

The application has been tested by the students. Last test was conducted in May 2014. 30 volunteers gave us written feedback via our questionnaire. Basically, each prior feedback helped us to redesign some elements to become more intuitive and to prioritize future features implementation. One of the interesting findings from the feedback is that different students prefer different functionalities of the application. There does not seem to be a consensus as to which functionality is the most useful one. Generally, they like it. Actually the most wanted service is notification of new emails and other personal data (such as exam results) available in the UIS.

4 Conclusions and Future Work

Initial survey and constant feedback from freshmen and older students helped us to identify their needs. Based on them, we created a list of services and their interconnections. We designed a uniform solution for a faster and more comfortable retrieval of various types on information. We implemented this idea in an application, which has been published on Google Play. During nearly one year, it has gained a positive response among our students.

We hope to expand our solution to other buildings of our university in the future. However, the greatest potential lies in its re-implementation for other buildings, not only universities, but also the shopping centers, business centers, large corporate buildings, government buildings, hospitals and so on. Even more, whole city could be supported in a similar fashion eventually [4]. When using the internal navigation, users would appreciate seeing their exact location and orientation within the map of the building, which could for example be achieved after deployment of Beacons.

Acknowledgements

This work was partially supported the Scientific Grant Agency of the Slovak Republic, grant No. VG 1/0752/14.. We wish to express our appreciation to our students Cáder Lukáš, Dušek Martin, Dzurilla Jaroslav, Gášpár Roland, Londák Martin, Ševčík Michal and Toma Matej, who designed and implemented a prototype version within their Team Project course.

5 References

1. Tony Shu-Hsien Wang, Dian Tjondronegoro, Michael Docherty, Wei Song, and Joshua Fuglsang. A recommendation for designing mobile pedestrian navigation system in university campuses. In Proc. of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration (OzCHI '13), ACM, pp. 3-12. (2013)
2. Ying Chen, Runtong Zhang, and Shouyi Zhang. Service Encapsulation-Based Model for Smart Campus. J. Electron. Commer. Organ. 10, 4, pp. 31-41. (2012)
3. Arsénio Reis, João Barroso, and Ramiro Gonçalves.Supporting accessibility in higher education information systems. In Proceedings of the 7th international conference on Universal Access in Human-Computer Interaction: applications and services for quality of life - Volume Part III (UAHCI'13), Vol. Part III. Springer-Verlag, pp. 250-255. (2013)
4. Toru Ishida. Activities and technologies in digital city kyoto. In Proceedings of the Third international conference on Information Technologies for Social Capital: cross-Cultural Perspectives (Digital Cities'03), Springer-Verlag, pp. 166-187. (2003)
5. Naomi Fujimura, Hitoshi Inoue, and Satoshi Hashikura.Experience with the educational ICT environment in Kyushu University. In Proc. of the 37th annual ACM SIGUCCS fall conference: communication and collaboration (SIGUCCS '09). ACM, pp. 167-172. (2009)
6. Fang Yuan, Ping Xiao, Qixin Liu, and Xiaolong Fu. Digital campus information portal content organization based on "information architecture". In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS '09). ACM, pp. 1330-1334. (2009)

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 78–81

Paralelná inferencia sémantickej siete

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 si kladie za snahu oboznámiť s možnosťou paralelnej rekurzívnej inferencie masívnej sémantickej siete realizovanej po častiach. To všetko za použitia modulácie inferovanej informácie pomocou bieleho šumu. Tým sa zabezpečí možnosť paralélnej lokálnej inferencie jednotlivých symbolov, so zachovaním vlastností, akoby bola inferencia realizovaná izolovane. Pritom sa zachováva kontext odvodenej informácie, a to aj naprieč krížovým referenciám.

Kľúčové slová: paralelná inferencia, inferencia po častiach, stochastická inferencia, sémantická sieť

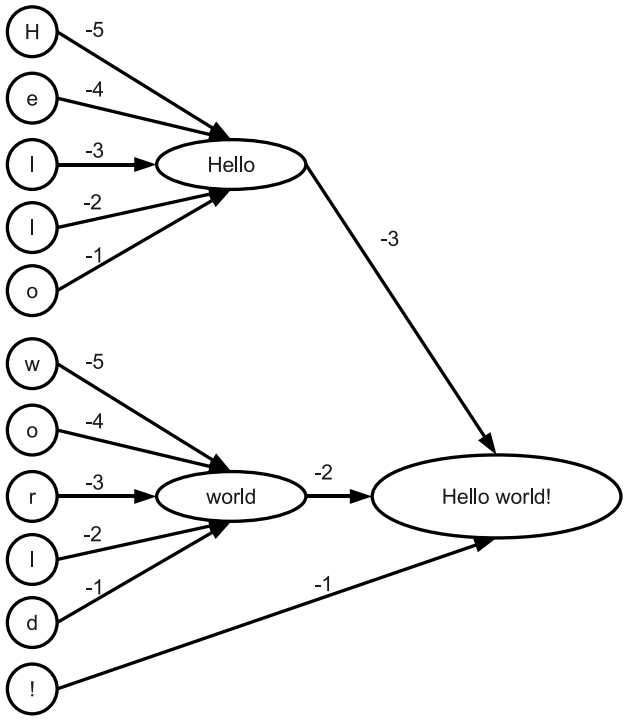
1 Úvod

V rámci projektu inteligentného sémantického vyhľadávania sa snažíme vyhľadávať v informáciach za pomoci sémantickej siete.

Vstupom do sémantickej siete sú dokumenty Wikipédie, tie sú tokenizované a prevedené do termov. Napríklad za pomoci zmeny kategórie znakov (prechod z čísla na alfa-znak, prechod na biely znak, interpunkčné znamienko, či prechod na ostatné znaky). Výsledkom je veľké množstvo termov, ktoré sú následne uložené do indexu $1term = 1symbol$. Medzi týmito termami sa definujú relácie vzhľadom na koreláciu daných symbolov.

V dôsledku veľkého množstva informácií vzniká následná potreba inferencie veľkého množstva relácií, čo je ešte násobené faktom, že inferencia pokračuje aj uzlami, ktoré nie sú priamo súčasťou dotazu, ale sú odvodené na základe štatistických informácií uložených v reláciach, alebo výberom relácií vybraných aplikačne (matematické výrazy). Vďaka tomu inferencia pokračuje do hĺbky - ma vlastnosti rekurzívnej inferencie a počet inferencovaných symbolov rastie exponenciálne.

Metódy ktoré sa snažili o postupnú inferenciu so sledovaním kontextu sa ukázali ako neefektívne a nepoužiteľné pri takýchto aplikáciach. Keďže sú limitované počtom vyhodnocovaných uzlov ako aj tým, že nevedia byť rekurzívne do dostatočnej hĺbky. Tým sme boli nútený hľadať takú inferenciu, ktorá sa v globále správa ako štandardná inferencia a však ktorá je schopná inferencovať celú sieť naraz (veľké množstvo uzlov) a súčasne po častiach (v každom okamihu je závislá iba od svojho okolia). Práve to nám umožňuje paralelnú inferenciu.



Obr. 1. Sémantická sieť

1.1 Sémantická sieť

Pre reprezentáciu zaindexovaných dokumentov používame sémantickú sieť. Úlohou sémantickej siete je reprezentovať sémantické vzťahy medzi konceptami [3]. Sémantická sieť je jednou z foriem používanou na reprezentovanie informácií, a je reprezentovaná vo forme orientovaného (ale aj neorientovaného) grafu. Koncepty sa uchovávajú vo vrcholoch grafu a hraný slúžia pre uchovávanie vzťahov medzi jednotlivými konceptami.

V našom prípade sémantická sieť používa vrcholy na reprezentáciu termov, ktoré sú výstupom indexácie, a hrany na reprezentáciu vzťahov medzi termami.

1.2 Kontext informácií

Pod kontextom informáciami chápeme textovú spojitosť, súvislosť. Ak opisujeme parkovisko na ktorom je modré a červené auto. Tak pri opätovnom hľadaní sa snažím rozlišovať fakt, že sa jedná o dve rôzne autá, a nie len jedno auto. Respektíve ak tvrdíme, že vták má krídla a lieta. Tak aj napriek tomu, že pštros je vtákom, automaticky to neznamená, že lieta. V kontexte keď vieme o aký druh sa jedná, vieme rozhodnúť či lieta, alebo nelieta.

1.3 Nemonotónne informácie

Nemonotónna logika - jedná sa o získavanie, uchovávanie a spracovávanie zrušiteľných znalostí [4]. V protikladu ku klasickej logike, kde sa pridávaním ďalších informácií pôvodne odvodené závery nemôžu zmeniť, tak v prípade nemonotónnej logiky sa naopak už odvodené závery môžu zmeniť, a to napríklad ak dôjde k zisteniu nových informácií, ktoré odporujú resp. sú v protiklade s predpokladanými závermi.

Nemonotónna logika je pre nás dôležitá pri inferencovaní dotazov nad sémantickou sieťou, a to z nasledujúcich dôvodov:

1. Sémantická sieť obsahuje veľké množstvo informácií a pre kompaktnosť reprezentácie tak veľkého množstva informácií je potreba ich zovšeobecňovania.
2. Inferencia sa deje paralelne, izolovane a po častiach.

V oboch prípadoch je potreba pracovať s neúplnými informáciami.

2 Stochastická inferencia

Rozhodli sme sa siahnuť po stochastických princípoch tak, aby nebola ovplyvnená presnosť, a tak aby inferencia závisela iba od svojho okolia. Taktiež sme sa snažili zachovať existujúce vlastnosti sledovania kontextu.

Biely šum Náhodný signál ktorý má rovnomernú spektrálnu výkonovú hustotu sa označuje ako biely šum [1]. V podstate ľubovoľná distribúcia hodnôt, ktorá má nulovú strednú hodnotu [2] sa dá považovať za biely šum.

Dôležitými vlastnosťami pre nás sú:

1. Stredná hodnota je nulová
2. Náhodný signál / náhodná distribúcia hodnôt
3. Rovnomerná výkonová spektrálna hustota

Tým, že stredná hodnota je nulová, informácie modulované za pomoci bieleho šumu nie sú ovplyvnené čo do amplitúdy za predpokladu, že použitý šum nebude korelovať s modulovanou informáciu. To je zabezpečené druhou podmienkou, ktorá hovorí o tom, že hodnoty sú náhodné, a teda pravdepodobnosť korelácie závisí od náhodnosti bieleho šumu. Rovnomerná výkonová spektrálna hustota nám zase zabezpečí, že výsledok nebude ovplyvnený ani v prípade krížových referencií. T.j.: takých referencií, kde ten istý term sa vyskytuje v dvoch rôznych kontextoch.

2.1 Indexácia a modulácia

Počas indexácie sa sledujú korelácie medzi uzlami siete a vytvárajú sa spojenia u ktorých sa eviduje: časové oneskorenie, to koľkokrát spojenie bolo aktivované a to koľkokrát spojenie malo byť aktivované a nebolo (pomocou spätnej revízie), ako aj ďalšie štatistické informácie.

Indexácia sa realizuje v niekoľkých fázach:

– **prvotná indexácia**, ktorá sa deje pri každom dokumente ktorý sa zaraďuje do sémantickej siete
– **počas vyhľadávania** za pomoci spätnej inferencie zvolených dokumentov
– **počas aktivácie** v prípade symbolov ktoré nie sú fixné, nemajú pevné spojenia a sú vyhodnocované za behu

Pri modulácii rozlišujeme či sa jedná o:

– **vstupnú vrstvu**: Požiadavka sa pretransformuje na množinu zoradených termov. Tie sú periodicky aktivované s rozfázovaním modulovaným pomocou bieleho šumu.
– **strednú resp. výstupnú vrstvu**: Symboly sú aktivované vstupnou vrstvou ako aj symbolmi zo strednej ale aj výstupnej vrstvy, a to vždy cez vstupné spojenia so zohľadnením časového oneskorenia a pravdepodobnosti. Pričom stochastická aktivácia založená na spomenutých informáciach je v podstate taktiež modulácia bielym šumom.

2.2 Vyhodnotenie vrátených výsledkov

Výstupná vrstva sémantickej siete je aktivovaná v rôznych intervaloch, a vďaka zvolenej modulácie môžme povedať, že stredná hodnota aktivácie výstupného symbolu je ekvivalentná relevancii hľadanej informácie. Inými slovami, ak všetky dokumenty ktoré boli aktivované minimálne raz zoradíme podľa ich strednej hodnoty, tak dostaneme zoznam dokumentov zoradených podľa ich relevancie voči dotazu.

3 Záver

V článku sa snažíme poukázať na fakt, že sémantická sieť môže byť inferovaná paralelne a po častiach, a to aj bez potreby inferencie všetkých uzlov za sebou. Taktiež poukazuje na spôsob, ako pracovať s neúplnosťou informácií počas inferencie za použitia bieleho šumu, ktorý nemá priamy vplyv na výsledky inferencie, ale umožňuje nám zachovávať kontext odvodených informácií.

Referencie

[1.] Carter Mancini, Bruce Ron Op Amps for Everyone. Texas Instruments (2009) ISBN 0080949487 p. 10-11
[2.] Jeffrey A. Fessler, On Transformations of Random Vectors, Technical report 314, Dept. of Electrical Engineering and Computer Science, Univ. of Michigan (1998)
[3.] John F. Sowa, "Semantic Networks". In Stuart C Shapiro. Encyclopedia of Artificial Intelligence (1987), Retrieved 2008-04-29
[4.] Nemonotónne rozhodovanie, Martin Pašmík, `http://www2.fiit.stuba.sk/~kapustik/ZS/Clanky0809/pasmik/index.html`

Modelovanie používateľa, komunikácia

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 85–90

Identita uživatele na sociálních sítích a v digitálních knihovnách

Adam Ondrejka, Jakub Stonawski, Petr Šaloun, Petr Haman a Veronika Zoltá

VŠB - Technická univerzita Ostrava, 17. listopadu 15, 708 33 Ostrava, Česká republika
{adam.ondrejka, jakub.stonawski}@gmail.com
{petr.saloun, petr.haman.st, veronika.zolta.st}@vsb.cz
http://www.vsb.cz

Abstrakt Vytvořili jsme vlastní metodu identifikace uživatelů různých typů služeb Internetu s dílčím možným využitím technik rozpoznávání obličejů. Shrnujeme náš původní přístup k hledání unikátní identity uživatele Internetu. Úžeji se věnujeme odhadu oblastí výzkumu autorů založenému na analýze jejich publikační činnosti a na veřejně dostupných metadatech z digitálních knihoven a na informacích, které o sobě autoři zveřejňují na sociálních sítích.

Keywords: identita uživatele, digitální knihovna, sociální síť, rozpoznávání obličejů

1 Úvod

Rozmach a zpřístupnění Internetu široké veřejnosti v poslední době umožnil lidem po celém světě zveřejňovat informace o své vlastní osobě, které ho svým způsobem popisují. Bohužel míst, kde o sobě může daná osoba publikovat je mnoho a situaci neulehčuje ani fakt, že stejně pojmenovaných může být mnoho lidí. Některé situace se pak velmi komplikují, jedním příkladem může být případ pořádání konferencí. Organizátor konference potřebuje pro každý přihlášený článek vybrat vhodné recenzenty. Tento úkol není obtížný v případech menších konferencí, kdy předseda výboru zná ostatní členy a počet přihlášených příspěvků není velký. Značně se situace ovšem komplikuje při větších konferencích, kdy již není možné vlastními silami udržovat přehled o oblasti působnosti jednotlivých recenzentů.

Cílem práce bylo vytvořit algoritmus, který nalezne autory publikací na sociálních sítích a podle veřejně dostupných informací také odhadne jejich oblast výzkumu. Právě to by poté mělo usnadnit organizátorům snadnější posuzování.

V další fázi jsme analyzovali efektivitu využití prvku rozpoznávání obličejů, jako verifikačního faktoru, který by měl přispět k větší úspěšnosti identifikace uživatelů.

Kombinaci metod získávání informací z webu a rozpoznávání obličejů považujeme jako hlavní přínos této práce, ačkoliv výzkum je stále ještě v procesu.

2 Stav poznání

Tato práce navazuje a částečně srhnuje poznatky popsané v našich předchozích publikacích [7,9].

V publikaci [6] Elie Raad používá podobných principů ve vektorovém prostoru. Jednotlivým klíčovým slovům navíc přidává váhu a porovnává také celé věty na profilech uživatelů. Výsledek, zda se uživatelé shodují nebo ne, je nakonec rozhodnut na základě skóre podobnosti mezi profily a ručně nastavenou hodnotou.

Mutschke a Haaseová v článku [2] provedli síťovou analýzu uživatelů na základě bibliografických záznamů obsahujících klíčová slova publikací a následnou aplikací analýzy sociálně-kognitivní sítě.

Z oblasti digitálních knihoven publikoval Martín v článku [1] techniky k nalezení podobných vědeckých článků pomocí jazykového modelu. Princip je založen na odhadu, zda jeden text dokáže vygenerovat slova z textu druhého. Odhad je založen na předchozí analýze abstraktů, klíčových slov a následovného strojového učení a může být alternativou, případně doplněním k porovnávání publikací ve vektorovém prostoru.

Technika rozpoznávání obličejů zatím nebyla v praktických situacích příliš využívána, zejména z důvodu nížší úspěšnosti. Pro naší práci je zajímavá výzkumná činnost, která byla využita samotnou sociální sítí Facebook, implementující svůj experimentální algoritmus rozeznávání tváří přátel uživatele [10]. Tento algoritmus poté síť využívala k automatickým návrhům označování osob na fotografiích. Tato implementace navíc otevřela otázku právního aspektu během rozpoznávání obličejů a vedla k diskuzi, jestli spadají obrazová data obličejů uživatelů do kategorie osobních údajů, potřebujících zvláštní ochranu [5].

Další z prací, která se tématu analýzy a detekce obličejů zaobírala je část projektu FaceBots [3], konkrétně studie *Friends with Faces* [4], jejíž cílem bylo vyvinout robota s funkcí rozpoznávání obličejů, který by byl navíc propojen do sociální sítě a byl tak schopen analyzovat sociální vztahy mezi detekovanými osobami.

3 Nalezení identity uživatele

Hlavní myšlenku pro hledání identity autora na sociálních sítích a v digitálních knihovnách jsme postavili na typech informací, které o sobě uživatel zveřejňuje a které ho popisují. Tyto dva typy jsme pojmenovali jako *obecné* a *unikátní*. Mezi *obecné* atributy můžeme zařadit např. *pohlaví*, *bydliště*, ale i *jméno a příjmení*. Jde o vlastnosti, které uživatele potvrzují a identifikují, ale nemůžeme o něm ještě rozhodnout, zda se jedna skutečně o námi hledanou osobu. Na druhou stranu atributy *unikátní* konkretizují. *Email* nebo *telefonní číslo* by měl skutečně používat jen jeden člověk, předpokládáme samozřejmě scénáře, kdy uživatel o sobě vyplňuje pravdivé informace. Případným dalším obecným atributem může být právě profilová fotografie zachycující *obličej uživatele*. Nalezením dostatečného počtu obecných atributů a alespoň jednoho unikátniho jsme

schopni poměrně přesně odhadnout a určit identitu uživatele napříč různými sítěmi. Níže vzorec použitý v algoritmu (vychází z návrhu popsaném v [7]):

$$sim_{u,p} = \begin{cases} \sum_{i=0}^{n} w_i \cdot sim(a_{i,u}, a_{i,p}) & \text{pokud } sim(a_{name}) > th_{name} \\ 0 & \text{jinak} \end{cases} \quad (1)$$

kde $sim(a_{name})$ je podobnost mezi jmény; th_{name} je prahová hodnota k určení, zda se jména shodují; n je počet porovnávaných atributů; w_i je váha porovnávaných atributů; a_p je množina atributů profilu uživatele na síti; a_u je množina atributů hledaného uživatele v databázi; $sim(a_{i,u}, a_{i,p})$ je podobnost atributů mezi profilem a vybraným uživatelem

Při porovnávání atributů se nedá spolehnout na prosté porovnávání řetězců. Jedna identická entita bývá na různých sítích popsána trochu jiným výrazem, např. telefonní číslo může být na jedné síti oddělené mezerami a na druhé ne. Adresa v digitálních knihovnách bývá take velmi často napsaná různě, příkladem budiž výrazy *VŠB – Technical University of Ostrava* a *VSB Tech. Univ. of Ostrava*. Pro určení shody textových řetězců jsme tedy použili algoritmus částečné shody, konkrétně *Levenstheinovu vzdálenost*. Detailnější popis algoritmu je popsán v publikaci [7].

Pokud navíc využijeme dalšího unikátních typů informací, konkrétně obličeje uživatele, jsme schopni porovnávání dále zdokonalovat především na sociálních sítích.

3.1 Digitální knihovny

V případě digitálních knihoven jsme analyzovali veřejná metadata o publikacích autorů. Jako klíčové se ukázaly atributy definující společenskou síť autora, která značně reflektuje jeho spojení s jinými uživateli na jiných sociálních sítích. Jde především o *spoluautory* a *instituce*.

3.2 Sociální sítě

Narozdíl od digitálních knihoven, sociální sítě jsou odlišné a specifické, uživatelé si vybírají, které informace na síti zveřejní. Ve většině případech se dají o uživateli zjistit *určující* atributy jako jsou adresa či pohlaví.

Většina unikátních atributů uživatele zpravidla na sociálních sítích zůstává veřejně nedostupná (emailová adresa, bydliště), ovšem profilovou fotografii, ukazující identifikovatelný obličej uživatele má velké procento profilů sociální sítě (viz [9], str. 55). Jako klíčové se zde ukázalo zkoumání spojení s dalšími autory a hledání shod v jiných sítích či autory z digitálních knihoven. Rovněž zaměstnání uživatele může být institucí z některých jeho publikací.

Využití analýzy profilové fotografie zde může být bráno jako podpůrný prostředek, v případech, kdy nebude dostatečně jistá identita uživatele (např. ostatní atributy jako jméno či instituce budou stejné).

4 Experiment

V rámci experimentu jsme ověřovali, zda pro 180 náhodně zvolených autorů různých národností bude správně nalezena jejich identita nebo ne. Z digitálních knihoven byly vybrány ACM Digital Library[1], IEEExplore[2] a SpringerLink[3] a ze sociálních sítí LinkedIn[4] a ResearchGate[5].

V první fázi jsme testovali hledání identit pouze pomocí atributů spoluatorů a institucí v digitálních knihovnách. Ze 180 autorů bylo správně nalezeno 118 autorů (sloupec "S"), 3 různí autoři byli nesprávně sloučeni pod jednu identitu (sloupec "NS") a 59 autorů bylo nesprávně identifikováno jako více autorů (sloupec "VA"). Správnost algoritmu byla v tomto případě 65,5 %, chybovost okolo 34,5 %.

Tabulka 1. Experiment hledání identit

	S	NS	VA
Spoluautoři	118	3	59
Spoluautoři + Socialní sítě	132	3	45
Spoluautoři + Socialní sítě + Klíčová slova	166	14	0

V druhé fázi jsme přidali porovnávání o data ze sociálních sítí. Správně bylo nalezeno 132 autorů, 3 autoři stále zůstali nesprávně sloučeni a 45 z nich zůstalo nesprávně vytvořeno více identit. Správnost algoritmu vzrostla na 73,3 % a chybovost kleska na 26,7 %.

