Using BM25F for Semantic Search

Jose R. Perez-Aguera, Javier Arroyo, Jane Greenberg, Joaquin Perez-Iglesias, Victor Fresno

Metadata Research Center, UNC, UCM, UNED

April 26 - 2010

Outline

Keyword-based Semantic Search

Keyword-based Semantic Web search engine development has become a major research area garnering much attention in the Semantic Web community over the last seven years.

Just for the sake of curiosity

It is possible to improve quality results in terms of relevance applying just classical IR approaches to RDF semantic structure?

Two main problems

- Indexing RDF triples using inverted indexes
- Ranking based retrieval for RDF objects

Two main problems

- Indexing RDF triples using inverted indexes
- Ranking based retrieval for RDF objects

Outline

Problem

How to store RDF triples in inverted indexes OR how to represent subjects, predicates, and objects information in a $n \times m$ matrix.

Solutions

- SIREN (based on XML indexing techniques)
- SEMPLORE model (based on the idea of artificial documents with fields)

Problem

How to store RDF triples in inverted indexes OR how to represent subjects, predicates, and objects information in a $n \times m$ matrix.

Solutions

- SIREN (based on XML indexing techniques)
- SEMPLORE model (based on the idea of artificial documents with fields)

Problem

How to store RDF triples in inverted indexes OR how to represent subjects, predicates, and objects information in a $n \times m$ matrix.

Solutions

- SIREN (based on XML indexing techniques)
- SEMPLORE model (based on the idea of artificial documents with fields)

We follow SEMPLORE model with some changes.

Index Structure based on SEMPLORE model

FIELD	CONTENT
text	plain text
title	keywords from the URI
obj	objects
inlinks	incoming link defined by a predicate
type	rdf:type

Table: Fields used to represent RDF structure in the inverted index.

Two dbpedia entries

- *The Godfather* http://dbpedia.org/page/The_Godfather
- *Francis Ford Coppola* http://dbpedia.org/page/Francis_Ford_Coppola

Triple

The Goodfather (Subject) - dbpprop:director (Predicate)- Francis Ford Coppola (Object)

Using inlink text to index the landing URL

The word *director* can be used as keyword to index the entry described by this URI http://dbpedia.org/page/Francis_Ford_Coppola.

Two dbpedia entries

- *The Godfather* http://dbpedia.org/page/The_Godfather
- *Francis Ford Coppola* http://dbpedia.org/page/Francis_Ford_Coppola

Triple

The Goodfather (Subject) - dbpprop:director (Predicate)- Francis Ford Coppola (Object)

Using inlink text to index the landing URL

The word *director* can be used as keyword to index the entry described by this URI http://dbpedia.org/page/Francis_Ford_Coppola.

Two dbpedia entries

- *The Godfather* http://dbpedia.org/page/The_Godfather
- *Francis Ford Coppola* http://dbpedia.org/page/Francis_Ford_Coppola

Triple

The Goodfather (Subject) - dbpprop:director (Predicate)- Francis Ford Coppola (Object)

Using inlink text to index the landing URL

The word *director* can be used as keyword to index the entry described by this URI http://dbpedia.org/page/Francis_Ford_Coppola.

Outline

Classical IR

For long time, search engines have been dealing with flat documents, that is, without structure.

Consequence

The main consequence of this approach is the fact that terms within a document are considered to have the same relevance (or value), disregarding their role in the document.

Simplification

This assumption implies a relevance model simplification based on **bag of words**, and, therefore, useful information is lost.

Classical IR

For long time, search engines have been dealing with flat documents, that is, without structure.

Consequence

The main consequence of this approach is the fact that terms within a document are considered to have the same relevance (or value), disregarding their role in the document.

Simplification

This assumption implies a relevance model simplification based on **bag of words**, and, therefore, useful information is lost.

Classical IR

For long time, search engines have been dealing with flat documents, that is, without structure.

Consequence

The main consequence of this approach is the fact that terms within a document are considered to have the same relevance (or value), disregarding their role in the document.

Simplification

This assumption implies a relevance model simplification based on **bag of words**, and, therefore, useful information is lost.

Structured IR

- Structured IR uses the document's structure to identify where the most representative terms of the document are (e.g. title, abstract,HTML or XML tags, etc)
- Boost factors are used to modify the impact of every term in the ranking function in order to take into account the document's structure.

