Abstrakt

Tato práce se zabývá způsoby vyhledávání informací. Nejprve čtenáře seznámí s modely pro vyhledávání dat a metodami pro hodnocení efektivnosti systémů pro vyhledávání informací. Poté přiblíží principy zpracování vstupních textů pro IR systémy s použitím seznamu stop slov a stemmeru. Dále ukazuje způsob rozšíření dotazů o synonyma pomocí thesauru. Nakonec přibližuje metody zohlednění výskytu frází v dotazech a představuje myšlenku ohodnocení dokumentu dle stupně podobnosti výskytu fráze.

Klíčová slova

Vyhledávání informací, vektorový model, relevance, měření efektivity vyhledávání, slovník synonym, analýza textu, stemming, vyhledávání frází

Abstract

This thesis deals with methods of information retrieval. Firstly, readers gets familiar with models for information retrieval and methods of retrieval evaluation. Then it put near the principles of input text processing for IR with use of stopword list and stemmer. Furthermore, it shows the way of query expansion with synonyms using the thesaurus. Finally, we show the methods of handling phrases appearance in queries and introduce the idea of ranking documents by degree of phrase occurrence similarity in document.

Keywords

Information retrieval, vector space model, relevance, retrieval evaluation, thesaurus, text analysis, stemming, phrase retrieval

Citace

Petr Hochmal: Vyhledávání informací v digitálních knihovnách, semestrální projekt, Brno, FIT VUT v Brně, 2010

Obsah

+ synonyma ...

přestrukturuj

1 Úvod

Už od dob, kdy lidstvo začalo psát knihy a shromažďovat je v knihovnách, měl člověk potřebu z těchto knih získávat informace. Avšak k tomu je nejprve potřeba zjistit, v kterých zdrojích potřebné informace hledat. Začaly vznikat katalogy, které umožňovaly vyhledávat podle jednoduchých kritérií (autor, název knihy, předmět zaměření, atp.) s přidanou informací o dokumentu (abstrakt, klíčová slova).

Ještě v dnešní době je možné se setkat v knihovnách s lístkovými katalogy. Myšlenku tohoto způsobu vyhledávání převzaly i mnohé vyhledávače, přesněji ty, zaměřené na hledání v elektronických katalozích. Pro údržbu báze dokumentů, nad kterými fungují tyto vyhledávače je však nutné vytvářet a spravovat i jejich metadata (informace popisující tyto dokumenty), což je náročná činnost.

Před několika lety zaznamenaly velký nárůst popularity fulltextové vyhledávače, které jsou schopny na uživatelský dotaz nalézt v obrovských bázích dat dokumenty obsahující text dotazu.

Trend ve vývoji vyhledávačů informací však míří směrem pochopení významu dotazu a vyhledání dokumentů, které obsahují i jinak formulovanou, ale požadovanou informaci.

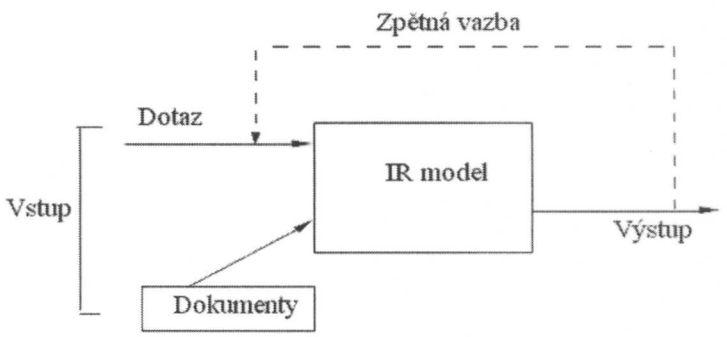
1.1 Struktura práce

Kapitola 2 seznamuje čtenáře s modely pro vyhledávání informací s důrazem na vektorový model, se kterým se pracuje dále. Ve třetí kapitole jsou uvedeny metody pro hodnocení výkonnosti systémů pro vyhledávání informací. Čtvrtá kapitola uvádí do problematiky zpracování textu, které je potřeba pro získání vlastností jak dokumentů, tak dotazů. Zaměřuje se na tokenizaci, tedy rozdělení textu na kandidáty na indexová slova, eliminaci častých slov a jejich úpravu na základní tvar. Dále představuje možnosti využití slovníku synonym pro možné zvýšení efektivity vyhledávání. V kapitole 5 se práce zabývá vyhledáváním frází, které čistý vektorový model neumožňuje a představuje ideu modifikace ohodnocení dokumentu dle podobnosti výskytu fráze.

2 Modely pro vyhledávání informací

Modely pro vyhledávání informací (dále jen IR modely) představují strukturu informací (z dokumentů) a procesy nad nimi, které vedou k získání znalosti o relevantnosti dokumentů, nad kterými vyhledáváme. Cílem vyhledávání je tedy vrátit uživateli na jeho dotaz seznam dokumentů, které odpovídají jeho informační potřebě. V tomto ohledu je dobré zmínit vyhledávání dat, které se zaměřuje na přesné uspokojení dotazu, to znamená nalezení dokumentů (zdrojů), které splňují exaktně podmínku obsahu dotazovaných dat.

V systému pro vyhledávání informací by se dal model pro vyhledávání informací zobrazit dle Obr. 2.1. Systém se skládá ze vstupu, jehož součástí je množina dokumentů, nad kterými se vyhledává, a dotazu, který specifikuje informační potřebu. Dále ze samotného IR modelu, výstupu, jež obsahuje množinu relevantních dokumentů, a volitelně zpětné vazby, která umožňuje uživateli pomocí výstupu specifikovat dotaz, čímž může dále zlepšovat výsledky vyhledávání.



