VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

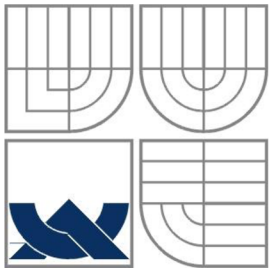
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SÉMANTICKÁ BLÍZKOST TERMÍNŮ

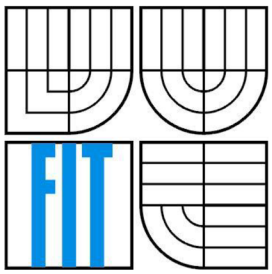
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE JÁN NOVÁK
AUTHOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SÉMANTICKÁ BLÍZKOST TERMÍNŮ

SEMANTIC SIMILARITY OF TERMS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — JÁN NOVÁK
AUTHOR

VEDOUCÍ PRÁCE — ING. MAREK SCHMIDT
SUPERVISOR

BRNO 2009

Abstrakt

Tato práce si klade za cíl zpracovat poznatky o automatickém rozpoznávaní termínů a metodách výpočtu podobnosti termínů podle spoluvýskytu a na základe těchto poznatků navrhnout a implementovat systém pro výpočet sémantické blízkosti termínů z rozsáhle kolekce dokumentů.

Abstract

The goal of this thesis is processing knowledge about Automatic Term Recognition and methods of computing term similarity according to co-occurence and on ground of this knowledge suggest and implement system for computing semantic similarity of terms from large collection of documents.

Klíčová slova

termín, latentní sémantická analýza, Random Indexing, SVD, automatické rozpoznávaní termínů

Keywords

Term, Latent Semantic Analysis, Random Indexing, SVD, Automatic Term Recognition

Citace

Novák Ján: Sémantická blízkost termínů, bakalářská práce, Brno, FIT VUT v Brně, 2009.

Sémantická blízkost termínů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pána Ing. Marka Schmidta.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Ján Novák
19. mája 2009

Poděkování

Za odborné vedení a cenné rady děkuji vedoucímu bakalářské práce Ing. Markovi Schmidtovi.

Obsah

1 Úvod

Automatické rozpoznávanie termínov aj zisťovanie blízkosti slov sú oblasti, ktorými sa zaoberá spracovanie prirodzeného jazyka (Natural language processing – NLP). Spracovanie prirodzeného jazyka je jednou z hlavných aplikácii umelej inteligencie. Cieľom je, aby mohol človek komunikovať s počítačom prirodzeným jazykom, napr. angličtinou alebo slovenčinou, namiesto špeciálne vytvorených jazykov a rozhraní. Používanie prirodzeného jazyka by veľmi zjednodušilo prácu s počítačom. Počítač musí nielen pochopiť a spracovať príkaz, ale aj vytvoriť správnu a zrozumiteľnú odpoveď. Ako príklad môže poslúžiť sémantický web, keď nebude potrebné do vyhľadávača zadávať heslá, ale skutočne zmysluplné otázky a systém poskytne relevantné odpovede.

Termín je slovo alebo slovné spojenie s presne vymedzeným významom používané záväzne v určitom vednom alebo pracovnom obore [1]. Pre systémy NLP zamerané na určité vedecké odbory je potrebné vytvoriť rozsiahle databázy termínov. Tie sa potom využívajú pri indexácii, klasifikácii, preklade alebo dolovaní údajov. Vyhľadávanie termínov môže prebiehať dvoma spôsobmi. Môžeme dať počítaču zoznam slov, ktoré považujeme za termíny. Druhý spôsob je, že počítač sám určí, ktoré výrazy považuje za termíny. Tomu sa hovorí automatické rozpoznávanie termínov. Hlavným problémom je rozoznať termíny od bežne používaných slov a fráz. Túto úlohu ďalej sťažuje časovanie, skloňovanie a používanie synoným, homoným a viacvýznamových slov v bežnom jazyku. Väčšina systémov nepracuje úplne automaticky, ale funguje ako automatizovaná podpora vyhľadávania vykonávaného človekom. Pre človeka je vyhľadávanie termínov náročný proces, najmä pri veľmi obsiahlych textoch. Automatické vyhľadávanie však zatiaľ neprináša dostatočné výsledky. Preto sa často používa kombinácia, keď sa automaticky vyberú vhodní kandidáti na termíny a človek z nich vyberie skutočné termíny.

Sémantická blízkosť sa tiež niekedy nazýva aj vzdialenosť alebo príbuznosť. Jedná sa o metriku pre termíny, ktorá sa určuje podľa podobnosti ich významu. Môže sa určovať aj pre dokumenty, keď vyjadruje podobnosť obsahov dokumentov. Blízkosť termínov znamená, ako veľmi súvisí termín A s termínom B. Je to obojstranný vzťah, teda platí, že blízkosť A a B je rovnaká, ako blízkosť B a A. Blízke slová sú napríklad synonymá alebo antonymá. Používa sa niekoľko techník. Podobnosť možno zistiť z ontológii, kde sa počíta napríklad ako najmenšia vzdialenosť dvoch uzlov v acyklickom grafe. Metódy, ktoré zisťujú blízkosť termínov na základe spoluvýskytu v texte, obvykle využívajú štatistické metódy, ako napríklad vektorové priestorové modely. Sémantická blízkosť termínov sa využíva pri konštrukcii koncepčných máp alebo v programoch, ktoré analyzujú pomocou prechádzania internetu náladu a trendy v spoločnosti. Nazývajú sa aj pavúky alebo crawlery. Napríklad služba Googlism dokáže na základe výsledkov vyhľadávača Google zistiť odpovede na otázky Čo to je?, Kde to je?, Kto to je?, Kedy to je?.

V tejto práci sa najprv zoznámime s postupmi pri automatickom rozpoznávaní termínov a výpočte sémantickej blízkosti termínov. Po zoznámení sa s týmito metódami pristúpime k návrhu a implementácií systému na výpočet sémantickej blízkosti termínov. Na záver budú zhodnotené výsledky pokusov na korpusoch, ktoré sú dostupné na FIT VUT v Brne. Na vyhodnotenie sa použijú niektoré štandardné metriky.

2 Metódy automatického rozpoznávania termínov

2.1 Postup pri extrakcii termínov

Štandardný postup pri extrakcii termínov zahŕňa niekoľko procedúr [16]. Nie vždy sa aplikujú všetky procedúry.

Postup:

1. lexikálna analýza
2. odstránenie nevýznamových a nešpecifických slov
3. lematizácia
4. porovnávanie slov so slovníkom
5. váženie

Pri lexikálnej analýze prebieha identifikácia jednotlivých slov a viacslovných výrazov v texte. Slová sa najčastejšie oddeľujú medzerou. Pri niektorých typoch slov je lexikálna analýza zložitejšia. Pri skratkách je treba odlíšiť bodku od bodky na konci vety. Pri slovách so spojovníkom nastáva dilema, či ho brať ako jedno alebo dve slová. Rozpoznanie viacslovných výrazov je oveľa zložitejšie ako pri samostatných slovách. Je potrebné vyriešiť niekoľko otázok. Ako napríklad či je dôležité poradie slov v súsloví alebo či musia nasledovať jednotlivé slová bezprostredne za sebou. Pri tomto postupe môžu byť za termíny označené aj viacslovné výrazy, ktoré v skutočnosti nie sú termíny. Niekedy sa preto na zlepšenie výsledkov používa aj slovník súsloví.

Odstraňovanie nevýznamových a nešpecifických slov prebieha najčastejšie pomocou negatívneho slovníka, kde sú tieto slová vymenované. V prípade neprítomnosti slovníka je možné filtrovanie slov na základe slovných druhov. Vylučujú sa slovné druhy, ktoré sa nevyskytujú v termínoch ako častice alebo spojky.

Lematizácia je redukcia slov na ich základný tvar. Pretože slová sa v texte vyskytujú v rôznych rodoch, časoch alebo pádoch, je potrebné zistiť ich základný tvar, aby bolo možné s nimi pracovať ako s jedným slovom. Lematizáciu vykonáva program, ktorý sa volá lematizátor alebo stemmer. Existuje niekoľko metód. Základné tvary sa môžu určovať na základe slovníka kmeňov alebo koreňov slov, odstraňovaním prefixov a sufixov slov alebo štatisticky na základe variety po sebe nasledujúcich písmen v slovách, keď sa pomocou frekvencie jednotlivých zhlukov písmen stanovuje, či sa jedná o prefix, koreň alebo sufix.

Vážením sa stanovuje relatívna hodnota, ktorá určuje význam výrazu. Váhy je možné určovať podľa rôznych kritérií, napr. podľa frekvencie výskytu alebo umiestnenia v texte (názov, abstrakt,

resumé, ...). Kritéria sa stanovujú podľa oblasti, pre ktorú je systém určený. Napríklad pri systémoch automatického rozpoznávania termínov zameraných na tvorbu indexov sa zisťuje selektívna sila indexovaného termínu. Tá vyjadruje schopnosť termínu vybrať z databázy dokumentov takú množinu, ktorá sa odlišuje od množiny získanej pomocou iného termínu. Niekedy sa označuje aj ako termhood. Termhood vyjadruje silu väzby termínu k danému konceptu [2]. Čím je termín pre daný koncept príznačnejší, tým je termhood väčší. Zisťuje sa najmä podľa frekvencie výskytov v jednotlivých dokumentoch. Pokiaľ sa nejaký termín vyskytuje v jednom, prípadne niekoľkých dokumentoch, výrazne častejšie ako vo zvyšných, je príznačný pre daný dokument, respektíve skupinu dokumentov.