V poslední, třetí, fázi experimentu jsme k sociálnícím sítím přidali porovnávání klíčových slov publikací a další atributy ze sociálních sítí, jako jsou zájmy, zkušenosti apod. Správně bylo nalezeno 166 autorů, bohužel se zvýšil počet identit více autorů sloučených pod jednu identitu na 14. Naopak žádnému autorovi nebylo vytvořeno více identit. Správnost v poslední fázi byla 92,2 %, chybovost klesla na 7,8 %. Nutno ovšem poznamenat, že se zvýšil horší scénář, kdy jednomu autorovi jsou přiřazeny profily a publikace jiných osob, což může výrazně zkreslovat další práci se získanými daty.

Prozatím odděleně byly zkoumány úspěšnosti algoritmu, sloužícímu k verifikaci uživatelů s využitím rozpoznávání obličejů. Bylo využito externí webové služby Betaface[6], která je schopna porovnávat obličeje a analyzovat jejich shodu. Nejprve byly provedeny testy, ke zjištění spolehlivosti algoritmů rozpoznávání této služby. Jako testovací data byly využity fotografie obličejů, poskytnuté Car-

[1] http://dl.acm.org/
[2] http://ieeexplore.ieee.org/Xplore/home.jsp
[3] http://link.springer.com/
[4] https://www.linkedin.com/
[5] http://www.researchgate.net/
[6] http://www.betaface.com/

negie Mellon University (Ralph Gross)[7], které zachycují obličeje osob, zachycené v odlišných fotografických podmínkách. V průměru byla vyčíslena úspěšnost Betaface algoritmů na 58 %, a jako největším problémem byla stanovena detekce obličejů, pokud je daný uživatel na fotografii zachycen pod velkým úhlem, případně nejsou viditelné primární objekty ve tváři (oči, ústa, nos,...). Viz výsledky v Tabulce 2.

Tabulka 2. Úspěšnost Betaface při hledání tváře ve fotografii [8]

Typ zachycení obličeje	Osoba A	Osoba B	Osoba C
běžný úhel snímání	100 %	100 %	67 %
velký úhel snímání	0 %	0 %	14 %
odlišné světelné podmínky	100 %	100 %	100 %

V další fázi byla rozpoznávací služba napojena na sociální síť Facebook. K náhodně vybraným fotografiím, zachycujícím osoby, které byly k dalším testům využity, byla připojena textová informace o jménu a příjmení osob, které jsou na fotografii. Toto mělo simulovat budoucí napojení na analyzované textové informace z předchozí fáze experimentu. Aby bylo možno tento oddělený experiment vyhodnotit, byl jako cíl stanoven, aby algoritmus dovedl rozeznat na náhodně vybraných fotografiích stávající přátele na Facebooku uživatele, a k novým lidem z fotografií se pokusit dohledat jejich profily na Facebooku a navrhnout je uživateli za nové přátele (pokud analyzovaná fotografie obsahovala textový popis připojený v předešlém bodě).

V sérii 20 náhodně vybraných fotografií různých osob bylo možno detekovat 10 nových přátel na sociální síti. Algoritmus byl schopen poskytnout 4 nová spojení, ale nedetekoval ani jedno chybné spojení (nedošlo k špatné identifikaci osob), tudíž je možno říci, že analýza obličejů se jako dodatečný verifikační prvek (k prvotním textovým informacím obsahujícím pouze jména osob) osvědčil. Z výsledků navíc vyplynulo, že Betaface byl schopen detekovat tváře všech deseti potencionálních nových přátel. Nemožnost navrhnout nové spojení se projevila teprve v druhé fázi experimentu, kdy se osoba vyhledávala podle jména na síti Facebook. Dohledatelné tyto osoby podle jmen byly, ale jejich profilové fotografie nebyly službou Betaface detekovány jako shodné s analyzovanou fotografií.

5 Závěr

Vytvořili jsme algoritmus k nalezení identity uživatele v digitálních knihovnách a na sociálních sítích. Výsledky našeho experimentu ukázaly, že identifikace uživatelů byla správná v 92 % případů z celkem 180 autorů.

[7] http://www.ri.cmu.edu/research_project_detail.html?project_id=418&menu_id=261

Ve druhé fázi jsme zkoumali možnosti praktického využití porovnávání obličejů. Bylo zjištěno, že stávající situace již umožňuje využití dostatečně spolehlivých algoritmů rozpoznávání. Tyto algoritmy jsou navíc veřejně dostupné, jako freeware webové služby, takže jejich zapracování do složitějších celků je možné.

Tato práce byla prvním krokem ve výzkumu hledající jednoznačnou identitu uživatele na Internetu nezávisle na typu webové stránky a ve výzkumu doporučování dle jednoznačné identity a stereotypu chování uživatele. Dalším krokem výzkumu bude zapracování verifikačního rozpoznávání obličejů do stávajících algoritmů identifikací uživatelů na Internetu, protože výsledky ukázali, že úspěšnost rozpoznávání obličejů je již na dostatečně vysoké úrovni. Hlavním přínosem této práce je především propojení dvou odlišných metod, a sice rozpoznávání obličejů a získávání informací z textu, k nalezení identity uživatele.

Reference

1. G. Hurtado Martín, S. Schockaert, Ch. Cornelis, and H. Naessens. Finding similar research papers using language models. In *2nd workshop on semantic personalized information management : retrieval and recommendation, Proceedings*, pages 106–113. University College Ghent, 2011.
2. P. Mutschke and A. Haase. Collaboration and cognitive structures in social science research fields. towards socio-cognitive analysis in information systems. *Scientometrics*, 52(3):487–502, 2001.
3. Shervin Emami Nikolaos Mavridis, Chandan Datta, Andry Tanoto, Chiraz BenAbdelkader, and Tamer Rabie. Facebots: Social robots utilizing facebook - human-robot interaction. In *ACM/IEEE 4th International Conference*, 2009.
4. Wajahat Kazmi Nikolaos Mavridis and Panos Toulis. Friends with faces: How social networks can enhance face recognition and vice versa. In *Computational Social Network Analysis*, pages 453–482. Springer-Verlag London Limited, 2010.
5. Office of the Data Protection Commissioner. Complaint against facebook ireland ltd.[online]. pages 101–105, 2011.
6. E. Raad, R. Chbeir, and A. Dipanda. User profile matching in social networks. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, pages 297–304, Sept 2010.
7. Petr Saloun, Adam Ondrejka, and Ivan Zelinka. Estimating users' areas of research by publications and profiles on social networks. In *Hypertext 2014 Extended Proceedings: Late-breaking Results, Doctoral Consortium and Workshop Proceedings of the 25th ACM Hypertext and Social Media Conference (Hypertext 2014), Santiago, Chile, September 1-4, 2014.*, 2014.
8. Petr Saloun, Jakub Stonawski, and Ivan Zelinka. Face recognition element as a supported tool for new contact suggestions in the social networks. In *International Conference on Advanced Engineering – Theory and Applications, AETA 2013, Vietnam, 2013*, pages 349–361, 2013.
9. Petr Saloun, Jakub Stonawski, and Ivan Zelinka. Recommending new links in social networks using face recognition. In *8th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2013, Bayonne, France, December 12-13, 2013*, pages 89–94, 2013.
10. Todd Zickler Zack Stone and Trevor Darrell. Autotagging facebook: Social network context improves photo annotation. In *Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference, 1-8, 2008*, pages 1–8, 2008.

Odhad expertízy vývojára na predchádzanie vzniku chýb v softvérovom projekte

Eduard Kuric, Karol Rástočný a Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
{eduard.kuric,karol.rastocny,maria.bielikova}@stuba.sk

Abstrakt. V tomto vizionárskom príspevku prezentujeme novú myšlienku na odhad expertízy vývojára na základe kvality jeho práce určenej z vývojových aktivít a výsledného zdrojového kódu. Odhad expertízy vývojára je dôležitý pri tvorbe softvéru, napr. pri hľadaní špecialistov s určitými schopnosťami, pri formovaní tímov (napr. vhodné dvojice pri programovaní v pároch), alebo pri hľadaní/porovnávaní kandidátov na pracovné pozície. Existuje veľa „konkurenčných" definícií pre expertízu vývojára. Pri odhadovaní expertízy je možné uvažovať množstvo vplývajúcich aspektov (napr. znalosť API, miera chýb), a preto sme sa sústredili na motiváciu a aktuálny stav poznania v tejto oblasti. V porovnaní s existujúcimi prístupmi sa zameriavame na využitie komplexnejších interakčných údajov na úrovni vzorov (odhaľovanie sledu vývojových aktivít vedúcich k chybe), a okrem použitia štandardných softvérových metrík, sa zameriavame na využitie analýzy zdrojového kódu na odhaľovanie „slabých miest".

Kľúčové slová: expertíza, vývojár, vývoj softvéru, kvalita, zdrojový kód, aktivita, interakcia

1 Úvod

Vývojári každý deň čelia) úlohám v nových doménach. Robia rozhodnutia, ktoré si vyžadujú širokú škálu informácií o softvérových projektoch, na ktorých pracujú. „Kvalita" rozhodnutí často závisí od ich znalostí, skúseností a zručností (expertízy).

Odhad expertízy vývojára je dôležitý pri tvorbe softvéru [12], obzvlášť pri efektívnom znovupoužití zdrojového kódu (softvérových artefaktov). Webové informačné systémy sú čoraz častejšie vytvárané (zostavované) vývojom riadeným vyhľadávaním (angl. search driven development). Vývojári sa stále viac spoliehajú na hotové (overené) softvérové artefakty, ktoré môžu znovupoužiť v svojich riešeniach. Web (pavučina) softvérových artefaktov (napr. CodeProject[1], Krugle[2], StackOverflow[3]) je rozsiahlym repozitárom zdrojového kódu, ktorý často poskytuje vývojárom útočisko pri

[1] CodeProject: http://www.codeproject.com
[2] Krugle: http://opensearch.krugle.org
[3] StackOverflow: http://stackoverflow.com

riešení ich cieľových úloh. Na podporu vývoja riadeného vyhľadávaním nepostačuje „prosté“ plnotextové vyhľadávanie/odporúčanie softvérových artefaktov (angl. fulltext search over software artifacts). Samozrejme, relevancia softvérových artefaktov vzhľadom na cieľovú úlohu je dôležitá, avšak nemenej dôležitá je ich dôveryhodnosť. Keď sa cieľový vývojár rozhoduje znovupoužiť „kus zdrojového kódu“ z externého zdroja, musí pritom často dôverovať práci neznámeho vývojára. Ak cieľový vývojár disponuje vierohodnou informáciou o renomé/expertíze neznámeho vývojára (napr. hodnotiace skóre založené na spätnej väzbe od iných vývojárov), potom sú jeho rozhodnutia o znovupoužití zdrojového kódu istejšie a často priamočiarejšie.

Systémy na odporúčanie odborníkov pri tvorbe softvéru pomáhajú lokalizovať/objaviť a odporučiť odborníkov s najväčšou expertízou pre softvérové artefakty. Odhad expertízy vývojára na fragment zdrojového kódu má dopad na produktivitu. Napríklad, vývojár, ktorý vie, ako použiť funkcie/metódy a triedy, ktoré sú súčasťou vývojovej úlohy, nepotrebuje často konzultovať a študovať dokumentáciu v porovnaní s vývojárom, ktorý sa v danom zdrojovom kóde nevyzná. Inými slovami, systémy na odporúčanie odborníkov sú napr. užitočné pri cielenom prerozdeľovaní úloh na zvýšenie efektivity práce. Pri údržbe softvérového systému je úloha pridelená najvhodnejšiemu vývojárovi, a to na základe odhadu jeho expertízy o softvérových artefaktoch (komponenty, moduly, aplikačné rozhrania), ktoré sú, či už priamou, alebo nepriamou súčasťou danej úlohy.

Manažéri v IT odvetví často čelia výzve na zlepšenie efektivity a kvality vývoja softvérových systémov. Množstvo ich aktivít (napr. plánovanie, priraďovanie úloh) môže byť podporených automatickým odhadom expertízy vývojárov, a to napr. pri hľadaní špecialistov s určitými schopnosťami, pri formovaní tímov (vhodná dvojice pri programovaní v pároch), alebo pri hľadaní/porovnávaní kandidátov na pracovné pozície.

V porovnaní s našou predchádzajúcou prácou, kde sme sa zameriavali na familiaritu vývojára so softvérovým artefaktom [4,5], v tomto príspevku sa zameriavame na kvalitu práce vývojára, ktorú odhadujeme jednak sledovaním procesu vývoja (napr. identifikovaním vývojových aktivít vedúcich k chybám v zdrojovom kóde) a tiež analyzovaním kvality vytvoreného zdrojového kódu (napr. analyzovaním chýb v kóde).

V rámci širšieho projektu PerConIK[4] máme navrhnutú a implementovanú infraštruktúru [3] na zber údajov počas vývoja softvéru v reálnom čase (napr. sledovanie aktivít programátora vo vývojom prostredí). Na projekte spolupracujeme so softvérovou spoločnosťou, čo napomáha úspešnému aplikovaniu dosiahnutých výsledkov do praxe. V súčasnosti nepretržite zbierame a analyzujeme aktivity vývojárov, jednak v softvérovej firme (komerčné prostredie) a jednak aktivity študentov pri výučbe programovania (akademické prostredie), ktoré umožňujú overovanie výsledkov a vzájomné porovnania (spôsob programovania vývojára v softvérovej firme vs. spôsob programovania študenta).

Našim cieľom/víziou je návrh a realizácia metódy na odhad expertízy vývojára na základe vývojových aktivít; štandardných softvérových metrík extrahovaných zo zdrojového kódu; histórie o chybách v zdrojovom kóde získanej z nástroja na evidenciu a sledovanie chýb (angl. issue tracking system); a analýzy zdrojového kódu na odhalenie

[4] PerConIK: http://perconik.fiit.stuba.sk

„slabých miest“ v zdrojovom kóde. Výstupom metódy bude ohodnotenie vývojárov v softvérovom projekte na základe množstva a dôležitosti vytvorených chýb a pravdepodobnosti vovedenia nových chýb do softvérového projektu.

2 Prístupy k odhadu expertízy vývojára

Existujúce prístupy na automatický odhad expertízy vývojára sa spravidla delia na základe informácií, ktoré sú použité v metódach odhadovania expertízy. Prevažne ide o *históriu interakcií*, *zmeny v zdrojovom kóde, záznamy o chybách v zdrojovom kóde,* a *skúsenosti s použitými technológiami*.

Interakcie. História interakcií (angl. interaction history) sú údaje, ktoré zachytávajú a opisujú aktivity vývojára so softvérovými artefaktami pomocou podporných nástrojov (napr. Mylyn[5]). Interakcie môžu byť napr. navigácia vývojára - výber/označenie (angl. selection) a zmena (angl. edit) vo fragmente zdrojového kódu vykonaná vývojárom vo vývojovom prostredí. História aktivít umožňuje lepšie preskúmať a pochopiť správanie vývojárov, ich úmysly/zámery a informačné potreby pri plnení úloh.

Fritz a kol. [1] navrhli metódu na automatické vytváranie modelu expertízy vývojára (angl. Degree-of-Knowledge model), ktorý zachytáva mieru familiárnosti vývojára so softvérovým artefaktom. Navrhnutý prístup pozostáva z dvoch hlavných komponentov, a to modelovanie dlhodobej znalosti na základe miery autorstva (angl. Degree-of-Authorship) a modelovanie krátkodobej znalosti na základe miery záujmu (angl. Degree-of-Interest). Zatiaľ čo miera autorstva je vyjadrená napr. počtom zmien vykonaných v cieľovom artefakte, tak miera záujmu sa odvíja od počtu interakcií s daným artefaktom. Výsledná expertíza vývojára je vypočítaná ako lineárna kombinácia faktorov, ktoré vplývajú na dlhodobú a krátkodobú znalosť.

Robbes a Röthlisberger [11] predstavili automatický proces na porovnávanie metrík na odhad expertízy, ktorý je založený na aktivitách vývojára. Autori zadefinovali dve metriky založené na aktivitách označenie/zmena zdrojového kódu, pričom v druhej z metrík, znižujú váhu aktivít podľa ich časovej hĺbky do minulosti. Autori tiež navrhli prístup, kde môže byť čas potrebný na dokončenie úlohy použitý na meranie efektívnosti potenciálnych metrík na odhadovanie expertízy.

Zmeny. Údaje uložené v repozitároch na verziovanie zdrojového kódu (angl. Version Control Systems) poskytujú okrem zdrojového kódu aj záznamy odovzdaní (angl. commit logs). Tieto informácie môžeme jednak priamo použiť napr. na identifikáciu systematických zmien (súbory A a B sú spravidla menené spolu v odovzdaní) a jednak nepriamo na určenie, ktorí vývojári a do akej miery poznajú zdrojový kód.

Existujúce prístupy [7,8,9] vychádzajú z pravidla „Riadok 10“ (angl. Line 10 Rule), ktoré stanovuje, že osoba, ktorá vykoná zmeny v zdrojovom kóde má odbornú znalosť o tomto kóde. Tieto prístupy reprezentujú expertízu vývojára ako monotónnu rastúcu funkciu. Teda akceptujeme predpoklad, že vývojár, ktorý kompletne nahradí implementáciu existujúcej metódy nemá vplyv na odbornú znalosť vývojára, ktorý je auto-

[5] Mylyn: http://www.eclipse.org/mylyn

rom pôvodnej implementácie. V skutočnosti by sa však mala odborná znalosť programátora pre daný zdrojový kód (element) zvyšovať pri vykonávaní zmien a znižovať pri vykonávaní zmien inými programátormi.

Odporúčač odborných znalostí (angl. Expertise Recommender) navrhnutý v [7] uvažuje znalosť vývojára pre zdrojový kód ako binárnu funkciu, pričom len jeden vývojár v čase je odborníkom pre fragment zdrojového kódu, t.j. vývojár, ktorý vykonal poslednú zmenu. Vyhľadávač odborných znalostí (angl. Expertise Locator) navrhnutý v [8] vylepšuje prehliadač odborných znalostí (angl. Expertise Browser) [9] uvažovaním vzájomných vzťahov medzi zdrojovými kódmi (komponentmi) a frekvencie zmien, ktoré boli uskutočnené súčasne.

Záznamy o chybách. Záznamy o chybách (angl. bug/issue reports) poskytujú informácie o neočakávanom správaní, alebo o chybách v zdrojovom kóde s ohľadom na softvérový projekt. Dané záznamy sú spravidla v štruktúrovanej forme a obsahujú napr. typ chyby, verziu, odosielateľa/adresáta reportu, prioritu, opis.

Nagwani a Verma [10] navrhli prístup na objavovanie a prideľovanie expertov na vyriešenie novo-vytvorených úloh zacielených na odstránenie pojednávajúcich chýb. Navrhnutý prístup je rozdelený na dve časti. Prvá časť rieši identifikáciu vhodných vývojárov na vyriešenie úlohy, druhá časť odhaduje expertízu vývojárov. Vzhľadom na to, že všetky potrebné informácie v záznamoch o chybách sú v textovej forme, autori používajú známe techniky z vyhľadávania informácií (TF/IDF, podobnosť). Cieľom je odporučiť expertov, ktorí majú skúsenosti s riešením podobných úloh, resp. úloh, v ktorých sa vyskytovali podobné chyby ako v nových/nevyriešených úlohách.

Wu a kol. [15] navrhli prístup s názvom DREX (angl. Developer Recommendation with k-nearest-neighbor search and EXpertise ranking). Cieľom je podobne ako v predošlom prístupe odporúčanie expertov pre nové úlohy. Metóda je založená na algoritme k-najbližších susedov (angl. k-nearest neighbors algorithm), pričom vstupom je podobnosť úloh a metriky na hodnotenie expertízy v sociálnych sieťach.

Skúsenosti s technológiami. Aplikačné programovacie rozhranie (API) reprezentuje štruktúrovaný informačný priestor, ktorý musia vývojári dobre poznať, rozumieť mu a byť schopní sa v ňom navigovať pri riešení úloh. Tento priestor je zvyčajne prepojený s dokumentáciou, manuálmi a príkladmi v podobe zdrojových kódov. Použitie knižníc tzv. tretích strán je dnes pri tvorbe softvéru prakticky nevyhnutné. Pri úlohách, ktoré sú cielené na údržbu systému, objavenie/identifikácia vhodného experta môže výrazne urýchliť a zjednodušiť ich dokončenie.

Expert Finder [6] a Expert Recommender [14] umožňujú identifikovať expertov na API vytvorené v programovacom jazyku Java. Prístup Teytona a kol. [13] je založený na myšlienke, že expertíza vývojára pre dané API je ovplyvnená počtom zmien vykonaných v zdrojovom kóde, ktorý používa cieľové API.

Napriek mnohým prácam, ktoré sa venujú odhadu expertízy vývojára, stále nedokážeme jednoznačne povedať, ktoré metódy (metriky) najspoľahlivejšie zachytávajú/reflektujú expertízu vývojára. Jedným z hlavných problémov je nedostatok údajov (angl. a clear baseline), na ktorých dané metriky vyhodnocovať a navzájom porovnávať. Ďalším problémom je veľké množstvo „konkurenčných" definícií pre expertízu vývojára. Pri odhadovaní expertízy je možné uvažovať množstvo vplývajúcich aspektov,

napr. znalosť programovacích jazykov/technológií/knižníc, zručnosti aplikovania návrhových vzorov, úroveň testovania, miera chybovosti zdrojového kódu, miera oboznámenia sa s dokumentáciou. Kvôli týmto a mnohým iným aspektom, spomenuté scenáre použitia odhadu expertízy vývojára si vyžadujú rôzne definície, prístupy a metriky.

V súčasnosti sa výskum čoraz viac koncentruje na použitie histórie interakcií ako nový/podporný zdroj na odhadovanie expertízy vývojára. Našim cieľom je návrh metódy na odhadovanie expertízy vývojára, ktorá odzrkadľuje kvalitu jeho práce. V porovnaní s existujúcimi prístupmi sa zameriavame na podrobnejšie zachytenie a využitie interakčných údajov na úrovni vzorov (odhaľovanie sledu vývojových aktivít vedúcich k chybe), a okrem extrahovania štandardných softvérových metrík, sa zameriavame tiež na využitie analýzy zdrojového kódu na odhaľovanie „slabých miest".

3 Odhad expertízy: kvalita práce vývojára

Našim cieľom je použiť existujúcu infraštruktúru v projekte PerConIK [3] a aktívne zbieranie údajov v softvérovej firme na návrh a realizáciu metódy na odhad expertízy vývojára, resp. kvality jeho práce určenej z vývojových aktivít a vytvoreného kódu. V súčasnosti zbierame rôzne aktivity vývojára vo vývojovom prostredí a webovom prehliadači, ako napr. vyhľadávanie v zdrojovom kóde/na Webe, refaktorovanie zdrojového kódu, ladenie, kopírovanie a vkladanie v zdrojovom kóde, navigáciu v zdrojovom kóde - prepínanie sa medzi súbormi, označovanie zdrojového kódu.

Našim zámerom je použiť tieto aktivity na odhaľovanie vzorov vedúcich k vzniku chyby (napr. sledy aktivít nad zdrojovým kódom v tvare Select+Find+Edit, Find+Copy+Paste). Uvedené plánujeme realizovať metódami strojového učenia - klasifikáciou – kde vstupom sú rôzne kombinácie vývojových aktivít a výstupom je typ chyby (napr. Bad practice, Smell, Security, Performance), a tiež závažnosť chyby.