Obr. 2.1: Systém pro vyhledávání informací [5]

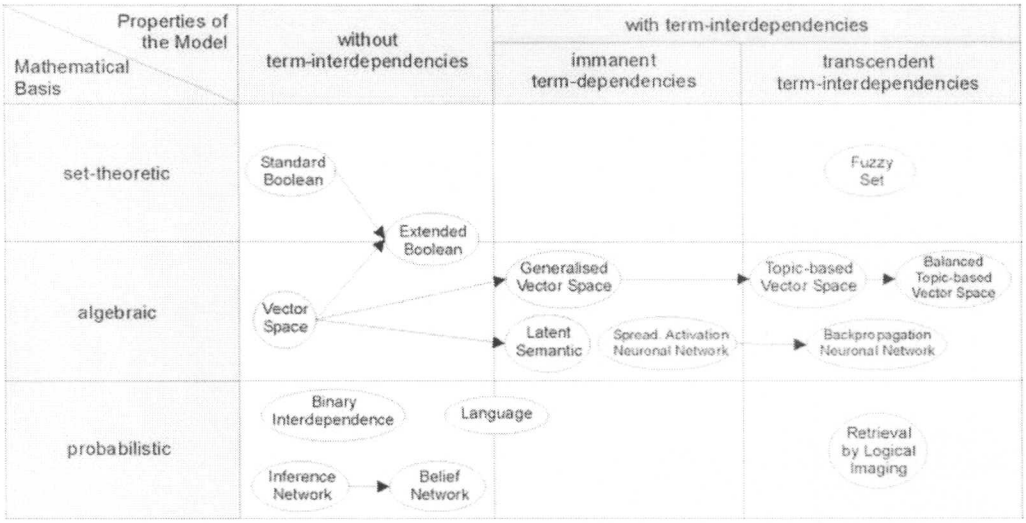
Modely pro vyhledávání informací se dle základního principu fungování rozdělují do tří skupin [1]:

- Množinové
- Algebraické
- Pravděpodobnostní

Základními zástupci těchto skupin jsou booleovský model, pracující s booleovskou algebrou nad množinami slov, vektorový model, kde je text reprezentován n-rozměrným vektorem a pravděpodobnostní model (známý jako model binárně nezávislého vyhledávání), pracující s teorií pravděpodobnosti. Další modely je možno vidět na Obr. 2.2. V následujících podkapitolách budou rozebrány tyto základní modely podrobněji.

Předem by bylo vhodné poznamenat, že výše zmíněné modely (booleovský a vektorový) jsou vhodné spíše jako teoretický základ pro další úpravy, s ohledem na jejich efektivnost vyhledávání (booleovský model) nebo výpočetní náročnost (vektorový model – práce s n-dimenzionálními vektory).

Jako zdroj pro popis modelů posloužil [1] – kapitola 2. V dalších kapitolách budeme uvažovat a pracovat s vektorovým modelem, nebude-li zmíněno jinak.



Obr. 2.2: Modely pro vyhledávání informací [8]

↑ popis obrázků v textu

2.1 Obecné principy

Před seznámením s konkrétními IR modely bude formálně charakterizován obecný IR model [1].

Model pro vyhledávání informací je čtveřice $\{\boldsymbol{D}, \boldsymbol{Q}, F, R(q_i, d_j)\}$, kde:

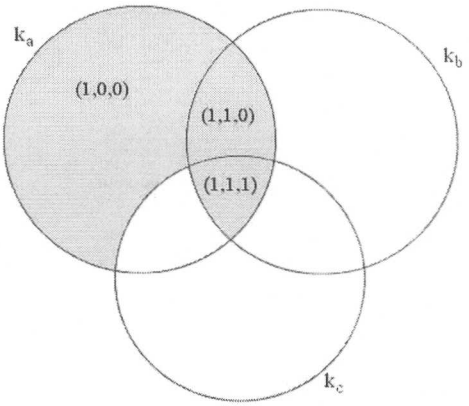
- ***D*** *je množina složená z logických pohledů reprezentující dokumenty.*
- ***Q*** *je množina složená z logických pohledů reprezentující uživatelský dotaz.*
- *F je systém pro modelování dokumentů, dotazů a jejich vztahů*
- *$R(q_i, d_j)$ je hodnotící (ranking) funkce, která dvojici $q_i \in D$ a $d_i \in Q$ přiřadí reálné číslo. Tedy ranking funkce definuje uspořádání dokumentů s ohledem na dotaz.*

V IR modelech je každý dokument popsán množinou reprezentativních klíčových slov, která se nazývají indexová slova (index terms). Tato slova by měla být schopna svojí sémantikou popsat samotný dokument. Obecně vzato, ne všechny slova mají stejnou vypovídací hodnotu o dokumentu. Proto je dobré indexovým slovům přiřadit různou váhu pro reprezentaci dokumentu.

Nechť k_i je indexové slovo, d_j je dokument, pak $w_{i,j} \geq 0$ je váha přiřazená páru (k_j, d_j). Máme tedy množinu $K=\{k_1,...,k_n\}$ obsahující všechna klíčová slova. Váha $w_{i,j}$ je pak přiřazena každému dokumentu pro každé indexové slovo. Pro ta indexová slova, která se nevyskytují v dokumentu, je $w_{i,j} = 0$.

2.2 Booleovský model

Tento model je jeden z nejstarších používaných pro vyhledávání informací. Jeho základem je teorie množin a booleovská algebra. Dotazy jsou dány booleovskými výrazy, které jsou aplikovány na každý dokument. Jako výsledek na dotaz vrací pouze ty dokumenty, které přesně odpovídají položenému dotazu. To je způsobeno samotným principem booleovské algebry. Neumožňuje tedy nalézt částečnou shodu. Jedná se tak spíše o model pro vyhledávání dat, kde je tato vlastnost žádoucí.



