TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Autoreferát disertační práce

Mgr. Jiří Vraný

Liberec 2009

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: P3901 Aplikované vědy v inženýrství
Studijní obor: 3901V025 Přírodovědné inženýrství

Lokalizace algoritmů pro řazení výsledků vyhledávání informací na webu

Localized algorithms for web pages ranking

Mgr. Jiří Vraný

Školitel: doc. RNDr. Pavel Satrapa PhD.
Pracoviště: Ústav nových technologií a aplikované informatiky

Rozsah práce:

Počet stran: 131
Počet obrázků: 19
Počet tabulek: 17

20.3.2009

Anotace

Tato disertační práce se zabývá problematikou algoritmů a technik, používaných pro uspořádání výsledků vyhledávání informací na webu, pomocí fulltextového vyhledávače. Konkrétně algoritmy pracujícími s kontextem vyhledávaného dotazu s podporou personalizace výsledků vyhledávání.

Práce se zabývá řešením otázky, zda je tyto algoritmy – vyvinuté a otestované pro anglický jazyk – možné použít také pro jiný jazykový model, konkrétně pro češtinu. Proces lokalizace se skládá z aplikace českého jazykového modelu, urychlení výpočtu pomocí numerických metod lineární algebry a paralelního výpočtu hodnotících vektorů.

V práci jsou uvedeny výsledky experimentálního ověření lokalizace dvou algoritmů. Výsledky jednotlivých experimentů prokázaly, že díky navrženým modifikacím lze originální algoritmy Intelligent Surfer a Topic Sensitive PageRank úspěšně použít i pro český jazyk.

Klíčová slova: fulltextové vyhledávání, PageRank, personalizace, lokalizace, řazení výsledků

Annotation

This dissertation thesis is concerned with the algorithms for web pages ranking, especially with the personalized algorithms. Those algorithms allow to modify basic ranking, based on hyperlink structure of the web, by a user preferences and search query context. Still, there is a lot of ordering done offline, in advance before the search process. That allows usage of the algorithms on large datasets.

This thesis is focused on the speedup of personalized algorithms by using the methods of numeric linear algebra, by applications of the language model and by the parallel computation. The main focus of the thesis is to verify a possibility of localization of the algorithms originally designed and tested on the English language. The Czech language model is used for the localization testing.

The results of the localization process of two personalized algorithms are included. A theoretical proposal of localization was experimentally tested on several datasets. The results of those experiments are included. The experiments proved that localization of the Intelligent Surfer and the Topic Sensitive PageRank for the Czech language is possible.

Keywords: fultext search, PageRank, personalization, localization, results ranking

Obsah

1 Úvod

WWW stránky uložené na serverech po celém světě v rámci sítě Internet představují nejrozsáhlejší soubor dokumentů v lidských dějinách. Přitom je celá tato rozsáhlá kolekce dosažitelná z jakéhokoliv z připojených počítačů a nemá žádnou centrální autoritu, která by měla na starost správu či katalogizaci dokumentů tak, jak ji známe například z knihoven.

Do jisté míry suplují tuto roli vyhledávací stroje pro WWW, které jsou v současnosti velmi rozvíjenou oblastí vědy o získávání informací. Jsou tak zároveň

i jednou z jejích nejznámějších aplikací u široké veřejnosti. Tato mezioborová disciplína v sobě spojuje poznatky z teorie informací, matematiky, teorie programování, ale i z humanitních oborů jako je lingvistika, psychologie či sociologie.

V současné době se počet webových stránek odhaduje na více než 25 miliard. Vyhledávání běžných dotazů v tak rozsáhlé kolekci pak pochopitelně vede k velkému množství výsledků. Ty je nutné uspořádat tak, aby na počátku byly stránky nejvíce kvalitní a relevantní k zadanému dotazu.

Tato disertační práce se zaměřuje na takové metody uspořádání výsledků, které umožňují provést ohodnocení jednotlivých stránek předem, ale zároveň pracují s kontextem uživatelského dotazu.

Základní algoritmy, využívající metod lineární algebry a teorie grafů, byly postupně doplněny o prostředky umožňující sémantickou či lexikální analýzu dat i dotazů. To s sebou na jednu stranu přináší nesporné výhody, ovšem je to zaplaceno jednak větší výpočetní náročností algoritmů a také tím, že řadící algoritmus přestává být nezávislý na textu dotazu. A právě díky tomuto omezení se nabízí otázka, zda jsou tyto teorie publikované a navrhované pro anglický jazyk, platné i v českém jazykovém prostředí.

2 Cíle disertační práce

Základní otázkou, jejíž řešení dosud nebylo publikováno, je zda jsou některé z algoritmů spojujících dohromady odkazovou analýzu a obsah dokumentů použitelné univerzálně. Tedy i pro další jazyky, nikoliv pouze pro angličtinu pro níž byly navrženy a testovány.

Na základě výsledků rešerše byly pro lokalizaci zvoleny dva algoritmy, které jsou nejvíce navázány na jazykový model, mají velmi dobré výsledky pro anglický jazyk a zároveň nabízí i další otevřené problémy k dořešení. Jsou to algoritmy Topic Sensitive PageRank (kapitola 3.2.) a Intelligent Surfer (3.3). Primárním cílem práce je ověřit lokalizovatelnost těchto dvou algoritmů.

Pro vlastní ověřování lokalizace je nezbytné navrhnout a implementovat experimentální systém, ve kterém bude možné lokalizované algoritmy testovat a experimentálně ověřit jejich funkčnost. Tento systém by měl být plnohodnotným fulltextovým vyhledávačem.

Experimentální systém by měl být schopen přijímat dotazy od uživatele, najít v testovací datové kolekci odpovídající dokumenty a ty setřídit pomocí zvolené hodnotící funkce. Mělo by být také možné jednoduchým způsobem tyto hodnotící funkce měnit, případně kombinovat. Vytvoření experimentálního systému je prvním ze sekundárních cílů této práce.

Princip personalizace je v obou algoritmech stejný v tom, že jde o opakované počítání PageRank vektoru s různým nastavením personalizačního vektoru. Přestože pro lokalizaci jako takovou není podstatné jakou metodou a jak rychle byl PageRank vektor spočítán, je celková rychlost výpočtu lokalizovaného algoritmu klíčovým faktorem určujícím, zda jde o řešení reálně použitelné, nebo řešení, které zůstane pouze teoretickou studií. Proto je druhým z vedlejších cílů práce, najít před samotnou lokalizací dostatečně efektivní metodu pro výpočet PageRank vektoru.

3 Současný stav problematiky

Pro uspořádání výsledků vyhledávání se v současné době využívají kombinace výsledků příslušného modelu (za předpokladu, že model výsledky uspořádá), dalších hodnotících funkcí, které vycházejí například ze syntaktické či sémantické analýzy dokumentu a algoritmů zabývajících se analýzou hypertextových odkazů.

Výsledné hodnocení stránky je pak složeno z desítek různých položek. Cílem je co nejpřesnější a zároveň nejstabilnější (neovlivnitelná spamem) hodnotící funkce. Algoritmy popsané v disertační práci provádějí většinu ohodnocení předem - tzv. offline. Při ohodnocení využívají zejména hypertextovou strukturu webu, kombinovanou s kontextem stránky. Proto se označují jako algoritmy odkazové analýzy.

Nejznámějším algoritmem odkazové analýzy je algoritmus PageRank. Lokalizace jeho personalizovaných modifikací je pak cílem této práce. Ve stručnosti si tedy popíšeme algoritmus PageRank i obě personalizované modifikace.

3.1 Pagerank

Tento algoritmus z roku 1998 je dílem autorů Brina a Page (odtud Pagerank) [11], kteří ho úspěšně použili ve struktuře vyhledávače Google [7]. Již v základním návrhu počítá s možností personalizace, tedy přizpůsobení výsledků podle konkrétních preferencí uživatele nebo i vyhledávače. Díky tomu bylo v této oblasti již publikováno několik prací, které se personalizací zabývají.

