gSemSearch: Objavovanie relácií v kolekciách textových a grafových dát

Michal Laclavík, Martin Šeleng, Marek Ciglan, Štefan Dlugolinský, Ladislav Hluchý

Ústav informatiky, Slovenská akadémia vied,
Dúbravská cesta 9, 845 07 Bratislava
michal.laclavik@savba.sk

Abstrakt. V článku opisujeme „work-in-progres“ algoritmy a užívateľské rozhranie zamerané na objavovanie relácií medzi entitami v kolekciách textových a grafových dát. Problém objavovania relácií v grafoch je zaujímavý nie len z hľadiska potenciálnych aplikácií ale aj z hľadiska dopytovania sa nad rozsiahlymi grafovými dátami, ktoré sú prirodzene prítomné v mnohých aplikáciách v čoraz väčšej miere.

Kľúčové slová: graf, sieť, objavovanie relácií

1 Úvod

Dáta vo forme grafov alebo sietí sa čoraz častejšie objavujú v rôznych aplikáciách ako prirodzená forma reprezentácie dát. Sú to napríklad:

- Sociálne siete: obsahujú množstvo grafových dát nie len ako prepojenie medzi priateľmi ale aj ich ďalšia interakcia nad artefaktmi ako správy, statusy, fotky a podobne.
- Emaily: [6] obsahujú takisto rozmer komunikačnej sociálnej siete, ktorá môže byť napojená na ďalšie objekty obsiahnuté v emailoch ako firmy, organizácie, dokumenty, linky, čas a podobne.
- Telekomunikácie: obsahujú sieť navzájom komunikujúcich ľudí formou hovorov, SMS s ďalšími metadátami ako čas alebo miesto.
- Internet: sieť odkazov a prepojení.
- Wikipédia: sieť prepojení a hierarchie jednotlivých tematických stránok ako aj jazykových mutácií
- LinkedData[2]: neustále sa rozširujúca sémantická sieť dát obsahujúca metadáta o ľuďoch, publikáciách, geografických miestach a iných entitách.

Dáta vo forme grafov reprezentované hranami a vrcholmi sú teda prítomné v dostupných dátach v čoraz väčšej forme. Na vyhľadávanie v týchto dátach je možné použiť grafové algoritmy alebo algoritmy pre siete malého sveta, pretože vyššie vymenované siete majú tieto vlastnosti.

V posledných rokoch nastal aj rozvoj grafových databáz:

- Triple stores: zahŕňajú databázy zamerané na ukladanie sémantiky vo forme trojíc. Sú to napríklad Virtuoso, Sesame, OWLim alebo SHARD[3].

[2] http://linkeddata.org/
[3] http://sourceforge.net/projects/shard-3store/

- Grafové databázy a grafové API : Neo4j, Virtuoso, SGDB[4], JUNG umožňujú manipuláciu s grafmi, ich traverzovanie alebo perzistentné uloženie údajov.
- Blueprints[5] je jednotné Java API pre prístup ku grafovým databázam, podobným spôsobom ako JDBC pre relačné databázy.

Pri dopytovaní sa nad grafovými dátami je dôležité hlavne rýchle traverzovanie grafu. Problémom je však, že vrcholy grafu sú prepojené rôzne a teda ide o náhodný prístup k vrcholom grafu. Všetky grafové databázy sa snažia čo najväčšiu časť dát obsiahnuť v operačnej pamäti. Dosiaľ neexistujú dostupné škálovateľné riešenia na dopytovanie a spracovanie grafov aj keď firmy ako Facebook alebo Google potrebujú riešiť spracovanie veľkých grafových dát. Google napríklad publikoval svoje riešenie Pregel [1] pre dávkové spracovanie grafov, podobne vznikajú open source riešenia (napríklad Hama[6]) na základe idei Pregel. Podľa našich informácií však neexistujú žiadne škálovateľné riešenia na real-time dopytovanie sa grafových dát.

2 Šírenie aktivácie a dopytovanie sa v reálnom čase

V našej práci sa snažíme využiť šírenie aktivácie na grafe multi-dimenzionálnej sociálnej siete podobne ako IBM Galaxy [2] kde bol predstavený koncept multi-dimensionálnej sociálnej siete na spracovanie textov. Šírenie aktivácie bolo použité napríklad aj na stránke Foaf.sk [3, 9, 10], alebo v poradenských systémoch [4, 10]. Na optimalizované šírenie aktivácie bola navrhnutá aj platforma Simple Graph Database SGDB [5], ktorá optimalizuje ukladanie informácií o vrcholoch a hranách v key-value store (súčasná implementácia používa Tokyo cabinet). Pre dopytovanie a traverzovanie grafu dosahuje lepšie výsledky ako napríklad Neo4j [5]. V našej budúcej práci by sme chceli vytvoriť škálovateľné riešenie pre grafové dopytovanie pomocou kombinácie SGDB a škálovateľného key-value store ako Apache Cassandra[7] alebo HBase[8].

