Dokumentografické systémy, fulltextové vyhledávání, filtrace, disambiguace, lemmatizace, indexy, tezaury, dotazování.

Thursday, May 30, 2013 8:21 AM

Dokumentografické systémy (DIS)

- vznik 50. léta 20. stol. za účelem automatizace postupů používaných v knihovnictví
- Nyní samostatná podčást IS
  - Faktografický IS - informace s definovanou vnitřní strukturou (nejčastěji tabulky)
  - Dokumentografický IS - informace v podobě textu v přirozeném jazyce bez pevné vnitřní struktury

Práce s DIS:

- *Zadání dotazu*
- *Porovnání*
- *Získání seznamu odpovídajících dokumentů*
- *Ladění dotazu*
- *Vyžádání dokumentu*
- *Obdržení textu*

Struktura DIS:

- Systém zpřístupnění dokumentu - sekundární informace o dokumentu (Autor, Název, ...)
- Systém dodání dokumentu - někdy není řešeno pomocí SW

Vyhodnocení dotazu

- přímé porovnávání náročné na čas

Ale při využití indexace a modelu dokumentu:

- nutné vytvořit model dokumentu
- ztrátový proces, založený na identifikaci slov v dokumentech
- výsledkem strukturovaná data vhodná pro porovnávání
- dotaz se upraví do odpovídající podoby a porovná se s modelem dokumentů

Text

Předzpracování

- vyhledávání nad modelem efektivnější, ale lze použít jen informace z modelu
- cíl: vytvořit model, zachovávající nejvíce info z původního textu
- Problém: nejednoznačností (ambiguity)
- dosud neřešené nároky na encyklopedické i asociativní znalosti

Porozumění textu

- Homonymie slov
  - jedno slovo může mít stejný tvar pro různé pády a další gram. jevy
    - kontroly: 1.p.m.č., 2.p.j.č. - není zřejmé jestli více kontrol nebo jedna kontrola
  - jeden tvar může mít různý význam
    - hnát - sloveso, podst. jm.
    - pět - číslovka, sloveso
- přiřazení je závislé na osobě, která dokument píše nebo čte
  - dva lidé mohou jednomu slovu přikládat zcela nebo částečně jiný význam
  - dva lidé si i pod stejným významem mohou představit jiný konkrétní předmět nebo množinu
    - máma, pokoj, ...
- výsledkem situace, kdy dva různí čtenáři nemusí přečtením získat stejnou informaci jako autor, ani

navzájem.
- Homonymie a nejednoznačnosti narůstají při přechodu od slov k větám.

Přesnost a úplnost

- důsledek nejednoznačnosti: žádný existující DIS nedává ideální výsledky
- Pro zobrazení odpovědi na dotaz lze určit
  - N_v (počet vrácených dokumentů) - O nich si DB myslí, že jsou relevantní, odpovídající dotazu
  - N_{vr} (počet vrácených relevantních dok.) - o nich si tazatel myslí, že uspokojí jeho požadavky
  - N_r (počet všech relevantních dok. v DB) - problematické u velkých DB
- Kvalita výsledné množiny se měří na základě:
  - Přesnost (Precision): $P = N_{vr} / N_v$
    - pravděpodobnost, že dokument zařazený v odpovědi je skutečně relevantní
  - Úplnost (Recall): $R = N_{vr} / N_r$
    - pravděpodobnost, že skutečně relevantní dokument je zařazený v odpovědi
- koeficienty jsou opět závislé na subjektivním názoru tazatele
- dokument vrácený na výstupu může uspokojovat požadavky dvou uživatelů, kteří položili stejný dotaz, různou měrou
- ideální případ: P == R == 1