2.2 Vyhľadávanie viacslovných termínov

Viacslovné výrazy sa zväčša zisťujú na základe slovných druhov. Vyskytujú sa v nich prevažne podstatné mená, v menšej miere aj prídavné mená. Problém nastáva pri rozlišovaní termínov od slovných spojení, ktoré sa bežne vyskytujú v texte. Na odlišovanie slúži metrika unithood.

Unithood vyjadruje silu spojenia aspoň dvoch plnovýznamových slov, ktoré odráža nejaký vzťah z reálnej skutočnosti [2]. Najčastejšie sa používa metrika mutual information. Počíta sa podľa vzorca:

$$I(x,y) = log_2 \frac{P(x,y)}{P(x)P(y)}$$

kde P(x) a P(y) sú pravdepodobnosti výskytu slov x a y. P(x,y) je pravdepodobnosť spoločného výskytu slov x a y. Zisťuje sa, ako často sa slová vyskytli samostatne a ako často spolu. Keď sa v texte objavujú častejšie spolu ako samostatne, zväčšuje sa pravdepodobnosť, že sa jedná o termín. Pri tejto metrike nezáleží na poradí slov v súsloví.

2.3 Frekvencia výskytu

Základnou metódou je výber termínov na základe frekvencie výskytu. Vyberá slová na základe počtu výskytov v danom korpuse. Za termíny sa považujú slová a slovné spojenia, ktoré sa v texte vyskytujú často. Niekedy je vhodné odfiltrovať aj výrazy s vysokou frekvenciou, pretože vzhľadom na ich početnosť sa bude pravdepodobne jednať o príliš všeobecné výrazy, než aby sa dali považovať za termíny. Metóda sama o sebe nie je veľmi úspešná, preto sa väčšinou používa spolu s morfologickou analýzou. Tá priradí slovám slovné druhy, na základe ktorých môžeme odfiltrovať nežiaduce slová. Pre vyhľadávanie termínov sú dôležité najmä prídavné a podstatné mená. Ostatné slovné druhy sa v termínoch zväčša nevyskytujú, takže sa nemusia brať do úvahy.

2.4 Backgroundový model

Backgroundový model sa používa spolu s frekvenčnou analýzou. K postupu sa pridáva porovnávanie so slovníkom. Je to zoznam slov a frekvencií ich výskytu vo všeobecnom texte. Väčšiu váhu majú výrazy, ktoré sú vo všeobecnom texte menej časté. Tieto výrazy sú totiž pravdepodobne špecifické pre spracovávaný dokument. Rozoznávame korpusový a doménový model. Korpusový model je vytvorený zo všeobecného textu. Doménový model je tiež tvorený zo všeobecného textu, ale berie do úvahy oblasť, z ktorej je skúmaný text.

3 Sémantická blízkosť

3.1 Latentná sémantická analýza

3.1.1 Princípy

Latentná sémantická analýza (LSA) je technika, ktorá sa venuje analýze vzťahov medzi dokumentmi a slovami. Pri aplikácii pri vyhľadávaní informácii sa nazýva aj latentná sémantická indexácia. Vychádza z distribučnej hypotézy. Podľa nej sa slová, ktoré majú podobný význam, vyskytujú v rovnakom kontexte, teda s rovnakými slovami [3].

LSA využíva distribučnú štatistiku na vytvorenie mnohodimenzionálneho vektorového priestoru, v ktorom sú slová reprezentované kontextovými vektormi. Od ich relatívnych smerníc sa odvádza miera sémantickej podobnosti slov. Pri LSA vektorový priestor predstavuje matica výskytov slov v kontextoch, v ktorej riadky reprezentujú slová a stĺpce reprezentujú kontexty. Ako kontexty sa môžu použiť slová, ale častejšie sa za kontext považuje celý dokument. Jednotlivé bunky matice znamenajú koľkokrát sa dané slovo vyskytlo v danom kontexte. V prípade dokumentov je táto matica väčšinou normalizovaná, aby sa kompenzovala rozdielna dĺžka dokumentov. Každý počet výskytov slova v dokumente je podelený počtom všetkých slov v dokumente. Riadky matice potom vytvoria vektory v mnohodimenzionálnom priestore tak, že prvky vektorov sú frekvencie výskytu slov v jednotlivých kontextoch a počet dimenzií priestoru je rovný počtu stĺpcov, teda počtu kontextov. Vektory nazývame kontextové vektory, lebo vyjadrujú kontext, v ktorom sa dané slovo vyskytuje. Keďže kontextové vektory reprezentujú distribučné profily slov, môžeme vyjadriť podobnosť slov pomocou metód vektorovej podobnosti, napríklad kosínusom uhlov medzi týmito vektormi. [3]

Keďže reálne korpusy môžu obsahovať veľké množstvo dokumentov, vektory budú mať veľký počet dimenzií. Potom by bol výpočet podobnosti vektorov príliš časovo náročný. Matica výskytu termínov v kontextoch je obvykle riedka (väčšina hodnôt je nulová). Je to spôsobené tým, že len veľmi malá časť slov v jazyku sa vyskytuje vo veľkom počte kontextov. Ostatné sa vyskytujú len v malom počte. V bežnej matici výskytov je až 99 % prvkov nulových [3]. Súvisí to so Zipfovým rozložením frekvencií výskytu slov v texte. Podľa tohto rozloženia sa najčastejšie vyskytujúce slovo objavuje v texte približne dvakrát častejšie ako druhé najfrekventovanejšie slovo, a to sa objavuje približne dvakrát častejšie ako štvrté a tak ďalej. Napríklad v Brown Corpuse tvorí 135 slov polovicu obsahu [17]. Na základe toho, že matica výskytov je riedka, redukuje sa počet dimenzií pomocou metódy singulárneho rozkladu (Singular Value Decomposition – SVD).

3.1.2 SVD

Singulárny rozklad je všeobecná metóda lineárnej algebry. Jeho definícia je: „Nech A je ľubovoľná štvorcová matica. Potom existujú ortogonálne matice U a V a diagonálna matica Σ, na ktorej diagonále sú vlastné čísla matice $\sqrt{A^T A}$ tak, že $A = U\Sigma V^T$. Rozklad matice A na matice U, V a Σ sa nazýva singulárny rozklad matice A. “ [4]

Keďže matica termínov v dokumentoch obvykle nebýva štvorcová, ale býva rádu m×n, kde platí m≠n, matica U bude mať rozmery m×m, matica Σ m×n a matica V n×n. Pri rozklade matice výskytu termínov v dokumentoch reprezentuje matica U kontextové vektory termínov, matica V reprezentuje maticu dokumentov. Σ je diagonálna matica m×n, ktorá obsahuje singulárne čísla $\sigma_1, \sigma_2, \ldots \sigma_{\min(m,n)}$. Tieto čísla sú usporiadané zostupne na hlavnej diagonále tak, že platí $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)}$. Toto zostupné usporiadanie znamená, že pre dostatočne presné výsledky nám stačí vypočítať prvých *k* najvyšších singulárnych čísel. Takto dostaneme k-aproximáciu hodnosti matice A. Vhodnú veľkosť čísla *k* treba určiť na základe experimentov. Pre veľké kolekcie dokumentov sa uvádza hodnota medzi 200 a 300 [4]. Takto sa redukuje mnohodimenzionálny priestor na priestor s dimenziou *k* a zároveň sa zachovávajú zhluky podobných si dokumentov a termínov.

Výpočet SVD je veľmi náročný. Rozklad matice m×n má zložitosť $O(\min(mn^2, m^2n))$ [5]. Pri LSA sa singulárny rozklad počíta iba pri indexácii a vyhľadávanie prebieha už vo vypočítanom rozklade. V súčasnej dobe boli vyvinuté techniky, ktoré zrýchľujú SVD. Niektoré využívajú aproximácie, napríklad pomocou metódy Monte Carlo [5]. V roku 2006 bola vypracovaná rýchla prírastková metóda, ktorá má pre r-aproximáciu matice veľkosti p×q časovú náročnosť O(pqr) (*r* je oveľa menšie ako rozmery matice) [6].

3.1.3 Výhody a nevýhody

Hlavnou výhodou LSA je, že využíva vektorový model, ktorý je matematicky dobre popísaný. Z toho vyplýva, že môže využiť celú škálu matematických nástrojov pre prácu s maticami a vektormi. LSA je schopná zistiť blízkosť slov, aj keď sa spolu v texte vôbec neobjavia. Toto je výhodné napríklad pri zisťovaní synoným, ktoré sa obvykle spolu nevyskytujú.

Hlavnou nevýhodou LSA je veľká náročnosť. Aj s použitím SVD trvá výpočet podobnosti termínov dlho, pretože samotné SVD je náročná operácia. Ďalšou nevýhodou je nemožnosť pridávať ďalšie kontexty. Pri pridaní ďalších kontextov je potrebné opäť počítať SVD. V poslednej dobe sa vyvinulo niekoľko metód, ktoré umožňujú pridávať nové dokumenty bez toho, aby bolo nutné opätovne počítať singulárny rozklad pre už spracovanú množinu kontextov. Medzi tieto metódy patrí SVD-updating a Folding-in.