3.1.1 Základní definice

Algoritmus vychází z algoritmů pro sčítání citací v publikacích. Základní myšlenkou algoritmu je„že stránka je důležitá, pokud na ní odkazují jiné důležité stránky“. Pagerank (PR) konkrétní stránky p - *Rank(p)*, by tedy mohl být součtem PR všech stránek, které na ni odkazují. Důležité stránky obvykle odkazují na více míst, proto pro zajištění proporcionality dostává každá stránka pouze zlomek ranku odkazující stránky. Jestliže N_u je počet odkazů vedoucích ze stránky u, pak odkaz *(u,v)* přináší stránce v pouze $Rank(u)/N_u$.

Vektor s hodnocením jednotlivých stránek je vypočeten jako vlastní vektor matice sousednosti $\mathbf{M}$ web grafu W. Tato $n \times n$ matice je extrémně řídká, protože pro celý web je $n \approx 2,5.10^{10}$, zatímco průměrný počet nenulových prvků na jednom řádku nepřekračuje řád desítek.

Sérií úprav, jejichž cílem je zajistit, aby matice byla stochastická a neredukovatelná, dojdeme od matice $\mathbf{M}$ až k výsledné Google matici $\mathbf{G}$.

$$\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{E} \tag{1}$$

Platí, že $\mathbf{E} = \mathbf{e}\mathbf{v}^T$, přičemž $\mathbf{v}$ je *personalizační vektor*, který umožňuje ovlivnit výsledný Pagerank libovolným směrem, ať už jsou důvody komerční či jiné.

Na stochastickou matici $\mathbf{G}$ lze nyní aplikovat Perron-Frobeniovu větu. Výsledný Pagerank vektor je levým Perron-Frobeniovým vektorem $\mathbf{x}^T$ matice $\mathbf{G}$.

Výpočet vlastního vektoru se provádí pouze jednou pro každou množinu stránek a vypočtený rank je pak používán pro uspořádání stránek při sestavování odpovědi.

Zásadním přínosem Pageranku je jeho nezávislost na textu dotazu. Právě ta umožňuje vypočítat hodnocení stránek offline, což byla v době vzniku převratná technologie.

Tradiční postup výpočtu PageRank vektoru je pomocí Mocninné metody. Bylo publikováno několik prací na téma jak tento výpočet urychlit. Nejzásadnějším alternativním způsobem výpočtu je řešení řídkého lineárního systému [10].

3.2 Topic sensitive Pagerank

Topic Sensitive PageRank (TSPR) vzniknul stejně jako některé další modifikace algoritmu PageRank v rámci projektu WebBase na Stanfordově univerzitě. Tento algoritmus propojuje PageRank s kontextem uživatelského dotazu a stránky za pomoci k speciálně zaměřených Rank vektorů. Jeho autorem je T. Haveliwalla [9].

Zaměřování jednotlivých vektorů dle určitého tématu se provádí pomocí množiny stránek, u kterých známe jejich téma. Tyto množiny použijeme pro sestavení tématického klasifikátoru T_j, kde $j \in < 1, k >$. Tématický klasifikátor T_j je množina stránek příslušejících tématu j a sestavením všech k množin rozdělíme web nebo jeho část do k tématických bloků. Personalizační vektor $\mathbf{v}_j$ určíme dle následujícího vzorce:

$$v_{ji} = \begin{cases} \frac{1}{T_j} \text{ pokud } i \in T_j \\ 0 \text{ v opačném případě} \end{cases}$$

Dělení skupiny dokumentů na jednotlivé bloky pomocí analýzy jejich obsahu je netriviální a výpočetně náročná operace. Proto byla pro algoritmus TSPR využita již připravená kolekce několika dokumentů s přesně určenými tématy - odkazový katalog Open Directory Project.

Stránky indexované v tomto katalogu jsou díky personalizačnímu vektoru na začátku výpočtu PageRank vektoru zvýhodněny a předpokládáme, že odkazují na další dokumenty související s příslušným tématem. Proporcionální předávání PageRanku pak zajistí, že i tyto propojené dokumenty budou ohodnoceny lépe než dokumenty s tématem nesouvisející a tudíž neodkazované. Tím je zajištěno zaměření PageRank vektoru dle zvoleného tématu. Kolekce základních dokumentů musí být dostatečně velká, aby obsáhla maximální možnou množinu propojených stránek. V praxi lze předpokládat, že určité množství odkazů povede i na stránky s jiným nepříbuzným tématem, ale dle publikovaných výsledků TSPR se zdá, že chyba způsobená těmito netématickými odkazy není pro kvalitu hodnocení kritická.

Offline výpočet takto personalizovaných PageRank vektorů je prvním krokem algoritmu TSPR.

Druhý krok algoritmu tedy musí logicky probíhat online, při zpracování uživatelského dotazu. TSPR využívá pro zpracování dotazu Q v jeho případném kontextu Q' pravděpodobnostního modelu. Kontext dotazu Q' vzniká v případě, že dotaz je položen tak, že uživatel označí část textu na stránce u. Pak kontext dotazu tvoří slova, která stránka u obsahuje. Tento přístup zadávání dotazů se ovšem v praxi prakticky nevyužívá, takže reálnějším případem bude zadání dotazu Q bez kontextu. V takovém případě platí, že $Q' = Q$.

Každé z k témat je reprezentováno vlastní třídou c_j, kde $j \in <1, k>$. Pomocí multinomického Bayesovského klasifikátoru vypočteme pro každé ze slov $q_i \in Q'$ pravděpodobnost jeho příslušnosti k třídě c_j podle vztahu:

$$P(c_j|Q') = \frac{P(c_j).P(Q'|c_j)}{P(Q')} \propto P(c_j) \prod_i P(q_i|c_j) \tag{2}$$

Pravděpodobnost $P(q_i'|c_j)$ vypočítáme za pomoci klasifikačního vektoru $\mathbf{D}_j$. Tyto vektory sestavíme během prvního kroku algoritmu offline a to tak, že položky příslušného vektoru tvoří frekvence výskytu všech slov, která obsahují stránky tvořící téma j. Velikost vektorů $\mathbf{D}_j$ je tedy pro jednotlivá témata různá a závisí na počtu slov v příslušných stránkách. Pravděpodobnost $P(c_j)$ se volí jako uniformní, ale Haveliwalla naznačuje možnost další personalizace pro jednotlivé uživatele právě pomocí této pravděpodobnosti.

Jestliže označíme množinu všech dokumentů, které vyhovují dotazu q jako Z, pak ohodnocení jednotlivých dokumentů $z \in Z$ provedeme takto:

$$s_{qz} = \sum_j P(c_j|q').r_{jd} \tag{3}$$

Ověření lokalizovatelnosti tohoto algoritmu pro české prostředí je jedním z cílů této disertační práce. Zhodnocení použitelnosti algoritmu pro reálné prostředí bude popsáno jako součást výsledků lokalizace.

3.3 Intelligent Surfer

Tento algoritmus opět vychází ze základního Pageranku a z principu náhodné procházky webovým grafem. Jeho autory jsou M. Richardson a P. Domingos[13]. Namísto náhodného chování předpokládají autoři u uživatele určitou inteligenci - tedy sledování odkazů, které jsou relevantní vzhledem k původnímu dotazu, případně náhodný skok na stránku, která ale opět je v relaci k původnímu dotazu. Odtud tedy název celého algoritmu - Intelligent Surfer (IS).

Základní myšlenkou tohoto algoritmu je nahrazení modelu zcela náhodně jednajícího uživatele modelem uživatele, který přemýšlí – tedy chová se inteligentně. Reálný uživatel nepřechází mezi stránkami zcela náhodně, tak jak to předpokládá Markovovský model náhodné procházky. Vybírá si stránky podle jejich obsahu a podle klíčových slov, která vyhledává. Pravděpodobnost, že skončí na stránce j, označme jako P_j. Tuto pravděpodobnost určíme tak, že pravděpodobnost náhodného skoku sečteme s pravděpodobností přechodu na další stránku pomocí některého z odkazů. Na rozdíl od klasického výpočtu PageRanku ale tuto pravděpodobnost vyjádříme v návaznosti na dotaz uživatele označený jako q.

$$P_q(j) = (1-\alpha)P_q'(j) + \alpha \sum_{i \in B_i} P_q(i)P_q(i \to j) \tag{4}$$

Kde B_i je množina příchozích odkazů na stránku i, $P_q(i \to j)$ pravěpodobnost přechodu na stránku j využitím odkazu, α je teleporační koeficient (stejně jako u PageRanku) a konečně $P_q'(j)$ je pravděpodobnost přechodu na jinou stránku přímým zadáním adresy (tedy teleportace).