Pri šírení aktivácie prehľadávame iba časť grafu, ktorá však dosť rýchlo narastá. Ide o siete malého sveta, ktoré majú krátke cesty prepojenia vrcholov, teda na niekoľko prechodov hranami, sa viem dostať do akéhokoľvek vrcholu v grafe. Preto aj pri rýchlom traverzovaní grafu, je potrebné optimalizovať algoritmus šírenia aktivácie, alebo iný algoritmus prehľadávania grafu, za účelom hľadania relácií. Spravidla väčšina algoritmov vychádza z traverzovania pomocou prehľadávania do šírky. Tu je nutné optimalizovať algoritmus na hĺbku prehľadávania, avšak algoritmus by mal vziať do úvahy aj topológiu grafu.

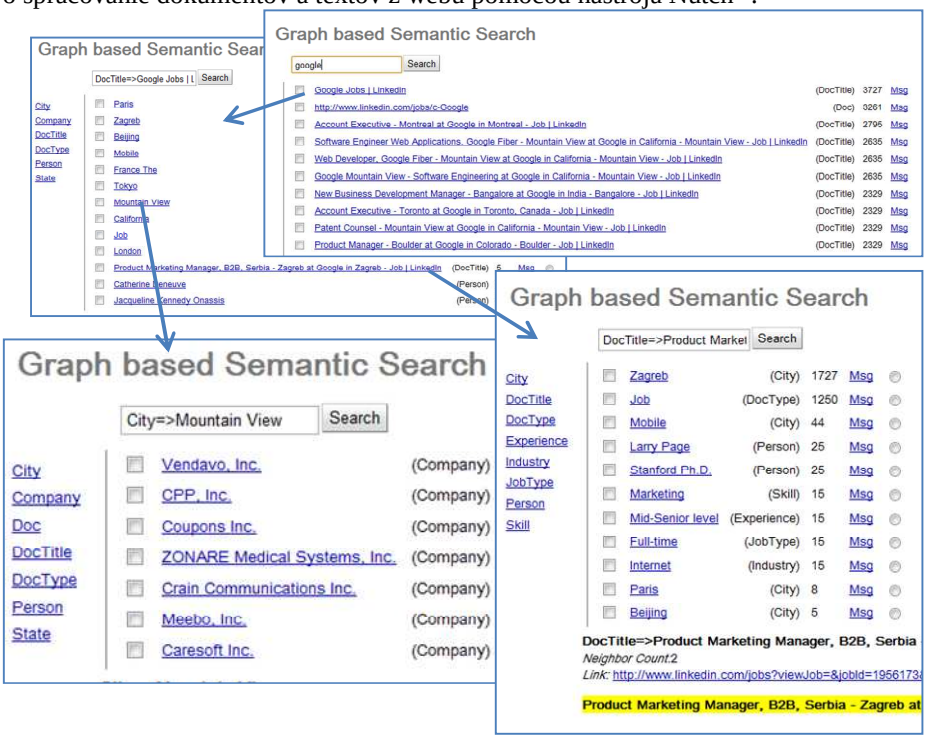
V našej súčasnej implementácii (demo pre Enron email corpus dostupné na webe[9]) [6] sme topológiu vzali do úvahy tak, že obmedzujeme celkový počet navštívených (spracovaných vrcholov), pričom ak má spracovávaný vrchol viac susedov ako zostávajúci počet vrcholov na spracovanie, tento vrchol sa preskočí a spracuje sa ďalší v poradí. Takýto algoritmus nám zabezpečí, že ukončí prehľadávanie do

[4] http://ups.savba.sk/~marek/sgdb.html
[5] https://github.com/tinkerpop/blueprints/wiki/
[6] http://incubator.apache.org/hama/
[7] http://cassandra.apache.org/
[8] http://hbase.apache.org/
[9] http://ikt.ui.sav.sk/esns/

stanoveného času, vyžadovaného aplikáciou, (napríklad 1/3 sekundy), avšak pri vrcholoch z väčším počtom susedov, môže zlyhať a nevráti relevantné výsledky. Je teda potrebné optimalizovať samotnú štruktúru grafu (definovať orientované, typové hrany) ako aj samotný algoritmus hľadania relácií (napríklad aktiváciu po typových hranách, alebo vynechanie určitých vrcholov, prípadne grafové transformácie).