Folding-in je pomerne jednoduchá metóda. Nové dokumenty sú pred pridaním do matíc U a V premietnuté do priestoru redukovaných dokumentov respektíve termínov, čím sa premietne

stav existujúcej databázy do týchto nových stĺpcových a riadkových vektorov. Nevýhodou je, že stav nových vektorov sa nedá premietnuť do existujúcej databázy, čím vzniká chyba.

3.2 Random Indexing

3.2.1 Princípy

Random Indexing bol vytvorený ako alternatíva k LSA. Je založený na práci Pentti Kanevru o riedkych distribučných reprezentáciách. Motivovaná je aj pozorovaním R. Hecht-Nielsena. Ten demonštroval, že v multidimenzionálnom priestore sa oveľa viac vyskytujú približne ortogonálne smery než tie skutočne ortogonálne. Z toho vyplýva, že môžeme použiť náhodné smery, aby sme vhodne aproximovali ortogonalitu [3]. Na základe tohto poznatku vzniklo niekoľko techník na redukciu počtu dimenzií. Najznámejšie sú Random Projection, Random Mapping a Random Indexing. Všetky tieto metódy sú založené na Johnson-Lindenstraussovej lemme. Tá hovorí, že keď premietneme body vektorového priestoru na náhodne vybraný podpriestor s dostatočne veľkým počtom dimenzií, vzdialenosti medzi bodmi budú približne zachované [3]. Preto môžeme počet dimenzií matice F redukovať jej vynásobením náhodnou maticou R :

$$F_{w\times d}R_{d\times k} = F'_{w\times k}$$

Dôležitým rozhodnutím je výber matice R. Pokiaľ by bola ortogonálna, bude platiť $F = F'$. Ak by bola približne ortogonálna, bude platiť $F \approx F'$. Najčastejšie sa používa Gaussovo rozloženie elementov náhodných vektorov v matici R. Avšak existuje aj jednoduchšia metóda. Skoro všetky elementy v týchto vektoroch budú nulové, čo znamená rozloženie s jednotkovou variáciou [3].

Hlavnou myšlienkou Random Indexing je akumulácia kontextových vektorov založených na výskyte slov v kontextoch. Každému kontextu je priradená unikátna reprezentácia, ktorá sa nazýva indexvektor. Indexvektor má dimenziu obvykle rádovo v tisícoch [3]. Je tvorený malým množstvom náhodne rozložených +1 a -1. Ostatné prvky sú 0. Vždy, keď sa slovo vyskytne v kontexte, je k jeho kontextovému vektoru pripočítaný indexvektor daného kontextu. To znamená, že slová sú reprezentované kontextovými vektormi, ktoré sú v podstate súčtom indexvektorov tých kontextov, v ktorých sa vyskytujú. Kontextami sú obvykle dokumenty alebo slová, môžu sa však využiť aj iné druhy kontextov .

Tento prístup je opačný než u LSA, kde najprv vytvoríme maticu spoluvýskytu a potom z nej extrahujeme kontextové vektory. Pri Random Indexing najprv vytvoríme kontextové vektory a potom z nich môžeme zostaviť maticu spoluvýskytu tak, že použijeme kontextové vektory ako riadky matice. Takto vytvorená matica spoluvýskytu bude aproximáciou matice vytvorenej pomocou metódy LSA. Rozdiel je v počte dimenzií vektorového priestoru. Pri LSA sa dimenzia rovná počtu kontextov,

čo môže byť rádovo v miliónoch alebo viac. Pri Random Indexing je počet dimenzií vektorového priestoru rovná dimenzii indexvektorov, čo býva obvykle rádovo v tisícoch. Výsledná matica je teda menšia aj bez použitia SVD, ale zároveň je podobná matici vytvorenej pomocou LSA.

3.2.2 Výhody a nevýhody

Hlavnou výhodou Random Indexing je menšia náročnosť metódy. Nie je potrebné najprv tvoriť maticu spoluvýskytu, ale stačí hneď vytvoriť kontextové vektory. Rovnako nie je potrebné používať techniku SVD na zníženie počtu dimenzií matice.

Ďalšou výhodou je jednoduché pridávanie ďalších kontextov bez toho, aby sme potom museli vykonávať časovo náročné operácie ako SVD. Jednoducho stačí vytvoriť nový indexvektor pre daný kontext a pripočítať ho ku kontextovým vektorom slov, ktoré sa vyskytujú v danom kontexte. Navyše pridanie kontextu nezvýši počet dimenzií. Ten sa pevne nastaví na začiatku ako parameter a neskôr sa už nemení.

Hlavnou nevýhodou je, že prípadne vytvorená matica spoluvýskytov je len približne ortogonálna. Teda nezodpovedá celkom skutočnosti. Je však natoľko podobná skutočnosti, že výsledky metódy Random Indexing sú porovnateľné s LSA.

3.3 Porovnanie Random Indexing a LSA

Na vyhodnotenie metód na zisťovanie podobnosti slov sa používa napríklad časť testu TOEFL (Test of English as a Foreign Language), ktorá je zameraná na vyhľadávanie synoným. Úlohou je vybrať zo štyroch možností tú, ktorá je synonymom pre zadané slovo. Random Indexing dosahuje úspešnosť 64,5% – 67% pri použití spoluvýskytu na základe slov. Pri využití lematizácie textových dát, čo redukuje počet jedinečných slov, bol najlepší výsledok 72%. LSA dosahovala úspešnosť 64,4%. Priemerný výsledok zahraničných uchádzačov na univerzitách v USA bol 64,5%. [3]

3.4 Ďalšie metódy výpočtu

Okrem LSA a Random Indexing existuje ešte veľa iných metód na zisťovanie sémantickej podobnosti. Niektoré vychádzajú z LSA ako napr. zovšeobecnená alebo pravdepodobnostná LSA. Ďalšie využívajú vyhľadávače (napr. Google), ako napr. Pointwise Mutual Inforamtion (PMI) alebo Normalized Google Distance (NGD).

NGD zisťuje, ako spolu súvisia termíny, podľa výsledkov vyhľadávača Google. Vychádza z predpokladu, že podobné termíny sa vyskytujú blízko pri sebe. Počíta sa podľa počtu výsledkov vyhľadávania, v ktorých sa termíny objavili spolu a v ktorých sa objavili zvlášť. Podobnosť sa zisťuje podľa vzorca:

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - \min\{\log f(x), \log f(y)\}}$$

kde M je počet stránok, ktoré našiel Google, f(x) a f(y) sú počty výskytov termínov x a y samostatne a f(x,y) je počet výskytov termínov x a y spolu. Pokiaľ sa dané výrazy nikdy nevyskytli spolu na jednej stránke, výsledok je nekonečno. Pokiaľ sa vždy vyskytli spolu, výsledok je 0. Nevýhodou oproti LSA a Random Indexing je, že slová, ktoré spolu súvisia, sa musia vyskytovať v texte pri sebe, čo nemusí vždy platiť. Napríklad synonymá sa väčšinou v bežnom texte neobjavujú pri sebe.

Stručný prehľad niektorých ďalších metód je možné nájsť napríklad na [13].

3.5 WordNet

Pre zisťovanie podobných slov sa využíva aj WordNet. Je to rozsiahla lexikálna databáza anglického jazyka. Nebola však vytvorená automaticky, ale ručne. Vývoj začal v roku 1995 na Princetonskej univerzite pod vedením profesora Georga A. Millera. Slová sú uchovávané v skupinách synoným, ktoré sa volajú synsety. Tie uchovávajú krátke definície a vzťahy s ostatnými synsetmi na základe sémantických relácii ako hypernymá, hyponymá a meronymá. WordNet rozlišuje medzi podstatnými a prídavnými menami, slovesami a príslovkami, pretože sa správajú podľa rozdielnych gramatických pravidiel. Pokiaľ má slovo viac významov, sú v rozdielnych synsetoch. WordNet je možné využiť aj ako ontológiu. Ontológie v zmysle informatiky sú formálne reprezentácie množiny konceptov v určitej doméne a vzťahov medzi týmito konceptmi. Vo WordNete koncepty predstavujú jednotlivé synsety.

Podobnosť slov sa dá zistiť pomocou niekoľkých metrík, napríklad podľa vzdialenosti uzlov (synsetov) v acyklickom grafe alebo ako lineárna podobnosť. Lineárna podobnosť sa počíta podľa informačného obsahu dvoch synsetov a informačného obsahu ich najviac špecifického spoločného predka. Počíta sa podľa vzorca:

$$sim(A,B) = \frac{2\log(P(A,B))}{\log(P(A)) + \log(P(B))}$$

kde P(A,B) je informačný obsah najviac špecifického spoločného predka uzlov A a B. P(A) a P(B) sú informačné obsahy synsetov A, resp. B. Informačný obsah jednotlivých synsetov je súčasťou databáze. Pokiaľ nemajú spoločného predka, je výsledok -1.0, inak je vrátená hodnota od 0 do 1, kde 1 znamená, že sa synsety A a B sú zhodné.