Informáciu o chybe v zdrojovom kóde získavame jednak značkovaním zdrojového kódu zo systému na sledovanie a evidenciu chýb v zdrojovom kóde, a tiež analýzou zdrojového kódu na odhaľovanie slabých miest (napr. použitím nástroja FindBugs[6]).

Plánujeme tiež použiť štandardné softvérové metriky zo zdrojového kódu. Tie je možné rozdeliť na metriky návrhu (angl. design metrics, napr. Halsteadova zložitosť), metriky kódu (angl. code metrics, napr. cyklomatická zložitosť) a ostatné metriky (napr. globálna hustota údajov). Podobne ako v prípade aktivít, použijeme metódy strojového učenia na klasifikáciu chyby, pričom vstupom sú hodnoty extrahované zo štandardných softvérových metrík a výstupom je typ a závažnosť chyby.

Pri experimentoch sa zameriame jednak na porovnanie úspešnosti viacerých prístupov strojového učenia (Naivebayes, Random Forest, Logistic regression), a tiež na rôzne kombinácie metrík založených na aktivitách a na štandardných softvérových metrikách. Očakávame, že pomocou našej metódy dokážeme odhaliť sledy aktivít vývojára a štandardné softvérové metriky, ktoré vedú k chybe v zdrojovom kóde s určením typu chyby a jej závažnosti. Natrénovaný model použijeme na odhad expertízy vývojára,

[6] FindBugs: http://findbugs.sourceforge.net/

a usporiadanie vývojárov v softvérovom projekte, pričom expertíza odzrkadľuje kvalitu práce vývojára - relatívne k ostatným vývojárom participujúcich v projekte.

Poďakovanie. Tento článok vznikol vďaka podpore v rámci OP Výskum a vývoj pre projekt: Výskum metód získavania, analýzy a personalizovaného poskytovania informácií a znalostí, ITMS: 26240220039, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

Literatúra

1. Fritz, T., et al.: A degree-of-knowledge model to capture source code familiarity. In: Proc. of the 32nd Int. Conf. on Softw. Eng. - Vol. 1. USA, ACM, 2010, pp. 385–394.
2. Girba, T., et al.: How developers drive software evolution. In: Proc. of the 8th Int. Workshop on Principles of Softw. Evolution. SA, IEEE CS, 2005, pp. 113–122.
3. Bieliková, M., Polášek, I., Barla, M., Kuric, E., Rástočný, K., Tvarožek, J., Lacko, P.: Platform independent software development monitoring: design of an architecture. In: Proc. of the 40th Int. Conf. on Current Trends in Theory and Practice of Computer Science. Slovakia, Springer LNCS, 2014, pp. 126-137.
4. Kuric, E., Bieliková, M.: Estimation of Student's Programming Expertise. In: Proc. of the 8th Int. Symposium on Empirical Softw. Eng. and Measurement. Italy, ACM, 2014. p. 4.
5. Kuric, E., Bieliková, M.: Webification of software development: user feedback for developer's modeling. In: Proc. of the 14th Int. Conf. on Web Engineering. France, Springer LNCS, 2014, pp. 550-553.
6. Ma, D.: Expert recommendation with usage expertise. In: Proc. of the 25th Int. Conf. on Softw. Maintenance. Canada, IEEE, 2009, pp. 535–538.
7. McDonald, D. W., Ackerman, M. S.: Expertise recommender: a flexible recommendation system and architecture. In: Proc. of the Conf. on Computer Supported Cooperative Work. USA, ACM, 2000, pp. 231–240.
8. Minto, S., Murphy, G. C.: Recommending emergent teams. In: Proc. of the 4th Int. Workshop on Mining Softw. Repositories. USA, IEEE CS, 2007, p. 5.
9. Mockus, A., Herbsleb, J. D.: Expertise browser: a quantitative approach to identifying expertise. In: Proc. of the 24th Int. Conf. on Softw. Eng. USA, ACM, 2002, pp. 503–512.
10. Nagwani, N., Verma, S.: Predicting expert developers for newly reported bugs using frequent terms similarities of bug attributes. In: Proc. of the 9th Int. Conf. on ICT and Knowledge Engineering, Bangkok, IEEE, 2012, pp. 113–117.
11. Robbes, R., Röthlisberger, D.: Using developer interaction data to compare expertise metrics. In: Proc. of the 10th Working Conf. on Mining Softw. Repositories, USA, IEEE Press, 2013, pp. 297–300.
12. Robillard, M., et al.: Recommendation Systems in Softw. Engineering. Springer Berlin Heidelberg, 2014.
13. Teyton, C.: Find your library experts. In: Proc. of the 20th Working Conf. on Reverse Engineering. Germany, IEEE, 2013, pp. 202–211.
14. Vivacqua, A., Lieberman, H.: Agents to assist in finding help. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, USA, ACM, 2000, pp. 65–72.
15. Wu, W.,: Drex: Developer recommendation with k-nearest-neighbor search and expertise ranking. In: Proc. of the 18th Asia Pacific Softw. Engineering Conf. Ho Chi Minh, IEEE, 2011, pp. 389–396.

Zabezpečenie udržateľnosti komunít v CQA systémoch orientáciou na odpovedajúcich používateľov

Ivan Srba a Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita,
Ilkovičova 2, 842 16, Bratislava, Slovensko
{meno.priezvisko}@stuba.sk

Abstrakt. Odpovedanie otázok v komunitách (angl. Community Question Answering – CQA) predstavuje efektívny spôsob, ako zdieľať znalostí v online komunitách. Súčasné CQA systémy charakterizuje stúpajúca miera diverzity ako v úrovni znalostí používateľov, množstva ich aktivít, tak aj v kvalite nimi vytváraného obsahu. Jedným z negatívnych dôsledkov tejto diverzity je narastajúci počet používateľov, ktorí produkujú veľké množstvo nekvalitného obsahu. Na základe analýzy profilov správania týchto používateľov poukazujeme na potrebu adaptívnych prístupov pre podporu spolupráce, ktoré vedú k zapojeniu celej komunity ako prostriedku pre jej dlhodobú udržateľnosť. Existujúca podpora odpovedania na otázky v CQA systémoch sa však primárne zameriava na pýtajúcich sa používateľov, čo vedie k zapájaniu len časti z celej komunity (väčšinou expertov). Navrhujeme preto nový koncept prístupov založených na pohľade odpovedajúcich používateľov, ktorý ilustrujeme na dvoch konkrétnych príkladoch.

Kľúčové slová: CQA, zdieľanie znalostí, adaptívna podpora, online komunity

1 Zdieľanie znalostí v online komunitách

Online komunity predstavujú v súčasnosti najrozšírenejšiu formu kolektívnej inteligencie, v rámci ktorej členovia týchto komunít zdieľajú enormné množstvo znalostí. Najznámejšie príklady online komunít môžeme nájsť v systémoch, ako sú napríklad wiki stránky (predovšetkým Wikipedia), stránky sociálnych sietí a v poslednej dobe aj systémy pre odpovedanie otázok v komunitách (angl. Community Question Answering – CQA), ako je napr. Yahoo! Answers alebo Stack Overflow.

CQA systémy predstavujú priestor, kde sa ľudia môžu pýtať najrozličnejšie otázky, na ktoré nenašli odpoveď pomocou štandardných vyhľadávacích nástrojov. Dôsledkom otvorenosti CQA systémov je preto vysoká miera diverzity, ktorá sa prejavuje ako v expertíze používateľov, v množstve a type nimi vykonaných aktivít, tak aj v samotnom obsahu. Táto diverzita na jednej strane umožňuje zdieľanie znalostí medzi ľuďmi s rôznou úrovňou expertízy, na druhej strane však prispieva k narastajúcemu počtu používateľov, ktorí vytvárajú veľké množstvo nekvalitného obsahu (predovšetkým pýtaním sa duplicitných alebo triviálnych otázok, ale aj poskytovaním veľkého množstva odpovedí s cieľom získať reputáciu). Tento problém je možné eliminovať prostredníctvom

adaptívnej podpory, ktorej cieľom je prispôsobovať CQA systém pre efektívne zdieľanie znalostí v komunite (napr. personalizovaným odporúčaním otázok). Existujúce prístupy k adaptívnej podpore sú však v prevažnej miere orientované na používateľov, ktorí sa pýtajú a prispievajú tak k tomu, že len úzka časť celej komunity (napr. experti, ktorí dokážu poskytnúť najkvalitnejšie odpovede) sa aktívne zapája do procesu odpovedania na otázky. Pre zabezpečenie dlhodobej udržateľnosti takéhoto ekosystému je však potrebné zapojiť do tohto procesu čo najväčšiu časť komunity [7].

Na základe podrobnej štúdie existujúcich prístupov, prípadovej štúdie správania sa používateľov v jednom z najväčších CQA systémov Stack Overflow s využitím dostupných dát aktivity v tomto systéme a tiež aj v kontexte skúseností s návrhom a používaním CQA systému Askalot v doméne vzdelávania vytvoreného na našej univerzite [6] poukazujeme na potrebu rozvíjať také prístupy, ktoré zohľadňujú zapojenie celej komunity a uvažujú adaptívnu podporu zdieľania znalostí z pohľadu odpovedajúcich používateľov (angl. answerer-oriented approaches). Tento nový koncept ilustrujeme na odporúčaní otázok používateľom podľa ich preferencii a na zvyšovaní kvality obsahu prostredníctvom minimalizácie duplicitných a podobných otázok.

2 Profily správania používateľov v kontexte CQA systémov

Diverzita používateľov v CQA komunitách sa stala predmetom viacerých štúdií s cieľom charakterizovať používateľov na základe ich profilov správania. Pre opis rôznych typov používateľov bolo pri tom použitých hneď niekoľko kategorizácií.

Predovšetkým je možné rozdeliť používateľov na tých, ktorí sa pýtajú a tých, ktorí naopak poskytujú odpovede. Prekryv medzi týmito skupinami sa líši v závislosti od konkrétneho CQA systému od 5,4% v systéme Naver Knowledge-iN [4] až po 21,4% v systéme Stack Overflow [3]. Autori v [1] rozšírili túto kategorizáciu o tretiu skupinu používateľov, ktorí prispievajú svojimi znalosťami prevažne formou diskusie.

V druhom pohľade je možné v CQA systémoch rozdeliť používateľom podľa miery ich aktivity na aktívnych používateľov (1% najaktívnejších používateľov poskytuje v priemere až 25% všetkých odpovedí [3], [4]) a pasívnych používateľov, ktorí využívajú znalosti v archíve otázok, ale aktívne do neho neprispievajú (angl. lurkers).

Nakoniec v treťom pohľade rozlišujeme používateľov podľa úrovne ich expertízy, ktorá priamo odzrkadľuje aj kvalitu nimi vytváraného obsahu. Pre identifikáciu expertov v rámci komunity bolo navrhnutých niekoľko metód, ktoré vychádzajú buď z algoritmov pre meranie centrality v grafe prepojených dokumentov [2] (napr. PageRank a HITS), alebo z klasifikácie na základe vlastností opisujúcich predchádzajúcu aktivitu používateľa (napr. percento poskytnutých odpovedí označených ako najlepších).

Keďže tieto tri základné kategorizácie sú navzájom paralelné, je možné ich navzájom kombinovať. Pre efektívne zdieľanie znalostí je nevyhnutné, aby sa v CQA komunite nachádzali niektoré špecifické typy používateľov (napr. aktívne odpovedajúci používatelia s vysokou mierou expertízy). V poslednej dobe sme však v CQA systémoch svedkami problémov, ktorých vznik úzko súvisí so zväčšujúcim sa počtom nežiaducich typov používateľov (napr. používatelia, ktorí sa pýtajú veľké množstvo nekvalitných otázok). Nárast počtu takýchto používateľov je ovplyvnený okrem iných faktorov aj

narastajúcou popularitou, a teda diverzitou používateľov CQA systémov. Dôsledkom tohto trendu je, že experti strácajú motiváciu zdieľať svoje znalostí, čo v konečnom dôsledku môže prispieť k ohrozeniu dlhodobej udržateľnosti celej komunity.

2.1 Prípadová štúdia v systéme Stack Overflow

S cieľom presnejšie opísať narastajúce problémy v CQA komunitách sme vykonali prípadovú štúdiu nad systémom Stack Overflow, ktorý je zameraný na riešenie otázok súvisiacich s informačnými technológiami. Systém Stack Overflow predstavuje jeden z najúspešnejších príkladov online komunity, v dôsledku čoho sa stal predmetom viacerých štúdií. Žiadna z nich však neanalyzovala problematiku udržateľnosti komunity v dôsledku narastajúceho počtu nežiaducich používateľov.

V prípadovej štúdii sme použili kvalitatívny prístup prostredníctvom analýzy otázok v časti Meta Stack Overflow (špecifická časť CQA systému Stack Overflow venovaná otázkam ohľadom fungovania systému samotného). V priebehu roku 2014 tu je možné sledovať rastúci trend otázok, ktoré poukazujú na negatívny vývoj komunity (význam tohto problému zdôrazňuje aj fakt, že sa ním zoberá aj položená otázka[1] ohľadom nárastu negatívnych aktivít a aj celkového negatívneho pocitu s výrazne najintenzívnejšou spätnou väzbou od komunity). Zároveň sme v štúdii využili kvantitatívny prístup formou analýzy nad dátovou sadou[2] zo systému Stack Overflow.

Výsledkom našej analýzy je identifikácia niekoľkých typov používateľov, ktoré v súčasnosti predstavujú najväčší problém analyzovanej komunity:

1. Skupina používateľov, ktorí vytvárajú veľké množstvo otázok bez snahy získať požadovanú znalosť z iných štandardných zdrojov (napr. pomocou vyhľadávacích nástrojov), pričom ich nezaujíma nič, len rýchle zodpovedanie svojej otázky (tiež označovaní ako *help vampires*). Komunite neprinášajú žiadnu pridanú hodnotu, ale naopak produkujú veľké množstvo duplicitného obsahu.
2. Druhou skupinou sú používatelia s veľmi nízkou úrovňou expertízy, ktorí vytvárajú triviálne a nekvalitné otázky (angl. *noobs*). Zahlcujú systém nekvalitným obsahom, ktorý nie je zaujímavý pre zvyšok komunity (kvalitu otázok je možné odvodiť na základe spätnej väzby poskytnutej komunitou, napr. z počtu pozitívnych hlasov).
3. Ako dôsledok vznikajúceho veľkého množstva nekvalitných otázok od predchádzajúcich dvoch skupín, sa začala formovať skupina používateľov, ktorí odpovedajú na tieto otázky s cieľom získať čo najvyššiu reputáciu (tzv. zberači reputácie). Táto skupina používateľov síce prispieva do systému (napr. tým, že odbremeňuje expertov), zároveň však svojím správaním podporuje vznik ďalších nekvalitných otázok.
4. Štvrtou skupinou sú používatelia, ktorí reagujú na pribúdajúci nekvalitný obsah pomocou veľkého množstva negatívnych hlasov bez dodatočného vysvetlenia, prečo nie je daná otázka nevhodná alebo nesprávne položená (angl. *haters*). Rovnaké správanie je možné sledovať aj pri moderátoroch, ktorí zodpovedajú za správu komunitného obsahu (kvôli čomu sú často označovaní aj ako *StackOverlords*).

[1] http://meta.stackoverflow.com/questions/251758/why-is-stack-overflow-so-negative-of-late/

[2] http://blog.stackexchange. com/category/cc-wiki-dump/

S výnimkou štvrtej skupiny používateľov (ktorú je možné regulovať zmenou pravidiel), je možné ostatné problémy riešiť prostredníctvom vhodnej adaptívnej podpory spolupráce, ktorá vedie k zapojeniu celej komunity do procesu odpovedania na otázky, pričom podporí žiaduce typy používateľov a zároveň eliminujú tie nežiaduce.

3 Adaptívne prístupy pre podporu zapojenia celej komunity

V doméne CQA systémov bolo navrhnutých niekoľko adaptívnych prístupov, ktorých cieľom je podporiť efektívne zdieľanie znalostí. Existujúce metódy sa však primárne zameriavajú na pýtajúcich používateľov. Ako príklad môžeme uviezť tzv. *smerovanie otázok* (angl. question routing), ktoré predstavuje odporúčanie nových otázok používateľom, ktorí by potenciálne mohli poznať odpoveď na danú otázku. Väčšina existujúcich prístupov sa zameriava na odporúčanie používateľom s najvyššou úrovňou znalostí (expertom) [7], a to bez ohľadu na to, akú úroveň znalostí vyžaduje zodpovedanie danej otázky. Dôsledkom takého prístupu je, že len malá časť celej komunity sa aktívne zapája do procesu odpovedania na otázky. Pre zabezpečenie trvalej udržateľnosti CQA komunít je však potrebné navrhovať také prístupy, ktoré uvažujú zapojenie celej komunity [7]. V našej práci sa to snažíme dosiahnuť prostredníctvom adaptívnych metód, ktoré zohľadňujú primárne pohľad odpovedajúcich používateľov (tzv. answerer-oriented approaches). V nasledujúcom texte predstavíme príklady dvoch takýchto metód.

Smerovanie otázok založené na diverzifikácii odporúčaní. Smerovanie otázok predstavuje personalizované odporúčanie, ktoré dokáže významne ovplyvniť proces odpovedania na otázky. Z tohto dôvodu má aj najväčší potenciál pomôcť pri riešení problémov súvisiacich s nežiaducimi skupinami používateľov.

Spomínané problémy môže zmierniť také smerovanie otázok k používateľom, ktoré zohľadňuje ich úroveň expertízy a preferenciu zložitosti otázok. To znamenaná, že pokiaľ je do systému vložená nová triviálna otázka, tak ju neodporučíme expertom. Preferujeme používateľov, ktorí majú nižšiu úroveň expertízy, ale zároveň dostatočnú na to, aby na danú otázku vedeli správne odpovedať. Navyše odporúčanie je možné vhodným spôsobom diverzifikovať tak, aby mali používatelia možnosť odpovedať aj na otázky mimo ich primárnej oblasti záujmu (na vhodnej úrovni expertízy) a eventuálne tak získať nové znalosti. Pre odporúčanie otázok aj pasívnym používateľom, ktorí aktívne do CQA systému neprispievajú, môžu vhodne poslúžiť dáta, ktoré sú dostupné o týchto používateľoch mimo samotného CQA systému (napr. blogy a sociálne siete).

Identifikovanie podobných a duplicitných otázok. Okrem personalizovanej podpory formou smerovania otázok je možné podporiť odpovedanie na otázky aj obmedzovaním vzniku podobných alebo duplicitných otázok už v čase ich vytvárania. Používatelia, ktorí majú záujem o zdieľanie znalostí, tak nemusia hľadať nové zaujímavé otázky vo veľkom množstve takých, pre ktoré už v systéme existuje odpoveď. Pri hľadaní podobných a duplicitných otázok je možné zohľadniť nielen samotný obsah (nadpis a text otázky), ale aj jej kontext, a to konkrétne históriu používateľa, ktorý sa danú otázku pýta. To umožňuje obohatiť tému otázky (identifikovanú napr. pomocou metódy LDA) o záujmy používateľa, ktorý ju položil. Predpokladáme, že vo väčšine prípadov nová otázka súvisí s témami, o ktoré sa daný používateľ pred tým aktívne zaujímal.

4 Záver a ďalšia práca

V súčasnosti sú CQA systémy považované za úspešný príklad kolektívnej inteligencie na webe. Ich mnohé pozitívne výsledky (množstvo zodpovedaných otázok, krátky čas do získania prvej odpovede, znalosti obsiahnuté v archívoch vyriešených otázok) významne prevyšujú problémy, ktorých dôsledky však v poslednom období signifikantne narastajú. Prostredníctvom prípadovej štúdie nad CQA systémom Stack Overflow sme identifikovali štyri skupiny používateľov, ktoré majú negatívny vplyv na efektívne zdieľanie znalostí. Ako riešenie týchto problémov navrhujeme zapojiť do podpory spolupráce v CQA systémoch také adaptívne prístupy, ktoré budú uvažovať zapojenie celej komunity ľudí a nielen úzkej skupiny expertov s vysokou úrovňou znalostí ako to môžeme pozorovať v súčasných systémoch. Existujúce prístupy sa orientujú primárne na pýtajúcich sa používateľov, pričom len minimálne zohľadňujú preferencie a očakávania používateľov, ktorí poskytujú odpovede. Prispievajú tým k tomu, že len malá časť komunity sa aktívne zapája do odpovedania na otázky. Predstavili sme preto nový koncept metód orientovaných na odpovedajúcich používateľov, ktorých cieľom je podporiť dlhodobú udržateľnosť CQA komunity. V ďalšej práci sa plánujeme zamerať na návrh a realizáciu metódy smerovania otázok, ktorá explicitne zohľadní tento koncept. Úspešnosť metódy overíme pomocou dátovej sady zo systému Stack Overflow a zároveň v komunite študentov a ich učiteľov v CQA systéme Askalot v prostredí našej fakulty.

Poďakovanie. Táto publikácia vznikla vďaka čiastočnej podpore projektov VG1/0675/11 a Kultúrna a edukačná grantová agentúra KEGA 009STU-4/2014.

Literatúra

1. Adamic, L.A., Zhang, J., Bakshy, E., Ackerman, M.S.: Knowledge sharing and yahoo answers. Proc. of the 17th int. conference on World Wide Web - WWW ’08. pp. 665–674. ACM Press, New York, USA (2008).
2. Aslay, Ç., O’Hare, N., Aiello, L.M., Jaimes, A.: Competition-based networks for expert finding. Proc. of the 36th int. ACM SIGIR conf. on Research and development in inf. retrieval - SIGIR ’13. pp. 1033–1036. ACM Press, New York, USA (2013).
3. Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., Hartmann, B.: Design Lessons from the Fastest Q&A Site in the West. Proc. of the 2011 annual conference on Human factors in computing systems - CHI ’11. pp. 2857–2866. ACM Press, New York, USA (2011).
4. Nam, K.K., Ackerman, M.S., Adamic, L.A.: Questions in, knowledge in? A Study of Naver’s Question Answering Community. Proc. of the 27th int. conference on Human factors in computing systems - CHI 09. pp. 779–788. ACM Press, New York, USA (2009).
5. Srba, I., Bieliková, M.: Adaptive Support for Educational Question Answering. In: Maillet, K. and Klobučar, T. (eds.) Proc. of the Doctoral Consortium at the European Conference on Technology Enhanced Learning 2013. pp. 109–114. CEUR, Paphos, Cyprus (2013).
6. Srba, I., Bieliková, M.: Askalot: odpovedanie otázok v komunite študentov a ich učiteľov. Proc. of Znalosti 2014. pp. 92-95 (2014).
7. Szpektor, I., Maarek, Y., Pelleg, D.: When Relevance is not Enough: Promoting Diversity and Freshness in Personalized Question Recommendation. Proc. of the 22nd int. conference on World Wide Web. pp. 1249–1259 (2013).

Prirodzený jazyk ako spôsob komunikácie v prostredí webu

Peter Macko

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 2, 842 16 Bratislava, Slovensko
peter.macko@stuba.sk

Abstrakt. Vyhľadávanie informácií na webe je aj v dnešnej informatizovanej dobe problémom, ktorý rieši množstvo výskumníkov. Dnešné prístupy sa najčastejšie sústredia na vyhľadávanie pomocou kľúčových slov, ktoré doplňujú kategorizáciu a v lepšom prípade fazety. To však pre komplikované otázky stále nie je dostatočné. Práve preto sa v našej práci zameriavame na spôsob vyhľadávania informácií pomocou prirodzeného jazyka používateľa. Tento jazyk je pre každého človeka prirodzený a prináša veľkú vyjadrovaciu voľnosť. Na to, aby sme však vedeli používať takýto jazyk na získavanie informácií, musíme byť schopní spracovať komplikované vety a mať dostatočne dobrú databázu, ktorú nám ponúka sémantický web.