Ranking functions

State of the art models have been adapted to this situation.

- BM25F
- LM for structured documents

... but this adaptation have some tricks.

The Problem

The linear combination of weigths for each field of the document is not enough if a saturation function, like $log(tf)$ or $\sqrt{tf}$ is used in the TF function

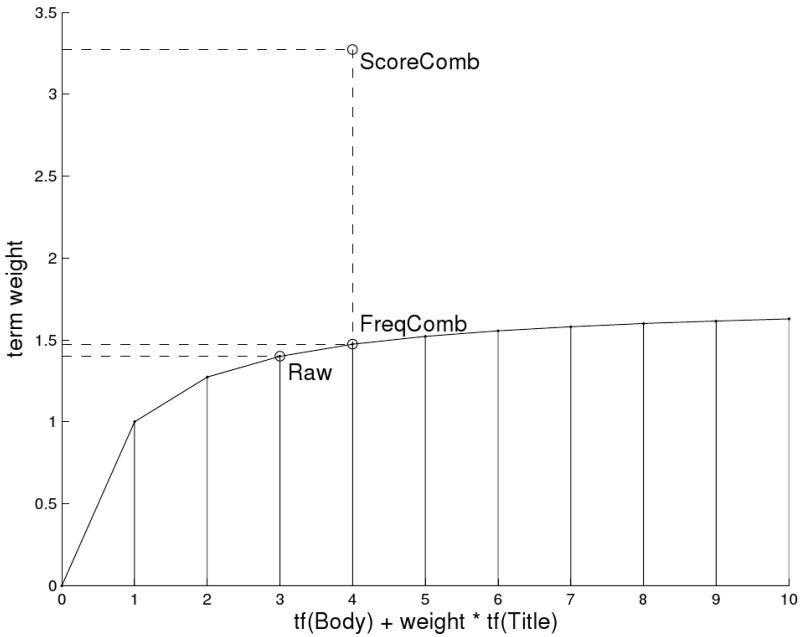

Figure: Source: Robertson et al.2004

Lucene's ranking function

The method used by Lucene to compute the score of an structured document is based on the linear combination of the scores for each field of the document.

$$score(q,d) = \sum_{c \in d} score(q,c) \tag{1}$$

where

$$score(q,c) = \sum_{t \in q} tf_c(t,d) * idf(t) * w_c \tag{2}$$

and

$$tf_c(t,d) = \sqrt{freq(t)} \tag{3}$$

Outline

The collection

- INEX evaluation framework fits good enough to the goal of evaluating Semantic Search systems with small changes
- We have mapped Dbpedia to the Wikipedia version used in the INEX contest
- Dbpedia entries contain semantic information drawn from Wikipedia pages

DBpedia entries: A sort of structured documents

- http://dbpedia.org/resource/The_Lord_of_the_Rings
- http://dbpedia.org/resource/Berlin
- http://dbpedia.org/resource/Semantic_Web

Statistics of the collections

Currently Dbpedia contains almost three millions of entries and the INEX Wikipedia collection contains 2,666,190 documents. As a result, our corpus only takes into account the 2,233,718 document or entities that result from the intersection of both collections.

Topics (Queries)

Given the corpus, INEX 2009 topics and assessments are adapted to this intersection. The result of this operation have been 68 topics and a modified assessments file.

Outline

Semantic Search Engines

- Sindice
- Watson
- Falcon
- SEMPLORE

Everybody is using Lucene, but are they using Lucene's ranking function? I don't know.

Using TITLE, DESCRIPTION and NARRATIVE from Topics

	MAP	P@5	P@10	GMAP	R-Prec
Lucene	.1560	.4147	.3368	.0957	.2100
LuceneF	.1200	.3971	.2971	.0578	.1632
BM25	.1746	**..4735**	**.3868**	.1081	.2257
BM25F	**.1822**	.4647	.3824	**.1170**	**.2262**

Table: MAP, P@5, P@10, GMAP, R-Prec for long queries. All this measures ranges from 0 to 1