Principem algoritmu je rozdělení celého původního grafu W na velký počet menších podgrafů. Klíčem pro rozdělení grafu je množina slov S – slovník všech

slov. Pro každé slovo $s_i \in S$ tvoří podgraf původního grafu W ty uzly, které obsahují v textu slovo s_i. Pro takto vygenerovaný podgraf provedeme výpočet PageRanku obvyklým způsobem.

Při sestavování matice G_{si} pro podgraf generovaný slovem s_i stanovíme jak pravděpodobnost přechodu pomocí odkazu, tak pravděpodobnost teleportace v závislosti na funkci $R_q(j)$. Což je funkce určující důležitost slova q ve stránce j. Pravděpodobnosti vyjádříme takto:

$$P'_q(j) = \frac{R_q(j)}{\sum_{n \in W} R_q(n)} \qquad P_q(i \to j) = \frac{R_q(j)}{\sum_{n \in F_i} R_q(n)} \tag{5}$$

Kde $n \in W$ je některý z uzlů webgrafu a F_i je množina odchozích odkazů ze stránky i.

Funkce $R_q(j)$ je definována takto:

$$R_q(j) = \begin{cases} 1 \text{ pokud } q \in j \\ 0 \text{ v opačném případě} \end{cases}$$

Výsledný QD-rank pro dotaz Q tvořený množinou slov $Q = q_1, q_2, .., q_n$ je pak lineární kombinací vektorů jednotlivých slov. Algoritmus omezuje výsledky pouze na ty dokumenty, které vyhovují všem slovům z dotazu Q.

Ověření možnosti lokalizace algoritmu IS a jeho urychlení tak, aby byla možná jeho reálná implementace je dalším z cílů této disertační práce.

4 Způsob řešení

4.1 CZDIS - lokalizovaný algoritmus Intelligent Surfer

Algoritmus IS přináší vylepšenou hodnotící funkci oproti standardnímu PageRanku, ale zároveň nabízí několik problémů k dopracování a dořešení, a to v oblasti škálovatelnosti a rychlosti výpočtu.

4.1.1 Snížení poměru U/N aplikací jazykového modelu a statistky

Protože s nárůstem počtu stránek N vzrůstá také počet slov a tím i počet unikátních dvojic dokument–slovo označený jako U, můžeme předpokládat, že poměr U/N se s rostoucím počtem stránek příliš nemění. Jedinou možností jak ovlivnit poměr U/N ve prospěch rychlosti výpočtu je tedy snížení počtu slov ve slovníku S. Toho lze dosáhnout postupnou aplikací pravidel jazykového modelu i využitím statistiky sledovaných textů.

Prvním krokem ke snížení velikosti slovníku S je vypuštění často opakovaných výrazů – stop slov. V oblasti získávání informací z textů jde o často využívanou operaci, používanou obvykle již během indexace. Jako stop slova označujeme výrazy, které se v konkrétním jazyce často opakují, ale nenesou žádnou významovou informaci.

Seznam českých stop slov má 257 položek a lze ho najít na [14]. Přestože se množina stop slov jeví jako poměrně malá, generuje obsáhlé podgrafy a její podíl na vytvoření U je nemalý (v poměru k její velikosti). Odstranění stop slov je i součástí původního řešení IS.

Další možností, jak snížit počet slov ve slovníku a přitom zachovat, či dokonce zlepšit kvalitu vyhledávání, je použít ve slovníku pouze základní tvary

slov – stemmy a lemmata. Lemma příslušného slova je jeho základní slovníková jednotka. V případě češtiny a podstatných či přídavných jmen jde obvykle o první pád jednotného čísla. Stem je pak kmenem či kořenem slova – zpravidla vzniká odstraněním přípony či předpony. Samotný stem pak nemusí být smysluplné slovo, v tom je hlavní rozdíl mezi stemem a lemmatem. U některých slov se mohou oba tyto základní tvary shodovat, u jiných ne.

Pro český jazyk je bohužel častější druhý případ. Čeština má velmi složité tvarosloví, řadu výjimek a nepravidelností. Lemmatizace se tak obvykle realizuje za pomoci slovníkových dat, protože algoritmicky je velmi složité ošetřit všechny výjimky. Český stemming je naproti tomu realizovatelný i algoritmem, na základě sady pravidel. Jedním z takových algoritmů je práce [8].

Posledním krokem, který povede ke snížení velikosti U, je omezení algoritmu pouze pro slova se střední frekvencí výskytu. Jde o opačný postup než v případě stop slov. Stop slova generují příliš rozsáhlé podgrafy a z hlediska uspořádání výsledek spíše zkreslují než naopak. Na opačném konci frekvenčního spektra stojí slova, která můžeme nazvat unikátní. To jsou slova, která se vyskytují jen ve zlomku dokumentů a generují na rozdíl od stop slov velmi malé podgrafy. Ty navíc velmi často tvoří pouze izolované body bez propojujících hran, protože jednotlivé stránky obsahující tato unikátní slova spolu nejsou propojeny. Tento fakt činí unikátní slova nezajímavými z hlediska analýzy odkazů a tedy z hlediska PageRanku.

4.1.2 Personalizační vektor

Abychom mohli provést výpočet příslušného rank vektoru, budeme nutně potřebovat personalizační vektor $\mathbf{v}_s$. Ten vygenerujeme z pravděpodobnosti P'. Před tím je ale nutné určit, jakou funkci $R_q(j)$ použijeme. Místo základní binární funkce využijeme normalizovanou váhu slova v dokumentu $w_{s,d}$:

$$w_{s,d} = fs_{s,d}.\log\frac{|D|}{|s \in D|} \tag{6}$$

Kde $fs_{s,d}$ je frekvence výskytu slova s v dokumentu d, $|D|$ je počet všech dokumentů a $|s \in D|$ počet dokumentů obsahujících slovo s.

Ze vztahu pro P' můžeme nyní za pomoci vztahu pro $w_{s,d}$ odvodit $\mathbf{v}$ pro příslušné slovo d. Platí, že $v_{si} = w_{s,d}(i)$. Počáteční nastavení vektoru řešení $\mathbf{x} : x_i = 1/|q|$, kde $|q|$ je velikost generované podsítě a $i \in < 1, |q| >$.

4.1.3 Formulace paralelního algoritmu

Řešení lineárního systému sice přináší zvýšení rychlosti výpočtu jednoho výpočetního kroku [10], a tím i celého výpočtu provedeného sekvenčně, ovšem algoritmus IS je ze své povahy přirozeně paralelní problém. Jde o klasický příklad SPMD - opakování stále stejného výpočtu na S různých datech. Paralelní výpočet několika podgrafů je tak logickým krokem k dalšímu zrychlení.

Výpočet je navržen jako master–worker algoritmus s dynamcikou dekompozicí problému. Pro minimalizaci I/O operací jsou vygenerované datové struktury drženy v paměti a zasílány výpočetním uzlům. Vypočtené výsledky ukládají výpočetní uzly do sdíleného adresáře hlavního uzlu.

Nyní zbývá zformulovat oba algoritmy, jak řídící master proces, tak výpočetní worker proces.

Master algoritmus

1. nahrát inverzní index dokumentů příslušný slovníku S
2. pomocí indexu vygenerovat n úloh
3. všem n dostupným klientům přidělit úlohu
4. vygenerovat n úloh
5. dokud počet vygenerovaných úloh není roven počtu slov ve slovníku S, přidělovat výpočetním uzlům další úlohy, jakmile oznámí dokončení předchozí.
6. během tohoto cyklu stále udržovat n vygenerovaných úloh pro zajištění optimálního zatížení všech výpočetních uzlů
7. počkat na dokončení výpočtu na všech uzlech
8. ukončit program na všech uzlech

Worker algoritmus

1. dokud Master neukončí program provádí následující kroky
2. načíst data úlohy
3. řešit lineární systém
4. uložit příslušný vyřešený vektor

4.2 TSPRL – lokalizace algoritmu Topic Sensitive PageRank

Podle publikovaných výsledků se zdá být algoritmus TSPR poměrně použitelným řešením. Určitou slabinou algoritmu je pouze volba kontextu dotazu, kde je použit teoretický model volby kontextu pomocí vyznačení slova v určité stránce.