Kľúčové slová: Sémantický web, SPARQL, dopyty, Liked Data, prepojené dáta, RDF, vyhľadávanie, neurónové siete

1 Prepojené dáta a ich využitie

Dnešní používatelia webu sú naučení na vyhľadávanie používať kľúčové slová. Tento spôsob si osvojili preto, že dnes používané prehliadače ich k tomu nútia. Vyhľadávače sa tak neprispôsobujú ich používateľom, ale používatelia sa prispôsobujú vyhľadávačom. A prečo tieto vyhľadávače používajú práve kľúčové slová? Je to z dôvodu štruktúry dnešného webu. Keďže absolútna väčšina dnes existujúcich stránok používa relačné databázy[1], nemajú vyhľadávače veľké možnosti, ako zistiť o ich obsahu viac. Jediné, čo týmto vyhľadávacím strojom ostáva, je postupné preliezanie stránok a hľadanie obsahu a následná indexácia týchto dát. Tieto metódy však neumožňujú vyhľadávačom zmapovať entity zaznamenané na stránkach a ich vzájomné väzby.

Dnes však k slovu prichádzajú nové spôsoby ukladania dát a medzi nimi aj sémantický web. Sémantický web prináša do ukladania dát nový rozmer a tým je prepojenosť. Vďaka tomu, že sémantický web je založený na prepojených entitách, môžu autori webových stránok tieto entity znova používať a týmto spôsobom

[1] http://db-engines.com/en/ranking

obohacovať nielen svoje, ale hlavne globálne dátové sady. Používateľ, ktorý sa bude snažiť niečo dozvedieť, napríklad o aute menom *Jaguár,* bude jednoznačne vedieť určiť, že má záujem o informácie o tejto entite a vyhľadávač mu nebude zobrazovať výsledky týkajúce sa zvieraťa menom jaguár.

2 Existujúce prístupy v oblasti vyhľadávania informácií

Ako sme spomenuli, dnes najčastejším spôsobom, ako vyhľadávať dáta, je forma kľúčových slov. Tento prístup však nestačí nielen nám, ale alternatívne spôsoby vyhľadávania riešia viacerí výskumníci.

Prvé metódy, ktoré sa snažili o podporenie používateľov pri vyhľadávaní náročných dopytov, sa datujú už do roku 1972, kedy vedci okolo projektu Apollo 11 vytvorili vyhľadávač pre dáta nazbierané na mesiaci [13]. Spôsob vyhľadávania v tomto riešení bol založený na analýze vety napísanej používateľom, pričom bol použitý efektívny bezkontextový parsovací algoritmus (angl. *efficient context-free parsing algorithm*) [1].

Ďalším odlišným spôsobom tvorby zložitých otázok pre vyhľadávanie je menu navigácia[5, 9]. Používateľ pri tomto princípe využíva menu na výber pokračovania otázky. Menu sa tvorí na základe predchádzajúceho slova a šablóny vety, ktorá je vybraná.

S príchodom sémantického webu prišlo k rozšíreniu výskumu v tejto oblasti, hlavne vďaka tomu, že dáta v takejto podobe o sebe vedia veľa povedať. Vznikli tu riešenia, ktoré vedia odpovedať na zjednodušené SQL dopyty [1, 5], riešenia, ktoré vedia pracovať s pseudo-prirodzeným jazykom[10, 12] a konvertovať ho na jazyk SPARQL[2]. Najkomplexnejším zástupcom tejto skupiny riešení je PANTO [12]. Toto riešenie pracuje so slovníkom WordNet[3], ktorého synonymá využíva na riešenie slovníkového problému. Okrem toho na analýzu vety využíva StandfordParser[4], vďaka ktorému riešenie pozná štruktúru vety.

Veľký prínos v tejto oblasti má riešenie spoločnosti IBM pod označením Watson [4]. Toto riešenie je založené na vyhľadávaní dát, na základe otázky, v neštruktúrovaných dokumentoch. Pričom pomocou následnej analýzy vybraných dokumentov sú vyhľadané možné odpovede na položenú otázku.

Keďže v tejto oblasti existuje množstvo riešení, ktoré je náročné navzájom porovnať vznikla iniciatíva *Question answering over linked data* [11]. Táto iniciatíva poskytuje dátové množiny aj pripravené otázky, na ktorých je možné nové riešenia testovať.

[2] http://www.w3.org/TR/rdf-sparql-query/
[3] http://wordnet.princeton.edu/
[4] http://nlp.stanford.edu/software/lex-parser.shtml

3 Metóda vyhľadávania pomocou jazyka používateľa

Naša metóda, OntoSearch berie do úvahy dáta a štruktúru, ktorú obsahuje aktuálny dátový zdroj. Preto pozostáva z dvoch fáz:

1. Predspracovanie dátovej množiny
2. Preklad používateľových dopytov

3.1 Predspracovanie ako kľúč k úspechu

Metóda sa nefixuje na konkrétnu dátovú množinu. To znamená, že vie pracovať s rôznymi dátovými množinami, ktoré majú rôznu štruktúru a obsah. Na to, aby sme takéto správanie dosiahli, vytvorili sme fázu predspracovania. V tejto fáze sa tvoria dva lexikóny:

1. Lexikón tried a vlastností,
2. Lexikón hodnôt.

Lexikón tried a vlastností vzniká zo štruktúry daného úložiska a následne je rozšírený o výrazy z databázy WordNet. Každému slovu v databáze je priradená váha akou sa viaže s daným pojmom v databáze. Napríklad slovo *auto* je synonymom slova *automobil*, ktoré sa vyskytuje v sade, a preto má vysokú váhu. V prípade, ak používateľ použije slovo auto, bude toto slovo preložené na objekt automobil.

Vo fáze predspracovania je ešte vytváraný druhý lexikón, a to lexikón hodnôt, ktorý obsahuje hodnoty predikátov v databáze. Tie sa získavajú nielen z názvu entity, ale takisto z často využívaných parametrov, ako je napríklad *rdfs:label*. Ak by sme mali v dátovej sade entitu *Image_12342,* pri vyhľadávaní by nám takéto niečo veľmi nepomohlo. Keď však do lexikónu zoberieme aj hodnotu jeho predikátu *rdfs:label*, ktorý obsahuje používateľovi prívetivejší text, *žlté auto*, tak náš vyhľadávač bude schopný vyhľadať aj takéto výrazy.

3.2 Preklad z jazyka používateľa na jazyk SPARQL

Po tom, ako je ukončená fáza predspracovania, je možné prekladať dopyty. Naša metóda, zobrazená na obrázku 1, pracuje tak, že dostáva na vstupe prirodzený jazyk. Tento následne pomocou lexikónov a neurónovej siete prekladá do jazyka SPARQL.

Po tomto preklade je veta v jazyku lexikónu a ďalšia súčasť ju konvertuje do trojíc. Pre túto konverziu sa používa séria pravidiel, ktoré na základe predchádzajúcej časti vety transformujú vetu na trojice. Takisto sa v tejto fáze spracúvajú aj dodatočné časti dopytu, ako je výber parametrov, ktoré sa vo výsledkoch zobrazujú používateľovi a *where* časť dopytu.

Neurónová sieť na preklad textu

V oblasti prekladu sme sa inšpirovali riešeniami, ktoré sa využívajú v oblasti prekladu dvoch prirodzených jazykov [2, 6, 7]. Riešením vhodným pre našu metódu

je *Vracajúca sa neurónová sieť založená na jazykovom modeli*. Tento model je vyhotovený tak, aby si pamätal, a teda zohľadňuje nielen aktuálne prekladané slovo, ale aj slová, ktoré ho predchádzali.

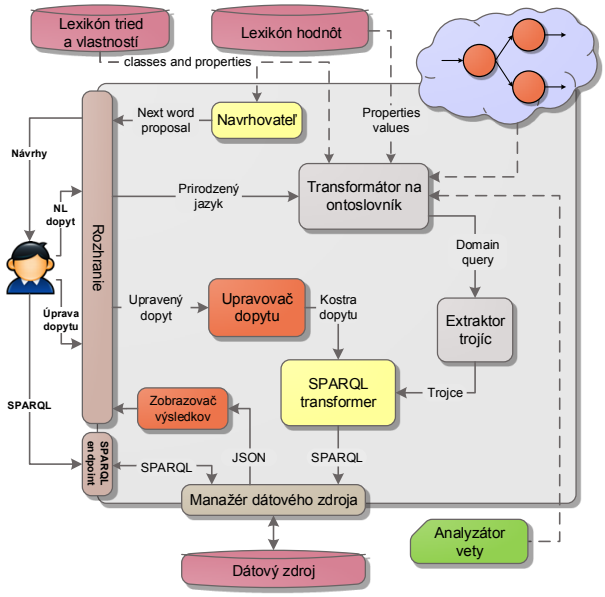


Fig. 1. Schéma fungovania metódy prekladu prirodzeného jazyka

4 Overenie a ďalší plán

Naša metóda je už druhou verziou, do ktorej sme pridali preklad pomocou neurónovej siete. Predchádzajúci model, podľa ktorého sme vytvorili aj prototyp, sme overili v niekoľkých experimentoch. Boli zamerané na presnosť prekladu so SPARQL expertami, ale aj na pohodlnosť písania a špeciálne časti riešenia. Našu metódu sme porovnali s metódou OWLPath. Pri tomto experimente sme zaznamenali zlepšenie oproti porovnávanej metóde hlavne v oblasti rýchlosti písania dopytov.

Nasledujúci plán pre overenie novej metódy je porovnanie s našou predchádzajúcou metódou. Následne chceme zostaviť aj niekoľko ďalších experimentov, v ktorých budeme vyhodnocovať úspešnosť nášho prekladu, pričom ju budú overovať aj SPARQL experti. Okrem iného sa chceme zapojiť do iniciatívy *Question answering over linked data*, vďaka čomu sa budeme vedieť porovnať s viacerými konkurenčnými riešeniami.

Poďakovanie. Tento článok vznikol vďaka čiastočnej podpore projektu VEGA Vranic - Pokročilé metódy v evolúcii softvéru: varianty, kompozícia a integrácia; Advanced Methods in Software Evolution: Variants, Composition, and Integration, the Scientific Grant Agency of the Slovak Republic, grant No. VG 1/1221/12.

Literatúra

1. Atzori, M., Zaniolo, C.: SWiPE: Searching Wikipedia by Example. In: *Proc. of the 21st Int. Conf. Companion on World Wide Web*, ACM Press, Lyon, (2012), pp. 309–312.
2. BENGIO, Y. et al. Neural Probabilistic Language Models. Berlin: Springer Berlin Heidelberg, 2006 roč. CXCIV, s. 137-86. ISBN 978-3-540-30609-2.
3. Earley, J.: An efficient context-free parsing algorithm. In: *Communications of the ACM*, New York USA, ACM Press, (1970), s. 94–102.
4. Ferrucci, D.A., Introduction to „This is Watson“. In: IBM Journal of Research and Development, IBM, (2012), s. 1:1 - 1:15
5. Kasneci, G. et al.: NAGA: Searching and Ranking Knowledge. In: *Proc. of the 2008 IEEE 24th Int. Conf. on Data Engineering*, IEEE Computer Society, (2008), s. 953–962.
6. MIKOLOV, T. et al. Extensions of recurrent neural network language model. Prague: IEEE, 2011, s. 5528 - 5531.
7. MIKLOV, T. *Statistické jazykové modely založené na neurónových sítích*. Brno: Brnenská Technická Univerzita, 2012.
8. Tennant, H. Ross, K. Thompson, C.: Usable Natural Language Interfaces Through Menu-based Natural Language Understanding. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston, Massachusetts, USA, ACM Press, (1983) s. 154–160.
9. Thompson, C., Martin, S.: Using a Menu-based Natural Language Interface to Ask Map- and graph-valued Database Queries. In: *Proceedings of the 1985 ACM Annual Conf. on The Range of Computing: Mid-80's Perspective*, Denver, Colorado, USA, ACM Press (1985), s. 328–338.
10. Valencia-García, R. et al.: OWLPath: An OWL Ontology-Guided Query Editor. In: *Systems, Man and Cybernetics, Part A: Systems and Humans*. IEEE Systems, Man, and Cybernetics Society, (2011), s. 121–136.
11. Vanessa Lopez, Christina Unger, Philipp Cimiano, Enrico Motta: Evaluating question answering over linked data. In *Web Semantics Science Services And Agents On The World Wide Web. Elsevier,* (2013), s. 3-13.
12. Wang, C., Xiong, M., Qi Z., Yong Y.: PANTO: A Portable Natural Language Interface to Ontologies. In: *4TH ESWC, INNSBRUCK*, Innsbruck, Springer-Verlag, (2007), s. 473–487.
13. Woods, W., Kaplan, R, Nash-Webber, B.: The lunar sciences natural language information system: Final report. Bolt Beranek and Newman, (1972).

Pohľad na používateľský zážitok učiaceho sa v integrovaných webových vzdelávacích systémoch

Jozef Tvarožek, Róbert Móro, Martin Labaj, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 2, 842 16 Bratislava, Slovensko
{meno.priezvisko}@stuba.sk

Abstrakt. Počas riešenia príkladov poskytovaných webovými vzdelávacími systémami sa používateľ nielen sústredí na vytvárané riešenie a text problému, ale v súčasných systémoch má zároveň k dispozícii ďalšie personalizované a sociálne nástroje, napríklad vo forme widgetov umiestnených vedľa vzdelávacieho objektu. Okrem príkladov sú typicky poskytované aj výučbové texty (vysvetlenia), otázky a ďalšie typy objektov a medzi nimi sa počas riešenia môže prepínať. V tomto príspevku sa zaoberáme používateľským zážitkom v rámci navzájom integrovaných vzdelávacích systémov, nielen z hľadiska riešenia príkladov, ale zároveň z hľadiska nástrojov a informácií dostupných na rovnakej stránke v okolí vzdelávacieho objektu, dostupných v ostatných častiach systému, ale aj v iných vzdelávacích systémoch, či na otvorenom Webe. Dôležitou súčasťou pre vyhodnocovanie používateľského zážitku je sledovanie pohľadu, ktorý môže zároveň slúžiť ako ďalší zdroj informácií pre modelovanie znalostí používateľa a podobne. V príspevku prezentujeme systém ALEF prepojený so systémom Peoplia a v rámci nich aplikáciu rámca pre používateľský zážitok.

Kľúčové slová: používateľský zážitok (UX), vzdelávacie systémy, pohľad

1 Používateľský zážitok a vzdelávacie systémy

Technologický pokrok a rozšírenie Internetu umožnili rozvoj online vzdelávacích prostredí [2], ktoré namiesto pasívneho čítania a pozerania posúvajú do popredia vzdelávacieho procesu aktívnu interakciu s rôznorodým vzdelávacím obsahom, ako aj sociálnu interakciu s ďalšími študentmi alebo učiteľmi. Učenie programovania obzvlášť vyžaduje aktívnu činnosť – programovanie, riešenie úloh/cvičení. Podporné vzdelávacie systémy, ktoré sa používajú aj na univerzitách, umožňujú nielen zbierať študentské riešenia, ale ich aj automatizovane testovať, vyhodnocovať a poskytnúť spätnú väzbu [5]. V posledných rokoch sa objavujú webové vzdelávacie systémy pre programovanie, ako napr. [4], podporujúce študentov pri písaní a analýze zdrojového kódu, ako aj pri spolupráci.

Ďalšou etapou výskumu je zapojenie podrobných údajov o práci študentov, najmä čo sa týka analýzy úprav, kompilácií a spúšťaní [3], správnosti riešení [10], a dát zo senzorov, napr. pohľadu očí [1]. Výzva spočíva v obrovskom množstve dostupných

údajov s nejasnou interpretáciou, napr. pri pohľade očí programátora, ktoré je navyše často potrebné kombinovať pre dosiahnutie úplného obrazu o práci študenta. Je to spôsobené cibuľovitým nabaľovaním prvkov rozhrania vzdelávacích prostredí, keď okolo zadania programátorskej úlohy a editora zdrojového kódu môžu byť rozmiestnené ďalšie nástroje (často vo forme widgetov). Študent tak môže editovať kód (kurzor myši je v okne editora) a zároveň čítať súvisiaci vzdelávací text umiestnený vo widgete mimo editora. Bez kombinácie údajov z myši, klávesnice a pohybu očí nie sme tento typ interakcie schopní odhaliť.

Informácie z týchto senzorov sú preto kľúčové pri vyhodnocovaní používateľského zážitku študentov pri učení a riešení programátorských úloh. Pod používateľským zážitkom (UX, angl. *user experience*) rozumieme všetky komplexné interakcie študentov s rozhraním vzdelávacieho prostredia vrátane študentmi poskytovanej (priamej i nepriamej) spätnej väzby. Tým, že získame lepší obraz o týchto interakciách, môžeme presnejšie modelovať znalosti študentov [6] a následne im rozhranie prispôsobovať (personalizovať), odporúčať im ďalšie zaujímavé vzdelávacie objekty (vysvetlenia, cvičenia/príklady) alebo poskytovať vhodnú pomoc.

V tomto článku opisujeme bližší pohľad na prácu študenta pri učení a riešení programátorských úloh s cieľom lepšie pochopiť prácu používateľa pri učení. Zahŕňame rôzne indikátory, od sledovania pohľadu používateľa na samotný editor riešenej úlohy, až po prepínanie kariet prehliadača. V druhej časti tohto článku opisujeme vrstvenú štruktúru webového používateľského rozhrania prepojených vzdelávacích systémov a doterajšie sledované indikátory. V tretej časti opisujeme infraštruktúru sledovania pohľadu vo webovom vzdelávaní a jej využitie.

2 Webové rozhranie v prepojenom vzdelávaní

Vzdelávacie systémy ALEF[1] a Peoplia[2] sú vytvorené a používané na Fakulte informatiky a informačných technológií pri výučbe. Systém ALEF poskytuje obsah vo forme vzdelávacích objektov doplnený o kolaboratívne, anotačné, sociálne a ďalšie funkcie. Systém Peoplia umožňuje riešenie a vyhodnocovanie programátorských úloh s príslušnými funkciami prehľadov pre učiteľa, motivačných rebríčkov a odznakov pre študentov a ďalších sociálnych funkcií.

Tieto systémy sú z pohľadu študenta prepojené najmä používateľským rozhraním [7] – študent pracuje v systéme ALEF, zvolí si vzdelávací objekt (typu *vysvetlenie*, *cvičenie*/*príklad*, *otázka*) a pracuje s jeho obsahom. *Príklady*, ktoré sú okrem formy samostatných objektov vložené na vhodných miestach vo *vysvetleniach*, umožňujú v programátorských kurzoch (funkcionálne, logické a procedurálne programovanie) interaktívnu tvorbu programátorského riešenia zabezpečenú vloženým komponentom zo systému Peoplia.

Na takúto formu integrácie vzdelávacích systémov teda môžeme z pohľadu používateľského rozhrania nahliadať v nasledujúcej cibuľovito nabalenej štruktúre:

[1] https://alef.fiit.stuba.sk
[2] https://www.peoplia.org/fiit/

1. **Editor pre vkladanie riešenia príkladu**. Časť stránky určená pre vytváranie programovacieho riešenia študentom. Je poskytovaná prepojeným systémom.
2. **Vzdelávací objekt obsahujúci zadanie príkladu.** Časť stránky obsahujúca zároveň predchádzajúcu časť (editor), ktorá zobrazuje zadanie a dodatočné informácie (napríklad voliteľnú pomôcku) potrebné pre riešenie príkladu. Môže byť poskytovaná systémom, v ktorom študent práve pracuje, alebo aj prepojeným systémom.
3. **Widgety týkajúce sa aktuálneho objektu**. Nástroje umiestnené v okolí aktuálneho vzdelávacieho objektu alebo priamo v ňom, ktoré poskytujú metainformácie o danom objekte, napríklad tagy pridané používateľmi, chyby hlásené v danom objekte, atď.
4. **Systémové widgety**. Nástroje umiestnené v okolí aktuálneho objektu s informáciami týkajúcimi sa viacerých objektov alebo celkovej aktivity študenta v systéme. Spadajú sem widgety obsahujúce definície pojmov (zdieľané naprieč všetkými vzdelávacími objektmi v systéme), zobrazujúce skóre používateľa a pod.
5. **Ďalšie karty prehliadača so vzdelávacími objektmi**. Študent si počas riešenia príkladu môže prirodzene otvoriť ďalšie objekty v ostatných kartách prehliadača, napríklad s *cvičením* s triviálnejšou alebo podobnou verziou práve riešeného príkladu alebo s *vysvetlením* témy týkajúcej sa príkladu.
6. **Ďalšie karty prehliadača s otvoreným Webom**. Študent môže počas práce vo vzdelávacom systéme taktiež prehliadať otvorený Web, čiže vyhľadávať ďalšie informačné zdroje týkajúce sa aktuálne študovanej témy. Tie získava napríklad vyhľadávaním vo webovom vyhľadávači alebo prostredníctvom komunikácie so spolužiakmi, často prostriedkami mimo dosahu samotného vzdelávacieho systému.

V najvyšších úrovniach (**5** až **6**) môžeme uplatniť sledovanie prepínania kariet prehliadača. Ak používateľ-študent prečíta vzdelávací objekt, prepne sa do vyhľadávača, navštívi viacero stránok na otvorenom Webe a vráti sa na vzdelávací objekt, navštívené stránky pravdepodobne súviseli so vzdelávacím objektom a preto tento objekt môžeme obohatiť o obsah nachádzajúci sa na ďalších navštívených stránkach. Pre sledovanie takéhoto prepínania medzi kartami obsahujúcimi stránky z iných webových systémov je však potrebné rozšírenie prehliadača nainštalované u používateľa.

Zároveň však môžeme sledovať prepínanie kariet prehliadača v rámci samotného vzdelávacieho systému a to bez potreby inštalácie rozšírenia. Ak napríklad väčší počet používateľov počas riešenia daného *príkladu* navštívi určité *vysvetlenie*, môžeme predpokladať, že toto vysvetlenie súvisí s látkou potrebnou pre vyriešenie tohto príkladu a môžeme ho odporúčať neskorším študentom riešiacim daný príklad.

