Metody dotazování pro textové databáze

Jan Martinovič *
jan.martinovic@vsb.cz

Abstrakt V dnešním světě existují velké kolekce dokumentů. Proto vznikají různé systémy využívající booleovských, vektorových a dalších modelů k reprezentaci dokumentů. Každý z těchto modelů pro svá omezení nedovoluje uživateli nalézt všechny očekávané dokumenty. Tato diplomová práce popisuje některé přístupy k vylepšení vektorového dotazu. Věnuje se rozšíření vektorového dotazu založeného na aglomerativním shlukování a následně jsou navrženy metody pro oddálení nerelevantních dokumentů od dotazu za pomoci dotazování na vývoj tématu. Na základě provedených testů jsou porovnány různé metody aglomerativního shlukování a vysloven závěr, zda lze pomocí oddálení nerelevantních dokumentů od dotazu zlepšit výsledky vektorových dotazů.

Klíčová slova: shlukování dokumentů, vývoj tématu, přesnost, úplnost, dotaz, vektorový

1 Úvod

V dnešním světě existují velké kolekce textových dokumentů. S rozvojem internetu nabývají tyto kolekce stále větších rozměrů. Proto byly vytvářeny systémy, které usnadňují práci s těmito kolekcemi (*dokumentografické informační systémy*).

Dokumentografický informační systém (dále jen DIS) lze volně popsat jako systém s databází textů či dokumentů. Obor Information Retrieval, se zabývá možnostmi uložení, vyhledávání a údržbou informací. Informací se zde myslí texty, obrázky, zvukové nahrávky, video a ostatní multimediální data [5]. V naší práci se budeme zabývat nestrukturovanými textovými dokumenty.

Pro vyhledávání v kolekcích dokumentů existují různé systémy využívajících booleovských, vektorových, pravděpodobnostních a dalších modelů k reprezentaci dokumentů, dotazů, pravidel a procedur umožňujících určit shodu mezi požadavkem uživatele (*dotaz*) a dokumenty. Každý z těchto modelů obsahuje řadu omezení. Tato omezení neumožňují uživateli nalézt všechny dokumenty, které očekává. Mezi očekávanými dokumenty (*relevantní dokumenty*) nalezneme i dokumenty špatné (*nerelevantní*) a některé relevantní dokumenty nejsou v seznamu vůbec obsaženy.

V této práci popíšeme využití rozšíření vektorového dotazu pomocí shlukovacích metod a metod hledání vývoje tématu. Cílem metod hledání vývoje tématu je k dokumentu, který zadá uživatel, vyhledat seznam dokumentů tématicky souvisejícími se zadaným dokumentem. Popíšeme jak rozšířit vektorový dotaz o nové termy. Dále uvedeme metodu využívající vývoj tématu k oddálení nerelevantních dokumentů od dotazu a přiblížení dokumentů relevantních.

Základní výsledky této práce jsme prezentovali na konferenci Znalosti 2004 [1] a workshopu DATESO 2004 [2].

* Katedra informatiky, FEI, VŠB - Technická Univerzita Ostrava, 17. listopadu 15, 708 33, Ostrava-Poruba

2 Hodnocení efektivity

Základními ukazateli vyhledávacích systémů jsou [6]:

- rychlost zpracování požadavků,
- uživatelský komfort, projevující se zejména ve formě interakce se systémem,
- způsob kladení dotazů a poskytování odpovědí,
- schopnost poskytnout informace o relevantních dokumentech.

Míra schopnosti poskytnout relevantní dokumenty se vyjadřuje pomocí těchto ukazatelů: *koeficient přesnosti* – P a *koeficient úplnosti* – R:

$$P = \frac{|A \cap B|}{|B|}$$

$$R = \frac{|A \cap B|}{|A|}$$

Koeficient úplnosti lze chápat jako pravděpodobnost, že relevantní dokument byl vybrán, koeficient přesnosti jako pravděpodobnost, že vybraný dokument je relevantní. Ideální případ by byl, kdyby oba koeficienty byly rovny 1. Ukazuje se ale, že tohoto ideálního případu nelze v praxi dosáhnout.