Kritéria

- **Kritérium predikce**
  - při formulaci dotazů je třeba uhádnout, které termy (slova) byly v dokumentu autorem použity pro vyjádření dané myšlenky
    - problémy způsobují
      - synonyma - autor používá synonyma, které si tazatel nemusí při dotazu uvědomit
      - překrývající se význam slov
      - opisy jedné situace jinými slovy
  - částečné řešení - zařazení tezauru, který obsahuje
    - hierarchie slov a jejich významů
    - synonyma slov
    - asociace mezi slovy
  - tazatel může tezaurus využít při formulaci svých dotazů
  - při ladění dotazů má uživatel tendenci postupovat konzervativně
    - v dotazu často zůstávají ty části, které uživatele napadly na začátku a mění se jen podružné části, které nekvalitní výsledek nemusí zásadně ovlivnit
  - vhodné je uživateli pomoci s odstraněním nevhodných částí dotazu, které nepopisují relevantní dokumenty a naopak s přidáváním formulací, které relevantní dokumenty popisují
- **Kritérium maxima**
  - tazatel obvykle není schopen (ochoten) procházet příliš mnoho dokumentů do té míry, aby se rozhodl, zda jsou pro něj relevantní nebo ne
  - obvykle 20-50 podle velikosti
  - potřeba nejen dokumenty rozlišovat na odpovídající/neodpovídající dotazu, ale řadit je na výstupu podle míry předpokládané relevance
  - v důsledku kritéria maxima se při ladění dotazu uživatel obvykle snaží zvýšit přesnost
    - malé množství dokumentů v odpovědi, obsahující co největší poměr relevantních dokumentů
  - některé oblasti použití vyžadují co nejvyšší přesnost i úplnost
    - např. právnictví

<u>Modely dokumentografických systémů</u>

Úrovně modelů:

- rozlišují (ne)přítomnost slov v dokumentech
- rozlišují frekvence výskytů slov
- rozlišují pozice výskytů slov v dokumentech

Boolský model

- vznik 50. léta 20. stol., automatizace postupů používaných v knihovnictví
- Databáze obsahuje dokumenty, dokumenty popisovány pomocí termů, reprezentace dokumentu pomocí množiny termů (obsažených v dokumentu, popisujících význam dokumentu)
- **Indexace**
  - Přiřazení množiny termů, které jej popisují ke každému dokumentu
    - Ruční - nekonzistence
    - Automatická - konzistentní, ale bez porozumění textu
    - Řízená - předem daná množina termů
    - Neřízená - množina termů se mění s přibývajícími dokumenty
  - Tezaurus - vnitřně strukturovaná množina termů
    - Synonyma s preferovanými termy
    - Hierarchie užších/širších termů
    - Příbuzné termy
    - ...
  - Stop-list - nevýznamová slova
  - Příliš obecná slova nejsou pro identifikaci dokumentů vhodná, příliš specifická slova také ne
  - dotaz vyjádřen logickým výrazem: AND, OR, NOT
  - Příklad dotazu: počítač AND NOT osobní
  - Víceslovné termy: počítač AND NOT osobní počítač
  - Organizace indexu:
    - Invertovaný seznam - pro každý term je seznam dokumentů, ve kterých se vyskytuje
    - Zpracování dokumentů na vstupu - vznikne posloupnost dvojic <dok_id,term_id>
    - Setřídění dle term_id,dok_id
- **Nevýhody**
  - formulace dotazů je spíše uměním než vědou
  - nemožnost ohodnotit vhodnost vystupujících dokumentů
  - všechny termy v dotazu i v identifikaci dokumentu jsou chápány jako stejně důležité
  - nemožnost řízení velikosti výstupu
  - některé výsledky neodpovídají intuitivní představě

Vektorový model

- vznik 70. léta 20. stol., cca o 20 let mladší než Booleovské DIS
- snaha minimalizovat nebo odstranit nevýhody Booleovských DIS
- Struktura:
  - databáze obsahuje dokumenty
  - dokument popisován pomocí množiny termů
  - term je slovo nebo sousloví
  - reprezentace dokumentu pomocí vektoru vah termů

Tezaury

In Microsoft SQL Server 2005, full-text queries can use a thesaurus to find synonyms of search terms. For each supported language, there exists a single thesaurus file.