Obr. 2.3: Tři konjunktivní komponenty pro dotaz $[\, \boldsymbol{q} = \boldsymbol{k_a} \wedge (\boldsymbol{k_b} \vee \neg\, \boldsymbol{k_c}) \,]$

Váhy indexových slov jsou binární, tedy $w \in \{0,1\}$. Dotaz je tvořen indexovými slovy a operátory *and, or* a *not*. Je ho tedy možné transformovat na disjunkci konjunktivních vektorů. Např. dotaz $[\, \boldsymbol{q} = \boldsymbol{k_a} \wedge (\boldsymbol{k_b} \vee \neg\, \boldsymbol{k_c}) \,]$ může být zapsán jako $\left[\, \vec{q}_{dnf} = (1,1,1) \vee (1,1,0) \vee (1,0,0) \,\right]$ viz Obr. 2.3 [1].

Hlavní výhodou booleovského modelu je jeho jednoduchost, nevýhodou je však jeho neschopnost pracovat s jemnějšími vahami indexových slov.

2.3 Vektorový model

U předešlého, booleovského, modelu jsme si uvedli, že binární váhy jsou příliš limitujícím faktorem. Tuto nectnost řeší vektorový model díky svému jemnému výpočtu vah, respektive způsobu určování podobnosti dokumentů. Jak bylo výše zmíněno, každé indexové slovo má přiřazenu váhu. Ve vektorovém modelu je každý dokument, nebo dotaz reprezentován t-rozměrným vektorem, jehož dimenze jsou složeny z vah všech indexových slov ($t = |K|$). Tyto vektory jsou využity k výpočtu *stupně podobnosti* dokumentu a dotazu (nebo jiného dokumentu). Díky tomuto je možné získat i částečnou shodu ve vyhledávání a podle stupně podobnosti řadit výsledky vyhledávání s ohledem na relevanci.

Shrnutím dosavadních poznatků získáme:

- Váhu $w_{i,j}$ přiřazenou dvojici (k_i, d_j), $w_{i,j} \in \mathbb{R}^{0+}$
- Váhu $w_{i,q}$ přiřazenou dvojici (k_i, q), $w_{i,q} \in \mathbb{R}^{0+}$
- Vektor $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q}), t = |K|$
- Vektor $\overrightarrow{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j}), t = |K|$

Vektory $\vec{q}$ a $\overrightarrow{d_j}$ budou použity pro výpočet stupně podobnosti mezi dotazem q a dokumentem d_j. K tomuto lze použít výpočet kosinu úhlu mezi těmito dvěma vektory [7],

$$sim(d_j, q) = \frac{\overrightarrow{d_j} \cdot \vec{q}}{|\overrightarrow{d_j}| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \qquad (2.1)$$

Místo rozhodování o relevantnosti dokumentu, vektorový model určuje stupeň podobnosti dokumentu a dotazu. Můžeme tedy určit mez, nad kterou budou dokumenty hodnoceny

dle podobnosti jako relevantní. Nejprve je nutné definovat výpočet vah pro indexová slova, který v podstatě určuje míru chování modelu.

V nejjednodušším případě může být váha $w_{i,j}$ rovna frekvenci slova i v dokumentu j. Ovšem přesnější je použití normované frekvence,

$$ntf_{i,j} = \frac{freq_{i,j}}{max_l\, freq_{l,j}} \tag{2.2}$$

kde $freq_{i,j}$ je frekvence i-tého indexového slova v j-tém dokumentu a max určuje nejfrekventovanější slovo, tím získáme frekvenci slova v intervalu $<0,1>$ nezávisle na jeho počtu výskytů. V praxi se spíše používá následující modifikace výpočtu frekvence slova [2],

$$ntf_{i,j} = a + (1-a)\frac{freq_{i,j}}{max_l\, freq_{l,j}} \tag{2.3}$$

Která vyzdvihuje slova obsažená v dokumentu od těch nevyskytujících se. Pak tedy platí:

- $w_{i,j} = 0, pro\ k_i \notin d_j$
- $w_{i,j} \in < a, 1 >, pro\ k_i \in d_j$

Váhu indexového slova dále ovlivňuje výskyt tohoto slova napříč dokumenty, tedy slovo, jež je obsaženo pouze v jednom nebo několika málo dokumentech je cenné. Naopak slovo vyskytující se ve všech dokumentech nemá o jedno konkrétním dokumentu žádnou vypovídací váhu. K tomuto nám slouží takzvaná inverzní dokumentová frekvence slova. Její výpočet je definován rovnicí

$$idf_i = \log\frac{N}{n_i} \tag{2.4}$$

kde N je počet všech dokumentů a n_i je počet dokumentů obsahujících indexové slovo k_i.

Dostáváme se tedy k celkovému výpočtu vah, a to:

$$w_{i,j} = a + (1-a)\frac{freq_{i,j}}{max_l\, freq_{l,j}} \times \log\frac{N}{n_i} \tag{2.5}$$

Jak již ze složení rovnice vyplývá, tato strategie výpočtu vah se nazývá tf-idf schéma.

Pro výpočet vah se využívá stejného principu. Dle [1], je pro dobré výsledky doporučeno použití koeficientu $a=0.5$:

$$w_{i,j} = 0.5 + \frac{0.5\, freq_{i,j}}{max_l\, freq_{l,j}} \times \log\frac{N}{n_i} \tag{2.6}$$

Existují další způsoby výpočtu vah, kterými se zde ale nebudeme zabývat. Více informací v [2], kapitola 6.4.3.

