Evaluation of IR Models

Anjana Tejaswini Kalava
50338176
anjanate@buffalo.edu

Abstract

The goal of this project is to implement, evaluate various IR models and to improve the search results for the query by tweaking the parameters, schema, query parsers and boosting parameters. By improving these results, we use the TREC_EVAL tool to improve the Mean Average Precision (MAP).

1. Introduction

In this project we make use of the twitter data that has been populated in three languages namely, English, German and Russian. This data has been indexed using Solr on 2 cores with the following similarity models:

- Vector Space Model (VSM)
- BM25 Model

The search results from these have been given as input to the Trev eval tool to get the MAP scores for the respective models.

2. Mean Average Precision (MAP)

Mean Average Precision is the mean of the average of the precision values obtained for the set of top k documents retrieved after search. It is basically the average area under the precision-recall curve for the specific set of queries. This MAP is most standard among the TREC community and is a good measure of stability and discrimination.

3. Models

3.1 Vector Space Model:

A Vector Space Model (VSM) is an algebraic model which retrieves the documents based on the cosine similarity between the query vector and the document vector. In this project to implement VSM we use the Classic Similarity Factory provided by Solr in the schema file.

<similarity class="solr.ClassicSimilarityFactory">

</similarity>

We add the above lines to the schema file and obtain a VSM.

3.2 BM25:

BM25 also known as the Okapi Best Matching 25 is a non-binary model that is supported by Solr and is the default similarity model in it. It is a probabilistic model that has two variables that can be tweaked b and k1. By default, these values are set to 0.75 and 1.2 respectively. In Solr we use the BM25 Similarity Factory to implement this model.

<similarity class="solr.BM25SimilarityFactory">

<str name="b">0.6</str>

<str name="k1">0.7</str>

</similarity>

4. Evaluation of IR Models

In each of the models the following changes have been done to increase the score of the search results and obtain a higher MAP value.

- Query parses have been used to boost the query. There are two parsers, dismax and edismax, of which edismax provided better results.

- Using edismax as the query parser, the field qf which is to increase or decrease the importance of the field has been used as follows

text_en+text_de+text_ru

- Classic tokenizer has been used in place of the standard tokenizer
- http, https, rt, t.co have been added to the stop words list for each language
- Synonyms have been added to the text file
- A copy field for each language has been generated to remove the stop words

4.1 VSM:

This model provided the MAP result as follows

```
num_rel                 all     225
num_rel_ret             all     119
map                     all     0.6828
gm_map                  all     0.6079
Rprec                   all     0.6970
```

4.2 BM25:

Apart from the above changes, in bm25 the parameters b and k1 have been tweaked to improve the MAP score. Below are some of the values used and their MAP scores produced accordingly.

b	k1	MAP
0.75	1.2	0.6969
0.6	1.3	0.7006
0.6	**0.7**	**0.7050**
0.8	2.0	0.6973

```
num_rel                 all     225
num_rel_ret             all     125
map                     all     0.7050
gm_map                  all     0.6327
Rprec                   all     0.7215
```

5. Result

The following are the MAP values obtained for BM25 AND VSM models:

Model	MAP
BM25	0.7050
VSM	0.6828

6. Conclusion:

As we can see from the above results we can conclude that the MAP value of BM25 is higher than that of the VSM model.

MAP(BM25) > MAP(VSM)

Referred Links:

https://lucene.apache.org/solr/guide/6_6/the-dismax-query-parser.html

https://lucene.apache.org/solr/guide/6_6/the-extended-dismax-query-parser.html

https://lucene.apache.org/solr/guide/7_5/other-schema-elements.html#OtherSchemaElements-Similarity

https://lucene.apache.org/solr/7_0_0/solr-core/org/apache/solr/search/similarities/package-summary.html

https://sease.io/2020/03/introducing-weighted-synonyms-in-apache-lucene.html

https://medium.com/@pablocastelnovo/if-they-match-i-want-them-to-be-always-first-boosting-documents-in-apache-solr-with-the-boost-362abd36476c