K zjednodušení informací o efektivitě systému byly vytvořeny metody, které naměřenou přesnost a úplnost zobrazují do 1-dimenzionálního prostoru. Jednou z nich je Van Risjbergen's F-míra [4]:

$$F_\beta = 1 - \frac{1+\beta^2}{\frac{\beta^2}{R} + \frac{1}{P}} = \frac{(1+\beta^2)\,RP}{\beta^2 P + R} = \frac{(1+\beta^2)\,|A \cap B|}{\beta^2 |A| + |B|},$$

kde β indikuje poměrovou důležitost mezi přesností a úplností.

3 Vektorový model

Vektorový model [6] dokumentů pochází ze 70. let. Dokumenty a uživatelské dotazy jsou ve vektorovém modelu reprezentovány pomocí vektorů.

Pokud bylo pro indexaci n dokumentů použito celkem m různých termů $t_1 \ldots t_m$, potom je každý dokument d_i reprezentován vektorem:

$$d_i = (w_{i1}, w_{i2}, \ldots, w_{im}),$$

kde w_{ij} je váha termu t_j v dokumentu d_i. Váha s největší hodnotou odpovídá termu s největší důležitostí.
Indexový soubor vektorového modelu reprezentuje matice:

$$D = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1m} \\ w_{21} & w_{22} & \ldots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{nm} \end{pmatrix},$$

kde i–tý řádek odpovídá i–tému dokumentu, a j–tý sloupec j-tému termu.

Dotaz ve vektorovém modelu můžeme reprezentovat pomocí m–místného vektoru vah:

$$q = (q_1, q_2, \ldots, q_m)\,,$$

kde $q_j \in \langle 0, 1\rangle^m$.

Na základě dotazu q můžeme pro každý dokument d_i spočítat *koeficient podobnosti*. Tento koeficient si můžeme představit jako "vzdálenost" vektoru dokumentu a vektoru dotazu. Pro výpočet podobnosti jsme použili *kosonovou míru*:

$$Sim(q, d_i) = \frac{\sum_{k=1}^{m} (q_k w_{ik})}{\sqrt{\sum_{k=1}^{m} (q_k)^2 \sum_{k=1}^{m} (w_{ik})^2}}$$

Další informace o vektorovém modelu jsou uvedeny v [6].

4 Shluková analýza

Úkolem shlukové analýzy je zjistit, zda mezi objekty existují skupiny objektů se stejnými nebo podobnými vlastnostmi. Tyto skupiny objektů se nazývají *shluky*. My se zabýváme shlukováním dokumentů, které se dá rozdělit do dvou kroků: vytvoření shluku a vyhledání relevantního shluku [3]. Spojováním podobných dokumentů do shluků lze dosáhnout zvýšení rychlosti vyhledávání ve vyhledávacích systémech. Důvod, proč se provádí analýza shluků, je obsažen v tzv. hypotéze o shlucích [6]:

Úzce vztažené dokumenty směřují k tomu, že jsou relevantní vůči týmž požadavkům.

Procesu, při kterém se hledá ideální rozklad množiny dokumentů do shluků, ve kterých jsou navzájem podobné dokumenty, se říká *shlukování*. Shluk je tedy tvořen množinou navzájem si podobných dokumentů.

4.1 Metody založené na matici podobnosti

Tyto metody pracují obvykle v čase $O(n^2)$ nebo vyšším (n je počet dokumentů). Podobnostní matici Sim_C pro kolekci C lze popsat takto:

$$Sim_C = \begin{pmatrix} sim_{11} & sim_{12} & \ldots & sim_{1n} \\ sim_{21} & sim_{22} & \ldots & sim_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ sim_{n1} & sim_{n2} & \ldots & sim_{nn} \end{pmatrix},$$

kde i-tý řádek odpovídá i-tému dokumentu a j-tý sloupec j-tému dokumentu.

