Západočeská univerzita v Plzni
Fakulta aplikovanch věd
Katedra informatiky a výpočetní techniky

Automatická indexace a vyhledávání dokumentů

KIV/IR

David Pivovar
pivovar@students.zcu.cz

Plzeň, 2019

Obsah

1. Zadání

Cílem semestrální práce je naučit se implementovat komplexní systém s využitím hotových knihoven pro preprocessing. Vedlejším produktem bude hlubší porozumnění indexerům, vyhledávacím systémům a přednáškám. Systém po předchozím předzpracování zaindexuje zadané dokumenty a poté umožní vyhledávání nad vytvořeným indexem. Vyhledávání je možné zadáním dotazu s logickými operátory AND, OR, NOT. Výsledek dotazu by měl vrátit top x (např. 10) relevantních dokumentů seřazených dle relevance.
Projekt by měl být implementován v jazyce Java s využitím poskytnutých podpůrných interfaců a abstraktních tříd. Výstupem bude jeden jar soubor obsahující javaDoc, přeložené i zdrojové soubory projektu a dokumentace popisující jednotlivé funkčnosti semestrální práce s důrazem kladeným na nadstandardní funkčnosti a vlastní řešení. Semestrální práce se bude testovat indexací dokumentů a vyhledáváním relevantních dokumentů nad vytvořeným indexem (tyto data nesmíte dále šířit a používat pro jiné účely). Pro porovnání jednotlivých semestrálních prací mezi sebou bude nakonec zveřejněna tabulka s naměřenými hodnotami. Proto je třeba aby váš projekt obsahoval třídy z projektu Interface (při doplnění Třídy Index lze spuštěním TestTrecEval získat výsledný soubor (relevantní dokumenty) a pokud lze spustit evaluační skript (trec_eval.8.1/trec_eval) tak výsledek evaluace (popis viz README). Hlavní evaluační metrika je Mean Average Precision (MAP). Pomocí této hodnoty budeme řadit vaše výsledky. Pokud nechcete zveřejnit své výsledky, dejte nám to vědět emailem.

2. Indexace

Při indexaci projde každý dokument nejprve preprocessingem. Poté se vytvoří invertovaný index a spočte tf-idf váha. Dokumenty se mouhou doindexovat i zpětně, připadně aktualizovat nebo odebrat.

2.1 Preprocessing

Nejprve se text převede na lower case. Poté se text tokenizuje, jako první datumy, webové adresy a odstraní se html tagy. Pak se provede stemming a odstraní se akcenty. Na závěr se odstraní stop slova.

2.2 Inverted index

Inverted index představuje Hash mapa. Klíč je term, hodnota je seznam dokumentů, ve kterých se term vyskytuje a s jakou frekvencí.

2.3 Tf-idf

Tf-idf je metodika hodnocení relevance při vyhledávání textu. Tf-idf hodnota pro každý term v dokumentu je spočtena jako tf*idf, kdy tf je četnost slova v dokumentu a idf je převrácená četnost slova ve všech dokumentech.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

3. Vyhledávání

Při vyhledávání se query nejprve tokenizuje (projde preprocessingem stejně jako při indexci). Následně se vytvoří seznam termů pro logické operace AND/OR/NOT (operandy se z query odstraní). Vypočte se tf-idf váhový vektor a jeho normalizovaný tvar.

Pro každý dokument, který obsahuje term z query se vytvoří tf-idf váhový vektor. Pokud dokument některý term neobsahuje, je na dané pozici vektoru 0. Poté se provedou logické operace AND/OR/NOT. Následně se všechny váhové vektory normalizují.

Nakonec se pro každý dokument vypočte skóre, výsledky se seřadí od největšího skóre a ohodnotí.

3.1 Logické operace AND/OR/NOT

Každý dokument se vyhodnotí podle seznamu logických operátorů AND/OR/NOT. Nevyhovující dokument se z výsledků odstraní.

AND - v dokumentu musí být obsaženy oba termy (tj hodnota vektoru na pozici daných termů je nenulová)
OR - v dokumentu musí být obsažen alespoň jeden z termů
NOT - daný term nesmí být v dokumentu obsažen

Pokud by po provedení logických operací nezbyl žádný dokument, logické operace se neprovedou.

3.2 Výpočet skóre

Skóre se vypočte pomocí cosine similarity. Provede se skalární součin váhového vektoru query a dokumentu, který se vydělí součinem jejich normalizovaných tvarů.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

4. Spuštění

Aplikace se spouští příkazem **java jar Indexer.jar [path]**
Evaluace se spouští příkazem **java jar IndexerEval.jar**

Ovládání aplikace **Indexr.jar** po načtení dat:

q	query	
a	path	add document
u	path	update document
r	docId	remove document
h		help
e		exit

Pro spuštění je potřeba mít nainstalovanou javu 1.8.
V adresáři, kde se nachází jar soubor, musí být soubor *stopwords.txt*, který obsahuje stop slova, adresář *TREC* s daty *czechData.bin* a *topicData.bin*. Do tohoto adresáře se ukládají výsledky. Pro spuštění evaluace musí být adresář *trec_eval.8.1* s evaluační aplikací.

5. Výsledky evaluace

num_q	all	50
num_ret	all	3719126
num_rel	all	762
num_rel_ret	all	760
map	all	0.1804
gm_ap	all	0.0527
R-prec	all	0.1999
bpref	all	0.1842
recip_rank	all	0.4056
ircl_prn.0.00	all	0.4480
ircl_prn.0.10	all	0.3453
ircl_prn.0.20	all	0.2870
ircl_prn.0.30	all	0.2534
ircl_prn.0.40	all	0.2203
ircl_prn.0.50	all	0.1891
ircl_prn.0.60	all	0.1600
ircl_prn.0.70	all	0.1291
ircl_prn.0.80	all	0.1025
ircl_prn.0.90	all	0.0560
ircl_prn.1.00	all	0.0478
P5	all	0.2480
P10	all	0.1920
P15	all	0.1707
P20	all	0.1540
P30	all	0.1347
P100	all	0.0652
P200	all	0.0394
P500	all	0.0202
P1000	all	0.0112

6. Závěr

V rámci semestrální práce jsem implementoval požadovanou základní funkcionalitu a některou nadstandardní.
Evaluaci jsem spustil na ZČU serveru ares.fav.zcu.cz. Mean Average Precision (MAP) při evaluaci vyšlo 0.1804.

Implementovaná nadstandartní funkčnost:

- ošetření html tagů
- tokenizace datumů
- doindexování dokumentů
- dokumentace v LaTeXu