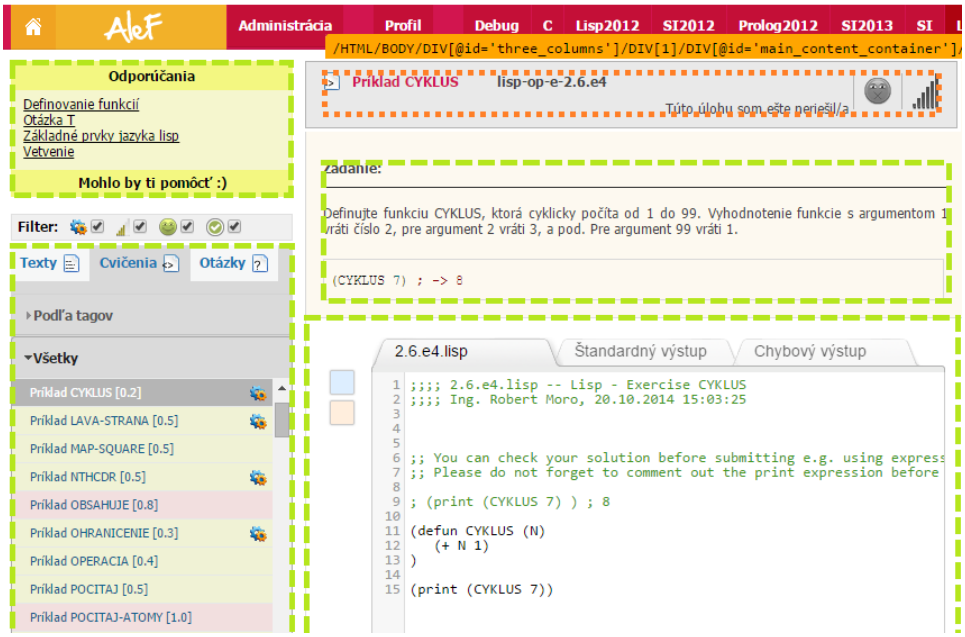
Na nižších úrovniach, úrovniach widgetov (**3** až **4**) a úrovniach samotného obsahu (**1** až **2**), môžeme sledovať bežne používané indikátory explicitnej a implicitnej spätnej väzby, napríklad pozíciu myši, klikanie, stláčanie kláves. Tieto indikátory však nemusia poskytovať dostatočný obraz o aktivite používateľa, napríklad widget poskytujúci definície pojmov poskytuje prehľad najčastejších pojmov bez ďalšej priamej interakcie (klikania, stláčania kláves) s daným widgetom. Študent tak môže widget použiť (napríklad oboznámiť sa s definíciou určitého pojmu) bez toho, aby vôbec pohol myšou. V takom prípade musíme uvažovať samotný pohľad používateľa na dané fragmenty používateľského rozhrania [8].

3 Využitie sledovania pohľadu

Aby sme mohli zaznamenávať pohyb očí používateľov (študentov) a kombinovať ho s údajmi z iných vstupných zariadení, navrhli sme a implementovali infraštruktúru pre sledovanie používateľského zážitku v prostredí dynamických webových stránok [9], akými sú aj vzdelávacie systémy ALEF a Peoplia. Hlavnou výhodou tejto infraštruktúry oproti existujúcim riešeniam je možnosť si vizuálne priamo na webovej stránke (vo vzdelávacom systéme) zadefinovať potenciálne oblasti záujmu (editor zdrojového kódu, text zadania, jednotlivé widgety a pod.; pozri Obr. 1).

Navrhnuté riešenie je pritom odolné voči zmene polohy ako aj veľkosti zvolených oblastí záujmu. Okrem toho je nami implementovaná infraštruktúra nezávislá na zariadení na sledovanie pohľadu (podporuje zariadenia Tobii X2-30 a The Eye Tribe, pričom je ľahko rozšíriteľná o ďalšie), ako aj na webovom prehliadači, v ktorom prebieha interakcia používateľa (študenta) s webovou stránkou (vzdelávacím systémom).

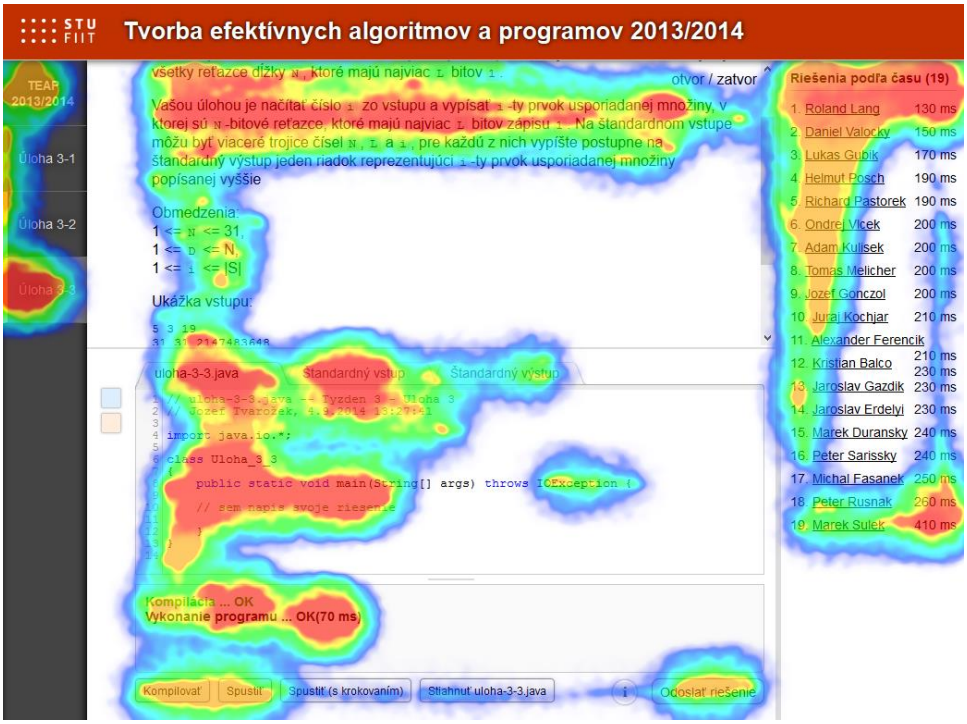
Pre zvolené oblasti záujmu sa následne z nazbieraných údajov o pohybe očí vyhodnocujú metriky ako počet fixácií (t. j. počet pozretí do zvolenej oblasti), celkové trvanie fixácií (t. j. celkový čas, ktorý sa používateľ pozeral do zvolenej oblasti) či priebeh trvania a počtu fixácií v čase. Tieto vypočítané metriky spolu s pôvodnými dátami sa sprístupňujú systémom ALEF a Peoplia pomocou jednotného aplikačného rozhrania (API).



Obr. 1. Pohľad na časť rozhrania systému ALEF s integrovaným editorom zdrojového kódu zo systému Peoplia. Čiarkovane (zelenou farbou) sú zvýraznené zadefinované oblasti záujmu (widget s odporúčaniami, menu, zadanie úlohy, editor), bodkovane (oranžovou farbou) je po nadídení kurzorom myši zvýraznená potenciálna oblasť záujmu.

Priame prepojenie infraštruktúry pre sledovanie používateľského zážitku so vzdelávacím systémom umožňuje analýzu vstupných prúdov dát (pohľadu očí, pohybov myši, senzorov emócií, atď.) vzhľadom na konkrétne vzdelávacie objekty a ich obsah. Napríklad pri programovaní je zvyčajne obsah editoru zdrojového kódu neustále upravovaný a statické zaznamenávanie súradníc na obrazovke bez prepojenia s aktuálnym obsahom je schopné analyzovať len všeobecné trendy používania (napr. čítanie zadania problému vs. písanie kódu riešenia). Prepojenie s obsahom navyše umožňuje analyzovať konkrétne spôsoby a postupy pri učení, ako napr. spôsob čítania a vytvárania zdrojového kódu, pasáže zdrojového kódu, ktoré študentovi spôsobujú ťažkosti, resp. čítanosť jednotlivých pasáží učebného textu (angl. tzv. *read-wear*).

Zozbierané údaje (Obr. 2) môže výhodne využiť aj samotný učiteľ, ktorý získa okamžitý prehľad o aktuálnom stave študenta, príp. agregáciou všetkých pohľadov v študijnej skupine získa priebežnú situáciu pri riešení úloh študentov na vyučovacej hodine, a môže tak lepšie zacieliť výklad.



Obr. 2. Vizualizácia pohľadu študenta pri riešení programátorskej úlohy v systéme Peoplia. Záznam čítania môže napríklad napovedať, ktorým častiam zadania sa študent venoval málo a následne príklad nevyriešil správne alebo naopak, pre úspešné vyriešenie neboli potrebné.

4 Záver

Tu opísané indikátory a metriky sme overili v prepojených systémoch ALEF a Peoplia v kurzoch funkcionálneho a logického programovania v letnom semestri 2013/2014. Zaznamenali sme 243 vyriešených prepojených príkladov vo funkcionálnom a 198 v logickom programovaní. Overili sme aj využitie prepínania medzi kartami na odporúčanie objektov a využitie prehliadania Webu na obohacovanie obsahu.

Pilotný zber údajov o pohľade očí vo vzdelávacom systéme pri programovaní nám dáva nové možnosti skúmať spôsoby ako začínajúci programátori (študenti) pracujú so zdrojovým kódom a ako sa tieto spôsoby menia pri napredovaní v učení.

Poďakovanie. Tento článok vznikol vďaka čiastočnej podpore projektu KEGA 009STU-4/2014 a podpore MŠVVaŠ SR v rámci OP Výskum a vývoj pre projekt: Univerzitný vedecký park STU Bratislava (UVP STU Bratislava), ITMS 26240220084 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

Literatúra

1. Busjahn, T. et al.: Eye tracking in computing education. In: Proc. of the 10th Annual Conf. on Int. Computing Education Research, pp. 3–10. ACM Press, New York (2014)
2. Downes, S.: E-learning 2.0. eLearn magazine, 10(1) (2005)
3. Helminen, J., Ihantola, P., Karavirta, V.: Recording and Analyzing In-browser Programming Sessions. In: Proc. of the 13th Koli Calling Int. Conf. on Computing Education Research, pp. 13–22. ACM Press, New York (2013)
4. Hwang, W.-Y., Wang, C.-Y., Hwang, G.-J., Huang, Y.-M., Huang, S.: A Web-based Programming Learning Environment to Support Cognitive Development. Interacting with Computers. 20(6), 524–534 (2008)
5. Ihantola, P., Ahoniemi, T., Karavirta, V., Seppälä, O.: Review of Recent Systems for Automatic Assessment of Programming Assignments. In: Proc. of the 10th Int. Conf. on Comp. Educ. Res., pp. 86–93. ACM Press, New York (2010)
6. Kardan, S., Conati, C.: Comparing and Combining Eye Gaze and Interface Actions for Determining User Learning with an Interactive Simulation. In: UMAP '13: Proc. of the 21th Int. Conf. on User Modeling, Adaptation, and Personalization, LNCS 7899, pp. 215–227. Springer, Berlin, Heidelberg (2013)
7. Labaj, M., Šimko, M., Tvarožek, J., Bieliková, M.: Integrated Environment for Learning Programming, In: EC-TEL '14: Proc. of the 9th European Conf. on Technology Enhanced Learning, LNCS 8719, pp. 498–501. Springer (2014)
8. Labaj, M.,: Implicit Feedback Based Recommendation and Collaboration. In: Information Sciences and Technologies Bulletin of the ACM Slovakia. pp. 41–42. (2011)
9. Móro, R., Daráž, J., Bieliková, M.: Visualization of Gaze Tracking Data for UX Testing on the Web. In: Hypertext '14 Extended Proceedings: Late-breaking Results, Doctoral Consortium and Workshop Proceedings of the 25th ACM Hypertext and Social Media Conference, vol. 1210. CEUR-WS (2014)
10. Návrat, P., Tvarožek, J.: Online Programming Exercises for Summative Assessment in University Courses. In: CompSysTech '14, Ruse, Bulgaria, to appear. Springer (2014)

Smerovanie dizertačných projektov

Hľadanie vzorov pri práci s počítačovou myšou: Vizuálna analýza ťahov

Peter Krátky, Daniela Chudá

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

Abstrakt. Charakteristiky práce s počítačovou myšou môžu byť všeobecne dostupnými biometrickými charakteristikami využitými pri zabezpečení systémov. Charakteristiky opisujúce pohyb využívané vo výskumných pre určenie, resp. potvrdenie identity, sú aplikáciou známych veličín. Avšak ak chceme odhaliť ďalšie špecifické vlastnosti tzv. ťahov s myšou, je potrebná hlbšia vizuálna analýza pohybu v priestore a čase. V našej práci sa zameriavame hľadaniu ďalších vzorov pohybu s myšou, ktoré by mohli zvýšiť presnosť určenia/potvrdenia identity, a to pomocou vizualizácie nameraných dát. Uvádzame tiež prvotné vyhodnotenie kvality nájdených charakteristík pre verifikáciu/identifikáciu.

Kľúčové slová: biometrické charakteristiky, dynamika práce s myšou, vizuálna analýza ťahov

1 Úvod

Ľahko a lacno získateľnými biometrickými charakteristikami môžu byť charakteristiky práce s počítačovou myšou. Pre účely verifikácie používateľa boli takéto charakteristiky použité vo viacerých výskumných prácach, pričom chybovosť verifikácie dosahovala úroveň medzi jedným a desiatimi percentami [1, 2, 4, 6].

Tieto charakteristiky sa väčšinou viažu na akcie vykonávané používateľom, ako napríklad kliknutie, pohyb myšou, rolovanie. Dĺžka kliknutia sa v našej predošlej práci ukázala ako vlastnosť s najlepšou rozlišovacou schopnosťou [3], vyššou ako ktorákoľvek vlastnosť pohybu. Avšak akcia pohybu skrýva veľké množstvo charakteristík, ktorých kombinácie môžu byť veľmi efektívne.

Využívané charakteristiky bývajú štandardné veličiny opisujúce pohyb, ako rýchlosť [4], zrýchlenie, zakrivenie, uhlová rýchlosť [2], ale aj veličiny so sofistikovanejším výpočtom, ako ťažisko trajektórie, koeficient trhania [1], uhol zakrivenia, vzdialenosť zakrivenia [6], počet inflexných bodov krivky [5].

V uvedených výskumných prácach nie sú charakteristiky odpozorované na kvalitatívnej úrovni, ale predovšetkým sú to veličiny využité z iných oblastí. V našej práci sa venujeme hľadaniu ďalších potenciálnych vzorov pohybu myši na základe vizuálnej analýzy tohto pohybu. K tomuto účelu sme implementovali nástroj pre vizualizáciu trajektórie pohybu, či zmien rýchlosti a smeru prostredníctvom grafov.

2 Údaje pre analýzu

Uskutočnili sme experiment, na základe ktorého sme získali údaje o práci s počítačovou myšou. Experiment sme navrhli tak, aby používatelia vykonávali relatívne rovnakú aktivitu, a teda ťahy podobného charakteru. Implementovaným nástrojom, v ktorom používatelia aktivitu vykonávali bola hra *pexeso*.

Získali sme údaje od 12 ľudí, ktorí priemerne vykonali priemerne 375 ťahov.

2.1 Spracovanie nameraných údajov do ťahov

Údaje o pohybe s myšou sú zaznamenávané s frekvenciou danou zariadením a ukladané vo forme textového súboru. Textový súbor predstavuje sekvenciu udalostí interakcie s používateľským rozhraním. Jeden záznam predstavuje jednu udalosť danú typom udalosti (zatlačenie/uvoľnenie tlačidla, pohyb, atď.), časom v milisekundách a pozíciou kurzora x a y na ploche. Ukážka úseku zo súboru:

```
m;1397061616051;507;571
m;1397061616075;506;571
m;1397061616085;506;570
m;1397061616127;506;569
m;1397061616257;505;569
m-down;1397061617295;505;569
```

Uvedený súbor so záznamami je z hľadiska pohybu zhlukom všetkých pohybových udalostí s myšou. Charakteristiky pre takúto dlhú sekvenciu by boli degradované. Potrebujeme rozdeliť dlhú sekvenciu na ucelené akcie, ktoré vyjadrujú jednu cielenú aktivitu používateľa. Vhodným celkom sa zdá byť ťah s myšou. Ide o sekvenciu pohybu myšou zakončenú kliknutím alebo dostatočne dlhým časovým intervalom nečinnosti. Zachytáva jeden zámer používateľa, a teda predpokladáme aj relatívnu homogenitu charakteristík v rámci bodov ťahu. Okrem toho predpokladáme podobnosť s inými ťahmi na začiatku, v strede a na konci ťahu.

3 Vizuálna analýza ťahov

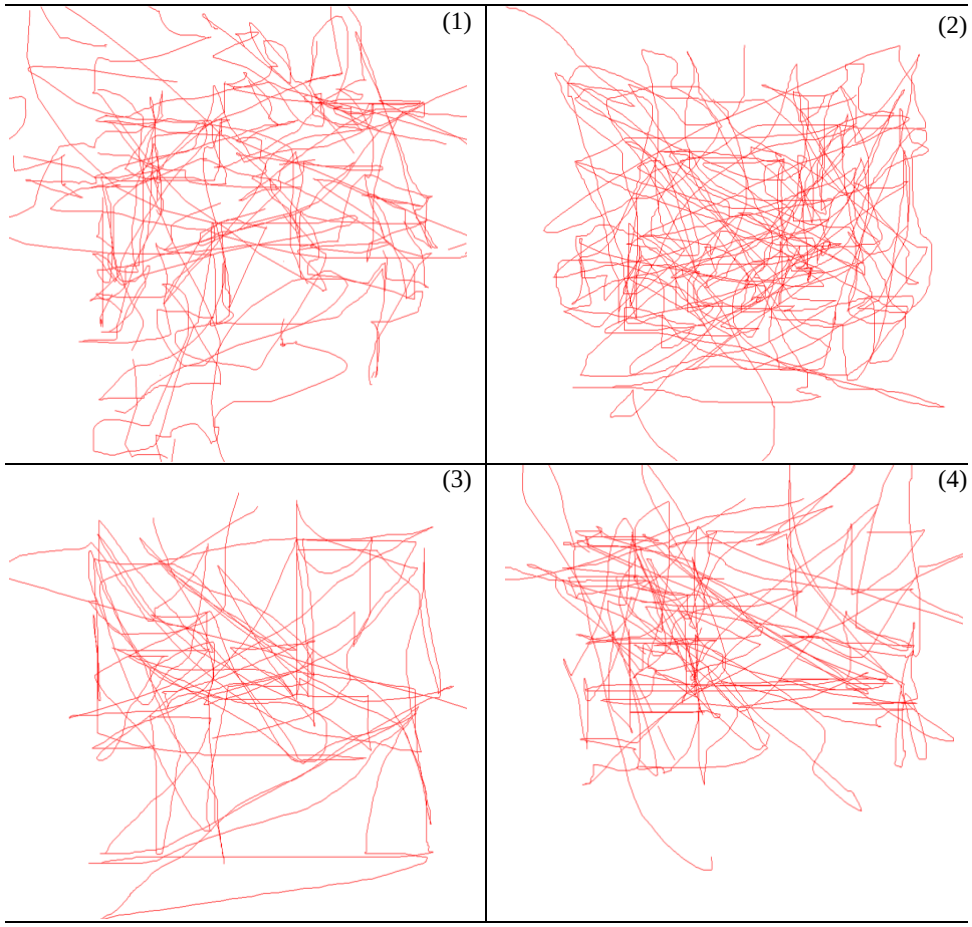
Vytvorili sme nástroj, ktorý vizualizuje ťahy v dvojrozmernom priestore a okamžité hodnoty niektorých veličín. Prezerali sme celkový nameraný pohyb u všetkých používateľov, množinu ťahov sme ďalej zúžili na ťahy podobnej dĺžky a smeru, u vybratých používateľov sme vizuálne analyzovali aj jednotlivé ťahy.

3.1 Analýza nehomogénnej skupiny ťahov v priestore

V prvom rade sme nechali vykresliť celkový pohyb s myšou pre jednotlivých používateľov. Pre štyroch z nich je znázornený na Obr. 1. Pohyb pre prvého používateľa je viac roztiahnutý do strán, obsahuje veľa oblúkov, často krát s ostrou zmenou smeru.

Počas premýšľania zrejme pohyboval pomaly kurzorom. Pohyb pre druhého taktiež obsahuje veľa oblúkov, sú však plynulejšie. Vodorovné ťahy majú mierny náklon. Tretí používateľ vykonal minimum pohybu navyše, ťahy si držia smer. Štvrtý vykonával striktne rovné vodorovné ťahy. Niektoré jeho ťahy obsahujú dlhú úzku slučku, čo možno značí odkladanie kurzora mimo čítaný obsah stránky.

Charakteristikami celkového pohybu by mohli byť: množstvo ostrých zmien smeru (na jednotku dĺžky), percentuálna časť zakriveného pohybu (ďalej ako *vlnitosť*), pokrytá plocha (navštívené štvorce mriežky), najhustejšie pokryté miesto (štvorec mriežky), vzdialenosť od stredu pri ponechaní kurzora.



Obr. 1. Celkový nameraný pohyb s myšou pre štyroch rôznych používateľov.

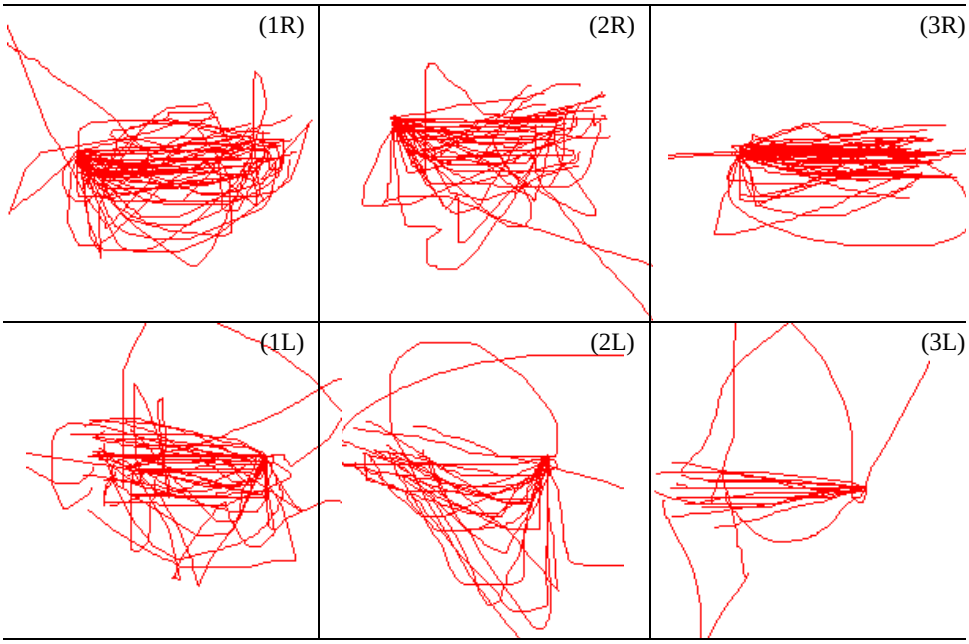
3.2 Analýza homogénnej skupiny ťahov v priestore

V ďalšom kroku sme nechali vykresliť ťahy s podobnou dĺžkou a podobným smerom. Pomocou algoritmu sme hľadali ťahy dlhšie ako 50 pixlov. Najviac vzájomne podobných ťahov malo vodorovný smer. Na Obr. 2 sú znázornené podobné vodorovné ťahy

u vybraných používateľov, pre lepšiu ilustráciu sú rozdelené na ťahy so smerom doprava (horný rad) a doľava (dolný rad). Prvý dvaja používatelia sa výrazne odlišujú od tretieho. Ťahom pokrývajú veľkú plochu smerom dole, čiastočne hore. Druhý používateľ má výrazný posun dole pri začiatku pohybu, čo možno vidieť predovšetkým pri pohybe vľavo. Tretí používateľ vykonáva priamočiary pohyb bez oblúkov navyše.

Okrem toho, pri vykreslení jednotlivých ťahov bolo možné u niektorých používateľov pozorovať krátky oblúk na konci ťahu, čo zrejme značí korekciu pohybu pri nasmerovaní kurzora na zamýšľané miesto.