3 gSemSearch

V našej predchádzajúcej práci [6][7][8] sme vytvorili prototyp, ktorý extrahuje multi-dimenzionálne sociálne siete z emailov. V súčasnosti sme rozšírili riešenie o spracovanie dokumentov a textov z webu pomocou nástroja Nutch[10].



Obr.1. Vyhľadávanie v pracovných ponukách LinkedIn. Vpravo hore: Fultextové vyhľadávanie vo vrcholoch grafu. Po kliknutí na linku začíname navigáciu v relevantných objektoch (vrcholoch) grafu. Vľavo hore: Po kliknutí na „Google Jobs“ (vpravo hore) sa zobrazia relevantné objekty ako mestá dokumenty alebo ľudia (vľavo hore). Vľavo dole: Zobrazenie relevantných objektov k mestu „Moutine View“ s obmedzením na firmy. Vpravo dole: Zobrazenie relevantných objektov k pracovnej ponuke.

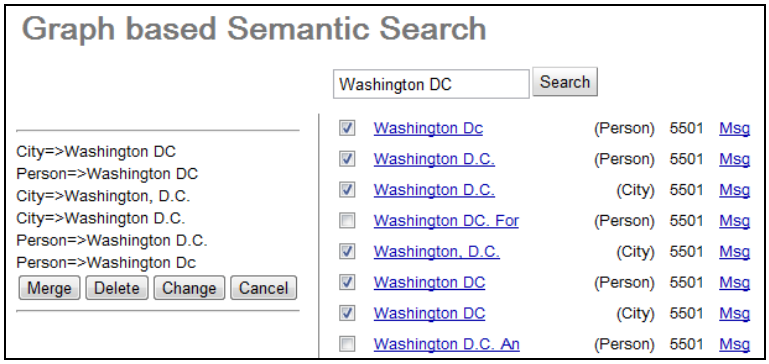
Pomocou nástroja Ontea[11] (vo forme plugin pre Nutch alebo priamo ako jar pre Hadoop) sa nájdu objekty vo forme párov kľúč – hodnota, ktoré sa pre jeden spracovaný dokument (emailová správa, webová stránka) organizujú do stromov.

[10] http://nutch.apache.org/
[11] http://ontea.sourceforge.net/

Takýto strom je potom zaradený do grafu (siete), kde pár kľúč - hodnota reprezentuje vrchol grafu [8]. Na tieto grafy aplikujeme šírenie aktivácie tak ako bolo opísané v [7]. Výsledkom je že pre vrchol grafu dostaneme najrelevantnejšie vrcholy (objekty) obsiahnuté v email archívoch. Pričom tieto vrcholy nemusia byť priamo spojené s vrcholom z ktorého vychádza aktivácia. Pri týchto operáciách dochádza k real-time dopytovaniu grafových dát čo je potrebné optimalizovať.

Na obrázku 1 je zobrazené užívateľské rozhranie ktoré umožňuje odhaľovanie relácií pomocou algoritmov šírenia aktivácie. Rozhranie je implementované ako GWT aplikácia a je oddeliteľné od implementácie grafovej databázy, pretože sa pripája na grafovú databázu pomocou Blueprints API. Podobne je možné implementovať iný algoritmus šírenia aktivácie alebo vyhľadávania relácií v grafe.

Okrem vyhľadávacieho rozhrania a navigácie v grafe podporuje rozhranie aj užívateľskú interakciu a manipuláciu grafových dát. Užívateľ môže zlučovať, vymazávať alebo meniť typ vrcholov grafu (obrázok 2). Tieto zmeny sa okamžite odrazia aj na vyhľadávaných reláciách. Okrem toho je tieto zmeny na grafe možné použiť na dodatočné vylepšenie extrakcie entít z textových dát.



Obr.1. Užívateľská interakcia s dátami na grafe. Pri vyhľadaní mesta „Washington DC“ dostanem viacero vrcholov reprezentujúcich toto mesto s rôznymi napojeniami na iné objekty. Pri označení objektov môžem tieto zlúčiť do jedného objektu (vrcholu grafu), vymazať niektoré objekty alebo zmeniť typ nesprávne identifikovaných objektov, napríklad Person na City. Takáto zmena dát má okamžitý dôsledok na výsledky vyhľadávania relácií.

Dôležitou funkciou gSemSearch je aj overenie správnosti nájdenej relácie. Napríklad keď k človeku alebo firme vyhľadáme telefónne číslo, je dôležité aspoň čiastočne overiť či je to naozaj číslo človeka alebo firmy ktorému chceme zavolať. Toto je možné zobrazením relevantného dokumentu, ktorý je najbližšie vyhľadávaným objektom a spravidla by ich mal obsahovať. S touto vlastnosťou je možné optimalizovať a vytvárať nové algoritmy hľadania relácií s ich čiastočným subjektívnym overením správnosti. V budúcnosti plánujeme aj rozšírenie vyhľadávania o aktiváciu viacerých vrcholov grafu. V súčasnosti je možné aktivovať iba jeden vrchol grafu.