Hlavní výhodou vektorového modelu je jeho schéma výpočtu vah indexových slov, které zvyšuje schopnosti vyhledávání, dále možnost částečné shody ve vyhledávání s dotazem a určování podobnosti dokumentů, umožňující ve výsledku řadit výsledky hledání dle získaného skóre.

Nevýhodou je, že indexová slova v dokumentech jsou uvažována jako vzájemně nezávislá, nelze tedy zohlednit sousednost slov.

2.4 Pravděpodobnostní model

Tento model se snaží zachytit problém vyhledávání informací pomocí teorie pravděpodobnosti. Myšlenka je následující. Na daný dotaz existuje množina dokumentů, obsahující právě ty dokumenty, které jsou relevantní a žádné jiné. Stanovme si tuto množinu dokumentů jako ideální odpověď. Pokud bychom měli přesný popis této odpovědi, nebyl by problém získat relevantní dokumenty. Můžeme tedy uvažovat proces dotazování jako proces hledání popisu ideální odpovědi. Úkolem je tedy vytvořit počáteční odhad množiny relevantních dokumentů R a uživatelskou interakcí zlepšovat popis ideální odpovědní množiny.

Nechť je $P(R|\vec{d_j})$ pravděpodobnost, že je dokument d_j relevantní k dotazu a $P(\overline{R}|\vec{d_j})$ pravděpodobnost, že dokument d_j není relevantní k dotazu. Pak je podobnostní funkce vyjádřena následovně:

$$sim(d_j, q) = \frac{P(R|\vec{d_j})}{P(\overline{R}|\vec{d_j})} \tag{2.7}$$

S použitím Bayesova pravidla:

$$sim(d_j, q) = \frac{P(\vec{d_j}|R) \times P(R)}{P(\vec{d_j}|\overline{R}) \times P(\overline{R})} \tag{2.8}$$

Hlouběji do teorie pravděpodobnostního modelu zasahovat nebudeme. Více podrobností lze nalézt v publikacích [1], kapitola 2.5.4 a zejména v [2], kapitola 11.

3 Měření efektivity

Jedním z hlavních pojmů ve vyhledávání informací je *relevance* (důležitost, významnost). Relevance určuje, zda nalezené dokumenty obsahují pro uživatele informace, jaké svým dotazem žádá. Je to tedy poměrně subjektivní měřítko kvality závislé na kontextu informací. Systémy pro IR jsou hodnoceny podle relevance dokumentů k dotazu.

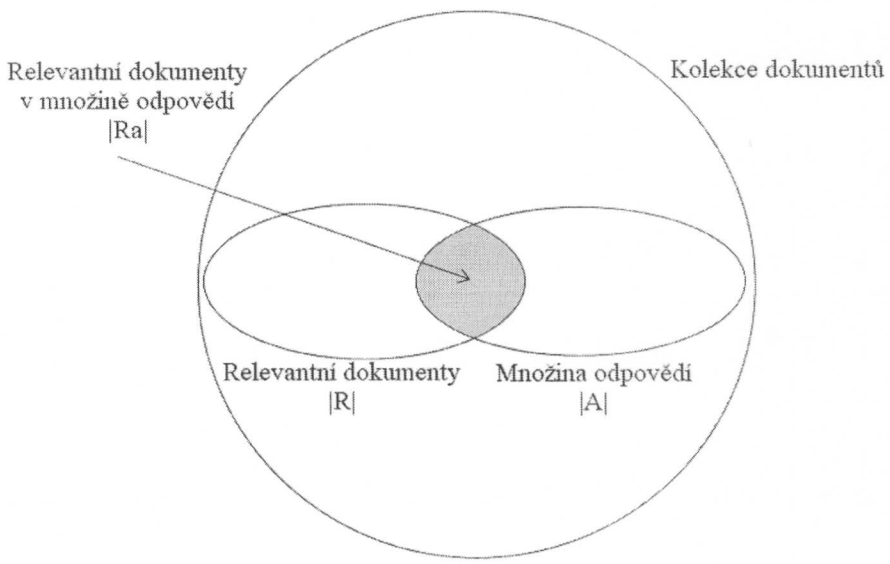
3.1 Testovací kolekce dokumentů

K tomu, abychom mohli vzájemně porovnávat efektivnost různých IR modelů, musíme jejich efektivnost vyhledávání testovat pomocí stejných dat. K tomuto slouží testovací kolekce zaměřené na různé obory IR a obsahující různorodá data. Tyto kolekce obsahují jak samotné dokumenty, tak dotazy pro vyhledávání a samozřejmě seznam relevantních dokumentů pro každý jeden dotaz. Tato specifikace, které dokumenty jsou relevantní je provedena odborníky na dané téma a považuje se za správné. Takto tedy můžeme na dané kolekci dokumentů provádět pomocí definovaných dotazů vyhledávání a poté v porovnání s referenčními odpověďmi určit, míru průniku množinu odpovědí s množinou referenčních odpovědí.

Tvorbou těchto testovacích kolekcí se zabývá například TREC (Text REtrieval Conference) a jejich data jsou k nalezení na < http://trec.nist.gov/data.html >.

3.2 Úplnost a přesnost

Důležitými pojmy pro měření efektivity jsou úplnost (angl. recall) a přesnost (angl. precision). Uvažujme například dotaz specifikující touhu po informacích *I* a pro tyto informace množinu relevantních dokumentů R. Pomocí IR systému (který je hodnocen) získáme množinu odpovědí A. Pak je množina Ra průnikem množin R a A (jak je vidět na Obr. 3.1).



