Project 3: Evaluation of IR Models

Abhishek S. Dalvi
UBITname:- adalvi
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
adalvi@buffalo.edu

1 Introduction

The goal of this project is to implement BM-25, Language Model and Divergence from Randomness IR models in Solr, evaluate the IR system and improve the search results based on by understanding the models, the implementation and the evaluation using TREC evaluation.

2 Implementation

All three IR models were implemented by manually changing the respective schema files using the similarity tags.

For BM-25; class solr.BM25SimilarityFactory is used which specifies two parameters k1 and b.

Figure 1: BM-25 Implementation

```
http://54.226.217.81:8983/solr/bmmodel/admin/file?wt=json&_=15727398327

<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO NOT EDIT -->
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.BM25SimilarityFactory">
    <str name="b">0.9</str>
    <str name="k1">0.5</str>
  </similarity>
  <fieldType name="_nest_path_" class="solr.NestPathField" maxCharsForDocValue
  <fieldType name="ancestor_path" class="solr.TextField">
    <analyzer type="index">
      <tokenizer class="solr.KeywordTokenizerFactory"/>
    </analyzer>
    <analyzer type="query">
      <tokenizer class="solr.PathHierarchyTokenizerFactory" delimiter="/"/>
```

For DFR; class solr.DFRSimilarityFactory is used with parameters “BasicModelG” plus “Bernoulli” first normalization ,“H2” second normalization.

Figure 2: DFR Implementation

```
http://54.226.217.81:8983/solr/IRF19P2_2/admin/file?wt=json&_=15727398327

<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO NOT EDIT -->
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.DFRSimilarityFactory">
    <str name="normalization">H2</str>
    <str name="afterEffect">B</str>
    <str name="basicModel">G</str>
  </similarity>
  <fieldType name="_nest_path_" class="solr.NestPathField" maxCharsForDocValues=
  <fieldType name="ancestor_path" class="solr.TextField">
    <analyzer type="index">
      <tokenizer class="solr.KeywordTokenizerFactory"/>
    </analyzer>
    <analyzer type="query">
```

For LM; class solr.LMDirichletSimilarityFactory is used with parameters with smoothing parameter mu.

Figure 3: LM Implementation

```
http://54.226.217.81:8983/solr/IRF19P2_3/admin/file?wt=json&_=157273983279

<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO NOT EDIT -->
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.LMDirichletSimilarityFactory">
    <str name="mu">2000.0</str>
  </similarity>
  <fieldType name="_nest_path_" class="solr.NestPathField" maxCharsForDocValues=
  <fieldType name="ancestor_path" class="solr.TextField">
    <analyzer type="index">
      <tokenizer class="solr.KeywordTokenizerFactory"/>
    </analyzer>
    <analyzer type="query">
```

3 Strategies to Improve Models

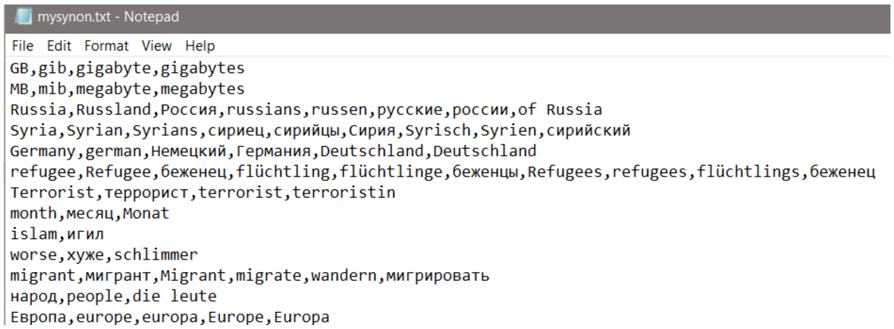
3.1 New “text” field

A field called ‘text’ is created which copies from text_en, text_ru, text_de, thus a common field which helps in searching. Operations like boosting can be performed on a single field where as in absence of this field, boosting will mentioned on all 3 fields.