U těchto shlukovacích metod se vytváří hierarchie rozkladů zadaných dokumentů. V průběhu výpočtu se vytváří shlukovací hladiny, na kterých jsou body spojovány do shluků. Hierarchické metody se dají rozdělit do dvou skupin:

aglomerativní – Na startu těchto metod je každý dokument brán jako jeden shluk, postupně se dokumenty spojují (shlukují) dohromady. Výpočet je ukončen v momentě, kdy všechny dokumenty tvoří dohromady jediný shluk.

divizivní – Pracují přesně opačně než aglomerativní metody. Na startu těchto metod tedy tvoří všechny dokumenty jeden shluk. Shluky se postupně rozpadají až do chvíle, kdy je každý bod samostatným shlukem.

5 Vylepšení odpovědi ve vektorovém modelu

5.1 Definování vývoje tématu

Cílem hledání vývoje tématu je k zadanému dotazu vyhledat seznam dokumentů tématicky souvisejících se zadaným dotazem. Dotazem se zde myslí, jak dotaz zadaný pomocí termů nebo také dokument, který je určen za relevantní.

Příkladem hledání vývoje tématu může být název operačního systému počítače. Zadaný dotaz bude pojednávat o nejnovější verzi operačního systému. My tento dotaz chceme rozšířit o dokumenty, které budou postupně sledovat vývoj tohoto operačního systému.

Vytvoření hierarchie dokumentů pomocí shlukování nám napomáhá k zjištění vývoje tématu. Při zjišťování vývoje, procházíme hierarchii zdola-nahoru, dokud nejsou splněny vyhledávací kritéria, kterými může být například počet dokumentů vrácených uživateli, či podobnost nově do vývoje přidaného dokumentu s dotazem.

5.2 Algoritmus získání vývoje tématu

Pro získání vývoje tématu z hierarchie shluků definujeme algoritmus TOPIC, který používá počet dokumentů ve vývoji, jako omezující kritérium.

Algoritmus TOPIC:

1. Určíme počet dokumentu, který chceme vrátit.
2. Nalezneme listový shluk obsahující označený relevantní dokument.
3. Postupujeme o úroveň výš v hierarchii.
4. Provedeme postupný průchod vedlejšího shluku. V průchodu nejdříve projdeme shluk vytvořený na nejbližší podhladině. Každý dokument, na který narazíme, přidáme do výsledné kolekce. Pokud počet dokumentů ve výsledné kolekci je roven požadovanému počtu dokumentů, ukončíme prohledávání.
5. Pokračujeme bodem 3.

5.3 Uspořádání odpovědi ve vektorovém modelu

Odpovědí na dotaz ve vektorovém modelu je kolekce dokumentů, která je uspořádána podle koeficientu podobnosti dotazu a dokumentu. V této části si představíme metodu, která mění toto uspořádání pomocí informací o vývoji tématu zjištěných ze shluků dokumentů. Geometricky změna tohoto uspořádání znamená oddálení nerelevantních dokumentů od dotazu a přiblížení dokumentů relevantních. Pro tuto změnu uspořádání jsme navrhli dva algoritmy a pojmenovali je SORT-ONE a SORT-EACH.

Algoritmus SORT-ONE spojí kolekci získanou vektorovým dotazem s kolekcí vývoje tématu tak, že postupně přidává dokumenty z jedné a pak z druhé kolekce:

1. Provedeme vektorový dotaz a získanou kolekci dokumentů označíme C_V.
2. Vybereme dokument D_V, který nejlépe charakterizuje kolekci C_V.
3. K dokumentu D_V nalezneme pomocí algoritmu TOPIC kolekci vývoje C_T. Počet dokumentů ve vývoji bude pro jednoduchost odpovídat počtu dokumentů v celé kolekci dokumentů.
4. Označíme výslednou setříděnou kolekci C_S a určíme počet dokumentů, který má obsahovat - *count*.