4 Záver

Pomocou vytvoreného prototypu je možné vyhľadávať relácie medzi objektmi, ktoré sú obsiahnuté v textoch alebo sieťach. Vytvorený prototyp gSemSearch je dostupný ako open source. Zatiaľ sme ho overovali iba na grafoch a sieťach extrahovaných z emailov a webu, ale predpokladáme že by mohol pomôcť pri vyhľadávaní v akýchkoľvek grafových dátach. Prototyp sme testovali aj na dátach zo simulácie správania sa davu v meste, kde grafové dáta reprezentovali udalosti v systéme.
Výhodnou vlastnosťou riešenia sa nám javí aj možnosť interakcie užívateľa s dátami, ktoré má okamžitý dopad na vrátenie výsledkov vyhľadávania relácií.

Pre použitie na textových dátach by bolo pravdepodobne vhodné doplniť manuálne značkovanie zaujímavých objektov, ktoré by sa potom objavili v dátach.
Dôležité pre použiteľnosť riešenia je aj jeho škálovateľnosť. Súčasná implementácia, ktorá využíva Blueprints in-memory úložisko je schopná pri 4GB operačnej pamäte alokovanej pre Java VM pracovať s dátami o veľkosti približne 3 milióny vrcholov.

Poďakovanie: Táto práca vznikla s podporou projektov TRA-DICE APVV-0208-10, SMART II ITMS: 26240120029, VEGA 2/0184/10 a Projekt ITMS: 26240220029.

Literatúra

1. Grzegorz Malewicz, Matthew H. Austern, Aart J.C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. 2009. Pregel: a system for large-scale graph processing - "ABSTRACT". In *PODC '09*. ACM, New York, NY, USA, 6-6. DOI=10.1145/1582716.1582723 http://doi.acm.org/10.1145/1582716.1582723
2. Judge, J., Sogrin, M., Troussov, A.: "Galaxy: IBM Ontological Network Miner" In: *Proceedings of the 1st Conference on Social Semantic Web*, Volume P-113 of Lecture Notes in In-formatics (LNI) series (ISSN 16175468, ISBN 9783-88579207-9). (2007)
3. Peter Vojtek a Mária Bieliková: Vhodnosť lokálneho ohodnocovania grafu v Sociálnej sieti obchodného registra SR; In *WIKT 2009 proceedings*. Editor František Babič, Ján Paralič. - Košice : CIT, FEEI TU Košice, 2009. ISBN 978-80-89284-42-9, p. 59-64.
4. Troussov, A., Parra, D., and Brusilovsky, P. Spreading Activation Approach to Tag-aware Recommenders: Modeling Similarity on Multidimensional Networks. In: D. Jannach, et al. (eds.) *Proceedings of Workshop on Recommender Systems and the Social Web at the 2009 ACM conference on Recommender systems*, RecSys '09, New York, NY, October 25, 2009.
5. Ciglan M., Nørvåg K.: SGDB - Simple graph database optimized for activation spreading computation, *Proceedings of GDM'2010* (in conjunction with DASFAA'2010)
6. Michal Laclavík, Štefan Dlugolinský, Marcel Kvassay, Ladislav Hluchý: Email Social Network Extraction and Search; In NextMail 2011 workshop, In *The 2011 IEEE/WIC/ACM WI-IAT 2011* - Los Alamitos : IEEE Computer Society, 2011, p. 373-376. ISBN 978-0-7695-4513-4, DOI 10.1109/WI-IAT.2011.30
7. Michal Laclavík, Marcel Kvassay, Štefan Dlugolinský, Ladislav Hluchý: Use of Email Social Networks for Enterprise Benefit; In IWCSN 2010, *IEEE/WIC/ACM WI-IAT*, 2010
8. Michal Laclavík, Ladislav Hluchý: Využitie sociálnych sietí pri vyhľadávaní v emailoch. In WIKT 2010 - Bratislava : ÚI SAV, 2010, p. 68-71. ISBN 978-80-970145-2-0
9. SUCHAL, J.: On Finding Power Method in Spreading Activation Search. In: SOFSEM 2008: Volume II – Student Research Forum, 2007, p. 124-130.
10. Suchal, Jan - Navrat, Pavol: Full Text Search Engine as Scalable k-Nearest Neighbor Recommendation System. In: Artificial Intelligence in Theory and Practice III IFIP Advances in Information and Communication Technology, 2010, Volume 331/2010, 165-173.