4 Návrh

4.1 Požiadavky na systém

Prvou požiadavkou na systém je, aby dokázal vyhľadať termíny v texte samostatne, bez nutnosti použiť dopredu pripravený slovník. Pokiaľ je to možné, všetky termíny by mali byť rozpoznané v jednom prechode. Keďže slová sa v bežnom texte nevyskytujú len v jednom tvare, ale sú rôzne skloňované a časované, systém musí pracovať s ich základnými tvarmi. Je potrebné zabezpečiť, aby sa s viacslovnými výrazmi pracovalo ako so samostatným výrazom a nie ako so súborom slov.

4.2 Vyhľadávanie termínov

Po preštudovaní dostupných materiálov som sa rozhodol generovať termíny pomocou frekvencie výskytu s použitím morfologického analyzátora. Backgroundový model som sa rozhodol nepoužiť, aby som urýchlil spracovanie dokumentov. Využitie tohto modelu je vhodné skôr pre systémy na tvorbu indexu, čo nie je cieľom tejto práce. Niektoré testovacie dáta sú navyše príliš všeobecné texty, ktoré nie sú vhodné pre backgroundový model. Rovnako sa nebude zisťovať unithood pri viacslovných výrazoch, pretože by bolo veľmi náročné ho počítať v jednom prechode korpusu. Viacnásobný prechod korpusu by znateľne zvýšil časovú náročnosť. Všetkým slovám v texte bude najprv priradený slovný druh a základný tvar. Na základe týchto informácii budú potom generované unigramy, digramy a trigramy podľa nasledujúcich pravidiel:

- unigramy sú jednoslovné výrazy, ktoré sú označené ako podstatné mená
- digramy sú dvojslovné výrazy, ktoré sa skladajú z:
  - prídavného mena , ktoré je nasledované podstatným menom
  - dvoch podstatných mien
- trigramy sú trojslovné výrazy, ktoré sa skladajú z:
  - dvoch prídavných mien, ktoré sú nasledované podstatným menom
  - prídavného mena, ktoré je nasledované dvojicou podstatných mien
  - trojice podstatných mien

V prípade digramov a trigramov sa musia tieto slovné druhy vyskytovať bezprostredne za sebou. Keďže testovacie údaje sú v angličtine, je prípustné, aby sa medzi podstatnými menami vyskytovala spojka „of“. Záleží na poradí slov. Takže spojenia “book review” a “review of book” sa budú považovať za dva rozdielne termíny.

Získané slovné spojenia sú potom triedené na základe frekvencie výskytu. Odstránia sa spojenia, ktoré sa vyskytujú veľmi málo alebo naopak veľmi často. Tieto hodnoty budú zadávané priamo užívateľom ako parameter.

Popri termínoch sa generuje ešte zoznam ostatných významných slov. Ako významné slová som uvažoval slovesá a prídavné mená. Tieto slová môžu tiež pomáhať určovať blízke slová. Napríklad sloveso „šoférovať“ sa objavuje obvykle spolu s rôznymi druhmi vozidiel ako napríklad auto alebo autobus. Z toho je možné usudzovať, že slová, ktoré sa vyskytujú v okolí slova „šoférovať“, sú sémanticky blízke, pretože sa jedná o rôzne druhy vozidiel. Táto technika bude mať význam najmä pri metóde Random Indexing, kde sa berie do úvahy okolie výrazu a nie celý dokument. Pri LSA majú tieto slová význam v prípade, že sa každá veta považuje za samostatný dokument. Pokiaľ sa za kontext bude považovať celý dokument, významné slová strácajú zmysel, pretože nie je možné určiť, ku ktorým termínom sa viažu.

4.3 Zisťovanie sémantickej blízkosti termínov

Pre zisťovanie sémantickej blízkosti termínov som sa rozhodol použiť Random Indexing a latentnú sémantickú analýzu. Každá metóda vyžaduje rozdielny prístup. Zistené podobnosti termínov sa ukladajú do matice. Kvôli rýchlosti sa na výpočet podobnosti termínov z kontextových vektorov použije jednoduchý skalárny súčin. Výsledky nebudú ovplyvnené, pretože všetky matice a kontextové vektory sú normalizované.

Pri metóde LSA sa vytvorí index, ktorý má podobu normalizovanej matice výskytov termínov a významných slov v dokumentoch. Matica bude normalizovaná tak, že hodnota výskytu výrazu v dokumente sa podelí počtom všetkých indexovaných výrazov v dokumente. Táto matica sa bude redukovať pomocou metódy SVD. Z výsledku tejto metódy sa vytvoria kontextové vektory. Ich skalárne súčiny budú prvkami matice blízkosti termínov. V nej budú len termíny a nie významné slová.

Pri Random Indexing sa netvorí matica výskytov ani neprebieha SVD, ale priamo sa akumulujú kontextové vektory. Z nich sa vytvorí matica podobností termínov rovnako ako v prípade metódy LSA.

4.4 Testovanie

Metódy na zisťovanie sémantickej blízkosti sa obvykle testujú na synonymickom teste. Bežným príkladom je časť testu TOEFL zameraná na synonymá. Tieto testy sú však skoro vždy určené len pre prídavné mená. Keďže samostatné prídavné mená sa medzi termínmi nevyskytujú, nie je možné využiť tento test.

Ďalším používaným prostriedkom je asociačný test. Na testovanie sa používa asociačná norma. Tá je podobná tezauru, aj keď obvykle nebýva taká rozsiahla. Tvorí sa tak, že človek dostane slovo a poskytne niekoľko slov, ktoré s ním súvisia. Takto bude zadaných niekoľko slov. Norma vznikne z odpovedí skupiny ľudí, kde sa vyberajú najčastejšie zvolené slová. Problém je, že medzi

odpoveďami sa vyskytujú okrem podstatných mien aj slovesá alebo prídavné mená. Preto treba odstrániť slová, ktoré sa nevyskytujú medzi termínmi vygenerovanými systémom. Potom môže pre niektoré slovo zostať menej asociácií, ako je počet systémom vygenerovaných podobných slov. Porovnávanie teda nemusí prebiehať na všetkých podobných slovách, ale len na prvých *n*, kde *n* je počet asociácií v norme. Ako asociačná norma sa použije Free Association Norms z University of South Florida.

Ďalšou používanou metódou je porovnávanie s iným slovným modelom. Existujú dve možnosti definovania podobnosti, a to podľa rozsahu podobnosti okolo slova alebo podľa počtu blízkych slov. Ako druhý slovný model na porovnanie som vybral WordNet, ktorý je dostupný aj na FIT VUT v Brne. Porovnávať sa bude na základe zhody piatich najbližších výrazov pre každý termín. Nebude pritom záležať na poradí týchto výrazov.

Na testovanie využijem korpusy, ktoré sú v súčasnej dobe dostupné na školských serveroch. Prvou databázou dokumentov je GIGAWORD. Je to databáza novinových článkov v angličtine. Obsahuje viac ako 5 miliónov dokumentov. Po spracovaní programom TreeTagger má celá databáza veľkosť približne 39 GB. Druhým korpusom je GENIA, ktorá obsahuje medicínske články. GENIA má asi 2000 dokumentov, veľkosť je 9 MB. Je uložená vo formáte XML. Všetky termíny sú označené príslušným tagom. Na základe týchto tagov budú extrahované termíny. Parsovanie XML dokumentu prebehne mimo systém.

Výstupom systému bude súbor, kde bude termínu priradených 5 najbližších termínov. Tie budú vybrané pomocou matice podobností. V prípade GIGAWORDU budú porovnávané s rovnakými údajmi získanými z WordNetu metódou lineárnej podobnosti (viď 3.5). Porovnávanie bude prebiehať len na množine termínov, ktoré sa vyskytujú v korpuse aj vo WordNete. Výsledok bude zhoda vybraných výrazov vyjadrená v percentách. Pri asociačnom teste sa bude skúmať, koľko z týchto výrazov sa vyskytuje v asociačnej norme.

V prípade GENIE sa bude zisťovať, koľko z 5 najbližších termínov patrí do rovnakej kategórie ako pôvodný termín. Podobne ako v prípade asociačného testu sa najprv odstránia výrazy, ktoré nie sú medzi tými, ktoré sú prítomné aj v systéme. Rovnako sa nebude porovnávať viac termínov, ako je ich počet v danej kategórii. Kategórie sú súčasťou XML súboru, v ktorom je uložení GENIA. Porovnávanie GENIE s WordNetom alebo pomocou asociačného testu neposkytne prakticky žiadne merateľné údaje, pretože väčšina termínov v GENII je vysoko špecializovaná a nenachádzajú sa ani v asociačnom teste, ani vo WordNete.

5 Implementácia

5.1 Štruktúra interných súborov

Systém používa rôzne druhy súborov pre vnútornú reprezentáciu údajov. Systém uchováva dokumenty, zoznam slov a termínov a súbory s maticami a niektoré ďalšie pomocné súbory. Všetky súbory okrem tých, ktoré boli vytvorené pomocou niektorých externých knižníc, sú textové súbory.

Dokumenty majú niekoľko vnútorných reprezentácií. Každá zodpovedá inej fáze spracovania. Prvou formou sú dokumenty uložené v textových súboroch. Každý súbor sa chápe ako jeden dokument. Všetky súbory sú uložené v jednom adresári, prípadne v ďalších v ňom vnorených adresároch.