From <http://msdn.microsoft.com/en-us/library/ms142491(v=SQL.90).aspx>

Tezaurus je vnitřně strukturovaná množina termů:
-synonyma s preferovanými termy
-hierarchie užších/širších termů
-asociace mezi slovy

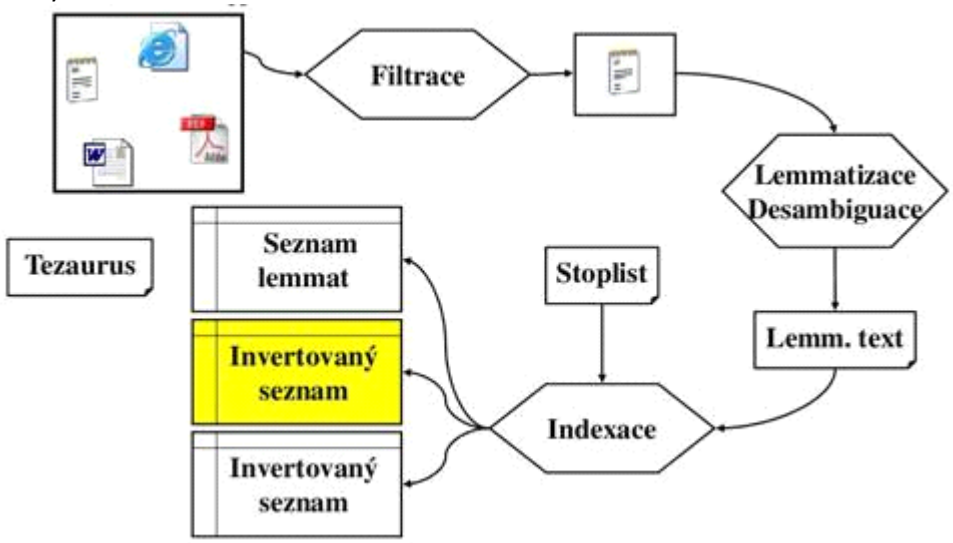
Fulltextové vyhledávání

- Odlišné od principů běžného vyhledávání
  - neprohledávají se striktně strukturovaná data, kde má každý sloupec každé tabulky předem

daný význam
    - prohledávají se volně psané texty, kde může být stejná událost popsána více autory rozdílně - různá slova stejného významu, různé slovní obraty a opisy
- DB systémy využívají svých prostředků rozšiřitelnosti a dodávají standardně prostředky, které vyhledávání v textových datech umožňují
- Rozdílné přístupy a možnosti
    - neexistuje objektivně nejlepší řešení
    - výsledky navíc podléhají subjektivním názorům tazatelů
- Samotná formulace dotazu, který by vrátil všechny dokumenty, které tazatele zajímají a žádné jiné, obvykle nelze zformulovat
    - spolu s vyhovujícími - relevantními - odpověďmi se obvykle vrací i odpovědi nerelevantní
- **Problémy**
    - Homonyma - ptá se tazatel dotazem "koruna" na finanční, lesnické či panovnické dokumenty?
    - Synonyma
        - Vyhovuje dokument o "krychlích" dotazu na dokumenty o "kostkách"?
        - Vyhovuje dokument o "stromech" dotazu na "souvislé grafy bez cyklů"?
    - Hierarchie významů
        - Zvíře - Savec - Šelma - Medvěd
        - Tiskovina - Časopis
    - Ohebnost slov - Jít, Jde, Jdu, Jdou, ...
- Striktní booleovská logika není pro formulaci dotazů příliš vhodná
    - Dokument buďto vyhovuje nebo nevyhovuje
    - Dotazování v textech vyžaduje třídit odpovědi podle předpokládané vhodnosti pro tazatele - je potřebné mít možnost definovat míru shody dotazu s dokumentem

- Pozn. Možná dobré vědět něco o algoritmech prohledávání řetězců viz PT: Knuth-Morris-Pratt, Boyer-Moore, Brute Force, Rabin-Karp…

Předzpracování

- Databáze obvykle používají některý z booleovských modelů reprezentace dokumentů
    - nejlépe odpovídá běžným dotazům
    - relativně snadno se implementuje
    - dotazy jsou ve formě booleovských formulí, ve kterých operandy tvoří jednotlivá slova - řada různých modifikací
- 

Jednotlivé kroky předzpracování fulltextového vyhledávání:

Filtrace

- *Filtrace* - odstraní formátovací značky a nechá čistý ASCII text

Disambiguace

- *Desambiguace* - určí význam slova podle kontextu

- "pět chválu" ... sloveso pět
- "pět vozidel" ... číslovka pět

Lemmatizace

- *Lemmatizace* - určí základní tvar slova a gramatický tvar v dokumentu, často nahrazen pomocí stemmeru, který hledá kmen slova

Indexace

- *Indexace* - vytvoří pomocné seznamy lemmat a dokumentů a invertovaný soubor
  - dvojice *[id_dok, id_lemmatu]* setříděné dle id_lemmatu a zbavené duplicit
  - dnes obvykle více informací, např. pětice *[id_dok,č_odstavce,č_věty,č_slova,id_lemmatu]* - dovoluje vyhodnocování tzv. proximitních omezení na vzdálenost slov v dokumentu

Large OBjects (LOB)

- pro podporu vyhledávání je potřeba nad textovým sloupcem vytvořit index - invertovaný soubor
- běžné textové sloupce jsou pro tyto účely krátké a nevyhovující
  - obvykle se takto indexují sloupce některého z LOB (Large OBject) typů
- LOBy
  - standardní typy pro ukládání objemných dat na serveru, definováno v SQL-92 Full
  - až 4GB dat
    - BLOB - standardní binární typ
    - CLOB - znakový typ v univerzální znakové sadě
    - NCLOB - znakový typ v národní znakové sadě
  - v Oracle navíc externí typ BFILE
    - pouze pro čtení
    - samostatný binární soubor uložený vně databáe v OS
  - v MS SQL
    - Image - binární data do velikosti 2 GB
    - Text - textová data do velikosti 2 GB
    - NText - textová data v národní znakové sadě do vel. 1 GB
- Ve sloupcích tabulky je uložen pouze deskriptor (tzv. LOB lokátor), odkazující na samostatně uložená data

Dotazování

Oracle fulltext

- Filtrování vstupních dokumentů
  - **NULL_FILTER** - pro textové dokumenty TXT, HTML, XML
  - **INSO_FILTER** - pro binární dokumenty
  - **CHARSET_FILTER** - pro konverzi získaných dokumentů do znakové sady databáze
- Druhy fulltextových indexů
  - **CONTEXT**
    - základní typ indexu pro vyhledávání v textových datech
    - vhodný pro větší dokumenty
    - synchronizace indexu s daty je nutno provést explicitně
  - **CTXCAT**
    - vhodný pro menší dokumenty a jejich úryvky
    - může být zkombinován s dalšími netextovými sloupci pro kombinované dotazování
    - synchronizace indexu s daty se provádí automaticky se změnami v tabulce
  - **CTXRULE**
    - postaven na množině předdefinovaných dotazů
    - slouží pro klasifikaci dokumentů do skupin podle toho, kterým dotazům vyhovuje
- Uložení dokumentů
  - **NORMAL_DATASTORE** - text je v jednom sloupci jednoho řádků
  - **MULTI_COLUMN_DATASTORE** - text ve více sloupcích jednoho řádku
  - **URL_DATASTORE** - text je na internetu, dostupný přes URL ve sloupci

- Příklad vytvoření indexu nad textovým sloupcem:

```
CREATE INDEX myindex ON doc(htmlfile)
INDEXTYPE IS ctxsys.context
PARAMETERS('datastore ctxsys.default_datastore
filter ctxsys.null_filter
section group ctxsys.html_section_group');
```

- Spolu s novými typy indexů databáze implementují nové operátory pro porovnávání dotazu s textem
- Operátory vrací číslo - očekávanou míru shody obsahu textu s tazatelovými požadavky

Příklad dotazu:

```
strSql = "SELECT SCORE(1), file_name, filesize FROM my_doc " & _
    WHERE CONTAINS(content," & Search.Value & ", 1) > 0 " & _
    ORDER BY SCORE(1) DESC"
```

From <http://www.codeproject.com/Articles/12188/Full-text-search-with-Oracle-Text>

From <https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>