Obr. 3.1: Průnik množin relevantních dokumentů a množiny odpovědí [1]

Úplnost vyjadřuje, jak velká část relevantních dokumentů byla nalezena [1].

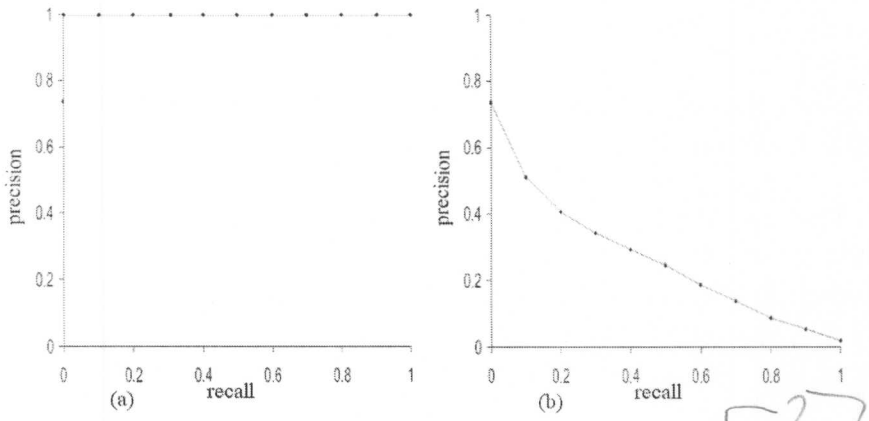
$$Recall = \frac{|Ra|}{|R|} \tag{3.1}$$

Přesnost vyjadřuje, jak velká část nalezených dokumentů je relevantní [1].

$$Precision = \frac{|Ra|}{|A|} \tag{3.2}$$

Jak vysoká úplnost, tak i přesnost je důležitá pro IR systém. V ideálním případě by obě charakteristiky měli nabývat hodnoty 1, tj. tedy výsledek je složen ze všech a pouze relevantních dokumentů (viz. Obr. 3.2).

Takto můžeme ohodnotit množinu neohodnocených odpovědí (výsledek vyhledávání pomocí booleovského modelu). U množiny ohodnocených odpovědí nás však zajímá i přesnost hodnocení (ranking funkce), to znamená provést hodnocení s ohledem na uspořádání odpovědí. Toho dosáhneme výpočtem přesnosti pro různou úroveň úplnosti (např. postupně 10%, 20%,…,100%). Pak tedy uvažujeme v rovnici 3.2 prvních *x*% výsledků z množiny A (s vlivem na Ra).



Obr. 3.2: (a) ideální a (b) obvyklé ohodnocení IR systému

Při hodnocení pro celou kolekci dotazů je možné spočítat průměrnou přesnost dle vzorce [1]:

$$AP(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q} \tag{3.3}$$

Kde N_q je počet dotazů a $P_i(r)$ je přesnost pro dotaz i na úrovni úplnosti r.

4 Zpracování textových dat

IR modely jsou vesměs univerzální, tedy nezávislé na datech, ve kterých mají umožňovat vyhledávání. Z toho důvodu je třeba provést vhodné zpracování textu, díky kterému získáme z dokumentu vhodná indexová slova. Toto zpracování je obvykle silně zaměřeno na konkrétní jazyk použitý v textu, protože charakteristiky slov i ostatních řetězců se mezi jazyky (skupinami jazyků) významně liší. Kvalita, se kterou jsou získána indexová slova, přímo ovlivňuje kvalitu chování celého IR systému.

4.1 Analýza textu

K získání indexových slov je třeba text dokumentu vhodně analyzovat a podle toho rozdělit na jednotlivé tokeny. Budeme uvažovat čistý text bez formátovacích značek nebo podobných struktur a zároveň se nebudeme zabývat získáním tohoto textu z konkrétních zdrojů.

Tomuto tématu se také věnuje [2], kapitola 2.2.

4.1.1 Tokenizace

První krok, který je třeba vykonat, je rozdělení textu na úseky znaků, které vnímáme jako slova, čísla, nebo jiné znaky (řetězce znaků), které mají daný význam a odstranění nadbytečných znaků jako jsou interpunkční znaménka. Tyto úseky tvoří kandidáty na další zpracování a v konečném důsledku tvoří indexová slova (termy).

Nejsnazším způsobem, ale ne ideálním, by bylo nahradit všechny nepísmenné znaky bílými a pak podle bílých znaků (mezera, odřádkování, tabulátor, atp.) rozdělit text na tokeny. Tím bychom sice dospěli k získání vhodných tokenů, ale zároveň bychom tím ztratili mnohé jiné informace, které mohou uživatele zajímat. Dále jsou velkou komplikací čísla a jejich různorodý způsob vyjádření a použití. Zvažme, jak má být zpracováno datum, které může být zapsáno například jako „1.1.2010“. Má se rozdělit na 3 tokeny „1“, „1“ a „2010“ podle tečky, má být zachováno v původním tvaru nebo tečku vyjmout a vytvořit celistvý „112010“. Z pohledu vyhledávání by asi bylo ideální datum rozpoznat v několika tvarech a převést na jednotný tvar. Ale toto je jen jeden případ z mnoha, u kterých už nemusí být jasná odpověď.

Univerzální řešení, které by neznevýhodňovalo určité typy řetězců, asi neexistuje. Vždy by se našly takové případy (např. zdrojové kódy), že by je proces tokenizace nezpracoval korektně. Je třeba se rozhodnout mezi řešením takovým, které se bude snažit poskytnout dobrý podklad pro vyhledávání (tj. co nejobecnější tvar řetězce) a tím, jenž uchová původní tvar řetězce.

Během zpracování textu se tokenizér potýká i s vlastnostmi jako je velikost písmen a v případě českého jazyka i diakritika.