Druhá forma vzniká po zisťovaní slovných druhov a základných tvarov slov. Dokumenty reprezentuje jeden súbor, v ktorom sú uložené trojice, ktoré sa skladajú zo slova, jeho slovného druhu a základného tvaru. Každá trojica je na samostatnom riadku. V tomto súbore sú jednotlivé dokumenty uzatvorené v tagoch <DOC> a <\DOC>. Tieto tagy sú tiež na samostatných riadkoch.

Tretia forma vzniká po extrakcii termínov a významných slov. Všetky dokumenty sú uložené v jednom súbore. Každý dokument reprezentuje zoznam termínov a ostatných významných slov uzatvorený do tagov <DOC> a <\DOC>. Každé slovo, viacslovný výraz alebo XML tag je na samostatnom riadku.

Zoznamy termínov a významných slov sú textové dokumenty, kde sú všetky termíny, respektíve významné slová, zo všetkých dokumentov, ktoré sa budú ďalej používať. Každé slovo alebo viacslovný výraz je na samostatnom riadku.

Matice sú textové súbory, ktoré obsahujú len čísla. Na prvom riadku súboru sú rozmery matice. Ďalšie riadky obsahujú samotnú maticu. Jednotlivé stĺpce sú oddelené tabulátormi. Riadky sú oddelené znakmi pre koniec riadka.

5.2 Spracovanie dokumentov

Prvým krokom programu je spracovanie vstupnej množiny dokumentov. Tá môže mať tri rôzne formy (viď 5.1). Prvou formou sú textové súbory. Tie sú v adresári, ktorý sa predáva ako parameter programu. Pri tejto forme sa ako prvý krok vytvorí zoznam súborov. Ako druhý krok nasleduje určovanie slovných druhov a základných tvarov.

Na zisťovanie slovných druhov a základných tvarov som použil program TreeTagger. TreeTagger je nástroj pre určovanie slovných druhov a základných tvarov slov. Bol vyvinutý Helmutom Schmidom v Ústave počítačovej lingvistiky na Univerzite Stuttgart. Program bol úspešne testovaný na angličtine, nemčine, francúzštine, taliančine a na ďalších jazykoch [7]. Je upraviteľný aj

pre ostatné jazyky, pre ktoré je dostupný lexikón a cvičný korpus. TreeTagger pre každé slovo, ktoré sa vyskytuje v dokumente, vygeneruje trojicu, ktorá sa skladá z pôvodného tvaru slova, slovného druhu a základného tvaru. V prípade, že základný tvar je neznámy, ako tretí prvok sa vygeneruje tag "<unknown>". V takom prípade sa bude za základný tvar slova považovať pôvodný tvar.

Po spracovaní TreeTaggerom vznikne druhý typ súborov (viď 5.1). Takto spracované súbory sú potom predané generátoru termínov a významných slov, ktorý odfiltruje nepodstatné slová. Tu je možné určiť, či sa budú oddeľovať jednotlivé vety ako samostatné dokumenty. Rozlišovanie viet má vplyv na výsledok. Napríklad pri metóde Random Indexing pri nerozlišovaní viet môže kontextové okno zasahovať do dvoch viet, ktoré spolu nesúvisia. Program na rozpoznávanie termínov a významných slov vytvorí tretí typ súborov (viď 5.1), v ktorom sú termíny a kľúčové slová v každom dokumente a ďalšie súbory, v ktorých je zoznam slov, ktoré budú brať do úvahy pri indexovaní (termíny a významné slová), a zoznam termínov. Tieto súbory sa potom spracovávajú buď pomocou metódy LSA, alebo pomocou metódy Random Indexing. Výsledkom obidvoch metód je matica, v ktorej sú podobnosti termínov. V tejto matici sa potom vyhľadávajú jednotlivé podobnosti. Program, ktorý vyhľadáva podobnosti, potrebuje ako parameter maticu, súbor so zoznamom termínov a jeden termín, keď sa zisťuje najbližšie slová, alebo dva, pokiaľ sa zisťuje blízkosť týchto dvoch slov.

Vstupom programu môžu byť dokumenty v ľubovoľnej z vyššie uvedených foriem. Ak sú dokumenty zadané v prvej forme, prebiehajú všetky fázy spracovania. Ak v druhej forme, neprebieha morfologická analýza, ale priamo vyhľadávanie termínov a významných slov. Ak sú dokumenty zadané v tretej forme, nie je možné určiť, či sú výrazy v dokumentoch termíny alebo významné slová, preto všetky výrazy budú vo výslednej matici. Toto správanie je možné ovplyvniť zadaním zoznamu slov, ktoré majú byť v indexe, ako parametra programu. Pokiaľ je zadaný zoznam významných slov aj termínov, neprebieha už ich extrakcia, ale priamo sa pristúpi k tvorbe indexu. Rovnako pri tomto type vstupných údajov už nie je možné rozlišovať vety.

5.3 Vyhľadávanie termínov

Vyhľadávanie sa skladá z generátorov unigramov, digramov, trigramov a ostatných významných slov, ktoré nie sú termíny. Každý generátor je samostatná trieda a je ju možné nahradiť odvodenou triedou. Toto umožňuje ľahkú modifikovateľnosť programu. Samostatnou triedou je aj trieda na modifikáciu a filtrovanie slov. Slová sú filtrované na základe dĺžky a znakov, ktoré obsahujú. Akceptujú sa len slová, ktoré obsahujú písmená, číslice a biele znaky a zároveň majú aspoň 3 znaky. Pri úprave sa všetky písmená v slove zmenia na malé a číslice sa zmenia na číslicu „0".

Program prechádza postupne morfologicky analyzované dokumenty a jednotlivým generátorom odovzdáva základný tvar slova a slovný druh. Ak je základný tvar neznámy, použije sa pôvodný tvar slova. Do súboru s dokumentmi sú zapísané všetky termíny a významné slová bez

ohľadu na frekvenciu výskytu. Po spracovaní všetkých dokumentov sa vyberú výrazy, ktoré budú v indexe. Výber prebieha na základe frekvencie výskytu. Užívateľ si parametrom určí minimálnu a maximálnu požadovanú frekvenciu. Na základe týchto údajov prebehne filtrovanie, ktorého výsledkom je zoznam termínov, pre ktoré sa bude tvoriť matica podobností, a zoznam slov, ktoré budú v indexe.

Keďže v dokumentoch môže byť veľmi veľké množstvo slov a výrazov, ktorých frekvenciu je potrebné počítať, program má vysokú pamäťovú náročnosť. Tento problém sa rieši tak, že po spracovaní každých 250 000 dokumentov sa odstránia výrazy, ktoré sa vyskytli príliš málo (rádovo v percentách z minimálnej frekvencie). Odstránenie týchto málo početných výrazov má zanedbateľný vplyv na výsledok.

5.4 LSA

Pri tejto metóde potrebujeme najprv vytvoriť maticu výskytov termínov v dokumentoch. Postupne prechádzame dokumenty a počítame výskyty termínov a významných slov v nich. Keďže celá matica je príliš veľká, aby sa zmestila do pamäte, zapíšeme vždy po určitom počte dokumentov časť matice do pomocného súboru, ktorý nazveme segment. Počet dokumentov v jednom segmente je možné nastaviť parametrom. Nakoniec tieto segmenty spojíme do výslednej matice. Jednotlivé prvky matice sú normalizované už pri zápise do segmentov.

Túto maticu je potom potrebné zmenšiť pomocou SVD. Použil som na to SVDLIBC. Je to knižnica napísaná v jazyku C. Základ tvorí knižnica SVDPACKC. Autormi sú Michael Berry, Theresa Do, Gavin O'Brien, Vijay Krishna a Sowmini Varadhan. Využíva Lanczosovu Single-Vector metódu.

Po spracovaní pomocou SVD vytvoríme kontextové vektory tak, že vynásobíme maticu termínov a diagonálnu maticu. Zo skalárnych súčinov kontextových vektorov vytvoríme maticu podobností termínov. V tejto matici budú len termíny a nebudú tu ostatné významné slová, ktoré sú v indexe. Matica sa bude tvoriť postupne po skupinách slov. Veľkosť skupiny je možné nastaviť parametrom. Čím väčšia skupina, tým rýchlejšie sa spočíta výsledná matica, čo si ale vyžaduje viac pamäte.

5.5 Random Indexing

Pri tejto metóde sa nemusí tvoriť matica výskytov termínov v dokumentoch, ale je možné robiť priamo kontextové vektory. Rozhodol som sa využiť knižnice Apache Lucene a Semanticvectors.

Lucene je opensource knižnica na získavanie informácii z textu. Vytvoril ju Doug Cutting s podporou Apache Software Foundation. Bola vydaná pod Apache Software License. Pôvodne bola napísaná v jazyku Java, neskôr bola prepísaná aj do iných programovacích jazykov.

Semanticvectors je knižnica napísaná v Jave. Aplikuje metódu Random Projection na index, ktorý bol vytvorený pomocou knižnice Apache Lucene. Knižnica bola vytvorená na Pittsburghskej univerzite.

Údaje do Apache Lucene indexu musia byť vkladané takým spôsobom, aby bolo možné pracovať s každým výrazom, teda aj s viacslovným, ako so samostatnou jednotkou. Pri vkladaní v textovej podobe sa táto vlastnosť nedá zaručiť. Preto vkladám údaje do indexu pomocou vlastného prúdu tokenov.