5. Provedeme následující třídění:

```
v_index = 0 // index prvního dokumentu v kolekci C_V
t_index = 0 // index prvního dokumentu v kolekci C_T
while C_S neobsahuje count dokumentů do
    while nedojde k přidání dokumentu do C_S do
        if C_S neobsahuje dokument C_V[v_index] then
            přidej dokument C_V[v_index] do C_S
        v_index = v_index + 1 // přesun na další dokument
    end
    if C_S obsahuje count dokumentů then
        ukonči třídění
    while nedojde k přidání dokumentu do C_S do
        if C_S neobsahuje dokument C_T[t_index] then
            přidej dokument C_T[t_index] do C_S
        t_index = t_index + 1 // přesun na další dokument pak
    end
end
```

6. Kolekci C_S zobrazíme uživateli.

Algoritmus závisí na výběru dokumentu charakterizujícího kolekci výsledku. Tento dokument může uživatel vybrat sám z kolekce dokumentů, kterou mu nabídneme po provedení vektorového dotazu, nebo jej můžeme vybrat automaticky. Pro automatický výběr dokumentu jsme navrhli metodu metoda DOC-FIRST: jako reprezentant se vybere dokument s největší hodnotou koeficientu podobnosti k dotazu.

Algoritmus SORT-EACH, přesune dokumenty v kolekci získané vektorovým dotazem tak, aby dokumenty ze stejného vývoje tématu byly za sebou:

1. Provedeme vektorový dotaz a získanou kolekci dokumentů označíme C_V.
2. Označíme výslednou setříděnou kolekci C_S a určíme počet dokumentů, který má obsahovat - *count*.
3. Určíme, kolik rozšiřujících dokumentů má obsahovat vývoj tématu k zadanému dokumentu. Tuto hodnotu označíme *level*.
4. Provedeme následující třídění:

```
foreach dokument D_V v C_V do
    if C_S je prázdná then
        vlož D_V do kolekce C_S
        goto Continue
    end
    K dokumentu D_V nalezneme pomocí algoritmu TOPIC
        kolekci vývoje C_T. Počet dokumentů ve vývoji
        bude level + 1 (dokument D_V).
    foreach dokument D_T v C_T mimo dokument D_V do
        if dokument D_T je v C_S then
            zařaď dokument D_V za D_T do C_S
            goto Continue
        end
```

end
if nebyl dosud dokument D_V zařazen **then**
vlož D_V do kolekce C_S
label: Continue
end

5. Kolekci C_S zobrazíme uživateli.

6 Testování

6.1 Postup testování

Pro testování jsme použili kolekci dokumentů nazvanou "Medlars Collection" (dostupnou na ftp://ftp.cs.cornell.edu/pub/smart). Kolekce obsahuje 1033 anglických abstraktů z oblasti medicíny (velikost 1.03 MB). Dále kolekce obsahuje sadu 30 dotazů, z nichž jsme vybrali 24, které na vektorový dotazu vracejí alespoň 100 dokumentů.

Vytvořili jsme indexy shluků dokumentů pomocí různých metod shlukování lišících se v metodě přepočtu matice podobnosti. Použité metody byly tyto: nejbližšího souseda, nejvzdálenějšího souseda, Wardova, průměrová a metoda mediánová.

Definovali jsme následující testovací metody pro úpravu vektorového vyhledávání:

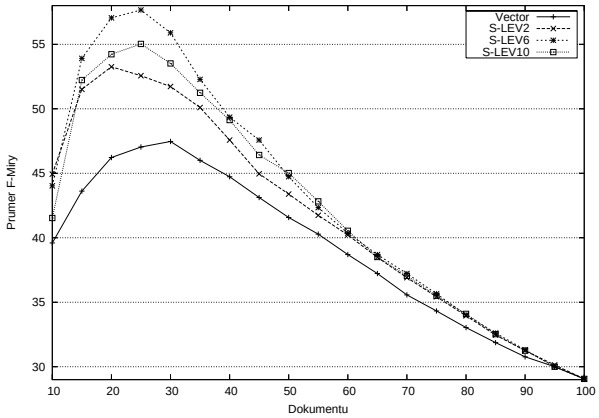
- **SORT-ONE s DOC-FIRST** (dále S-FIRST): Výsledek vektorového dotazu je upraven pomocí metody SORT-ONE s metodou výběru dokumentu DOC-FIRST.
- **SORT-EACH s *level*=2, s *level*=6 a s *level*=10** (dále S-LEV2, S-LEV6 a S-LEV10): Výsledek vektorového dotazu je upraven pomocí metody SORT-EACH s *level*=2, s *level*=6 nebo s *level*=10.
- **Rozšíření pomocí KLD** (dále E-KLD): Nalezneme metodou UP-DONW-2 (viz. [1]) shluk nejvíce podobný dotazu. V tomto shluku pomocí metody KLD nalezneme 5 rozšiřujících termů a přidáme je k původnímu dotazu. Pomocí tohoto nového dotazu provedeme vektorový dotaz.

Pro vytvořené hierarchie shluků jsme provedli následující test s každou výše uvedenou metodou:

1. Metodu jsme provedli na každém z 24 dotazů.
2. Pro každý výsledek dotazu jsme spočítali hodnotu přesnosti a úplnosti a z nich F-míru s $\beta = 1$ pro prvních 5,10, ... 100 získaných dokumentů.
3. Vypočítali jsme průměrnou hodnotu F-míru ze všech dotazů pro prvních 5,10, ... 100 získaných dokumentů.

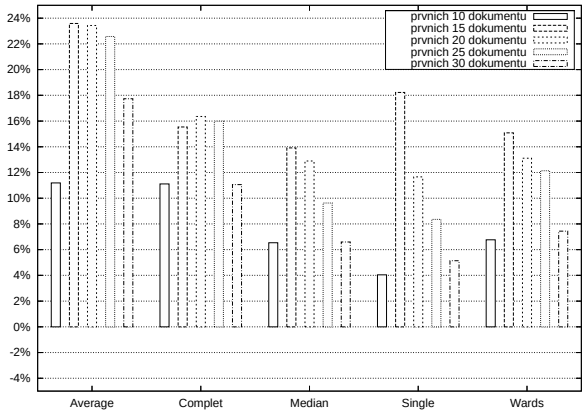
6.2 Zhodnocení výsledků testování

Testy jsme provedli s metodami S-LEV2, S-LEV6 a S-LEV10, které provádí přeskládání výsledku vektorového dotazu podle vývoje dokumentů zjišťovaného ve shlucích. Ukázalo se že pomocí tohoto postupu lze dosáhnout zlepšení až k 22% F-míry oproti původnímu vektorovému dotazu. Toto zlepšení se projevuje v testech provedených u prvních 10-30 dokumentů. Vzájemným srovnáním jednotlivých metod jsme zjistili, že nejlepších výsledků se dosáhlo u metody S-LEV6. Zhoršení u metody S-LEV10 lze přisoudit velikosti kolekce dokumentů, kdy při 1033 dokumentech, shluky které obsahují samostatná téma



Obrázek 1: Srovnání vektorového dotazu s metodami S-LEV2, S-LEV5 a S-LEV6 u indexu vytvořeného metodou průměrů.

jsou malé. Příklad grafu srovnávajícího jednotlivé metody u indexu vytvořeného metodou průměrů je na obrázku 1. Procentuální vylepšení F-míry metodou S-LEV6 oproti původnímu vektorovému dotazu u jednotlivých shlukovacích metod je uvedeno na obrázku 2. Je zde patrné, že nejlepší výsledky se dosahují opět u metody průměrů. Za další vhodné shlukovací metody lze označit medotu nejvzdálenějšího souseda a Wardovu metodu.