Velikost písmen

Zdálo by se, převod všech písmen na malé, je účinné řešení. Ale pokud uvážíme dvojici slov *FIT* a *fit*, zájemce o studium informatiky asi nebude rád, když mu vyhledávací systém vrátí dokumenty o cvičení a zdravé výživě. Je tedy na rozvaze jak se má systém chovat, zda obecně

a konvertovat všechna velká písmena na malá, nebo zda má zachovávat sémantiku skrytou ve velikosti písmen a měnit velikost například jen písmen slov na začátku věty.

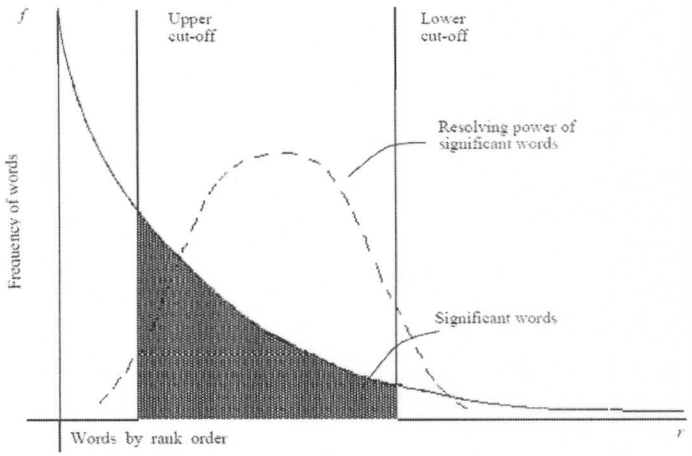
Diakritika

Jedná se o podobný případ jako u velikosti písmen. Pro „univerzální" řešení můžeme odstranit všechnu diakritiku, což je, vezmeme-li v úvahu, kolik uživatelů píše bez použití diakritiky, řešení, které má dobrý vliv na množství získaných výsledků. Potom se ale sémanticky rozdílná slova jako jsou například *koš* a *kos* stanou nerozlišitelná. Opět je tedy na rozhodnutí, zda má systém poskytovat více odpovědí, ale s rizikem, že nebudou zcela relevantní, nebo má uživatel zadávat svůj dotaz korektně k jeho sémantice.

4.1.2 Eliminace častých slov

Častá slova, jak jsme si již zmínili v kapitole 2.3 při výpočtu *idf* mají velice nízkou, spíše až žádnou vypovídací hodnotu o textu v němž se vyskytují. V tomto ohledu by nám tato slova ani moc nevadila, ale například při výpočtu normované frekvence slov (rovnice 2.3) může v jednom dokumentu enormně se vyskytující slovo zbytečně znevýhodnit tento dokument od ostatních. Další nevýhodou častých slov je prostor, který zabírají v indexu (obzvláště pak v pozičním indexu slov v dokumentech).

Řešením jak se vypořádat s negativy spojenými s častými slovy, je tato slova z kandidátů na indexová slova odstranit. K tomu potřebujeme seznam těchto častých slov (obvykle se nazývají stop slova – stopwords), který buďto můžeme získat (např. zdroje na [10]), nebo sami vytvořit z dostatečně velké báze dokumentů. Slova s nízkou *idf* (dle [4] < 0.2) jsou vhodnými kandidáty na seznam stop slov. Dnešní trend však vede k používání malého seznamu stop slov (desítky místo stovek), který obsahuje převážně spojky atp. [2].



Obr. 4.1: Charakteristika význačných slov [5]

4.1.3 Lemmatizace a Stemming

Slova v dokumentech mají z gramatických důvodů různé tvary, kdy slova s různými koncovkami, mají víceméně stejný význam (například student, studentka, studenti). Abychom mohli nad těmito tvarově i významově podobnými slovy efektivně vyhledávat, je potřeba je převést na společný základní tvar. Tímto docílíme v konečném důsledku i menšího slovníku IR systému, takže jeho prostorové nároky budou taktéž menší.

K získání základního tvaru slova je možné využít lemmatizace, kdy se využívá slovníkové a morfologické analýzy slov. Tento proces dosahuje dobrých výsledků, avšak pouze v případě, že daný tvar je obsažen ve slovníku.

Další možností jak se dopracovat k základnímu tvaru slova je proces zvaný stemming. Ten pomocí primitivní heuristiky analyzuje slovo a odřízne jeho příponu, případně ještě doplní do pravděpodobně korektního tvaru. Výstup stemmingu nemusí plně odpovídat gramaticky správnému základnímu tvaru slova (kořene). Spokojíme se s tím, že pravděpodobně významově stejná a tvarově podobná slova převede do jednotné formy. Nejznámější je Porterův stemmer (ukázka viz Obr. 4.2) navržený pro angličtinu, jehož princip se převzal i pro jiné jazyky. Stemmery pro různé jazyky jsou k dispozici na [10].

Stejně jako v předchozích krocích zpracování textu, i použití převodu na základní tvar slova má různý dopad na efektivitu vyhledávání. Pro některé dotazy může být výhodná generalizace slov, pro jiné může způsobit snížení přesnosti (viz kapitola 3.2).

Rule			Example		
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
SS	→	SS	caress	→	caress
S	→		cats	→	cat

Obr. 4.2: Ukázka principu fungování Porterova stemmeru [2]

4.2 Využití sémantických slovníků

Dalším způsobem, jak se můžeme pokusit zvýšit efektivitu vyhledávání je použití sémantických slovníků, konkrétně za účelem získání synonym slova. Jejich použití ve fázi zpracování dokumentů (indexace) nepřichází v úvahu. Informace o dokumentech můžeme zobecňovat, nikoliv však rozšiřovat. Správným krokem však může být rozšíření dotazu o synonyma jednotlivých slov.