Z hotového indexu sa počítajú kontextové vektory pre jednotlivé výrazy. Na vytvorenie kontextových vektorov som použil triedu TermTermVectorsFromLucene z balíka Semanticvectors. Za kontext slova sa nepovažuje celý dokument, ale slová, ktoré sa vyskytujú v jeho blízkosti. To sa nazýva kontextovým oknom. Jeho veľkosť definuje, koľko slov sa berie do úvahy. Hodnota sa zadáva parametrom. Minimálna hodnota je 2. Obvykle sa zadávajú hodnoty v rozmedzí 10 až 20. Pôvodná verzia tejto triedy potrebuje niekoľko úprav, aby zohľadňovala špecifiká tohto systému. Originálna trieda obsahuje filtrovanie termínov, ktoré sa vykonáva už pri tvorbe indexu, takže tu už nie je potrebné. Navyše filtrovanie v systéme prebieha na základe odlišných pravidiel. V tejto triede chýba filtrovanie na základe dĺžky slova a odstraňujú sa výrazy, ktoré obsahujú iné znaky než písmená. Systém však akceptuje aj slová, ktoré obsahujú číslice. Navyše pracuje aj s viacslovnými výrazmi, ktoré obsahujú biele znaky. Najdôležitejšou modifikáciou je prispôsobenie na viacslovné výrazy. V okolí viacslovných výrazov sa budú prirodzene vyskytovať jednotlivé časti týchto výrazov, ktoré sa musia ignorovať. Napríklad pri slovnom spojení "preteky automobilov" sa budú vyskytovať slová "preteky" a "automobil". Keby sme ich brali do úvahy, skresľovali by výsledky. Preto sa ignorujú všetky slová, ktorá sú podreťazcom práve skúmaného slova, alebo skúmané slovo je ich podreťazcom.

Posledným krokom je výpočet matice podobností termínov. Podobnosti termínov sa počítajú skalárnym súčinom kontextových vektorov. Výsledky sú následne ukladané do matice. Tu budú už len podobnosti termínov. Ostatné slová sú vynechané. Keďže matica môže byť veľmi veľká, počíta sa postupne. Nepočíta sa však po jednom slove, ale po celých skupinách slov. Veľkosť tejto skupiny je možné nastaviť parametrom. Čím bude väčšia, tým bude výpočet rýchlejší, ale zároveň porastú aj nároky na pamäť. Podobnosť dvoch termínov sa vyskytuje v matici dvakrát (pokiaľ sú odlišné). Systém ju vypočíta len raz, druhýkrát získa už vypočítanú podobnosť z výslednej matice.

6 Výsledky

6.1 Vyhodnotenie testov na GIGAWORDE

Podobnosť s WordNetom a asociačný test som robil len pre metódu Random Indexing. Metódu LSA som nevyužil kvôli vysokej časovej náročnosti pri veľkom počte dokumentov. Testoval som rôzne parametre pre dosiahnutie lepších výsledkov. Všetky testy prebiehali na školských serveroch. Jeden test trval približne 23 až 27 hodín v závislosti na počte termínov a významných slov a na vyťažení serveru. Nasledujúca tabuľka ukazuje výsledky.

Číslo testu	Minimálna frekvencia	Maximálna frekvencia	Kontextové okno	Delenie na vety	Podobnosť s WordNetom	Asociačný test
1	1 000	10 000 000	2	áno	3.430 %	3.222 - 2.275 %
2	1 000	10 000 000	10	áno	4.538 %	5.193 - 3.768 %
3	1 000	10 000 000	20	áno	4.497 %	4.847 – 3.676 %
4	1 000	10 000 000	10	nie	4.498 %	4.927 - 3.801 %
5	1 000	100 000	10	nie	3.727 %	4.307 - 3.113 %
6	1 000	100 000	10	áno	3.628 %	3.482 - 2.736 %
7	1 000	10 000	10	nie	3.076 %	4.276 - 3.227 %
8	1 000	1 000 000	10	nie	4.444 %	4.728 - 3.635 %

Tabuľka 6.1: Výsledok testov pre GIGAWORD

Na porovnanie bola vybraná množina približne 11 000 výrazov v teste číslo 7 až 18 000 v testoch 1 až 4. Vyberali sa výrazy, ktoré sa vyskytujú v korpuse aj vo WordNete. Zhoda výstupu systému s náhodne vygenerovanými slovami je od 0.02 do 0.05 %. Podobnosť s WordNetom vychádza veľmi nízka, ale aj napriek tomu je výrazne vyššia ako v prípade náhodne vygenerovaných slov. Malá zhoda výstupu systému s WordNetom je spôsobená niekoľkými faktormi. Za prvé sú to nepresnosti, ktoré vznikli v systéme. Medzi ne patria chybne priradené slovné druhy v programe TreeTagger a nemožnosť rozlíšiť homonymá. Na výsledky vplýva aj to, že GIGAWORD sú novinové články. Medzi jednotlivými slovami sú teda aj súvislosti, ktoré WordNet nezaznamenáva. Vo WordNete sa blízkosť slov určuje na základe iných vlastností slov, čo následne skresľuje výsledky. Ako príklad uvediem podobné výrazy pre slová „Aachen“, čo je mesto v Nemecku, a „smallpox“[1] pri teste číslo 2.

[1] kiahne

Random Indexing	WordNet
Cologne	city
Hamburg	metropolis
Dresden	urban center
Leipzig	municipality[2]
Dusseldorf	populated area

Tabuľka 6.2: Podobné slová pre "Aachen"

Random Indexing	WordNet
rabies[3]	pox[4]
influenza[5]	contagion[6]
polio[7]	contagious disease
hepatitis	dose[8]
anthrax	std

Tabuľka 6.3: Podobné slová pre "smallpox"

Z tabuľky je vidieť, že WordNet vybral k názvu mesta aj k chorobe všeobecné termíny, ktoré popisujú tento výraz. Systém vybral pre názov mesta iné mestá, ktoré patria tomu istému štátu. Pre názov choroby vybral názvy iných chorôb. Pri niektorých slovách je vidieť súvislosti na základe udalostí, ktoré sa udiali vo svete. Napríklad v teste číslo 1 pri slove „Bali“ vybral systém ako najpodobnejšie výrazy „Oklahoma City“, „USS Cole“, „suicide“, „nightclub“ a „apartment house“. Tieto výrazy (s výnimkou posledného) sa viažu s bombovými útokmi. Je medzi nimi vnútorná spojitosť a nie spojitosť na základe významu slov.

Na druhú stranu pre niektoré pojmy vygeneroval systém úplne odlišné slová, pri ktorých nie je žiadna zjavná súvislosť. Napríklad pre výraz „alert“ pri teste číslo 1 boli nájdené výrazy „old school“, „probability“, „school“, „catholic school“ a „learning“. Väčšinou dosiahli horšie výsledky všeobecné pojmy, ktoré sa vyskytujú vo veľkom počte rozdielnych kontextoch.

Zlé výsledky asociačného testu sú spôsobené podobnými problémami ako pri podobnosti s WordNetom. Asociačný test je silno závislí na prostredí, v ktorom sa pohybujú ľudia, ktorý ho vytvorili. Napríklad na slovo „terorizmus“ budú iné asociácie medzi Američanmi alebo Európanmi

[2] magistrát
[3] besnota
[4] kiahne, osýpky
[5] chrípka
[6] nákaza
[7] detská obrna
[8] dávka

a iné v arabských krajinách. V asociačnej norme sa vyhľadávalo prvých *n* slov pre každý výraz, kde *n* nadobúda hodnoty 1 až 5. V tabuľke je rozsah, v ktorom sa výsledky pre daný test pohybovali. Najlepšie výsledky vyšli, keď sa porovnávalo len prvé slovo. Naopak najhoršie boli pre 5 slov. Toto platí pre všetky testované kombinácie parametrov.

Pre zistenie vplyvu parametrov som porovnal výsledky pre prvé 4 testy, kde sa testovala rovnaká množina termínov. Porovnávalo sa prvých päť najbližších výrazov pre každý termín. Výsledok ukazuje nasledujúca tabuľka. Čísla testov sú podľa Tabuľka 6.1.

Číslo testu	**1**	**2**	**3**	**4**
1	-	23.690 %	18.266 %	20.243 %
2	23.690 %	-	44.118 %	45.856 %
3	18.266 %	44.118 %	-	45.600 %
4	20.243 %	45.856 %	45.600 %	-

Tabuľka 6.4: Podobnosť výsledkov pre rôzne parametre

Aj keď sa výsledky pre jednotlivé parametre značne líšia, ich porovnanie s WordNetom a asociačný test vracajú veľmi podobné výsledky. Najviac odlišný bol test číslo 1, kde sa ako kontext výrazu brali len bezprostredne susediace pojmy. Pre lepšiu predstavu si porovnáme výsledky pre výrazy „Aachen“ a „alert“.

test číslo 1	**test číslo 2**	**test číslo 3**	**test číslo 4**
Cologne	Cologne	Hamburg	Mannheim
Leipzig	Hamburg	Leipzig	Leipzig
Bonn	Dresden	Mannheim	Hamburg
Hamburg	Leipzig	Bonn	Dresden
Steffi Graf	Dusseldorf	Dusseldorf	Weimar

Tabuľka 6.5: Výsledky pre slovo „Aachen“ pre rôzne parametre

test číslo 1	**test číslo 2**	**test číslo 3**	**test číslo 4**
old school	security measure	security measure	fear
probability	vigilance	warning	vigilance
school	interrogation	threat	order
catholic school	suspect	fear	area
learning	hunt	force	attack

Tabuľka 6.6: Výsledky pre slovo „alert“ pre rôzne parametre

Z príkladov je vidieť, že výsledky pre všeobecnejšie slovo „alert“ sa líšia viac ako pre špecifické slovo „Aachen“. V prípade slova „Aachen“ boli skoro všetky slová názvy nemeckých miest. Aj keď sa výsledky úplne nezhodujú, výrazy patria do rovnakej kategórie. Pri slove „alert“ je vidieť, že najviac sa opäť líši test číslo 1.