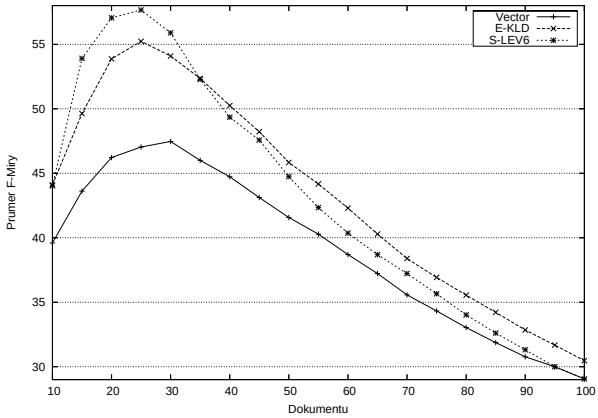


Obrázek 2: Srovnání shlukovacích metod s metodou S-LEV6.

Dalším testem, který jsme provedli bylo srovnání výše uvedených testů u metod S-LEV6 a E-KLD. Jako příklad uvádíme graf na obrázku 3. Z grafu je patrné, že výsledky obou metod jsou srovnatelné. U metody S-LEV6 oproti E-KLD nemusíme provádět prohledávání stromu shluků.

7 Závěr

Z výsledků provedených testů vyplývá, že metoda DOC-UP-DOWN2 je nejvhodnější pro výběr dokumentu, který nejvíce reprezentuje zadaný dotaz. Dále jsme zjistili, že pomocí metod, které přeskládávají výsledek vektorového dotazu za pomoci vývoje tématu



Obrázek 3: Srovnání vektorového dotazu s metodami E-KLD a S-LEV6.

zjišťovaného v hierarchiích shluků lze dosáhnou až 22% zlepšení F-míry. U metod pro rozšíření dotazu o 5 termů lze dosáhnout zlepšení přesahujících hranici 16%.

Výraznější zlepšení oproti původnímu vektorovému dotazu se nejvíce projevuje v prvních 10–30 získaných dokumentech.

Z popsaných shlukovacích metod se ukázaly nejlepší metody, u kterých nedochází ke tvorbě řetězců. V případě, že při vytváření hierarchie shluků dochází k tvorbě řetězů je velikost indexu velká. Nevhodná se projevila metoda nejbližšího souseda, u které je velikost vytvořeného indexu výrazně větší než u ostatních shlukovacích metod. Nejlepší se jeví vytváření hierarchií shluků pomocí metody průměrů, která v námi provedených testech dosahovala nejlepších výsledků.

Z výsledků testů srovnávajících různé metody shlukovaní a různé metody vylepšení odpovědi vektorového dotazu na naší testovací kolekci dokumentů, můžeme doporučit používat shlukovací metody a námi navržené metody pro oddálení nerelevantních dokumentů od vektorového dotazu. Vhodnými jsou shlukovací metody, u kterých nedochází k tvorbě řetězů, ale vzniklá hierarchie shluků je co nejvíce pravidelně rozložená. Z metod pro vylepšení vektorového dotazu se jako vhodná ukázala metoda SORT-EACH.

Literatura

1. Dvorský J., Martinovič J., Pokorný J., Snášel V.: Vyhledávání témat v kolekci dokumentu, Znalosti 2004.
2. Dvorský J., Martinovič J., Snášel V.: Query Expansion and Evolution of Topic in Information Retrieval Systems, DATESO 2004.
3. Christis Faloutsos, Dayglas Oard: A Survey of Information Retrieval and Filtering Methods, University of Maryland, College Park, MD 20742.
4. Tsunenori Ishioka: Evaluation of Criteria for Information Retrieval, The National Center for university Entrance Examinations, Japan.
5. Gerald J. Kowalsi, Mark T. Maybury: Information Storage and Retrieval System, Theory and implementation, Secon Edition, Kluwer Academic Publishers, 2000.
6. Pokorný J., Snášel V., Húsek D.: Dokumentografické informační systémy. Karolinum, Skriptum MFF UK Praha, 1998.