Lokalizace algoritmu má dva hlavní problémy. Tím prvním je nalezení dostatečně velké báze základních adres, potřebných pro sestavení tématických klasifikátorů T_k. Druhý problém je vytvoření frekvenčních slovníků potřebných pro trénování Bayesovského klasifikátoru. Při řešení druhého problému je ale již možné využít nalezené základní adresy k získání potřebných textových dat.

4.2.1 Nalezení báze pro vytvoření tématických klasifikátorů

Haveliwalla využil pro sestavení tématických klasifikátorů Open Directory Project. Jako hodnotu k zvolil 16, což je počet hlavních kategorií ODP. Z celkem 3 milionů adres, které ODP v roce 2002 obsahoval, použil 280 tisíc stránek, které byly součástí datové kolekce WebBase, kterou měl k dispozici pro testy.

Jestliže v roce 2002 byla odhadovaná velikost webu cca 2 miliardy stránek [15] a WebBase kolekce měla 120 milionů stránek, tvořila použitá báze 0,01% velikosti webu a 0,23% velikosti testovací kolekce.

V současné době obsahuje ODP 4,6 milionů stránek v různých jazycích. Českých stránek eviduje ovšem pouze 26 tisíc. Pokud vyjdeme z odhadované

k	*téma*	*počet URL*
1	dům a bydlení	2304
2	eshopy	873
3	hry	9621
4	kultura	19176
5	obchod	4713
6	počítače	2842
7	instituce	5075
8	společnost	27565
9	sport	10926
10	cestování	7089
11	věda	5537
12	volný čas	11012
13	zdraví	1782
14	zpravodajství	3928
	celkem	112443

Tabulka 1: Velikost jednotlivých tématických klasifikátorů T_k.

velikosti českého webu, představuje báze tvořená stránkami v české části ODP opět 0,01% ze 300 milionů. Nejrozsáhlejším katalogem českých stránek ovšem není ODP, ale katalog Seznamu. Tento katalog obsahuje více než 110 tisíc adres, což představuje asi 0,03% webů v českém jazyce. Spojením těchto dvou zdrojů dohromady, dostaneme po odečtení duplicitních adres, 112 tisíc URL.

Nově vytvořené kategorie byly poté použity pro algoritmus TSPRL. Odkazy nejsou mezi jednotlivé kategorie rozděleny rovnoměrně, počty URL v jednotlivých sekcích ukazuje tabulka 1. Tyto počty jsou zároveň počtem nenulových prvků personalizačních vektorů $\mathbf{v}_k$.

Uvedených 112 tisíc URL posloužilo také jako startovací množina stránek pro vytvoření datové kolekce *CZdat2*, která sloužila jako testovací množina pro experimenty s lokalizovanou verzí TSPR.

Počet URL dostupných pro vytváření tématických klasifikátorů vzniklých sloučením obou zdrojů je k celkovému počtu českých stránek v řádově stejném poměru jako u originálního algoritmu. Protože testovací množina CZdat2 byla výrazně menší než WebBase, tvoří URL získané báze 2,18% všech testovacích dat, což je výrazně více než tomu bylo u Haveliwallova experimentu. Zaměřování pomocí této báze by tak mělo teoreticky přinést srovnatelné, nebo dokonce lepší výsledky.

4.2.2 Určení příslušnosti dotazu k určitému tématu

Abychom se mohli rozhodnout, kterou z kategorií použijeme pro zaměřování dotazu, musíme pochopitelně nejprve určit do jaké kategorie dotaz pravděpodobně patří. V originálním algoritmu je k tomu využito Naivního Bayesovského klasifikátoru.

Pro určení, s jakou pravděpodobností spadá kontext dotazu Q do kategorie j reprezentované třídou c_j tedy využijeme vztahu (2) popsaného v kapitole 3.2. Jestliže stejně jako v originálním algoritmu nastavíme pravděpodobnost

jednotlivých tříd uniformní na hodnotu $P(c_j) = 1/k$, zbývá nám určit pravděpodobnost, s jakou jednotlivá slova přísluší k určité třídě - $P(q_i|c_j)$.

Na rozdíl od originálního algoritmu provedeme určení pravděpodobností $P(q_i|c_j)$ offline. Využijeme k tomu frekvenční slovníky vytvořené ze slov, která obsahují stránky tvořící bázi určité kategorie. Tyto slovníky poslouží jako základní prvky pro konstrukci unigramového jazykového modelu. Slovník S_j si definujeme jako množinu všech slov (termů) v bázi kategorie j. Každý slovník bude zároveň reprezentován vektorem $\mathbf{f}_j$. Tento vektor definujeme jako n-rozměrný a n jako počet slov v kompletním slovníku báze, který můžeme definovat jako:

$$S = \bigcup_k S_j.$$

Pro jednotlivé prvky vektoru $\mathbf{f}_j$ pak platí, že:

$$f_{ji} = \begin{cases} \phi(i) \text{ pokud je slovo } i \in S_j \\ 0 \text{ v opačném případě} \end{cases}$$

Kde $\phi(i)$ je frekvence výskytu slova i ve slovníku S_j. Z takto sestavených vektorů následně sestrojíme matici $\mathbf{D}$, pro kterou platí, že její j-tý sloupec je tvořen vektorem $\mathbf{f}_j^T$. Normalizací prvků matice $\mathbf{D}$ dle vztahu:

$$d_{m,n} = d_{m,n}/sum(d_m)$$

získáme matici $\mathbf{D}_{norm}$, jejíž jednotlivé prvky tvoří hledané pravděpodobnosti $P(q_i|c_j)$. Řádek matice pak tvoří pravděpodobnostní vektory pro jednotlivá slova $q \in S$.

V případě báze datové kolekce CZdat2 byla matice $\mathbf{D}_{norm}$ (n, k) rozměrná, přičemž $k = 14$ a $n = 838238$, což byl celkový počet slov ve slovníku báze definované celkem 112443 stránkami.

4.2.3 Paralelní výpočet jednotlivých vektorů

Výpočet jednotlivých, tématicky zaměřených vektorů je přirozeně paralelním problémem, protože algoritmus TSPR jako celek lze snadno rozložit na k sekvenčních problémů. U těchto dílčích úloh navíc není nutná žádná synchronizace a komunikace během výpočtu. Pro tuto základní paralelizaci výpočtu tedy potřebujeme pouze mít k dispozici k výpočetních uzlů + sdílený adresář pro zdrojová a výsledná data.

Jednotlivé personalizační vektory $\mathbf{v}_j$ pro $j = 1 \ldots k$ sestavíme za pomoci tématických klasifikátorů T_j takto:

$$v_{ji} = \begin{cases} \frac{1}{T_j} \text{ pokud } i \in T_j \\ 0 \text{ v opačném případě} \end{cases}$$

4.2.4 Algoritmus ohodnocení výsledků dotazu

Ohodnocení výsledků dotazu Q je online proces, kterému předchází sestavení několika různých indexů offline. URL všech stránek v kolekci CZdat2 (či obecně ve zpracovávaných datech) je nutné zaindexovovat do klasického inverzního indexu dokumentů a dále je potřeba vytvořit index bázových slov, který ke každému slovu přiřadí příslušný řádkový vektor matice $\mathbf{D}_{norm}$.

název kolekce	*stránek*	*odkazů*	*velikost slovníku*
California	9664	16773	–
tul	45531	150789	315498
CZdat	998037	3822430	1805572
CZdat2	4980930	34998687	5598268

Tabulka 2: Datové kolekce použité pro experimenty.

Posledním indexem potřebným pro vyhodnocení dotazu je index jednotlivých tématických ranků, který pro každý z dokumentů v kolekci CZdat2 přiřazuje buď 15 různých hodnot tématického hodnocení (14 kategorií + obecné), případně pouze hodnotu rankingu pro kategorii *obecné*, která je vypočtena jako normální (nezaměřený) PageRank.