3.2 Synonyms for multi lingual search

A custom **mysynon.txt** is created which contained translations of important terms(eg. refugee, Russia, Syria).The synon.txt is referred in both index and query analyzer in fields text_en, text_de, text_ru, text using solr.SynonymGraphFilterFactory class. solr.FlattenGraphFilterFactory is used and is needed for synonyms on the index.

Figure 3: mysynon.txt




```
GB,gib,gigabyte,gigabytes
MB,mib,megabyte,megabytes
Russia,Russland,Россия,russians,russen,русские,россии,of Russia
Syria,Syrian,Syrians,сириец,сирийцы,Сирия,Syrisch,Syrien,сирийский
Germany,german,Немецкий,Германия,Deutschland,Deutschland
refugee,Refugee,беженец,flüchtling,flüchtlinge,беженцы,Refugees,refugees,flüchtlings,беженец
Terrorist,террорист,terrorist,terroristin
month,месяц,Monat
islam,игил
worse,хуже,schlimmer
migrant,мигрант,Migrant,migrate,wandern,мигрировать
народ,people,die leute
Европа,europe,europa,Europe,Europa
```

Using synonyms gave a better score for all the models, therefore the important terms translations-synonyms were enough and synon.txt didn't needed to be increased in terms.

3.3 Slop queries

Slop queries using edismax parser can be included which searches for terms within the slop of the query terms i.e the displacement of words cannot be more than the mentioned slop (e.g. slop of 3 on 'Hello World' means 'Hello' and 'World' can only have a maximum term distance of 3).

The documents itself being small, slop didn't matter much, in fact reduced the accuracy (as tweet lengths are not as long as a book), therefore, it wasn't considered.

3.4 Boosting Camel Notations words (Proper Nouns)

Boost terms can be used for Proper nouns, therefore more weight and relevance will be given to documents having those terms. But again, tweets being small in size a very small boost should be kept (which will have no impact), as a high boost might give more relevance to a less relevant tweet just because the tweet has mentioned those Proper Nouns (E.g. of failed case: Tweets with hashtags "#Syria #Russia" more relevant even if it doesn't match the query context).

3.5** Tuning k1 and b in BM-25 model

Parameter k1 (Range:- more than 0, usually not more than 3) controls the non-linear term frequency normalization (saturation). Therefore, if a term appears frequently in a very long documents like books; then the term is not so relevant. Therefore, k1 should be set to a high value. On the other hand for short documents (tweets), if a term appears frequently in the document then it is probably important thus a small k1.

Parameter b (Range:- 0 - 1) controls to what degree document length normalizes term frequency values. Thus, a document which isn't a very specific should have a(news article) should have a larger b while a document very specific (Architecture specification) should have a smaller b.

Using this intuition various values were tried.

Table 1: BM-25 Tuning Parameter

k1	b	Map Score
1.2	0.75	0.6521
0.6	0.8	0.6609
0.9	0.8	0.6386
0.4	0.85	0.6447
0.3	0.9	0.6766
0.5	0.9	0.6840

The dataset being tweets a low k1 and high b works the best as tweets being small in size there isn't much saturation(k1) we have to consider and tweets being not exactly very specific in nature, a high b is considered.

4 Results

By tuning the models; the best MAP values for each model were found; which are shown in the images below.

MAP for BM-25 model:- 0.6840

runid	all	bmmodel
num_q	all	15
num_ret	all	280
num_rel	all	225
num_rel_ret	all	122
map	all	0.6840
gm_map	all	0.6175
Rprec	all	0.6699
bpref	all	0.6787

MAP for DFR model:- 0.6769

runid	all	DFR
num_q	all	15
num_ret	all	280
num_rel	all	225
num_rel_ret	all	121
map	all	0.6769
gm_map	all	0.6055
Rprec	all	0.6676
bpref	all	0.6759

MAP for Language model:- 0.6837

runid	all	LM
num_q	all	15
num_ret	all	280
num_rel	all	225
num_rel_ret	all	123
map	all	0.6837
gm_map	all	0.6104
Rprec	all	0.6754
bpref	all	0.6903