6.2 Vyhodnotenie testov na GENII

Testy na GENII som vykonával metódami Random Indexing a LSA. Výsledky je možné vidieť v nasledujúcej tabuľke.

Číslo testu	Metóda	Minimálna frekvencia	Maximálna frekvencia	Kontextové okno	Výsledok
1	Random Indexing	2	1 000	6	24.288 – 21.179 %
2	Random Indexing	5	1 000	20	27.682 – 24.534 %
3	Random Indexing	10	1 000	2	31.808 – 28.599 %
4	Random Indexing	10	1 000	20	31.986 – 29.581 %
5	Random Indexing	20	1 000	2	36.292 – 34.029 %
6	Random Indexing	20	1 000	6	34.007 – 33.601%
7	Random Indexing	20	1 000	10	33.392 – 32.602 %
8	Random Indexing	20	1 000	20	34.974 – 32.191 %
9	Random Indexing	50	1 000	20	32.151 - 37.022 %
10	LSA	2	1 000	-	28.844 - 27.595 %
11	LSA	5	1 000	-	34.591 - 31.049 %
12	LSA	10	1 000	-	37.850 - 35.499 %
13	LSA	20	1 000	-	42.223 - 38.740 %
14	LSA	50	1 000	-	48.115 - 42.739 %

Tabuľka 6.7 Testy pre GENIU

Výsledky sa počítali podobne ako pri asociačnom teste. Teda skúmal sa výsledok postupne pre prvý až pre prvých päť najbližších výrazov. V tabuľke sú uvedené rozmedzia, v ktorých sa pohybovali výsledky. Prvá hodnota je len pre prvý a druhá pre prvých päť termínov. Najlepšie výsledky boli pri porovnávaní iba prvého najbližšieho výrazu. Potom postupne klesali až k hodnote pre prvých päť výrazov, kde boli najnižšie. Jedinou výnimkou bol test číslo 9, kde test pre prvý termín dopadol najhoršie a postupne stúpal až po prvých päť termínov. V tomto prípade bolo vybraných len 171 termínov, čo môže skresľovať výsledok. Je vidieť, že lepšie výsledky sú pri nižších minimálnych frekvenciách.

V tomto prípade výsledky nie sú ovplyvnené chybami pri určovaní slovných druhov, pretože tie neboli systémom určované. Nakoľko sa jedná o odborné termíny, výskyt homoným medzi nimi je veľmi nepravdepodobný. Nepresnosti môžu byť spôsobené malým počtom dokumentov. Horšie výsledky dosiahli pri nižšej minimálnej frekvencii, kde je väčšia množina termínov.

Metóda LSA poskytuje lepšie výsledky. Rýchlostne sa LSA vyrovnala Random Indexing, čo je spôsobené malým počtom dokumentov. Najdlhší test trval len asi 20 minút. Podobnosť výsledkov LSA a Random Indexing nepresiahla 10 %.

Aj malá zmena rozsahu frekvencií mala výrazný vplyv na počet vybraných termínov, ako je možné vidieť v nasledujúcej tabuľke.

Minimálna frekvencia	**Počet termínov**
2	10 003
5	3 002
10	1 224
20	509
50	171

Tabuľka 6.8 Počet vybraných termínov

Menila sa len minimálna frekvencia, maximálna bola pri všetkých testoch nastavená na hodnotu 1000. Túto hodnotu nedosahuje žiadny termín, takže nemala žiadny vplyv na výsledky. Pri malom počte dokumentov som nepovažoval za nutné odstraňovať príliš často sa vyskytujúce výrazy.

7 Záver

Cieľom práce bolo vytvoriť systém na zisťovanie sémantickej podobnosti termínov. Pre rozpoznávanie termínov som zvolil metódu výberu na základe slovných druhov a frekvencie výskytu. Slovné druhy sa zisťovali pomocou programu TreeTagger. Hlavným dôvodom bola možnosť využitia pre rôzne jazyky. Pre výpočet blízkosti termínov som vybral Random Indexing z dôvodu rýchlosti. Túto metódu som implementoval pomocou knižníc Apache Lucene a Semanticvectors. Na porovnanie som implementoval aj metódu latentnej sémantickej analýzy. Jej najzložitejšou časťou je singulárny rozklad matice výskytov slov v dokumentoch. Na jeho výpočet som použil knižnicu SVDLIBC. Systém pozostáva z niekoľkých programov. Jednotlivé programy sú implementované v jazyku Java. Spojené sú pomocou skriptu pre shell. Systém je určený predovšetkým pre operačný systém Linux.

Vyhodnotenie úspešnosti systému sa ukázalo ako veľmi náročná úloha, pretože na výber podobných slov má veľký vplyv oblasť, z ktorej sú skúmané dokumenty. Najbežnejšie používané testovanie na synonymá sa nedalo použiť, pretože synonymické testy sú postavené na prídavných menách a nie na termínoch. Porovnávaním s existujúcou ontológiou nie je možné určiť kvalitu výsledkov. Tie hovoria iba to, ako veľmi sa výstup systému podobá na ontológiu. Testovaním sa zistilo, že výstup systému sa nepodobá údajom z WordNetu ani z asociačného testu. Zlepšenie výsledkov je vidieť pri testovaní medicínskych článkov v korpuse GENIA, keď sa zisťovalo, či vybrané podobné slová spadajú do rovnakej kategórie. Pri tomto teste bol limitujúcim faktorom malý počet dokumentov. Metóda LSA dosiahla lepšie výsledky ako Random Indexing. Časová náročnosť bola pri obidvoch metódach približne rovnaká. To sa však mení pri rozsiahlejších kolekciách dokumentov, kde náročnosť metódy LSA rastie oveľa rýchlejšie, najmä kvôli tvorbe matice výskytov a výpočtu singulárneho rozkladu. Testy zároveň ukázali, že parametre systému nemajú veľký efekt na výsledky testov. Veľkosť kontextového okna má vplyv najmä pri všeobecných slovách. Pri viac špecifických termínoch sa výsledky príliš nelíšili. Minimálna a maximálna frekvencia ovplyvňuje najmä počet získaných termínov. V tomto sa prikláňam k väčšiemu počtu. Pre jednotlivé termíny je viac možností, z ktorých sa vyberajú podobné výrazy, čo zvyšuje šancu, že sa medzi nimi nachádzajú vhodné pojmy.

Systém dokáže samostatne určiť termíny a zistiť ich sémantickú blízkosť, čím spĺňa požiadavky naň kladené. Najväčší priestor na zlepšovanie vidím pri rozpoznávaní termínov. Vhodné by bolo doplnenie výpočtu unithoodu. Týmto by sa znížil počet nadbytočných termínov. Pri metóde Random Indexing by boli vhodné úpravy, ktoré by umožňovali jednoduché pridávanie a odoberanie dokumentov tak, aby sa následne nemuseli nanovo vytvárať kontextové vektory.

Literatúra

[1] Wikipedia. Termín (názov) [online] [cit. 2009-05-10].
Dostupné na URL: <http://sk.wikipedia.org/wiki/Term%C3%ADn_(n%C3%A1zov)>

[2] Kageura, Kyo: Methods for Automatic Term Recognition. [online] [cit. 2009-05-11].
Dostupné na URL: <http://research.nii.ac.jp/~kyo/papers/term-recognition.ps>

[3] Sahlgren, Magnus: An Introduction to Random Indexing. [online] [cit.2009-05-11].
Dostupné na URL: <http://www.sics.se/~mange/papers/RI_intro.pdf>

[4] Krátky, Michal: Využití SVD pro indexování latentní sémantiky. [online] [cit. 2009-05-11].
Dostupné na URL: <http://www.cs.vsb.cz/arg/techreports/lsi-svd_ma.pdf>

[5] Michael P. Holmes, Alexander G. Gray, Charles Lee Isbell Jr.: Fast SVD for Large-Scale Matrices. [online] [cit. 2009-05-11].
Dostupné na URL: <http://bigml.wikispaces.com/file/view/Holmes.pdf>

[6] Brand, Matthew: Fast Low-Rank Modifications of the Think Singular Value Decomposition. [online] [cit. 2009-05-05].
Dostupné na URL: <http://www.merl.com/publications/TR2006-059/>

[7] TreeTagger. [online] [cit. 2009-05-04]. Dostupné na URL:
<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

[8] Strachota, Tomáš: Automatická tvorba rejstříku publikace. bakalářska práce, Brno, FIT VUT v Brne, 2008.