Online zpracování dotazu Q pak bude probíhat dle následujícího algoritmu:

1. Dotaz Q rozdělit na jednotlivá slova q_i.
2. Otestovat existenci jednotlivých slov $q \in Q$ v indexu bázových slov. Pokud bylo slovo v indexu nalezeno, spárovat slovo a příslušný řádek matice $\mathbf{D}_{norm}$ obsahující pravděpodobnosti jeho výskytu v jednotlivých třídách. Pokud slovo v indexu bázových slov není, použít třídu *obecné* a pravděpodobnost příslušnosti k této třídě nastavit na 1.
3. Opakováním kroku 2. přiřadit pravděpodobnost $\forall q_i \in Q$.
4. Pomocí inverzního indexu dokumentů sestavit množinu výsledků V. Do této množiny zahrnout pouze stránky obsahující všechna slova z dotazu Q.
5. Ohodnotit jednotlivé výsledky podle vzorce (3): $s_{qv} = \sum_j P(c_j|q).r_{jd}$.

5 Experimentální ověření

5.1 Testovací data

Pro experimenty provedené v rámci experimentálního ověření byly použity celkem čtyři různé datové kolekce. Základní přehled těchto datových kolekcí přináší tabulka 2.

5.2 CZDIS - lokalizovaný algoritmus Intelligent Surfer

5.2.1 Redukce slovníku S o unikátní slova

Pro odstranění stop slov využíváme jejich seznam, někdy označovaný jako *negativní slovník*. Pro slova, která byla v teoretické části označena jako slova unikátní, takovýto slovník k dispozici nemáme. Pro jeho vygenerování je nejprve nutné stanovit si kritérium hodnocení, podle kterého rozhodneme, zda je slovo unikátní či nikoliv.

Jako kritérium unikátnosti byla v této práci zvolena frekvence výskytu slova v dokumentech. Slovník generovaný kolekcí CZdat, obsahuje celkem 1805572

datová kolekce	*velikost S size*	*U*	*poměr U/N*
tul neredukovaná	315498	6853644	150
tul bez stop slov	315268	6630472	145
tul pouze stemy slov	242371	6053120	133
tul_100 - bez unikátních sl.	8903	4789301	105
CZdat neredukovaná	1805572	157689876	158
CZdat bez stop slov	1805342	152555094	153
CZdat pouze stemy slov	1387907	136080806	137
CZdat_100 - bez unikátních sl.	100538	106512965	107

Tabulka 3: Redukce velikosti slovníku S aplikací jazykového modelu.

slovních tvarů. Z nich pouze 23651, tedy 1,31%, se vyskytuje ve více než 500 dokumentech z této kolekce, která celkem obsahuje 998037 dokumentů. Tato slova můžeme označit jako běžná.

Pro další hledání kritéria unikátnosti tedy zohledníme slova, která se vyskytují v méně než 500 dokumentech. Pokud na tato slova použijeme Paretův princip 80/20 [12] pro vyřazení 20% nejunikátnějších slov, dostaneme se právě k číslu 100 jako kritériu pro unikátnost slova. Jako unikátní slova tedy budou vyřazena taková slova, která se vyskytují ve 100 a méně dokumentech.

Postupnou aplikací všech tří teoretických kroků - seznamu stop slov, stemmeru a vypuštění unikátních slov, dojde k postupné redukci slovníku S. K největší redukci vede krok třetí, tedy vypuštění unikátních slov. Výsledky aplikace jazykového modelu zobrazuje tabulka 3. Jednotlivé kroky jsou na sobě nezávislé a je možné je provést idividuálně.

Pro urychlení celého procesu by bylo přínosnější nejprve provést redukci slovníku o unikátní slova a teprve po tomto kroku provádět stemming. Ten by se prováděl na několikrát menší množině slov. Převedení slov na stemy ovšem vede k tomu, že řada původně unikátních slov je nyní převedena na stejný tvar a generují tak svou podsíť, která je zajímavá pro další výpočet.

5.2.2 Efektivita paralelního výpočtu

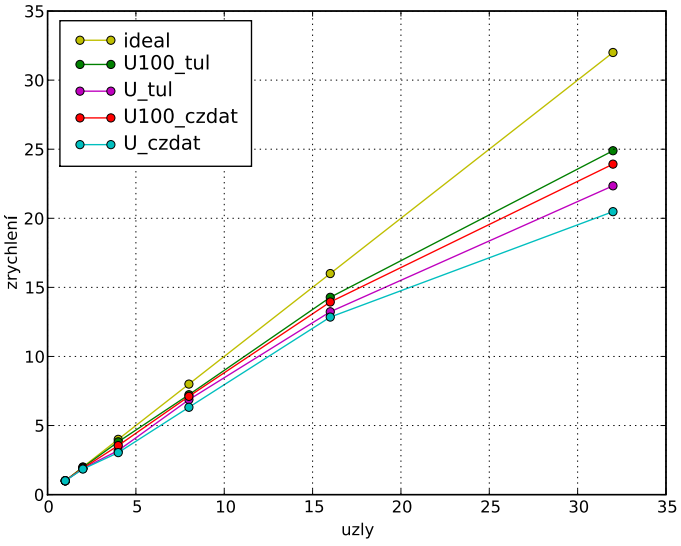
Výpočet byl proveden celkem na čtyřech testovacích slovnících. Vždy byl použit slovník úplný a poté slovník redukovaný. Pro porovnání byl nejprve celý algoritmus proveden jako sekvenční výpočet na jednom uzlu. Dále byl vypočten také klasický PageRank vektor a to jako sekvenční výpočet lineárního systému. Následně byl algoritmus CZDIS spuštěn paralelně, přičemž byl po každém výpočtu zvýšen počet výpočetních uzlů na dvojnásobek a to až do počtu 32.

Časy jednotlivých výpočtů pro kolekci CZdat jsou uvedeny v tabulce 4. Uvedený čas představuje celkový čas paralelního výpočtu včetně komunikace a sestavování dílčích úloh. Poměr zrychlení výpočtu, vypočtený jako podíl celkového času paralelního a sekvenčního výpočtu, je pak zobrazen v grafu na obrázku 1. Experimentální výsledky potvrzují teoretický předpoklad, že redukcí slovníku na slova generující větší podsítě dojde ke snížení režie celého výpočtu.

Malé podsítě snižují efektivitu algoritmu, protože je jich příliš mnoho a náklady na vytvoření a komunikaci jsou výrazně větší než vlastní čas výpočtu. To se projevuje horší škálovatelností algoritmu – master nestíhá dostatečně rychle

	CZdat_100		CZdat	
počet uzlů	zrychlení	čas (s)	zrychlení	čas (s)
1	1,00	56,131	1,00	939,666
2	1,91	29,388	1,85	507,928
4	3,55	15,811	3,04	309,101
8	7,11	7,895	6,32	148,681
16	13,94	4,027	12,85	73,126
32	23,92	2,347	20,48	45,882

Tabulka 4: Čas a zrychlení výpočtu pro slovník CZdat.



Obrázek 1: Zrychlení (speedup) výpočtu.

	tul_100	*CZdat_100*	*tul*	*CZdat*
pagerank	0,162	14,820	0,162	14,820
sekvenční výpočet	4,688	56,131	81,789	939,67
teoretický poměr U/N	79	80	99	102
reálný poměr	28,936	3,787	504,870	63,41
čas výpočtu na 32 proc.	0,188	2,347	3,659	45,88
reálný poměr 32 proc.	1,163	0,158	22,589	3,10

Tabulka 5: Poměr PageRank/CZDIS.

generovat příslušné dílčí úlohy a některé výpočetní uzly musejí čekat na přidělení úlohy. S rostoucím počtem výpočetních uzlů tak sice stále klesá výpočetní čas, ale bohužel i efektivita celého výpočtu.

U redukovaných slovníků je škálovatelnost lepší, nicméně i zde s rostoucím počtem výpočetních uzlů přestává zrychlení výpočtu kopírovat ideální křivku a začíná klesat.

Jak můžeme vidět z výsledků v tabulce 5, teoretický poměr U/N a reálný poměr mezi časy výpočtu PageRank/CZDIS se poměrně výrazně liší. Teoretický poměr pro výpočet určený ze vztahu $0.75 * U/N$ [13] je v intervalu mezi 79 pro redukovaný slovník tul a 102 pro neredukovaný slovník CZdat. Reálný poměr je ve třech případech ze čtyř menší a výpočet tedy probíhá rychleji než teoretický odhad.