Slovník synonym (neboli thesaurus) obsahují dvojice *slovo – jeho synonymum*. To nám dává možnost získat pro každé slovo seznam jeho synonym. Můžeme pak rozšířit dotaz o tato synonyma. Z pohledu zachování tvaru dotazu, by bylo vhodné místo pouhého přidání synonym, nahradit původní slovo množinou synonym tohoto slova (včetně něj). Například:

{term_1, term_2} => {{term_1, synonymum_termu_1,...},{term_2, synonymum_termu_2,... }

I přesto, že vektorový model vyhledávání nebere ohled na pořadí nebo sousednost slov v dokumentu / dotazu, je vhodné tuto strukturu dotazu ponechat pro další využití a až v konečné fázi zpracování převést dotaz na vektor.

Rozšíření dotazu o synonyma může vést k lepším výsledkům vyhledávání, ale zároveň může uživateli poskytovat výsledky na pro něj nevyhovující výrazy. Proto by mělo být použití rozšíření dotazu o synonyma volbou, kterou může zakázat.

České slovníky synonym vhodné k použití (např. ve formátu hunspell) jsou k nalezení například na [9], stejně jako slovníky pro mnohé jiné jazyky.

5 Vyhledávání se zohledněním frází

Uživatelé obvykle pro vyhledávání používají slovní spojení (fráze), které očekávají, že bude nalezený dokument obsahovat. Uvažujeme-li použití vektorového modelu pro vyhledávání, víme, že se při zjišťování podobnosti dokumentu a dotazu nezohledňuje sousednost slov. V této kapitole si představíme metody, které by mohly vykazovat dobré výsledky při vyhledávání pomocí dotazu obsahujícím fráze. Zajímavý způsob zohlednění vztahů slov představuje dokument [6], kde je závislost mezi indexovými slovy zohledněna pomocí asociačních pravidel.

5.1 N-gramy slov

Jedním z poměrně jednoduchých způsobů, jak zohlednit sousednost slov a tím i napomoci vyhledávání frází je použití n-gramů slov. Jiný výraz pro n-gram je v angličtině n-words (například biwords, triwords, atp.). Princip je takový, že v průběhu indexace dokumentů nebudeme uvažovat, že *1 token = 1 indexové slovo*, ale indexové slovo bude složeno z n-tice po sobě jdoucích tokenů. Dle [2] již použití tri-gramu vykazuje dobré výsledky při vyhledávání frází.

Jako příklad si uveďme dotaz *„fit vut brno"*. Při použití bigramů rozdělíme dotaz na dvojice:

„fit vut" a
„vut brno"

a tyto dvojice následně použijeme jako indexová slova pro vyhledávání. Stále je sice šance získání falešných pozitiv (s ohledem na celou frázi), ale jak bylo poznamenáno výše, při použití trigramů to není častý jev.

Problémem však je, že použití n-gramů velmi zvyšuje velikost slovníku indexových slov. Navíc při indexovém slovu složeném z např. trigramu není jedno slovo přirozeně zpracováno (bylo by nutné ho vyhledávat v každém trigramu) a tak je stále potřeba udržovat v indexu jednotlivá slova. Dále abychom mohli zohlednit dvojslovné fráze, je třeba udržovat také v slovníku indexových slov bigramy.

Při zpracování dotazu můžeme uvažovat 2 způsoby frázování – implicitní a explicitní. Při použití *explicitních* frází uživatel vymezí fráze například uvozovkami. Poté slova mimo uvozovky uvažujeme jako jednoduchá indexová slova a uvnitř uvozovek je rozdělujeme na co největší používané n-gramy). V případě implicitního frázování rozdělíme dotaz na 1 až n-tice slov a ty poté najednou použijeme pro vyhledávání. V tomto případě by bylo vhodné uvažovat vyšší váhu indexového slova se zvyšujícím se stupněm n-gramu, aby dokumenty obsahující fráze získaly vyšší váhu v množině odpovědí.

5.2 Poziční indexy

Vektorový model vyhledávání potřebuje pouze s informací, ve kterých dokumentech se indexové slovo vyskytuje. Abychom ale mohli zohlednit sousednost, nebo lépe vzdálenost slov v dokumentu, je třeba si uchovávat pro každou dvojici *{indexové slovo, dokument}* pozice výskytů indexového slova. Nazvěme tyto seznamy pozičními indexy.

Myšlenka použití pozičních indexů je následující. Uvážíme-li, že uživatel svůj dotaz píše takovým stylem, jak očekává, že se bude vyskytovat v textu, měli bychom stupeň podobnosti

dokumentu s dotazem ovlivnit také podobností vztahu slov dotazu se slovy v dokumentech. Tedy brát v úvahu vzdálenost slov v dotazu. Uvažovat pouze sousednost slov nemusí být uspokojující. Mezi v dotazu dvěma sousedními slovy může být v textu například zájmeno, které význam fráze příliš neovlivňuje, ale znemožní zohlednění sousednosti slov, která ho obklopují.