Charakteristikami jedného ťahu by mohli byť: pokrytá plocha pod krivkou, vzdialenosť najväčšieho oblúku od začiatočného bodu ťahu (percentuálne, vzhľadom na dĺžku ťahu), pomer dĺžky ťahu voči dĺžke spojnice, pomer dĺžky záverečného úseku ťahu voči spojnici začiatočného a koncového bodu úseku.

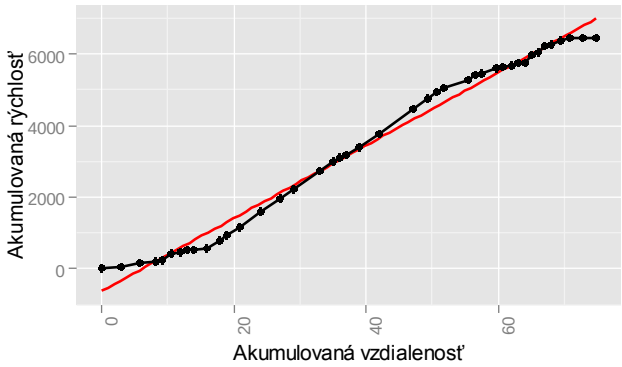


Obr. 2. Podobné ťahy s myšou pre troch rôznych používateľov v smere doprava (horný rad) a doľava (dolný rad).

3.3 Analýza ťahu s ohľadom na čas

V tejto časti analýzy sme nechali vykresľovať graf, ktorý má na osi x vzdialenosť od začiatku ťahu a na osi y akumulovanú rýchlosť. Často sa objavoval vzor ako na Obr. 3, kde rýchlosť na začiatku stúpa, po krátkej dobe sa ustáli a na konci klesá.

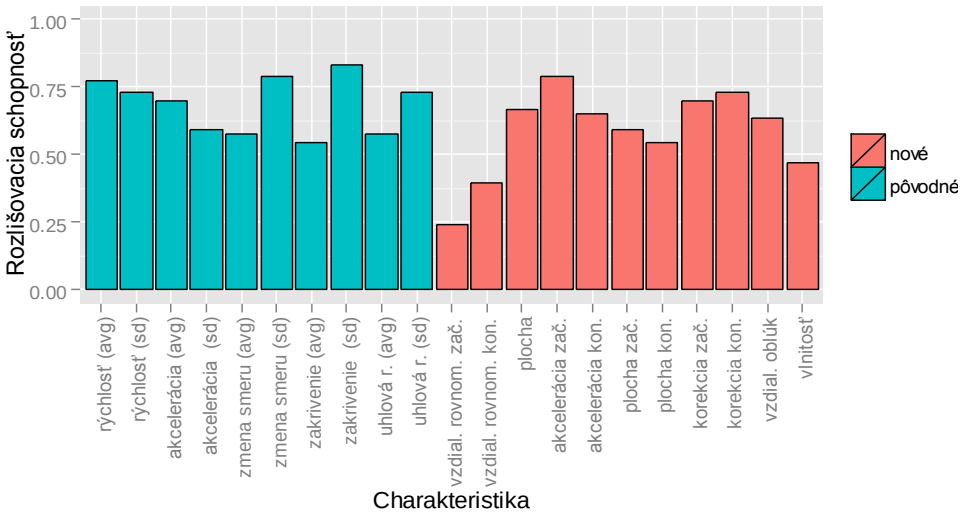
Ďalšími charakteristikami ťahu by mohli byť: vzdialenosť od začiatku, kedy sa rýchlosť ustáli a vzdialenosť od konca, kedy sa začne spomaľovať. Vyjadrenie takejto charakteristiky môžeme určiť ako vzdialenosť, v ktorej krivka akumulovanej rýchlosti pretne regresnú priamku reprezentujúcu rovnomerný pohyb. Na stanovenom začiatočnom/koncovom úseku by sme zasa mohli vypočítať akceleráciu.



Obr. 3. Graf akumulovanej rýchlosti v závislosti od akumulovanej vzdialenosti. Regresná priamka reprezentuje rovnomerný pohyb.

4 Vyhodnotenie nájdených charakteristík

Pre nasledujúce charakteristiky ťahu sme vyčíslili kvalitu pre rozlíšenie používateľov, rovnako ako boli vyčíslené hodnoty základných charakteristík v [3] (viď Obr. 4): začiatok/koniec rovnomerného pohybu, plocha pod krivkou ťahu, akcelerácia na začiatku/konci, plocha pokrytá na začiatku/konci krivky ťahu, korekcia na začiatku/konci ťahu, vlnitosť ťahu. Zdá sa, že väčšina potenciálnych charakteristík by mohla významne pomôcť pri autentifikácií/identifikácií. Charakteristikou zrejme nebude začiatok a koniec ustáleného pohybu, úroveň 50% nedosiahla ani vlnitosť.



Obr. 4. Hodnoty rozlišovacích schopností jednotlivých charakteristík pôvodných podľa [3] a novo identifikovaných.

5 Záver

Vizuálne sme analyzovali pohyb s myšou, ktorý sme si pomocou jednoduchého nástroja nechali vykresliť. Údaje sme získali z experimentu nastaveného tak, aby všetci používatelia vykonávali aktivitu podobného charakteru, zjednodušenú oproti orientácií v bežnom používateľskom rozhraní. Detailne sme prezreli celkový nameraný pohyb s myšou v priestore pre jednotlivých používateľov, vyhľadali sme ťahy s podobným charakterom a hľadali v nich opakujúce sa vzory, taktiež sme analyzovali pohyb v čase. Našli sme niekoľko potenciálnych charakteristík, ktoré vo výskumných prácach sledované neboli. Pre celkový pohyb to sú napríklad množstvo ostrých zmien pohybu, percentuálna časť rovného/zakriveného pohybu. Pre ťahy z hľadiska priestoru to sú napríklad plocha pod krivkou, vzdialenosť oblúku od spojnice, a z hľadiska času začiatok/koniec rovnomerného pohybu, akcelerácia na začiatočnom/koncovom úseku ťahu. Aj keď údaje boli namerané za zjednodušených podmienok, veríme, že vymenované charakteristiky sa budú v nejakej miere vyskytovať aj pri bežnej práci. V tom prípade predpokladáme, že bude potrebné namerať väčšie množstvo ťahov, keďže veľa ťahov bude zrejme spôsobovať šum. Uviedli sme prvotné vyhodnotenia kvality vybraných charakteristík z hľadiska verifikácie/identifikácie používateľa, ktoré ukazujú možnosť použitia väčšiny z nich.

Poďakovanie. Táto práca bola čiastočne podporená projektom VEGA VG1/0971/11 Získavanie, spracovanie, vizualizácia textových informácií na základe analýzy relácií podobnostiam projektom APVV-0208-10 Kognitívne cestovanie po digitálnom svete webu a knižníc s podporou personalizovaných služieb a sociálnych sietí.

6 Literatúra

1. Feher, C., et al.: User identity verification via mouse dynamics. In: *Information Sciences.* (2012), vol. 201, s. 19–36.
2. Gamboa, H., & Fred, A.: A behavioral biometric system based on human-computer interaction. In: *Proceedings of SPIE*. Orlando, FL, (2004).
3. Chudá, D., Krátky, P.: Usage of computer mouse characteristics for identification in web browsing. In: *CompSysTech ’14 Proceedings of the 15th International Conference on Computer Systems and Technologies*, (2014). (In print).
4. Pusara, M., & Brodley, C. E.: User re-authentication via mouse movements. In: *Proceedings of the 2004 ACM workshop VizSEC/DMSEC ’04*, NY, (2004), s. 1-8.
5. Schulz, D.: Mouse curve biometrics. In: *2006 Biometrics Symposium: Special Session on Research at The Biometric Consortium Conference*. (2006), s. 1–6.
6. Zheng, N., Paloski, A., Wang, H.: An efficient user verification system via mouse movements. In: *Proceedings of the 18th ACM conference on Computer and communications security - CCS ’11*, New York, NY, (2011), s. 139–150.

Spracovanie prúdu údajov pomocou transformácie opakujúcich sa sekvencií na symboly

Jakub Ševcech[1]

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

Abstrakt. Spracovanie prichádzajúceho prúdu údajov prináša veľa ohraničení a výziev spojených s obmedzenou pamäťou a nutnosťou spracovania údajov v jedinom prechode. V našej práci študujeme vlastnosti prichádzajúceho prúdu údajov a možnosť použitia opakujúcich sa sekvencií v takomto prúde ako prostriedku na jeho reprezentáciu a ďalšie spracovanie. Navrhujeme reprezentáciu časových radov, kde opakujúce sa sekvencie sú transformované na symboly, pričom sa sústreďujeme na špecifiká transformácie prichádzajúceho prúdu údajov na symboly s ohľadom na variabilitu prúdu údajov, pamäťové obmedzenia a obmedzenie na jediný prechod cez údaje. Overenie plánujeme pomocou porovnania vlastností navrhnutej reprezentácie s inými bežne používanými reprezentáciami pri rôznych úlohách analýzy prúdov údajov ako aj statických kolekcií časových radov.

Kľúčové slová: časový rad, prúd údajov, symbolická reprezentácia

1 Úvod

V minulosti sa veľké úsilie výskumníkov venovalo výskumu vlastností a metód na spracovanie časových radov. Vzniklo množstvo metód na ich reprezentáciu, spracovanie a analýzu. Väčšina z týchto prístupov sa však sústreďuje na spracovanie statických kolekcií údajov napríklad pre potreby dopytovania sa databáz časových radov [5]. V ostatnej dobe sa však venuje zvýšený záujem prúdovému spracovaniu údajov, ktoré nachádza svoje uplatnenie v najrôznejších doménach od spracovania fyzikálnych meraní, cez analýzu finančných trhov až po monitorovanie aktivity vo webových aplikáciách.

Spracovanie prúdov údajov zdieľa so spracovaním statických kolekcií údajov množstvo problémov a úloh ale zároveň zavádza do spracovania množstvo ďalších výziev a obmedzení. V tejto práci sa zaoberáme spracovaním prúdov údajov, konkrétne sa zaoberáme reprezentáciou časových radov pomocou transformácie opakujúcich sa sekvencií na symboly pričom sa sústreďujeme na použiteľnosť takejto repre-

[1] Školiteľka: prof. Mária Bieliková

zentácie pri obmedzeniach spôsobených spracovaním prúdov údajov a jej ďalšími aplikáciami pri spracovaní a analýze prúdov údajov.

Pre spracovaní časových radov sa tieto častokrát nepoužívajú vo svojej surovej podobe (postupnosť udalostí alebo meraní) ale využívajú sa aj ich rôzne reprezentácie, ktoré majú za účel zefektívniť ďalšie spracovanie časových radov redukovaním ich dimenzionality alebo predspracovaním pre jednoduchšie použitie. V [4] autori prezentujú jednu z najčastejšie používaných reprezentácií časových radov Piecewise Aggregate Approximation (PAA) a porovnávajú jej vlastnosti s tromi inými často používanými reprezentáciami pričom ukázali, že robustné reprezentácie ako je napríklad Diskrétna Fourierova Transformácia dosahujú lepšie výsledky ako jednoduchšie metódy ako už spomínaná metóda PAA, ale tieto jednoduchšie reprezentácie poskytujú benefity ako je napríklad jednoduchosť na pochopenie, jednoduchosť implementácie a vo všeobecnosti lepšiu použiteľnosť pri spracovaní veľkých objemov údajov.

Podobne ako v prípade reprezentácií časových radov vzniklo aj množstvo rôznych metrík na porovnávanie časových radov, ktoré majú rôzne vlastnosti a dosahujú diametrálne odlišné výsledky pri spracovaní rôznych typov údajov [7].

Väčšina pozornosti pri navrhovaní reprezentácií časových radov a metrík na výpočet ich podobností sa venovala spracovaniu statických kolekcií údajov. Len málo prác sa však sústreďovalo na návrh nových alebo adaptáciu existujúcich metód pre prácu s potencionálne nekonečným prúdom údajov, čo je oblasťou, ktorej sa v našej práci chceme zaoberať.

2 Spracovanie prúdu údajov

V našej práci sa sústreďujeme na spracovanie potencionálne nekonečného prúdu prichádzajúcich údajov. Spracovanie prúdov údajov je charakteristické viacerými odlišnosťami od spracovanie statických kolekcií údajov a s nimi spojenými obmedzeniami. V práci [1] autori definujú model prúdu údajov na základe jeho charakteristických vlastností takto:

- Udalosti prichádzajú online, v čase ich vytvorenia.
- Systém nemá kontrolu nad poradím, v ktorom udalosti prichádzajú.
- Prúd údajov je potencionálne nekonečný.
- Akonáhle je udalosť spracovaná, je zahodená alebo archivovaná. Nemôže však byť jednoducho znovu získaná okrem prípadov, keď je uložená v pôvodnej alebo agregovanej podobe v pamäti, ktorá je však oveľa menšia ako je veľkosť prúdu údajov.

Tieto vlastnosti zavádzajú do spracovania prúdu dát niekoľko problémov a výziev, s ktorými sa musia vysporiadať metódy na ich analýzu [3]:

- Spracovanie súvislého prúdu dát s premenlivou rýchlosťou pribúdania udalostí. Systém na spracovanie údajov sa musí prispôsobovať a škálovať pre premenlivé množstvo prichádzajúcich údajov.
- Spracovanie potencionálne nekonečného prúdu dát s použitím len obmedzeného množstva pamäti.

- Inkrementálne spracovanie údajov bez možnosti viacerých prechodov cez dáta a značnými obmedzeniami na množstvo použiteľnej pamäte. Pri spracovaní udalostí teda nie je možné v ľubovoľnom čase pristupovať k už spracovaným častiam prúdu okrem prípadov ak je táto časť uložené v obmedzenej pamäti.
- Nutnosť poskytovať výsledky s garantovanou presnosťou pri prítomnosti pamäťových obmedzení a jedného prechodu cez dáta.
- Modelovanie posunutia spôsobeného variabilitou prichádzajúceho prúdu dát. Veľká variabilita a postupný vývoj prichádzajúceho prúdu dát vyžadujú postupnú úpravu vytváraných modelov, pričom metódy na ich vytváranie sa s týmito posunutiami musia dokázať vysporiadať, prípadne ich musia dokázať vizualizovať.

Na vysporiadanie sa s opisovanými problémami bolo navrhnutých viacero techník a metód ako je napríklad používanie posuvného okna, aproximatívne algoritmy alebo uchovávanie agregovaných údajov v pamäti. V našej práci študujeme možnosti využitia transformácie prichádzajúceho prúdu údajov do inej reprezentácie. Študujeme vlastnosti transformácie opakujúcich sa sekvencií v prichádzajúcom prúde údajov na symboly ako prostriedku na redukciu dimenzionality a na podporu ďalších úloh analýzy údajov.

3 Transformácia prúdu údajov na symboly a otvorené problémy

Spomedzi existujúcich reprezentácií časových radov predstavujú pomerne zaujímavú skupinu symbolické reprezentácie, ktoré transformujú časový rad na postupnosť symbolov. Transformácia na symboly okrem redukcie dimenzionality časového radu umožňuje spracovávať časové rady pomocou metód, ktoré vyžadujú pre svoju prácu údaje v diskrétnej reprezentácii ako sú napríklad Markovovské modely, suffixové stromy, hashovanie a podobne. Súčasné symbolické reprezentácie časových radov však predpokladajú možnosť viacerých prechodov cez spracovávané údaje počas ich transformácie pre optimálne definovanie symbolov reprezentujúcich jednotlivé sekvencie časového radu [2, 6]. Takáto transformácia nie je možná pri spracovávaní postupne prichádzajúcich údajov, ktoré so sebou okrem obmedzenia na jediný prechod prinášajú aj ďalšie obmedzenia opisované v predchádzajúcej časti. Jedným z najväčších obmedzení je variabilita prichádzajúcich údajov a objavovanie sa nových vzorov v prichádzajúcich údajoch, ktoré nie je možné reprezentovať existujúcou sadou symbolov a preto treba rozširovať abecedu symbolov na základe novo prichádzajúcich údajov, čo so sebou nesie nové problémy pri spracúvaní transformovaných údajov.

V našej práci navrhujeme metódu na transformáciu časového radu na symbolickú reprezentáciu založenú na symbolickej reprezentácii časových radov prezentovanej v [2]. Jednotlivé symboly reprezentujú skupiny podobných opakujúcich sa sekvencii v priebehu časového radu, pričom existujúcu metódu na transformáciu časového radu adaptujeme pre potreby spracovania prúdov údajov [8, 9]. Pri návrhu metódy na transformáciu prúdu údajov sa sústreďujeme na niekoľko hlavných problémov:

- Online transformácia prichádzajúcich údajov na symboly.

- Uchovávanie sady symbolov pri použití obmedzenej pamäte a potencionálne neobmedzeného prúdu údajov.
- Posunutie zhlukov sekvencií mapovaných na symboly s novými prichádzajúcimi údajmi.
- Spracovanie transformovaného prúdu symbolov pri neustále sa rozširujúcej abecede symbolov použitých na reprezentáciu časového radu.

Transformácia na navrhovanú reprezentáciu je založená na posúvaní okna definovanej dĺžky cez prichádzajúci prúd údajov a porovnávanie sekvencie ohraničenej týmto oknom so sadou vzorov, ktoré sa objavili v predchádzajúcich častiach prúdu. V prípade zhody sledovaného okna s predchádzajúcim vzorom je táto sekvencia transformovaná na symbol, ktorý reprezentuje nájdený vzor. V prípade ak sa nenájde zhoda s predchádzajúcimi vzormi je z danej sekvencie vytvorení nový vzor a tým sa rozšíri abeceda existujúcich symbolov. Každý takto vytvorený vzor/symbol reprezentuje zhluk podobných, opakujúcich sa sekvencií. Kľúčovou úlohou pri takejto transformácii sekvencií na symboly je proces porovnávania prichádzajúcich sekvencií so sadou vzorov a tvorba zhlukov sekvencií tvoriacich symboly pri obmedzeniach na jeden prechod, pamäťových obmedzeniach a s ohľadom na vyvíjajúci sa prúd údajov a posun symbolov s novo prichádzajúcimi podobnými sekvenciami. V súčasných metódach transformácie časových radov na symboly sa používajú iteratívne zhlukovacie algoritmy ako je napríklad K-means, ktoré vyžadujú niekoľko prechodov cez dáta [4]. V našej práci experimentujeme s inkrementálnymi algoritmami na vytváranie zhlukov sekvencií ako je napríklad opisovaný greedy algoritmus, ktorý vytvorí nový zhluk vždy, keď sa v prichádzajúcich dátach objaví sekvencia odlišná od všetkých doteraz pozorovaných sekvencií.

4 Očakávané prínosy a plán overenia

V práci sa sústreďujeme na návrh metódy transformácie časových radov na symbolickú reprezentáciu v prostredí práce s potencionálne nekonečným prúdom údajov. Navrhovaná metóda má za cieľ umožniť ďalšie spracovanie a analýzu prichádzajúcich prúdov údajov s ohľadom na obmedzenia, ktoré zavádzajú. Pomocou navrhovanej transformácie chceme dosiahnuť efektívnu redukciu dimenzionality prichádzajúceho prúdu údajov. Transformáciou prúdu udalostí (resp. meraní) na sekvenciu symbolov chceme umožniť použitie metód vyžadujúcich symbolickú reprezentáciu údajov aj pri spracovaní prúdu údajov, čo pri doterajších metódach sústreďujúcich sa na spracovanie statických kolekcií údajov nebolo možné.

V kontexte navrhovanej metódy plánujeme sériu experimentov na overenie efektívnosti symbolickej reprezentácie v porovnaní s inými bežne používanými reprezentáciami pri základných úlohách analýzy časových radov. V predbežných experimentoch porovnávajúcich úspešnosť klasifikácie s použitím rôznych metrík podobnosti časových radov v ich „surovej“ podobe a nami navrhovanej reprezentácie v spojení s Levenshteinovou vzdialenosťou sme dosiahli sľubné výsledky na rôznych typoch dátových vzoriek. V ďalšom kroku chceme tieto predbežné experimenty rozšíriť tak, aby sme dokázali zovšeobecniť naše pozorovania a aby sme dokázali identifikovať

vlastnosti údajov, pri ktorých navrhnutá reprezentácia dosahuje lepšie výsledky ako iné, porovnávané metódy.

Ďalej plánujeme overovať stabilitu generovanej abecedy symbolov s pribúdajúcim množstvom údajov na rôznych typoch údajov z reálnych aplikácií ako sú napríklad rôzne webové logy, mikroblogy alebo údaje o návštevnosti veľkých webových portálov. Jednou z kľúčových častí navrhovanej transformácie údajov, na ktorú sa chceme zamerať je spájanie podobných sekvencií do zhlukov a porovnanie vlastností rôznych zhlukovacích algoritmov a ich škálovateľnosti v prostredí spracovania prúdov údajov.

Poďakovanie. Táto práca vznikla vďaka čiastočnej podpore Vedeckej a grantovej agentúry Slovenskej republiky, číslo grantu VG 1/0752/14.

Literatúra

1. Babcock, B., Widom, J.: Models and Issues in Data Stream Systems. Proc. of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 1–16, ACM (2002).
2. Gaber, M. M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: a review. ACM Sigmod Record 34, no. 2, pp. 18-26, ACM (2005).
3. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery 26, no. 2, pp. 275-309, Springer (2013).
4. Das, G., Lin, K. I., Mannila, H., Renganathan, G., Smyth, P.: Rule Discovery from Time Series. Knowledge Discovery and Data Mining, vol. 98, pp. 16-22, (1998).
5. Keogh, E. J., Pazzani, M. J.: A simple dimensionality reduction technique for fast similarity search in large time series databases. Knowledge Discovery and Data Mining, pp. 122-133, Springer (2000).
6. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM SIGMOD Record, no. 30, vol. 2, pp. 151-162, ACM (2001).
7. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Mining and Knowledge Discovery, vol. 15, no. 2, pp. 107-144, Springer (2007).
8. Ševcech, J., Bieliková, M.: Data Streams Representation Using Repeating Patterns as Symbols. Advances in Intelligent Data Analysis XIII. First look track (accepted).
9. Ševcech, J., Bieliková, M.: Reprezentácia časových radov pomocou opakujúcich sa vzorov. Datakon (To appear)

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 126–129

Inteligentná analýza veľkých objemov dát

Petra Vrablecová[1]

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave
Ilkovičova 2, 842 16 Bratislava, Slovensko
petra.vrablecova@stuba.sk

Abstrakt. Spracovanie veľkých objemov dát otvára mnoho nových výziev – od vynaliezania nových výpočtových technológií, spôsobov ukladania a správy rozsiahlych dát po nové algoritmy a metódy ich analýzy, vizualizácie a prezentácie. Veľké objemy dát sú zdrojom cenných informácií v mnohých oblastiach a rôzne odvetvia ich využívajú pri biznis analýze. V energetickom sektore sú zdrojom veľkých objemov dát inteligentné merače elektrickej energie, ktoré pravidelne každú štvrťhodinu odosielajú údaje o spotrebe. Pre distribútora je presná predpoveď spotreby energie mimoriadne dôležitá, pretože mu umožní efektívne nakupovať energiu tak, aby dodržal regulatórne aj prevádzkové podmienky. V tejto práci sa zameriavame na metódy predikcie spotreby elektrickej energie a uvádzame koncept smerovania nášho budúceho výskumu v oblasti ich vylepšovania a adaptácie na spracovanie veľkých objemov dát.