Ovšem pro neredukovaný slovník tul je reálný poměr pětkrát větší než teoretický. Logickou otázkou je, proč? Důvodem je opět redukce slovníku S a režie spojená se zpracováním malých sítí. Pro neredukovaný slovník je nutno vytvářet řadu malých podsítí, což vede ke značnému zpomalení celého procesu.

Pro srovnání byl vypočten také poměr mezi rychlostí výpočtu algoritmu CZDIS na 32 procesorech a sekvenčním výpočtem PageRanku. Z výsledků tohoto výpočtu je zřejmé, že časová náročnost paralelního výpočtu je přibližně třikrát větší, než pro klasický výpočet PageRanku. Ten je pro nasazení algoritmu nutné provést v každém případě, abychom získali hodnotící funkci pro unikátní slova. Celkový čas výpočtu hodnotící funkce algoritmu CZDIS je tak přibližně čtyřnásobkem času potřebného pro výpočet PageRanku. To dává dobré předpoklady pro reálné nasazení algoritmu CZDIS. Otázka, do jaké míry by se tento poměr upravil v případě paralelního výpočtu samotného PageRanku, bude předmětem dalšího zkoumání. Stejně jako celý PageRank vektor je totiž možné počítat paralelně také některé z rozsáhlejších generovaných podgrafů v algoritmu CZDIS.

5.2.3 Kvalita lokalizované hodnotící funkce

Pro testování kvality hodnotící funkce byla v případě algoritmu CZDIZ i ve zbytku této práce použita stejná metodika jakou použili autoři algoritmu IS, s několika drobnými rozdíly. Počet testujících dobrovolníků byl zvýšen na 5 (v originále 3) a bodová škála byla rozšířena na pětistupňovou od 0, pro zcela nerelevantní výsledek, do 4, pro nejkvalitnější výsledek.

Do výsledků testování byla zahrnuta pouze množina stránek CZdat, protože se v průběhu testování ukázalo, že množina tul obsahuje příliš malé množství stránek a generované podsítě jsou tvořeny spíše izolovanými body a výsledná

hledaný výraz	CZDIS	PageRank
aktuální akce	24,6	28,4
české články	18,2	24,8
financování	28,4	16,8
jazyk	20,5	10,2
dopis	28,4	16,2
pole	23,4	14,2
rozvrh	27,3	19,8
univerzitní ústav	23,2	29,8
aplikace	34,2	20,5
číslicový	23,4	23,8
průměrné hodnocení	25,16	20,45

Tabulka 6: Průměrná známka pro zadané dotazy

hodnotící funkce je zkreslená.

Pro výpočet hodnotící funkce CZDIS byl využit redukovaný slovník S100 a k němu příslušející množina q-vektorů vypočtená již při experimentech s rychlostí výpočtu. Pro porovnání, a také jako řadící funkce pro slova mimo redukovaný slovník S100, byl vypočten také PageRank vektor příslušející k množině CZdat.

Každý z dobrovolníků nejprve zvolil dva náhodné dotazy, čímž byla vytvořena množina deseti různých náhodně vybraných dotazů. Dále pak každý postupně zadal všech 10 dotazů a následně hodnotil navrácené výsledky. Výsledků měl k dispozici 20 – první polovina bylo 10 nejlepších výsledků CZDIS, druhá 10 nejlepších výsledků PageRanku. Těchto 20 výsledků bylo zobrazeno v náhodném pořadí, aby nedošlo k ovlivnění testujícího. Každý z výsledků byl ohodnocen známkou od 0 do 4 a z těchto známek se pak vypočetlo výsledné hodnocení.

Výpočet výsledků probíhal ve dvou krocích – nejprve byly sečteny výsledky každého z dobrovolníků pro každou hodnotící funkci. Maximální možné skore bylo 40 bodů. Druhým krokem pak byl výpočet průměrného bodového hodnocení, které je zobrazeno v tabulce 6.

Také pro český jazykový model byly zopakovány výsledky původního algoritmu IS a hodnotící funkce dosáhla lepších výsledků než PageRank. Lze tedy říci, že lokalizace algoritmu byla úspěšná a že algoritmus je při použití vhodného slovníku přenositelný i na jiné jazyky než angličtinu.

5.3 TSPRL - lokalizovaný Topic Sensitive PageRank

V rámci testování lokalizovaného algoritmu byl nejprve proveden test efektivity paralelního výpočtu a poté proběhlo otestování samotné hodnotící funkce.

5.3.1 Efektivita paralelního výpočtu

Výpočet všech 14 vektorů byl nejprve proveden na jednom výpočetním uzlu sekvenčně. Měření času probíhalo stejně jako v předchozím případě pomocí časového razítka procesoru. Měřen byl tentokrát celkový čas běhu algoritmu, nikoliv samotný výpočet. Důvodem pro toto rozhodnutí byla rozdílná I/O režie při sekvenčním a paralelním výpočtu. Paralelní výpočet byl proveden na 14

sekvenční výpočet	231,70 s
paralelní výpočet na 14 uzlech	42,63 s
zrychlení paralelního výpočtu	5,43
efektivita paralelního výpočtu	0,39

Tabulka 7: Efektivita výpočtu algoritmu TSPRL na množině CZdat2.

uzlech clusteru Hydra. Opět byl měřen čas od okamžiku aktivace jednotlivých výpočetních skriptů příkazem cluster-fork, až do okamžiku uložení posledního vypočteného vektoru na disk hlavního uzlu clusteru.

Z naměřených časů uvedených v tabulce 7 vidíme, že režie I/O operací se negativně podepisuje na efektivitě paralelního výpočtu. Efektivita vypočtená jako podíl zrychlení a počtu procesorů je pouze 0,39. K největšímu zpoždění dochází před samotným zahájením výpočtu v době, kdy je soubor s maticí A (486 MB ve formátu PETSc.SEQAIJ) přenášen na jednotlivé výpočetní uzly.

I přes relativně nízkou efektivitu se paralelní výpočet provede 5,4 krát rychleji, takže při dostupné volné výpočetní kapacitě je vhodnou metodou pro výpočet jednotlivých vektorů. Pro rozsáhlejší datové struktury, než je CZdat2, by bylo nutné věnovat optimalizaci přenosů více pozornosti.

5.3.2 Kvalita lokalizované hodnotící funkce

Pro testování kvality byla využita stejná metodika jako u algoritmu CZDIS, tedy skupina 5 testujících dobrovolníků, která hodnotila jednotlivé výsledky vyhledávání pětibodovou škálou (viz. kap. 5.2.3). Testování probíhalo nad množinou CZdat2, která byla sestavena právě pro tento účel.

Stejně jako v případě algoritmu CZDIS, i tentokrát každý z testujících dobrovolníků nejprve náhodně zvolil dva dotazy. Celkem tedy bylo opět k dispozici 10 náhodně zvolených dotazů. Pro každý dotaz vrátil experimentální systém 10 nejlépe hodnocených výsledků dle tématického ranku a 10 nejlépe hodnocených výsledků s nezaměřeným PageRankem. Z hodnocení od jednotlivých dobrovolníků byl vypočten aritmetický průměr, který je uveden v tabulce 8 (sloupce TSPRL a PageRank).

U sedmi testovacích dotazů byla spokojenost dobrovolníků s uspořádáním pomocí funkce TSRPL lepší, než s uspořádáním pomocí funkce PageRank. Také celková průměrná známka pro TSPRL je lepší. Lze tedy říci, že i v případě TSPR algoritmu je jeho lokalizace do češtiny možná a že se podařilo potvrdit teoretický předpoklad dobrých výsledků hodnotící funkce TSPRL na testovací množině CZdat2.

5.4 Srovnání algoritmů CZDIS a TSRPL

Logickým závěrem experimentů je porovnat oba lokalizované algoritmy mezi sebou.