Mějme dotaz „term1 term2 term3". Pak uvažujme všechny dvojice slov *{a,b}*, tedy konkrétně *{term1, term2}, {term1, term3}, {term2, term3}*. Poté vypočítáme blízkost slov v dotazu $P_q(a,b)$, kde slova a,b tvoří dvojici:

$$P_q(a,b) = \min \left| pos_q(a) - pos_q(b) \right| \tag{5.1}$$

Dále určíme blízkost slov v dokumentu j $(a, b \in Q)$:

$$P_j(a,b) = \min \left| pos_j(a) - pos_j(b) \right| \tag{5.2}$$

Vypočteme koeficient odchylky blízkosti dvojice slov v dotazu vůči blízkosti slov v dokumentu:

$$DQPC_j(a,b) = \begin{cases} \dfrac{k - \left| P_j(a,b) - P_q(a,b) \right|}{k}, & pokud\ P_j(a,b) - P_q(a,b) \in \langle 0,k \rangle \wedge a,b \in d_i \\ 0, & v\ ostatních\ případech \end{cases} \tag{5.3}$$

kde k udává maximální vzdálenost dvojice slov.

Průměrný koeficient blízkosti dvojic slov v celém dotazu vůči dokumentu j vypočteme jako:

$$AverDQTP_j = \frac{\sum_{i=1}^{|N|} DQTP_j(N_{iA}, N_{iB})}{|N|} \tag{5.4}$$

kde N je množina dvojic (N_A, N_B) z dotazu.

Pro tento průměrný koeficient platí $AverDQTP_j \in \langle 0,1 \rangle$, kdy „1" znamená přesný výskyt fráze a „0" znamená, že dokument neobsahuje frázi. Hodnoty mezi „0" a „1" značí různou míru podobnosti fráze uvnitř dokumentu.

Pro zjednodušení je možné tento algoritmus pro zjištění podobnosti výskytu fráze aplikovat až po vyhledání dokumentu podle čistého dotazu na dokumenty s ohodnocením vyšším než je stanovená mez. Toto ohodnocení poté modifikovat podle $AverDQTP_j$. Např.:

$$simP(d_j, q) = 0.7sim(d_j, q) + 0.3AverDQTP_j$$

Tento způsob zohlednění výskytu frází je jen myšlenkou a jeho funkčnost na reálných datech zatím nebyla v praxi ověřena. Výpočet $AverDQTP_j$ bude pravděpodobně náročný, ale při vhodném předzpracování dat by mohl přinášet dobré výsledky.

6 Závěr

Cílem této práce bylo seznámit se s problematikou vyhledávání informací, zejména pak modely k tomu určenými a s nimi spojenými metodami pro hodnocení systémů pro vyhledávání informací. Dále jsem rozebral principy zpracování textu pro IR systémy s ohledem na efektivitu vyhledávání informací. S tím související je i použití slovníku synonym pro rozšíření dotazů. V poslední kapitole jsem se věnoval způsobům jak zohlednit výskyt frází při vyhledávání, kde jsem přednesl myšlenku ohodnocení dokumentů dle podobnosti výskytu fráze dané dotazem.

V navazující diplomové práci využiji získané informace, především o vektorovém modelu a způsoby zpracování textu. Pokusím se dále rozpracovat ideu metody podobnosti výskytu fráze. Dalšími kroky bude navržení a implementace systému pro vyhledávání s použitím vektorového modelu, slovníku synonym a českého stemmeru a navržení a implementace aplikace elektronické knihovny (konkrétně indexaci a vyhledávání školních diplomových prací).

zatím OK (SP)

Literatura

[1] BAEZA-YATES, Ricardo - RIBEIRO-NETO, Berthier. *Modern information retrieval*. New York: ACM Press, 1999. 513 s. ISBN 0-201-39829-X

[2] MANNING, Christopher D., RAGHAVAN, Prabhakar, SCHÜTZE, Hinrich. *An Introduction to Information Retrieval*. New York: Cambridge University Press, 2008. 496 s. 1. edice. ISBN 978-0-521-86571-5.

[3] PÁNEK, Karel. Architektury a modely webových strojů. *LUPA : server o českém internetu* [online]. 2002 [cit. 2009-12-28]. Dostupný z WWW: <http://www.lupa.cz/clanky/architektury-a-modely-webovych-stroju/>.

[4] PÁNEK, Karel. Šrotujeme text. *LUPA : server o českém internetu* [online]. 2002 [cit. 2009-12-30]. Dostupný z WWW: <http://www.lupa.cz/clanky/srotujeme-text/>.

[5] RIJSBERGEN, C.J. van. *INFORMATION RETRIEVAL* [online]. 1999 [cit. 2009-12-28]. Dostupný z WWW: <http://www.dcs.gla.ac.uk/~iain/keith/index.htm>.

[6] SILVA, Ilmério R., SOUZA, João Nunes, SANTOS, Karina S. . Dependence Among Terms in Vector Space Model. In *Database Engineering and Applications Symposium, International*. Los Alamitos, CA, USA : IEEE Computer Society, 2004. s. 97-102. Dostupný z WWW: <http://doi.ieeecomputersociety.org/10.1109/IDEAS.2004.1319782>. ISSN 1098-8068.

[7] WIKIPEDIA, Contributors. *Dot product* [online]. Wikipedia, The Free Encyclopedia, 2001-2009 [cit. 2009-12-26]. Dostupný z WWW: <http://en.wikipedia.org/w/index.php?title=Dot_product&oldid=332835230>.

[8] WIKIPEDIA, Contributors. *Information retreival* [online]. Wikipedia, The Free Encyclopedia, 2001-2009 [cit. 2009-12-29]. Dostupný z WWW: <http://en.wikipedia.org/w/index.php?title=Information_retrieval&oldid=334215389>.

[9] Dictionaries – OpenOffice.org Wiki, http://wiki.services.openoffice.org/w/index.php?title=Dictionaries&oldid=153923#Czech_.28Czech_Republic.29

[10] IR Multilingual Resources at UniNE, http://members.unine.ch/jacques.savoy/clef/index.html.

[11] Text REtrieval Conference (TREC) Data, http://trec.nist.gov/data.html

Seznam příloh

Bez příloh.