Sensibility test for BM25F. All this measures ranges from 0 to 1. *te* = *text*, *ti* = *title*, *in* = *inlinks*, *ob* = *obj*, *ty* = *type*, *all* = *allfields*

	te	te+ti	te+in	te+ob	te+ty	all
MAP	.1756	**.1867**	.1760	.1749	.1750	.1822
GMAP	.1084	**.1190**	.1098	.1080	.1080	.1170
P@5	.4529	.4559	.4500	.4500	.4559	**.4746**
P@10	.3882	**.3941**	.3897	.3853	.3853	.3824

Table: Sensibility test for BM25F. All this measures ranges from 0 to 1. *te* = *text*, *ti* = *title*, *in* = *inlinks*, *ob* = *obj*, *ty* = *type*, *all* = *allfields*

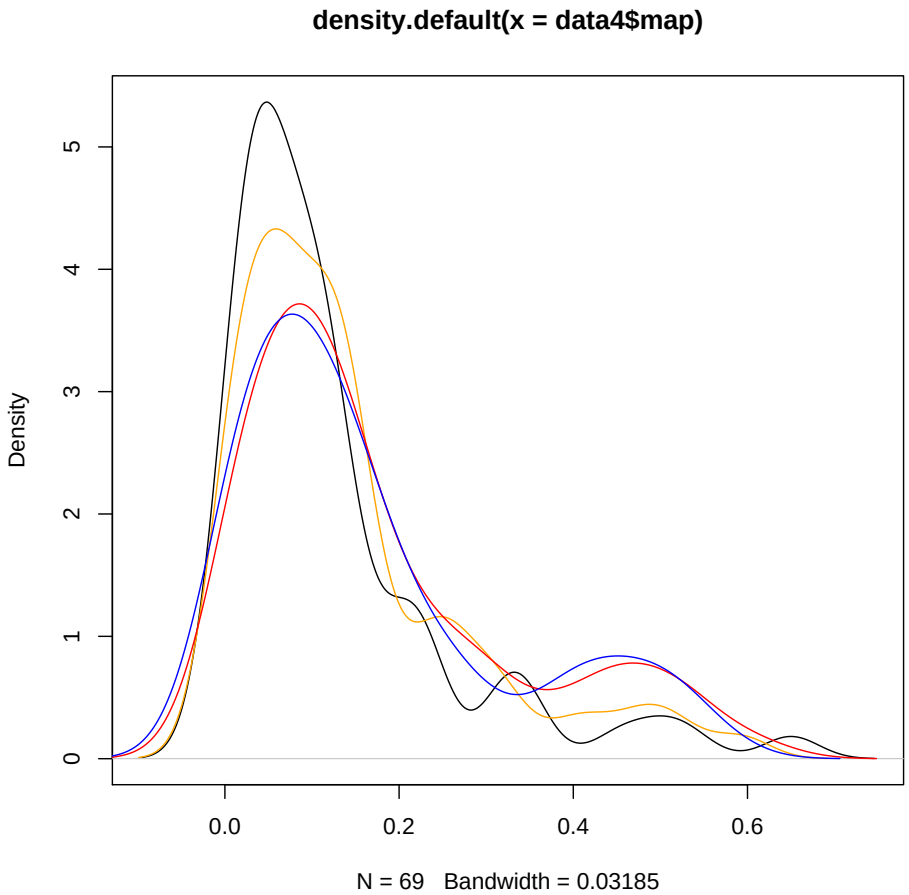

Figure: Density of the MAP values for different ranking approaches (BM25=blue, BM25F=red, Lucene=yellow, Lucene multifield=black)

Outline

Conclusions

- Lucene hurts the retrieval performance, while BM25F does not when we are working on structured information, which is very important for Semantic Search.
- IR ranking functions are not able to take profit from the semantic information contained in the fields with less text.

Future work

- It is necessary more work on how to adapt IR ranking functions to Semantic Search
- It is not trivial to use semantic information from the Web of data to improve search on the Web for keywords based retrieval
- It is important to identify what kind of information needs can be solved using semantic information

BM25F implementation for Lucene is available

Joaquín Pérez-Iglesias, José R. Pérez-Agüera, Víctor Fresno, Yuval Z. Feinstein: Integrating the Probabilistic Models BM25/BM25F into Lucene CoRR abs/0911.5046: (2009)
http://nlp.uned.es/ jperezi/Lucene-BM25/