Na datovou kolekci CZdat2 byl aplikován algoritmus CZDIS tak, jak byl představen v kapitole 4.1. Aplikací jazykového modelu vznikl z originálního slovník S o velikosti 5,5 mil. slov redukovaný slovník S_100, který měl pro kolekci CZdat2 velikost 225057 slov. Sekvenční výpočet jednotlivých vektorů na po-

dotaz / algoritmus	TSPRL	CZDIS	PageRank	Kombinace
horské kolo	32,2	20,2	21,8	32,2
vážná hudba	25,2	18,2	18,7	25,2
český ráj	21,2	17,8	17,2	21,2
lyžování	24,1	31,5	12,1	31,5
předpověď počasí	28,2	19,2	22,7	28,2
dovolená	17,6	25,2	20,2	25,2
kanárské ostrovy	29,8	22,1	18,2	29,8
rýma	18,8	25,6	14,3	25,6
počítačová simulace	27,4	19,3	23,2	27,4
jazyk	15,2	28,4	20,1	28,4
průměrná známka	23,97	22,75	18,85	27,47

Tabulka 8: Průměrná známka algoritmů TSPRL a CZDIS pro jednotlivé testovací dotazy.

horské kolo	
sport	0,42
obchod	0,19
cestování	0,18
eshopy	0,16

vážná hudba	
kultura	0,48
zpravodajství	0,21
eshopy	0,13
obchod	0,08

český ráj	
cestování	0,35
obchod	0,29
sport	0,11
zpravodajství	0,08
instituce	0,05

lyžování	
eshopy	0,36
obchod	0,22
sport	0,19
zpravodajství	0,12
cestování	0,09

předpověď počasí	
zpravodajství	0,68
cestování	0,11
věda	0,08
instituce	0,03

dovolená	
obchod	0,78
cestování	0,12
zpravodajství	0,06

kanárské ostrovy	
obchod	0,56
cestování	0,34

rýma	
zdraví	0,54
zpravodajství	0,25
obchod	0,12

počítačová simulace	
hry	0,38
počítače	0,18
eshopy	0,12
věda	0,09

jazyk	
počítače	0,19
zpravodajství	0,19
kultura	0,18
věda	0,16
zdraví	0,12
instituce	0,07

Tabulka 9: Pravděpodobnosti $P(c_j|q)$ pro jednotlivé testovací dotazy. V tabulkách jsou uvedeny pouze kategorie s $P >= 0,05$.

čítači Lexis trval 198 sekund. Vektory byly zpracovány do inverzního indexu, který pak sloužil programu IndexServer pro uspořádání výsledků.

Z výsledků uvedených v tabulce 8 je vidět, že i na množině CZdat2 se algoritmus CZDIS choval podobně jako v předchozím testu. V případě jednoslovných dotazů měl lepší hodnocení než oba zbývající algoritmy. Algoritmus TSPRL dosáhl lepšího výsledku u 6 dotazů – všechny byly složeny ze dvou slov. CZDIS pak dosáhl lepšího hodnocení u 4 jednoslovných dotazů a to poměrně výrazně. Průměrná známka obou personalizovaných algoritmů je lepší než průměrná známka normálního PageRanku. Sloupec nazvaný *kombinace* zobrazuje teoretické výsledky kombinovaného algoritmu spojujícího TSPRL a CZDIS (viz. dále).

6 Závěr

V disertační práci byly popsány výsledky experimentálního ověření možnosti lokalizace algoritmů pro uspořádání výsledků fulltextového vyhledávače, založených na kontextu dotazu. Podařilo se prokázat, že anglický jazykový model použitý pro návrh originálních algoritmů, je možné úspěšně nahradit modelem českého jazyka.

Konkrétně byla provedena lokalizace algoritmů Topic Sensitive PageRank a Intelligent Surfer. Obě lokalizované verze byly v rámci uživatelského testování hodnoceny skupinou pěti dobrovolníků. Návrh lokalizovaného algoritmu CZDIS byl přijat ke zveřejnění (duben 2009) na mezinárodní konferenci BIS 2009 [2].

Výsledky testů lokalizovaných verzí prokazují, že kvalita algoritmů nebyla lokalizací snížena. Hodnotící funkce, vzniklá kombinací obou lokalizovaných algoritmů by pak mohla být velmi kvalitním prostředkem pro uspořádání stránek a to jak pro jednoslovné, tak pro víceslovné dotazy (kapitola 6.3).

Dalším přínosem práce je urychlení výpočtu hodnotících vektorů lokalizovaných algoritmů. Toho bylo dosaženo nahrazením sekvenčního výpočtu vektoru paralelním řešením řídkého lineárního systému. Tento krok spolu s aplikací jazykového modelu umožňuje nasadit lokalizované algoritmy i v reálném provozu. Výsledky experimentu s paralelním výpočtem více hodnotících vektorů, byly publikovány ve sborníku mezinárodní konference INTED 2009 [1].

Pro účely testování lokalizovaných algoritmů byl vyvinut a implementován experimentální systém, který lze, po dořešení některých nedostatků odhalených během jeho používání, využívat i pro další výzkumy v oblasti ZIW.

6.1 CZDIS - Zhodnocení úspěšnosti lokalizace

Úspěšná implementace českého jazykové modelu ukazuje, že algoritmus inteligent surfer je použitelný jako hodnotící funkce také pro některé jiné jazyky než angličtinu. To bylo hlavním cílem této práce. Protože algoritmus IS byl kritizován jako příliš pomalý pro reálnou aplikaci, bylo dalším cílem zrychlení procesu výpočtu. Potvrdil se teoretický předpoklad, že vhodná aplikace jazykového modelu vede i ke zrychlení celého výpočtu, aniž by tím utrpěla kvalita hodnotící funkce. Paralelní výpočet rank vektorů pak vede k dalšímu zrychlení výpočtu do řádu minut, což činí z algoritmu CZDIS použitelné řešení pro řazení výsledků fulltextového vyhledávání založené na kontextu dotazu.

Stejně jako v případě původního algoritmu i v případě lokalizované verze dosahují víceslovné dotazy v průměru horšího hodnocení než PageRank, zatímco

u jednoslovných dotazů je tomu naopak. Důvodem je zkreslení, ke kterému dojde v okamžiku kdy z vektorů hodnocení pro jednotlivá slova sestavujeme jeden hodnotící vektor pro víceslovný dotaz. V situaci, kdy se tato dvě slova vyskytují jako ustálené slovní spojení, je zkreslení větší. Řešením tohoto problému spočívá v rozšíření základního unigramového slovníku i o nejpoužívanější bigramy, se kterými pak bude dále z hlediska algoritmu nakládáno stejně jako s unigramy.

Aby mohl být algoritmus IS označen za skutečně univerzální řešení, bude nutné ověřit jeho použitelnost pro více jazyků. Lze předpokládat, že je použitelný pro jazyky západního typu. Existuje nicméně řada dalších jazyků, pro které obecně platí, že text IR je netriviální problém, jako arabština či jazyky východní Asie, a nelze se domnívat, že by algoritmu IS měl tvořit výjimku.

6.2 TSPRL - zhodnocení úspěšnosti lokalizace

Podařilo se ověřit, že lokalizace algoritmu TSPR do češtiny je možná. V řadě případů bylo ovšem hodnocení obou porovnávaných algoritmů téměř stejné a přínos tématicky zaměřeného hodnocení nebyl tak velký, jak bylo očekáváno. Pokusme se nyní zodpovědět otázku proč. Odpověď je potřeba hledat v oblasti sestavení zaměřovaných vektorů – tedy v pravděpodobnosti přiřazení dotazu k jednotlivým třídám.

Výstup Bayesovského pravděpodobnostního modelu – pravděpodobnost s jakou dotaz Q patří do kategorie j, ukazuje pro jednotlivé testovací dotazy tabulka 9. Do jednotlivých tabulek byly zaznamenány pouze ty kategorie, pro které byla $P(c_j|q) >= 0,05$. Můžeme vidět, že zejména v případě jednoslovných dotazů nebyl pravděpodobnostní model příliš úspěšný. Kontext dotazu je v tomto případě příliš malý na to, aby mohl přesně určit téma. Dobře to demonstruje mnohovýznamové slovo „jazyk“.

Na algoritmu TSPR je dobře vidět rychlost vývoje v oblasti WWW. Za dobu, která uplynula od publikace originálního algoritmu TSPR, znalosti webmasterů v oblasti SEO hodně pokročily, a výsledek algoritmu TSPRL by byl poměrně hodně ovlivnitelný pomocí SEO technik, ať již etických či nikoliv. I pokud vynecháme z hodnocení problematiku SEO Spamu, může prostý obsah stránek hodně ovlivnit hodnocení pomocí TSPR(L). Například u dotazu „horské kolo“ se na stránce elektronického obchodu zaměřeného na prodej kol bude tento výraz vyskytovat u každé z položek v katalogu, zatímco u sportovního článku, který by mohl být z hlediska uživatele hodnocen jako kvalitnější zdroj, se hledané slovní spojení může vyskytnout jednou či dvakrát.