Kľúčové slová: veľké objemy dát, analýza časových radov, predpoveď spotreby elektrickej energie

1 Veľké objemy dát

Zber a analýza dát sú vo veľkom využívané v komerčných aj nekomerčných oblastiach. Príkladmi využitia sú odporúčanie produktov zákazníkom, personalizovaná reklama, prispôsobovanie obsahu používateľovi, výskum, simulácie a modelovanie rôznych situácií, predpovedanie udalostí. V dôsledku globalizácie a vývoja nových technológií dáta neustále pribúdajú, a preto sú potrebné automatizované prostriedky pre ich spracovanie. Navyše tento proces musí byť dostatočne rýchly, pretože informácie získané z dát sa rýchlo stávajú neaktuálne a strácajú svoju hodnotu.

Už v roku 2001 definovala spoločnosť Gartner veľké objemy dát pomocou tzv. 3V modelu, ktorý opisuje ich tri hlavné „veľké“ vlastnosti – objem, rýchlosť a rôznorodosť [5]. Táto definícia otvorila nové výzvy na poli spracovania veľkých objemov dát [2]. Výskum a vývoj nových riešení prebieha pre každú fázu životného cyklu dát –

[1] Školiteľ: doc. Ing. Viera Rozinajová, PhD., Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave

od zberu, ukladania, filtrovania až po analýzu, vizualizáciu, prezentovanie dát a rozhodovanie sa na základe získaných informácií.

Vďaka technologickému pokroku (rýchle procesory, pamäťové média) a pokrokom v distribuovanom počítaní (model MapReduce) sme schopní dostatočne rýchlo narábať s veľkými objemami dát. Ďalším krokom v automatizácii procesu ich spracovania je integrácia tejto rozvinutej platformy pre fyzické narábanie s dátami s nástrojmi pre ich analýzu, vylepšovanie a prispôsobovanie samotných analytických metód a algoritmov pre prácu s väčšími objemami dát.

V súčasnosti existuje viacero spoločností, ktoré sa zaoberajú biznis analýzou dát (najvýznamnejšie Microsoft, SAP, SAS, IBM, Oracle). Ponúkajú komplexné analytické nástroje, ktoré obsahujú rôzne metódy pre dolovanie v dátach a ich vizualizáciu. Ich používanie je rozšírené najmä v súkromnom sektore (bankovníctvo, poisťovníctvo, energetika, telekomunikácie), kde presné prognózy a znalosti o zákazníkoch pomáhajú firmám zlepšovať ich pozíciu na trhu.

2 Prognózovanie v energetickom sektore

V energetickom sektore, ktorý je poznačený prevádzkovými a regulatórnymi zmenami podmienok, sú presné predikcie dopytu elektrickej energie mimoriadne dôležité. Elektrickú energiu totiž nie je možné skladovať a jej výroba a spotreba musí nielen spĺňať ustanovenia regulátora ale i napĺňať očakávania spotrebiteľov. S nástupom inteligentných meračov energie, ktoré sú schopné odosielať údaje o spotrebe v štvrťhodinových intervaloch, získa distribútor veľké množstvo dát, pomocou ktorých bude možné ľahšie a presnejšie predpovedať spotrebu.

Hlavné faktory, ktoré vplývajú na spotrebu elektrickej energie sú:

- odberno-odovzdávacie miesto (úroveň napätia, typ – domácnosť/firma),
- cyklickosť – sezóna (ročné obdobie), deň (pracovný deň/sviatok/víkend), hodina (časť dňa – ráno/obed/večer/noc),
- počasie (teplota, vlhkosť).

Vedľajšie vplyvy, s ktorými treba pri predpovediach rátať sú tzv. obnoviteľné zdroje energie (solárne panely, veterné elektrárne a pod.), ktoré sa môžu vyskytovať v odberno-odovzdávacom mieste, ale aj čierne odbery. Z hľadiska distribútora elektrickej energie je podstatné vedieť predpovedať súhrnnú spotrebu energie pre všetky jemu pridelené odberno-odovzdávacie miesta (bilančnú skupinu).

Podľa vyhlášky MHSR č. 358/2013 sa budú inteligentné meracie prístroje postupne inštalovať koncovým odoberateľom podľa ich ročnej spotreby a do roku 2020 by malo byť v Slovenskej republike prístrojmi vybavených 80 % miest. Štúdia z roku 2012 [6] uvádza, že odberné miesta na nízkom napätí, ktorých sa zavedenie prístrojov týka, predstavujú približne 2,38 mil. zákazníkov. Príklady uvádzaných pozitív inteligentných meračov sú možnosť vytvorenia tarifných systémov zákazníkom na mieru, nižšia spotreba energie jej racionálnym používaním podľa taríf, ľahšia detekcia a vyčíslenie krádeží, rýchlejšie reakcie na výpadky, menej administratívnych úkonov

a fyzických návštev z dôvodu odpočtov, odpájania a znova pripájania elektromerov, úspory z rýchlejších platieb zákazníkov a *efektívny nákup energie*.

2.1 Metódy pre predpovedanie spotreby energie

Pre prognózovanie sa všeobecne využívajú ekonometrické metódy (regresná analýza) alebo extrapolačné (analýza časového radu). Ekonometrické modely zostavujú príčinné vzťahy medzi spotrebou a nezávislými premennými, ktoré ju ovplyvňujú (napr. počasie, ekonomické faktory). V Taliansku využili na dlhodobú predpoveď spotreby lineárny regresný model, ktorý uvažoval HDP a rast populácie [1]. Vzťahy medzi premennými nemusia byť vždy len lineárne.

Extrapolačné metódy odvádzajú predpovede spotreby elektrickej energie z hodnôt nameraných v minulosti. Najpoužívanejšie sú Box-Jenkinsova metodológia (ARIMA model) a vyhladzovanie časových radov (Holtovo exponenciálne vyrovnávanie). Keďže spotrebu elektrickej energie výrazne ovplyvňujú faktory ako počasie a striedanie ročných období, využívajú sa na jej predpoveď verzie modelov, ktoré ich zohľadňujú (SARIMA, Holt-Wintersove exponenciálne vyrovnávanie). Model SARIMA bol úspešne použitý pre krátkodobú predpoveď spotreby energie v Číne [3].

V našej práci by sme sa chceli zaoberať práve týmito metódami. Extrapolačné metódy sú pomerne jednoduché a vhodné na krátkodobú predpoveď (v prípade inteligentných meračov predpoveď na sériu 15-minútových intervalov dopredu). Hodia sa na predpoveď javov, v ktorých sa predpokladá, že trend sa bude ďalej vyvíjať podľa minulosti a nie je silno ovplyvňovaný náhlymi, prípadne častými, externými zmenami, čo v prípade spotreby elektrickej energie platí.

2.2 Inkrementálne počítanie predikcie

Pre výpočet predpovedí sú potrebné dáta celého časového radu. Vo všeobecnosti platí, že čím dlhší časový rad je k dispozícii, tým je možné vytvoriť presnejší model pre predpovedanie jeho budúcich hodnôt. Zároveň ale s väčším objemom dát sa zväčšuje, kvôli výpočtovej zložitosti metód, čas konštrukcie predikcie. V prípade predpovede spotreby elektrickej energie na základe dát z inteligentných meračov potrebujeme vedieť výsledok predpovede pred príchodom údajov za ďalšiu štvrťhodinu.
Predpokladáme, že počiatočný model predpovede natrénujeme na poskytnutých dátach za obdobie jedného roku. S príchodom nových dát bude možné model aktualizovať a dosahovať tak lepšiu presnosť predpovede. Proces konštrukcie predpovede sa skladá z nasledujúcich krokov:

1. očistenie dát,
2. odstránenie sezónnosti,
3. odhad trendu na ďalšie obdobie,
4. výpočet predpovede (vrátane sezónnosti),
5. aktualizácia modelu a sezónnych faktorov.

Jednotlivé kroky tohto procesu majú rôznu zložitosť a pri veľkých objemoch dát by ich vykonanie mohlo trvať príliš dlho. Preto sa v našej práci chceme zamerať na optimalizáciu jednotlivých krokov (najmä kroku 3) a vytvoriť metódu pre zostavenie predpovede založenú na inkrementálnom výpočte. To znamená, že pri príchode nových dát nebude potrebné prepočítavať celý model, ale využijeme výsledky z predchádzajúceho výpočtu. Možné zlepšenie vidíme vo výpočtoch existujúcich extrapolačných metód (exponenciálne vyrovnávanie založené na rekurzívnom výpočte, ARIMA modely, ktoré využívajú regresnú analýzu a kĺzavé priemery).

Očakávané prínosy a plán overenia

Vytvorením metódy pre predpovedanie spotreby elektrickej energie s inkrementálnym výpočtom očakávame umožnenie analýzy prichádzajúcich údajov z inteligentných meračov v *reálnom čase*. Inkrementálnym počítaním nielen zjednodušíme konštrukciu predikcie ale pravidelné a časté aktualizácie modelu zvýšia jej presnosť. Vylepšená metóda predikcie bude prínosom pre oblasť energetiky. Po nasadení modelu predpovede vytvoreného našou metódou do systému inteligentnej siete, bude distribútor schopný včas predpovedať spotrebu a zefektívniť svoje náklady, ale i prispôsobiť tarify výhodnejšie pre svojich zákazníkov.

Metódu predikcie spotreby plánujeme overiť porovnaním jej výsledkov s výsledkami tradičných extrapolačných metód. Budeme tak môcť zistiť jej presnosť voči pôvodným metódam. Simuláciou prúdu prichádzajúcich dát bude možné overiť, či je naša metóda odolná voči zahlteniu a má vhodnú časovú zložitosť.

Literatúra

1. Bianco, V., Manca, O., Nardini, S.: Linear regression models to forecast electricity consumption in Italy. Energy Sources, Part B: Economics, Planning, and Policy 8, 86-93 (2013)
2. CCC: Challenges and Opportunities with Big Data (2012), http://www.cra.org/ccc/files/docs/init/bigdatawhitepaper.pdf
3. He, H., Liu, T., Chen, R., Xiao, Y., Yang, J.: High frequency short-term demand forecasting model for distribution power grid based on ARIMA. IEEE International Conference on Computer Science and Automation Engineering (CSAE) 3, 293-297 (2012)
4. Hong, W.-C.: Intelligent Energy Demand Forecasting. Springer-Verlag, London (2013)
5. Laney, D.: 3D Data Management: Controlling Data Volume, Velocity and Variety. Gartner (2001)
6. Združenie Dodávateľov Elektriny: Posúdenie výhodnosti zavedenia inteligentných meračov elektriny v podmienkach SR. Accenture (2012)

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 130–135

OpenStack: cloudová platforma typu IaaS

Martin Bobák, Viet Tran, Ladislav Hluchý

Ústav informatiky, Slovenská akadémia vied,
Dúbravská cesta 9, 845 35 Bratislava

Abstrakt. Cieľom toho článku je oboznámiť čitateľa s problematikou cloudového počítania. Ide o technológiu, ktorá umožňuje ponúkať infraštruktúru, platformu, či softvér ako službu. Hlavnou výhodou cloudového počítania je využitie elastických zdrojov na virtuálnych strojoch. Tento koncept priniesol niekoľkonásobné zefektívnenie využitia virtuálnych strojov v dátových centrách.
OpenStack je v súčastnosti najdynamickejšie sa rozvíjajúca cloudová open-source platforma. V článku prezentujeme ako vyzerá životný cyklus virtuálneho stroja v OpenStack-u.

Kľúčové slová: OpenStack, cloudové počítanie, IaaS

1 Úvod

Cloudové počítanie predstavuje výpočtovú paradigmu 21. storočia. Jeho hlavným prínos je efektívnejšie narábanie so zdrojmi. Cloudy rozdeľujeme podľa dvoch základných kritérií:

1. ponúkaná služba: IaaS, PaaS a SaaS
2. prístupnosť: verejný, súkromný a hybridný cloud.

V súčasnosti najrozšírenejšia open-source cloudová platforma je OpenStack. V rámci tohto článku sme spravili demonštráciu toho, ako vyzerá životný cyklus virtuálneho stroja v tejto cloudovej platforme.

2 Cloudové počítanie

S rozvojom internetu sa postupne začína rozširovať nová výpočtová paradigma – cloudové počítanie. Ide o informatickú technológiu, ponúkanú cez Internet ako službu, v ktorej výpočty prebiehajú mimo jej používateľa. Popularita tohto konceptu vďačí obrovskému zvýšeniu výpočtovej sily procesorov a kapacity pamätí. V dnešnej dobe je možné rýchlo spracovať obrovské množstvo dát na pomerne lacných serveroch. Hlavným prínosom cloudových architektúr je v tom, že ponúkajú elastické zdroje, ktoré si aplikácia môže škálovať podľa potreby. Z pohľadu používateľov (zákazníkov) je výhodné, že platia len za zdroje, ktoré aktívne použili.

Pre cloudové počítanie je charakteristické:

1. **Spoločné zdroje** – zdroje sú v rámci cloudu zdieľané. Hocikto s prístupom do cloudu, môže narábať s týmito zdrojmi.
2. **Virtualizácia** – virtualizácia ponúka efektívny pohľad na infraštruktúru cloudu (abstrahuje od hardvéru). V cloude sa virtualizácia používa na rozdelenie fyzických serverov na virtuálne servery. Minimálny virtuálny server musí mať virtualizované tieto komponenty:
   - procesor
   - operačnú pamäť
   - disk
   - sieťová karta

   Virtualizácia sa dosahuje pomocou techniky hypervisor. **Hypervisor** umožňuje, aby na jednom fyzickom serveri bežalo paralelne viacero operačných systémov a zároveň, aby zdroje boli medzi nimi zdieľané. Zabalením virtuálnych komponentov do virtuálneho stroja sa zabezpečí úplná kompatibilita so všetkými štandardnými operačnými systémami, aplikáciami, či drivermi.
3. **Elasticita** – v koncepte cloudového počítania elasticita predstavuje dynamickú škálovateľnosť. To znamená, že v cloudovom počítaní aplikácie, bežiace v cloude, majú možnosť dostávať zdroje podľa ich aktuálnej potreby. Ideálny cloud poskytuje klientovi toľko výpočtových prostriedkov, koľko aktuálne potrebuje. Tým pádom pri nízkom zaťažení využíva len malú časť fyzického hardvéru. Pri vysokom zaťažení aplikácia využíva hardvér z viacerých serverov. Ideálne sa toto deje automaticky(v dôsledku čoho, používateľ platí len za tie zdroje, ktoré aktívne používal).
4. **Automatizácia** – virtuálne stroje sú poskytované a nasadzované automaticky[1].
5. **„Účet na mieru“** – ako sme už spomenuli predtým, pre cloudy je typické, že pri ich prenajímaní sa platí len za tie zdroje, ktoré používateľ aktívne používal. Z toho sa dá usúdiť, že cloudy sú ekonomicky výhodné.

2.1 Taxonómia cloudov

Existuje viacero kritérií, podľa ktorých je možné klasifikovať cloudy. Jedným z nich je **klasifikácia podľa ponúkanej služby**:

- **IaaS** – tento typ cloudov ponúka infraštruktúru ako službu. Je to najnižšia vrstva, v podstate ponúkajúca ako službu hardvér. Používateľ takéhoto typu cloudu má k dispozícii virtuálny stroj, ku ktorému pristupuje pomocou siete.
- **PaaS** – tento typ cloudov ponúka platformu ako službu. Pod platformou si môžeme predstaviť operačný systém, poprípade nejaké nástroje v rámci daného operačného systému (programovací jazyk, databázy...). Používateľ je v tomto modeli odbremenený od administrácie danej platformy, sústredí sa len na nasadenie či administráciu vlastných aplikácií. Na druhej strane takýmto zjednodušením platí závislosťou aplikácií na platforme ponúkanej cloudom (napríklad keď vyvíja aplikáciu, táto aplikácia je priamo závislá na platforme)

[1] Veľkým spoločnostia trvalo približne 60 – 90 dní, kým nainštalovali, nakofigurovali a nasadili servere do prevádzky. V cloudovom počítaní je to otázka minút maximálne hodín [3].

– **SaaS** – tento typ cloudu ponúka softvér ako službu. Donedávna sa softvér kupoval spolu s licenciu. Licencia bola viazaná na daný počítač a zväčša bola časovo obmedzená[2]. Model cloudov prichádza s nový konceptom ponúkania softvéru. Namiesto kupovania licencie softvéru, je možné si softvér prenajať. Používateľ teda platí len za používanie softvéru. Výhoda tohoto konceptu je v tom, že používateľ je odbremenený od spravovania softvéru, to má na starosti prevádzkovateľ danej služby.

Iným kritériom, podľa ktorého sa delia cloudy je **prístupnosť cloudu**:

– **Súkromné cloudy** – výpočtové zdroje a dáta nachádzajúce sa na tomto type cloudu nie sú prístupné pre verejnosť. Daný cloud sa používa iba na interné účely danej organizácie. Na jednej strane si organizácia môže vytvoriť cloud podľa svojich predstáv. Na druhej strane len ťažko (vo všeobecnosti) dokážu súkromné cloudy konkurovať známym verejným cloudom (či už komerčným, alebo nekomerčným)
– **Verejné cloudy** – poskytovateľ tohoto typu cloudu ponúka prostriedky cloudu ako služby zákazníkom.
– **Hybridné cloudy** – tento typ cloudu je kombináciou predchádzajúcich dvoch typov cloudov. Takýto cloud môže vzniknúť napríklad v situácii, keď sa súkromnému cloudu vyčerpajú zdroje.

3 OpenStack

OpenStack [1] je open-source cloudová platforma typu IaaS. Celá platforma je implementovaná v jazyku Python. Do tohoto projektu je zapojených viac ako 200 spoločností, medzi ktorými sú: AT&T, AMD, Brocade Communications Systems, Canonical, Cisco, Dell, EMC, Ericsson, Groupe Bull, HP, IBM, Inktank, Intel, NEC, Rackspace Hosting, Red Hat, SUSE Linux, VMware, a Yahoo!.

Dá sa povedať, že Openstack je v podstate operačný systém pre cloud. OpenStack sa skladá z nezávislých komponentov spravujúcich rôzne časti infraštruktúry, ktoré medzi sebou komunikujú pomocou aplikačných rozhraní. Jednotlivé prvky poskytujú výpočtové, sieťové a autentifikačné služby.

Výpočtové služby – Nova: k výpočtovým zdrojom sa pristupuje pomocou komponentu Nova. Pomocou tohto komponentu vieme povedať OpenStack-u aký virtuálny stroj potrebujeme (požiadavky sa zadávajú cez Nova-api). Samotné spojazdnenie stroja má na starosti Nova-service, ktorý tiež informuje o momentálnom stave jednotlivých virtuálnych strojov. Nova zabezpečuje (horizontálnu) škálovateľnosť. Stará sa tiež o spravovanie a automatizáciu zdieľania zdrojov vo virtuálnom prostredí. V súčastnosti podporuje 8 typov hypervisorov (napríklad VMware, Hyper-V, KVM...).

Dátové služby – Swift a Cinder: momentálne sa používajú tri typy modely úložísk. Najstarší model rozdeľuje úložisko do **blokov**. Jeho nástupcom je model, v ktorom úložisko reprezentujú jednotlivé **súbory**. Najnovší model úložiska je abstrakciou nad súbormi pomocou **objektov**. Pomocou komponentu **Swift** pristupujeme k objektovému úložisku. Týmto rozhraním sa dá vytvoriť distribuované úložisko na cloudovej

[2] Častokrát sa doplácalo za upgrady, či rôzne iné prevádzkové služby.

platforme. Komponent **Cinder** pracuje s blokovým typom úložiska. Cinder sa používa v situácii, keď virtuálny stroj obsahuje virtualizovaný pevný disk.

Sieťové služby – Neutron: tento komponent slúži na manažment siete (napríklad manažovanie IP adries, izolácia jednotlivých používateľov...). Neutron sa stará o flexibilitu OpenStack-u. Z pohľadu používateľa je Neutron abstrakciou siete – vytvára virtuálnu sieť.

Autentifikačné služby – Keystone: autentifikačný komponent Keystone umožňuje šifrovanú komunikáciou pomocou SSH protokolu. Na začiatku komunikácie sa overí používateľ, na základe čoho sa v systéme vytvorí časovo obmedzený token. Následne sa identita používateľa a jeho prístupové práva overujú pomocou tohoto tokenu. Tento komponent taktiež manažuje prístupové kľúče, ktorými prebieha autentifikácia používateľov.

Spravovanie virtuálnych image-ov – Glance: toto rozhranie ponúka dve základné služby:

1. nahrávanie a sťahovanie image-ov jednotlivých strojov
2. správa meta-dát jednotlivých image-ov

Grafické rozhranie – Horizon: toto rozhranie predstavuje alternatívu ku konzolovému manipulovaniu s platformou OpenStack pomocou GUI.

4 Životný cyklus virtuálneho stroja v prostredí OpenStack

Používateľ si na začiatku musí zvoliť aký typ virtuálneho stroja potrebuje (z pohľadu hardvéru). Jednotlivé hardvérové profily sa v rámci OpenStack-u sa označujú termínom *flavor*. Ich zoznam získame cez príkaz:

```
$ nova flavor-list
```

Výstupom tohoto príkazu je tabuľka, v ktorej sa nachádzajú základné informácie o flavor-och nainštalovaných na danej platforme. Ďalej sa musí používateľ rozhodnúť aký operačný systém bude bežať na danom virtuálnom stroji. Softvérové vybavenie je v OpenStack-u riešené pomocou image-ov. Ponuku jednotlivých image-ov dostane používateľ pomocou príkazu:

```
$ glance image-list
```

Výstupom tohoto príkazu je tabuľka, v ktorej sa nachádzajú základné informácie o image-och nainštalovaných na danej platforme. Výsledný virtuálny stroj je kombinácia hardvéru (flavor) a softvéru (image). Aby sme mohli k danému virtuálnemu stroju pristupovať musíme mu vytvoriť dvojicu kľúčov – súkromný a verejný kľúč. Pomocou týchto kľúčov budeme bezpečne komunikovať s daným virtuálnym strojom pomocou SSH protokolu. Toto vykonáme pomocou príkazu[3]:

```
$ nova keypair-add kluc.pub > kluc.priv
```

[3] Po vytvorení kľúča treba nastaviť prístupové práva tak, aby len vlastník súboru mohol zo súboru čítať a do súboru zapisovať. Zároveň ostatní používatelia nemajú žiadne práva. Toto docielime príkazom **`chmod 600 kluc.priv`**

Týmto sme k verejnému kľúču *kluc.pub* vytvorili súkromný kľúč *kluc.priv* Kľúče spravujeme pomocou príkazov:

```
$ nova keypair-list
$ nova keypair-delete <meno_kluca>
```

Prvý príkaz nám vráti zoznam kľúčov. Pomocou druhého príkazu vieme zmazať kľúč s daným menom. Teraz nám už nič nebráni vo vytvorení virtuálneho stroja. Táto operácia sa robí v rozhraní Nova nasledovným príkazom:

```
$ nova boot <meno_virtualneho_stroja>
        -image <Image_ID>
        -flavor <Flavor_ID>
        -key_name <verejny_kluc>
```

Overenie, či prebehlo vytvorenie nového virtuálneho stroja úspešne zistíme pomocou príkazu:

```
$ nova list
```

Ak už daný virtuálny stroj nepotrebujeme, zrušíme ho príkazom:

```
$ nova delete <ID_stroja>
```

Informácie o danom virtuálnom stroji získame príkazom:

```
$ nova show <ID_stroja>
```

K virtuálnemu stroju sa vieme pripojiť buď cez internetové spojenie[4] (konzola novnc), alebo cez klienta v Jave (konzola xvpvnc). Pred tým treba inicializovať konzolu virtuálneho stroja. To sa dá vykonať nasledovným príkazom:

```
$ nova get-vnc-console <ID_stroja> <novnc|xvpvnc>
```

S virtuálnym strojom vieme taktiež komunikovať pomocou SHH spojenia, ktoré inicializujeme príkazom:

```
ssh -i kluc.priv root@<IP_stroja>
```

5 Záver

Ako vidíme cloudy vytvárajú ilúziu nekonečných výpočtových zdrojov, ktoré sú sprístupnené používateľovi na požiadanie. Z pohľadu prevádzkovateľov je jedným z najväčších prínosov/výhod cloudov to, že vďaka virtualizácii dokážu omnoho efektívnejšie využiť fyzický hardvér. Z pohľadu používateľov cloudy predstavujú častokrát ekonomicky výhodnejšie riešenie, pretože používateľ platí len za tie zdroje, ktoré aktívne používal. To je dôsledok toho, že prostriedky sú dynamicky škálovateľné podľa potrieb aplikácií bažiacich v cloude.