[9] Apache Lucene. [online] [cit. 2009-05-04].
Dostupné na URL: <http://lucene.apache.org/java/docs/index.html>

[10] Semanticvectors. [online] [cit. 2009-05-04].
Dostupné na URL: <http://code.google.com/p/semanticvectors/>

[11] SVDLIBC. [online] [cit. 2009-05-04].
Dostupné na URL: <http://tedlab.mit.edu/~dr/svdlibc/>

[12] Wikipedia. Semantic Similarity [online] [cit. 2009-05-08].
Dostupné na URL: <http://en.wikipedia.org/wiki/Semantic_similarity>

[13] Wikipedia. Semantic Relatedness [online] [cit. 2009-05-07].
Dostupné na URL: <http://en.wikipedia.org/wiki/Semantic_relatedness>

[14] WordNet. [online] [cit. 2009-05-07].
Dostupné na URL: <http://wordnet.princeton.edu/>

[15] Sahlgren, Magnus: The Word-Space Model. [online] [cit. 2009-05-10].
Dostupné na URL: <http://www.sics.se/~mange/TheWordSpaceModel.pdf>.

[16] Schwartz, J.: Současný stav a trendy automatické indexace dokumentů. [online] [cit. 2009-05-11]. Dostupné na URL: <http://www.ikaros.cz/node/1300>

[17] Wikipedia. Zipf's Law [online] [cit. 2009-05-11].
Dostupné na URL: <http://en.wikipedia.org/wiki/Zipf%27s_law>

[18] Lin, Dekang: An Information-Theoretic Definition of Similarity. [online] [cit. 2009-05-13].
Dostupné na URL: <http://www.cs.ualberta.ca/~lindek/papers/sim.pdf>

Zoznam príloh

Príloha 1. Príklady vygenerovaných podobných výrazov

Uvedené sú len niektoré vhodné príklady. Všetky sú z korpusu GIGAWORD. Úplné výsledky testov pre GIGAWORD sú na priloženom CD.

Skúmaný výraz	**Najbližšie výrazy**				
	1	**2**	**3**	**4**	**5**
ACER	DELL	Apple	peripheral	disk	drive
aluminium	nickel	tin	zinc	aluminum	cash
diesel engine	jet engine	electric car	hybrid	plant	factory
diplomatic corps	royal family	executive council	cabinet	present	advisory board
European Central Bank	Bank of England	FED	Federal Reserve	Bundesbank	Bank of Japan
fascism	Nazism	veteran	battlefield	tyranny	totalitarianism
FBI	secret service	ATF	border patrol	DEA	Drug Enforcement Administration
financial aid	financial support	help	support	funding	effort
flash flood	flooding	mudslide	hailstorm	rainstorm	tornado
greenhouse gas	carbon dioxide	global warming	emission	fossil fuel	smokestack
heart attack	heart failure	cardiac arrest	complication	liver cancer	congestive heart failure
long jump	high jump	triple jump	pole vault	javelin	shot put
milky way	quasar	light year	black hole	supernova	astronomer
petrol bomb	molotov cocktail	gasoline bomb	firebomb	hand grenade	tear gas
photon	wavelength	bulb	atom	particle	neutrino
Toyota	Nissan	automaker	Ford	carmaker	auto maker
violin	piano	cello	clarinet	guitar	flute
Wednesday	Monday	Friday	Thursday	Tuesday	Saturday

Príloha 2. Užívateľský manuál systému

Systém je tvorený niekoľkými programami. Ovláda sa pomocou troch skriptov: install.sh, indexer.sh, termSimilarity.sh. Skript install.sh slúži na inštaláciu, indexer.sh slúži na tvorbu indexu a výpočet blízkosti termínov a termSimilarity.sh slúži na vyhľadávanie podobných slov a zisťovanie podobnosti slov.

Inštalácia

Projekt sa inštaluje pomocou skriptu install.sh. Ten skompiluje všetky zdrojové texty. Po spustení tohto skriptu je možné používať aj ďalšie dva. Skript nemá žiadne parametre. Pre spustenie je potrebné mať nainštalovaný program make, kompiler g++, javac a program ant.

Tvorba indexu

Index sa tvorí pomocou skriptu indexer.sh. Pre jeho spustenie je potrebné mať nainštalovaný Java VM, a interpret jazyka Python. Skript indexer.sh má 2 povinné parametre. Jeden je parameter -d <adresár>, ktorým sa zadáva pracovný adresár, kde sa budú ukladať pomocné súbory, matica podobností termínov a zoznam termínov. Druhý povinný parameter sa zadáva vstupná množina dokumentov. Môže byť zadaná v troch rôznych formátoch s parametrami -c, -tag alebo -xml. Prvým parametrom -c <adresár> sa zadáva množina dokumentov, ktorá je uložená v zadanom adresári vo forme textových súborov. V každom súbore je jeden dokument. Druhá forma je zadaná parametrom -tag <súbor>. V súbore sú uložené dokumenty po spracovaní programom TreeTagger. Jednotlivé dokumenty sú uzavreté v tagoch <DOC> a </DOC>. Tretia forma je zadaná parametrom -xml <subor>. Zadáva sa takto množina dokumentov, ktorá je v jednom súbore, v ktorom je zoznam termínov a významných slov v dokumentoch. Zoznam výrazov z jedného dokumentu je uzavretý medzi značkami <DOC> a </DOC>. Ostatné parametre sú nepovinné.

Zoznam parametrov skriptu indexer.sh:

-d <adresár>	pracovný adresár (povinný parameter)
-c <adresár>	adresár, kde sa nachádzajú dokumenty. Jeden dokument je v jednom textovom súbore.
-xml <súbor>	súbor, kde sa nachádzajú dokumenty. Jeden dokument je tvorený zoznamom slov v tomto dokumente. Zoznam slov v jednom dokumente je medzi značkami <DOC> a </DOC>.

-tag <súbor>	súbor kde sa nachádzajú dokumenty spracované programom TreeTagger. Všetky dokumenty sú v jednom súbore, v ktorom sú uzatvorené v tagoch <DOC> a </DOC>
-min <min_freq>	minimálny počet výskytu výrazu v dokumentoch, aby bolo zahrnuté v indexe (defaultne 10).
-max <max_freq>	maximálny počet výskytu výrazu v dokumentoch, aby bolo zahrnuté v indexe (defaultne 1000).
-xms <veľkosť>	minimálna veľkosť haldy v MB pre Java VM (defaultne 128).
-xmx <veľkosť>	maximálna veľkosť haldy v MB pre Java VM (defaultne 256).
-w <veľkosť>	veľkosť okna pri tvorbe matice podobností termínov a počet dokumentov v segmente pri LSA (defaultne 250).
-cw <veľkosť>	veľkosť posuvného kontextového okna pri počítaní term-term vektorovpri Random Indexing. Počet slov okolo výrazu, ktoré sa budú brať do úvahy pri tvorbe kontextových vektorov. Má zmysel iba pri metóde Random Indexing. (defaultne 10).
-dim <počet>	požadovaný počet SVD trojíc alebo dimenzií. Zmysel má len pri metóde LSA. Defaultne sú všetky dimenzie v matice.
-lsa	použije sa metóda LSA namiesto Random indexing
-sen	pri zadaní tohto parametra sa budú rozlišovať vety ako samostatné dokumenty. Tento parameter má zmysel len pri zadaní vstupnej množiny dokumentov s parametrom –c alebo -tag.
-terms <súbor>	zoznam termínov, pre ktoré sa tvorí matica podobností
-words <súbor>	zoznam výrazov, ktoré budú v indexe.
-h	výpis nápovedy
--help	výpis nápovedy

Zisťovanie podobnosti termínov

Podobnosť termínov sa zisťuje pomocou skriptu termSimilarity.sh. Skript má dva povinné parametre. Prvý je -d <adresár>, ktorý udáva adresár, v ktorom je výsledok skriptu indexer.sh.

Zoznam parametrov skriptu termSimilarity.sh:

-d <adresár>	adresár vytvorený skriptom indexer.sh. V tomto adresári sa musí nachádzať aspoň súbor termList.txt, v ktorom sa nachádza zoznam termínov a súbor simialrity.txt, ktorý obsahuje maticu podobností termínov. Je to povinný parameter. Musí byť uvedený ako prvý.

Druhý parameter je term, ktorý sa vyhľadáva. Druhý parameter je povinný.
Tretí parameter je term. Pokiaľ je zadaný, skript vráti podobnosť týchto dvoch zadaných termínov. Pokiaľ nie je zadaný druhý termín, skript vypíše zoznam podobností všetkých ostatných termínov s prvým termínov, usporiadaný od najväčšej podobnosti po najmenšiu.

Pokiaľ jeden z termínov nie je v zozname termínov v adresári danom prvým parametrom, skript vráti hodnotu -1.0.

Príloha 3. CD

Na priloženom CD sa nachádzajú nasledujúce adresáre

- /System - kompletný systém pripravený na inštaláciu. V adresári System/src sú zdrojové kódy a knižnice Apache Lucene a Semaneticvectors. V adresári System/TreeTagger je program TreeTagger vo verzii pre Linux. Okrem týchto adresárov obsahuje ešte skripty install.sh, indexer.sh a termSimilarity.sh, súbor README.txt a súbor build.xml na kompiláciu java súborov pomocou programu ant. Archív svdlibc.tgz obsahuje knižnicu SVDLIBC.
- /Dokumentacia- programová dokumentácia vygenerovaná pomocou programu JavaDoc.
- /Vysledky - výsledky testov na GIGAWORDe.
- súbor technicka_sprava.pdf – úplné znenie technickej správy vo formáte PDF.