Tento fakt je dobře patrný na výsledcích uvedených v tabulce 9. Lze vypozorovat, že téměř pro všechny testovací dotazy je velmi pravděpodobné zaměření na kategorie obchod či eshopy. Důvodem může být to, že při tvorbě komerčních projektů je obvykle věnována daleko větší pozornost SEO technikám. Klíčová slova se v textu vyskytují tak často, jak je to možné, stránky jsou často odkazovány v rámci různých výměnných sítí apod. Navíc je tato kategorie poměrně široká, tak jako je široké spektrum různých obchodů. Je zřejmé, že sestavení tématických klasifikátorů podle zařazení stránek do kategorií katalogů, není ideální volbou, pokud ponecháme volbu zaměřovacího vektoru na pravděpodobnostním modelu. Z hlediska vyhodnocování stránek, u kterých byla provedena optimalizace zohledňující hypertextové odkazy tedy tento algoritmus více méně kopíruje chování standardního PageRanku a optimalizované stránky dosahují lepšího hodnocení. Algoritmus TSPR by tak nebylo vhodné použít jako

samostatnou hodnotící funkci, ale pouze jako část kompozitní hodnotící funkce.

Pokud bychom použili naznačené řešení – ponechat uživatele, ať si zvolí konkrétní téma či témata sám, mohl by naopak tohoto faktu uživatel využít k odfiltrování obchodních stránek tím že kategorii obchod z výsledků vyřadí předem. Pokud se budeme držet principu, že průměrný uživatel nechce složité ovládací rozhraní, nechce se učit složitou syntaxi a dokonce si ani nechce příliš komplikovat práci zpřesňováním svých dotazů pro vyhledávač, ale přes všechnu tuto neochotu vyžaduje pokud možno ideálně řazené výsledky, bude nutné pro tvorbu jednotlivých tématických klasifikátorů využít jinou metodu než katalogy WWW stránek. Kromě ne vždy jasných kategorií, mají katalogy další podstatnou nevýhodu, a tou je postupné ukončování jejich provozu. Zatímco v roce 2000 byl katalog běžnou formou vyhledávání na Internetu, o deset let později majorita uživatelů vyhledává pomocí fulltextů a nově přicházející generace uživatelů již ani nemusí tušit, že se v minulosti nějaké katalogy používaly. Nové stránky se v katalozích neobjevují, stačí si porovnat velikost katalogu ODP v době publikace TSPR a dnes. Jinou metodou použitelnou pro tvorbu tématických klasifikátorů by se tak mohly stát algoritmy používané pro shlukování dokumentů či pro automatickou detekci tématu. Prověření, zda je efektivita těchto algoritmů dostatečná pro aplikaci na rozsáhlé datové kolekce bude předmětem dalších výzkumů.

6.3 Srovnání obou lokalizovaných algoritmů

Při testování na stejných datech byly oba algoritmy hodnoceny v průměru lépe než základní PageRank. Algoritmus TSPRL, se i přes nedostatky zmíněné v závěru kapitoly 4.2, zdá být vhodnou funkcí pro uspořádání víceslovných dotazů. Algoritmus CZDIS naopak dosahoval nejlepšího hodnocení u dotazů jednoslovných. Kombinací obou algoritmů by tak mohla vzniknout hodnotící funkce, která bude navázána na obsah stránky i uživatelské preference (personalizovaná) a zároveň bude univerzálně použitelná jak pro dotazy jednoduché, tak dotazy složené.

Základní možností kombinace obou algoritmů je výpočet hodnotících vektorů nezávisle a volba řadící funkce v okamžiku vyhodnocování dotazu, podle počtu jeho prvků. V porovnání s nepersonalizovaným PageRankem půjde sice o řešení náročnější na výpočetní výkon i prostorové nároky vytvořených indexů, ale výsledky uživatelských testů prokazují, že oba lokalizované algoritmy řadí výsledky k větší uživatelské spokojenosti. Teoretické hodnocení této kombinace obou algoritmů je uvedeno v tabulce 8. Kombinovaný algoritmus by dosáhl průměrné známky 27,5.

Shrnutí přínosů k rozvoji vědního oboru

V práci jsou navrženy dva lokalizované algoritmy pro řazení výsledků fulltextového vyhledávání na webu. Oba algoritmy jsou založené na kombinaci odkazové analýzy a kontextu dotazu. U algoritmů je popsána metoda urychlení výpočtu a proces lokalizace pomocí aplikace českého jazykového modelu.

Shrnutí přínosů pro praxi

Výsledky experimentů s lokalizovanými algoritmy potvrzují, že oba algoritmy jsou pro řazení výsledků kvalitnější funkcí, než nepersonalizovaný PageRank.

Také urychlení obou algoritmů aplikací numerických metod řešení a jazykového modelu dává dobré předpoklady pro jejich reálné nasazení v praxi.

Další práce a experimenty

Dalším pokračováním v tématu práce by mělo být dopracování algoritmu kombinujícího TSPRL a CZDIS v jedno řešení, tak jak bylo naznačeno v závěru kapitoly 5.4. Kvalita tohoto kombinovaného algoritmu by pak měla být opět ověřena uživatelským testem. V tomto testu by zároveň mělo dojít k rozšíření metodiky volby dotazů. Náhodná volba testujícími uživateli, by měla být doplněna o dotazy získané analýzou dat z Google trends.

Budoucí výzkum v oblasti ZIW by měl být zaměřen na problematiku automatické detekce tématu stránek z jejich obsahu, shlukování dokumentů dle témat a využití těchto prostředků pro uspořádání výsledků fulltextového vyhledávání.

Další cílem je dopracování experimentálního systému v transparentně použitelné prostředí, vypracování příslušné dokumentace a využití tohoto hotového prostředí pro výzkumnou činnost studentů FM v rámci závěrečných prací a projektů.

Reference

Vlastní publikace

[1] Vraný J.: "CZDIS - distributed algorithm for personalized web pages ranking", INTED2009 Proceedings, Valencie Španělsko 2009, ISBN: 978-84-612-7578-6

[2] Vraný J.: "Parallel algorithm for query content based webpages ranking", BIS2009 Proceedings, Poznaň Polsko 2009, Springer 2009 ISBN: 978-3-642-01189-4

[3] Vraný J.: "Řazení výsledků WWW vyhledávače", Doktorandský den ústavu NTI, Liberec 2008

[4] Vraný J.: "Personalizace algoritmu PageRank", Doktorandský den ústavu NTI, Liberec 2007

[5] Vraný J.: "Experimentální prostředí pro WebIR", Seminární přednáška, Seminář ústavu NTI, Liberec 2008

[6] Vraný J.: "Úvod do teorie získávání informací na webu", Seminární přednáška, Seminář ústavu NTI, Liberec 2007

Použitá literatura

[7] Brin S., Page L.: The anatomy of a large-scale hypertextual web search engine. In Proceedings of the Seventh International World Wide Web Conference, 1998.

[8] Dolamic L., Savoy J.: "Stemming Approaches for East European Languages". In Advances in Multilingual and Multimodal information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007

[9] Haveliwala T.H.: "Topic-sensitive PageRank". In Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, May 2002.

[10] Langville A., Meyer C.: "Deep inside Pagerank" Internet Mathmathematics, Vol. 1, No. 3, 2005, pp.335-400.

[11] Page L., Brin S.,Motwani R., Winograd T.: "The Pagerank citation ranking: bringing order to the Web", 1999 Stanford University technical report.

[12] Reed, W. J.: "The Pareto, Zipf and other power laws". 2001 Economics Letters 74 (1): 1519. doi:10.1016/S0165-1765(01)00524-9

[13] Richardson M., Domingos P.: "The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank". 2002 Cambridge

[14] Savoy, J.: Czech stop words 2008: http://www.unine.ch/info/clef/

[15] Search engine statistics 2002 http://searchengineshowdown.com/statistics/size.shtml

Poznámka: tento zkrácený seznam obsahuje pouze publikace důležité vzhledem k obsahu autoreferátu. Kompletní seznam literatury je součástí disertační práce.