V druhej časti článku sme sa venovali problematike životného cyklu virtuálneho stroja v OpenStack-u. V rámci nej sme sa venovali týmto problémom:

[4] Túto službu musí poskytovať image stroja.

1. vybratie softvéru a hardvéru virtuálneho stroja
2. vytvorenie kľúča, pomocou ktorého budeme pristupovať cez SSH spojenie k danému virtuálnemu stroju
3. vytvorenie (nabootovanie) nového virtuálneho stroja
4. vzdialené pristupovanie k virtuálnemu stroju
5. zrušenie virtuálneho stroja

5.1 Budúca práca – Optimalizácia behu aplikácií v cloudovom prostredí

Keďže OpenStack momentálne podporuje iba cloudy typu IaaS, tak jednou z jeho hlavných nevýhod je neprítomnosť nástroja, ktorý by optimalizoval beh jednotlivých aplikácií. V súčasnej implementácii OpenStack-u je možné dosť neefektívne narábať so zdrojmi. Spôsobuje to hlavne nepodporovanie cloudov typu PaaS a SaaS. V dôsledku čoho v OpenStack-u nie je implementovaná automatická škálovateľnost. Uvažujme napríklad tri aplikácie – Wordpress, databázový server a aplikáciu vykonávajúcu strojové učenie. Naivné spustenie by vyzeralo tak, že každá z týchto aplikácii by bežala na jednom virtuálnom stroji. Toto však nemusí byť optimálny plán behu týchto troch aplikácií. Ak by bolo možné spustiť Wordpress a databázový server na jednom virtuálnom stroji, potom by zostalo viacej zdrojov pre aplikáciu vykonávajúcu strojové učenie. Momentálne je táto úloha predaná používateľovi tejto platformy.

V súčastnosti pracujeme na nástroji, ktorý by pomohol riešiť túto úlohu. V prvej fáze sme si formalizovali celú problematiku v rámci čoho sme vytvorili abstraktný model pre multi-cloudové počítanie [2]. Tento matematický model nám umožňuje efektívne narábať s jednotlivými aplikáciami bežiacimi na virtuálnych strojoch. V budúcnosti chceme na základe tohoto modelu implementovať a otestovať nástroj, ktorý by realizoval optimalizáciu behu aplikácií v cloudovom prostredí. Optimalizácia by mohla byť realizovaná sadou logických pravidiel, ktoré by umožňovali posudzovať jednotlivé plány a na základe ktorých by sa vybral ten najefektívnejší. Túto ideu následne plánujeme rozšíriť do portálu, ktorý by ponúkal svojím používateľom „platformy“ a zároveň by jednotlivé platformy a aplikácie v nich bežali optimálne.

Poďakovanie. Článok bol podporovaný grantom VEGA 2/0054/12 a projektom APVV-0809-11.

Referencie

1. Openstack. `http://www.openstack.org`.
2. Martin Bobák Ladislav Hluchý and Viet Tran. Abstract model of k–cloud computing. In *International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2014*, Xiamen, China, August 2014.
3. Jothy Rosenberg and Arthur Mateos. *The Cloud at Your Service*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2010.

Hluchý, L., Bieliková, M., Paralič, J. (Eds.) : WIKT 2014, pp. 136–141

Paralelné skladanie veľkých dátových korpusov DNA

Peter Kubán, Mária Lucká

Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií, Ilkovičova 2, 842 16 Bratislava
{peter_kuban, maria.lucka}@stuba.sk

Abstrakt. Sekvenovanie a následné skladanie DNA zohráva dôležitú úlohu v medicíne pri diagnostikovaní a klasifikovaní chorôb a čoskoro sa stane dôležitou súčasťou personalizovanej medicíny. Sekvenované genómy ľudskej bunky predstavujú z bioinformatického hľadiska obrovské dátové korpusy, ktorých korektné a rýchle spracovanie je náročné na čas, pamäť a použitie primeraných softvérových nástrojov. V článku prinášame stručný opis niektorých metód spracovania a analýzy veľkých sekvencií DNA a prezentujeme naše skúsenosti zo skladania reálnych génov pomocou existujúcich riešení. Načrtávame prínos a užitočnosť paralelného spracovania, ktoré dokumentujeme výsledkami experimentov.

Kľúčové slová: bioinformatika, sekvenovanie DNA, de novo skladanie, skladanie DNA, mapovanie DNA, paralelný výpočet.

1 Úvod

Bioinformatika kombinuje pri skúmaní, analýze a spracovaní biologických dát počítačové vedy, štatistiku, matematiku a biológiu. Vznikla krátko po objavení prvých postupov použiteľných na sekvenovanie DNA. V pomerne krátkej dobe boli výskumníci schopní generovať veľké množstvá DNA sekvencií. So súčasnými technológiami je sekvenovanie mnohonásobne rýchlejšie, lacnejšie a produkuje enormné množstvo dát, ktoré sa v takomto rozsahu stávajú náročnými na spracovanie a analýzu.

Pojmom genóm sa v biológii označuje súhrn dedičných informácií organizmu. Je kódovaný buď pomocou DNA (deoxyribonukleová kyselina) alebo v prípade rôznych druhov vírusov aj pomocou RNA (ribonukleová kyselina). V bioinformatike sú sekvencie DNA jedným z najdôležitejších zdrojov informácií.

DNA je dvojvláknová makromolekula, v ktorej každé vlákno má opačnú orientáciu (smer). Je formovaná reťazcami jednoduchších molekúl, známych ako nukleotidy. Pre informatikov (bioinformatikov) je najdôležitejšie poznať tzv. bázy nukleotidov, podľa ktorých sa rozlišujú. Tieto bázy sú Adenín (A), Guanín (G), Cytozín (C) a Tymín (T) a teda DNA v informatike sú reprezentované ako reťazce nad abecedou $\Sigma = \{A, C, G, T\}$. Nukleotidy sú navzájom komplementárne - každý nukleotid sa viaže s nukleotidom na opačnom vlákne (A s T, C s G a naopak). Dĺžka genómu je pre rôzne organizmy odlišná. Malé organizmy, akými sú baktérie a vírusy, majú najkratšie známe genómy, ktoré dosahujú niekoľko tisíc nukleotidov (báz). Najdlhšie známe

genómy majú rastliny. Ľudský genóm dosahuje dĺžku približne 3,2 Gb (3,2 Giga báz = 3 200 000 000 báz).

2 Sekvenovanie DNA

Sekvenovanie DNA je proces určovania poradia nukleotidov, resp. báz, vo vlákne DNA. Môže byť použité na sekvenovanie individuálnych génov, väčších genetických regiónov, chromozómov alebo aj celých genómov. Takto získané časti genómu sa nazývajú fragmenty (angl. reads). Sekvenovanie zohráva dôležitú úlohu v medicíne pri diagnostikovaní a klasifikovaní chorôb a v blízkej budúcnosti sa stane dôležitou súčasťou personalizovanej medicíny pre jednotlivcov. Prvých 30 rokov po objavení metód sekvenovania DNA prevládali metódy, ktoré boli veľmi drahé a neefektívne. Až pred dokončením projektu ľudského genómu (Human Genome Project) sa začala o túto oblasť viac zaujímať aj komerčná sféra, čím sa vývoj výrazne urýchlil. Kým zosekvenovanie prvého ľudského genómu trvalo zhruba 10 rokov a stálo približne 3 miliardy amerických dolárov, v súčasnosti, s najmodernejšími prístrojmi, je to možné dosiahnuť v priebehu jedného dňa pri cene blížiacej sa k tisíc dolárom.

Sekvenovanie DNA je na ceste ku každodennému používaniu v medicíne [17]. Nové metódy sekvenovania (metódy druhej a tretej generácie) však so sebou prinášajú aj nové problémy a výzvy. O nových metódach sa tiež hovorí ako o masívne paralelných metódach, pretože na rozdiel od tradičných Sangerových metód (prvá generácia) [18] sekvenujú veľké množstvo fragmentov paralelne. Hlavnými výhodami druhej generácie sú rýchlosť a cena, ale nevýhodami sú krátke dĺžky fragmentov a omnoho väčšia chybovosť. Proces sekvenovania je tiež náhodný – nevieme, z ktorej časti genómu daný fragment pochádza a ani z ktorého vlákna (smeru). Tieto problémy sa obchádzajú zvyšovaním pokrytia, t.j. sekvenovanie prebieha viackrát, čím však vznikajú nové problémy s množstvom dát a ich spracovaním. Metódy tzv. tretej generácie sa snažia obísť nevýhody predchádzajúcich metód. Ich hlavným cieľom je zvýšiť priepustnosť (množstvo fragmentov za čas), zvýšiť dĺžku sekvencií (priblížiť sa alebo prekonať Sangerove metódy), znížiť chybovosť a zachovať klesajúcu cenu. Metódam sekvenovania sa hlbšie venujú napríklad články [7], [17], [18]. Na celom svete sa momentálne využíva viac ako 2000 moderných sekvenovacích prístrojov. V priebehu jedného roka sa vygeneruje viac ako 15 PB (petabajtov) genetických dát [15]. Existuje niekoľko dominantných spoločností [9] zaoberajúcich sa sekvenovaním, ktoré navrhli a zostrojili rôzne sekvenovacie systémy/platformy. K najvýznamnejším z nich patria: 454 Life Sciences [1], Illumina [4], Life Technologies [6], Pacific Biosciences [13], Oxford Nanopore [12].

Rôzne sekvenovacie platformy používajú odlišné postupy a technológie a preto z nich získané sekvencie majú odlišné špecifické vlastnosti (dĺžky fragmentov, chybovosť, rýchlosť sekvenovania, atď.). Napríklad hlavným obmedzením platformy 454 sú tzv. homopolyméry (po sebe idúce rovnaké bázy ako napr. AAA, GGG), pri ktorých vystupuje do popredia problém s určovaním ich početnosti. Dominantným typom chýb pre platformy 454 sú vloženia (angl. insertion - vloženie nesprávnej bázy),

zmazania (angl. deletion - zmazanie bázy). Výhodou oproti iným metódam druhej generácie sú dĺžky získaných fragmentov. Napríklad pri Illumina platformách sú dominantými chybami substitúcie (nahradenie báz) a homopolyméry sú menším problémom. Ale dĺžky fragmentov pri Illumina platformách sú omnoho kratšie. Porovnaním metód a vlastností niektorých súčasných sekvenátorov sa zaoberajú autori v práci [9].

3 Skladanie a mapovanie DNA

Skladanie DNA sekvencií znamená zarovnávanie a zlučovanie fragmentov do väčších DNA sekvencií so snahou zrekonštruovať pôvodnú sekvenciu (pôvodný genóm). Skladanie sekvencií je nevyhnutné, pretože súčasné technológie nedokážu zosekvenovať celý genóm naraz, ale len po menších častiach. Poskladané súvislé sekvencie sa nazývajú kontigy. Je dôležité rozlišovať medzi tzv. de novo skladaním a mapovaním. Kým de novo skladanie sa zameriava na rekonštrukciu genómov, ktoré nie sú podobné žiadnym iným, predtým zosekvenovaným genónom, mapovanie využíva porovnávaciu metódu, pri ktorej sa využíva už existujúca (referenčná) sekvencia rovnakého alebo blízkeho organizmu. V súčasnosti je skladanie pomalšie, výpočtovo náročnejšie a drahšie ako sekvenovanie [2], a preto mu viaceré tímy vedcov venujú pozornosť.

3.1 Mapovanie

Mapovanie DNA je použiteľné vtedy, ak existuje dôveryhodná referencia. Táto referencia sa využíva tak, že sa v nej hľadajú výskyty fragmentov použitím viacerých efektívnych algoritmov [16]. Hlavnými problémami sú dĺžky fragmentov a ich množstvo. Často sa zarovnáva veľké množstvo krátkych fragmentov (milióny až miliardy) k veľkej referenčnej sekvencii/sekvenciám. Takýto proces je výpočtovo a časovo náročný. Mapovanie sa využíva napríklad v populačnej genomike, pri hľadaní variácií a mutácií, v diferenciálnej genomike a už spomínanom skladaní.

Existuje viacero programových celkov, ktoré pomocou efektívnych metód a algoritmov riešia problém mapovania k referenčnej sekvencii. K najpopulárnejším patria: BWA (Burrows-Wheeler Aligner), Bowtie, s jeho novšou verziou Bowtie2 (http://bowtie-bio.sourceforge.net) a ďalšie.

3.2 De novo skladanie

Pri de novo skladaní nie je k dispozícii referenčná sekvencia a metóda, ktorá sa používa na skladanie, je prekrývanie fragmentov. Hľadanie prekryvov medzi veľkým množstvom fragmentov je náročné rovnako, ako aj následné uchovávanie tejto informácie. Pri de novo skladaní sa najčastejšie využívajú grafy a rôzne grafové algoritmy na ich prehľadávanie. Vrcholy grafov obsahujú fragmenty alebo ich časti a hrany medzi vrcholmi existujú vtedy, ak sa tieto fragmenty prekrývajú. Poskladaná sekvencia potom vzniká prechádzaním grafu. Takýto graf v praxi obsahuje milióny vrcholov

a hrán. Matematicky a výpočtovo je de novo skladanie náročné a patrí do kategórie NP-ťažkých problémov, pre ktoré neexistuje efektívne výpočtové riešenie [16]. De novo skladanie je v súčasnosti najpoužívanejší spôsob objavovania nových sekvencií. Softvér na skladanie genómov je často navrhnutý pre konkrétne dáta vybraného sekvenátora alebo sa dá upravovať pomocou nastaviteľných parametrov. S rastúcim množstvom dát bolo nutné navrhnúť nové metódy a algoritmy a na zrýchlenie času výpočtu bola nevyhnutná ich paralelizácia. Vznikli viaceré paralelné nástroje ako napríklad ABySS [19], PASHA [7] alebo Ray [2], ktoré využívajú výhody vysokovýkonných počítačov (angl. high-performance computers) so špecifickou architektúrou pre vybraný nástroj.

4 Veľké dátové korpusy a DNA

Nárast množstva dát v rôznych oblastiach a doménach priniesol nové výzvy a príležitosti v mnohých odboroch – od vedy a techniky až po biológiu. Veľké dátové korpusy (angl. Big Data) predstavujú nové príležitosti pre inovácie, produktivitu a pre vznik nových technológií. Výskumníci potrebujú spracovávať veľké množstvo prichádzajúcich dát a to veľmi rýchlo [15].

Objem dát vyprodukovaných sekvenovaním, mapovaním a analýzou genómov, postavil genomiku pred problém spracovania veľkých dátových korpusov. Sekvenovanie a analýza genómov mnohých ľudí (a iných organizmov) veľmi rýchlo pridáva stovky terabajtov dát [15].

Pre súčasné metódy sekvenovania a skladania sú hlavnými komponentmi: vysoko výkonné počítanie, disky (dátové úložiská) a siete. Najpomalšie komponenty sú práve disky a siete. Existuje veľa dôvodov znemožňujúcich úplnú paralelizáciu, napríklad z dôvodov sekvenčných častí a obmedzení algoritmov. Ďalšími obmedzeniami sú napr. prístup k zdieľaným zdrojom, vstupno/výstupné operácie a ďalšie [8].

Na prácu s takýmto objemom dát osobné počítače nepostačujú, a preto mnohé technológie a riešenia využívajú vysokovýkonné počítače a masívne paralelné prístupy. Využívajú pritom moderné technológie, akými sú napríklad MPI (angl. Message Passing Interface)/OpenMP (angl. OpenMulti-Processing), technológia MapReduce a jej implementácia Hadoop, a iné [10], [3], [11]. Pri našich experimentoch sme používali programy využívajúce technológie MPI/OpenMP (ABySS, PASHA, Ray).

5 Experimenty

Pre dostupné dáta zo zdravého tkaniva človeka (pokrytie 30x, platforma Illumina HiSeq, dĺžka fragmentov 100, počet fragmentov 1 353 241 584, binárny súbor o veľkosti ≈100GB) a rakovinového tkaniva človeka (pokrytie 60x, platforma Illumina HiSeq, dĺžka fragmentov 100, počet fragmentov 2 567 690 261, binárny súbor o veľkosti ≈200GB) sme odskúšali niekoľko existujúcich paralelných skladačov: ABySS [19], PASHA [7] a Ray [2]. Experimenty sme vykonávali na HPC klastri STU (https://www.hpc.stuba.sk) IBM iDataPlex, ktorý pozostáva z 52 výpočtových uzlov IBM iDataPlex dx360 M3. Z tohto počtu sú 4 osadené dvoma GPU akcelerátormi

NVIDIA Tesla. Výpočtový uzol má CPU 2x 6 jadrový Intel Xeon X5670 2.93 GHz. Operačná pamäť klastra je 48GB a lokálny disk má kapacitu 1x 2TB. Aj napriek veľkým pamäťovým možnostiam sa nám z dôvodu rôznych implementácií paralelizmu v týchto skladačoch, nepodarilo dokončiť skladanie na celých dátach z dôvodu nedostatku pamäte. Preto sme dáta rozdelili na menšie časti, podľa chromozómov, s použitím nástroja Samtools [5] (dáta už boli spracovávané a zarovnávané a preto obsahujú informácie o pozíciách a chromozómoch). Na vybranom chromozóme (chromozóm 22, počet fragmentov 15 198 275) zo zdravého tkaniva sme spustili spomínané skladače a získali poskladané sekvencie. Sekvencie sme následne porovnali s referenčnými sekvenciami a vyhodnotili použitím Perl skriptu assess_assembly.pl (http://www.plantagora.org). Najlepšie výsledky dosiahol ABySS ≈60% pokrytie referencie poskladanými sekvenciami, avšak očakávali sme výsledky pokrytia aspoň 80-85%. PASHA dosiahol ≈58% a Ray ≈55%. Najlepší výpočtový čas dosahoval Ray ≈ 3 hodiny, ABySS a PASHA potrebovali ≈ 4-5 hodín, pričom sme použili 48 procesorov.

6 Záver a budúca práca

V experimentoch sme zistili, že narábanie s veľkými biologickými dátami je výpočtovo a časovo náročné aj s použitím moderných paralelných nástrojov. Pri experimentoch sme narazili na viacero problémov (nedostatok pamäte, veľká pamäťová a časová náročnosť pri skladaní a následnom zisťovaní výsledného pokrytia, veľké množstvo krátkych kontigov a kontigov kratších ako dĺžky fragmentov). Preto sme použili menšie dátové súbory, avšak dosiahnuté výsledky napriek tomu neboli uspokojivé, keďže výsledné pokrytie vybraného chromozómu nebolo dostatočné. Preto chceme v budúcnosti vyhodnotiť výsledky aj iným spôsobom, a to zistením ako dobre sa dajú sekvencie namapovať k známym sekvenciám ľudských génov. Na toto porovnanie sme začali pracovať na tvorbe vlastného softvéru, v rámci ktorého navrhujeme vlastné algoritmy alebo sa pokúšame zrýchliť existujúce algoritmy zarovnávania a mapovania. V rámci tohto procesu sa snažíme využiť paralelizmus aj použitím grafických procesorov, ktorých výpočtový výkon v posledných rokoch dramaticky stúpa.

7 Poďakovanie

Tento článok vznikol vďaka podpore projektov VEGA 1/0752/14 "Inteligentná analýza veľkých údajových korpusov sémanticky-orientovanými a bio-inšpirovanými metódami v paralelnom prostredí" a vďaka podpore v rámci OP Výskum a vývoj pre projekt: Medzinárodné centrum excelentnosti pre výskum inteligentných a bezpečných informačno-komunikačných technológií a systémov, ITMS 26240120039, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja a vďaka projektu DNApuzzleDNA, FIIT STU, vďaka ktorému sme mali možnosť vykonávať výpočty na univerzitnom klastri STU.

Literatúra

1. 454 Life Sciences. [Online] http://www.454.com
2. Boisvert S., Laviolette F, Corbeil J. Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies. Journal of Computational Biology. Vol. 17, 11, pp. 1519-1533. (2010)
3. Hadoop. [Online] http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
4. Illumina. [Online] http://www.illumina.com
5. Li H., et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. Vol. 25, 16, pp. 2078-2079. (2009)
6. Life Technologies. [Online] https://www.lifetechnologies.com
7. Liu Y., et al. Parallelized short read assembly of large genomes using de Bruijn graphs. BMC bioinformatics. Vol. 12, 1, p. 354. (2011)
8. Mardis E. R. Next-generation DNA sequencing methods. Annual review of genomics and human genetics. Vol. 9, pp. 387-402. (2008)
9. Miller J. R., er al. Assembly Algorithms for Next-Generation Sequencing Data. Genomics. Vol. 95, 6, pp. 315-327. (2011)
10. MPI. [Online] http://www.open-mpi.org
11. OpenMP. [Online] http://openmp.org/wp
12. Oxford Nanopore. [Online] https://www.nanoporetech.com
13. Pacific Biosciences. [Online] http://www.pacificbiosciences.com
14. Perkel, J. M. Next-Gen Sequencing 2014 Update. http://www.biocompare.com/Editorial-Articles/155411-Next-Gen-Sequencing-2014-Update/. [Online] Biocompare. (2014)
15. Pla B., Jordi J. T. Big Data Challenges in Bioinformatics. www.jorditorres.org. [Online] http://www.jorditorres.org/wp-content/uploads/2014/02/XerradaBIB.pdf. (2014)
16. Pop M. Genome assembly reborn: recent computational challenges. Briefings in bioinformatics. Vol. 10, 4, pp. 354-366. (2009)
17. Reis-Filho J. S. Next-generation sequencing. Breast cancer research. Vol. 11. (2009)
18. Sanger F., Nicklen S. DNA sequencing with chain-terminating inhibitors. Proceedings of the National Academy of Sciences of the USA. Vol. 74, 12, pp. 5463-5467. (1977)
19. Simpson J. T., et al. ABySS: a parallel assembler for short read sequence data. Vol. 19, 6, pp. 1117-1123. (2009)

Index autorov

Ladislav Hluchý, Mária Bieliková, Ján Paralič (Eds.)

WIKT 2014: 9th Workshop on Intelligent
and Knowledge Oriented Technologies

Zborník príspevkov
1. vydanie
157 strán, 80 výtlačkov
Tlač Nakladateľstvo STU v Bratislave
Rok vydania 2014

ISBN 978-80-227-4267-2