CSE 435/535 Information Retrieval (Fall 2016)
Project 3: Evaluation of IR models

Group 52

1)Neha Lalaso Jagtap (50206620)

2)Vidhi Jatin Shah (50207090)

Implementation of retrieval models with default parameters-

We are creating three cores, each for a model.
We are running each query for its predefined language in all of the three cores and taking top 20 results for each query. We will now feed this output to trec_eval to calculate MAP value for each model.

1) Best Matching (BM 25)-

1) The Best Matching (BM25) algorithm is a probabilistic Information Retrieval (IR) model.
2) Made core train_newcore for implementing BM model. Added similarity field in the schema.xml. Following are the screenshots of similarity class for default Boolean Model and corresponding MAP value.

Parameters:

1) k1 - Controls non-linear term frequency normalization (saturation).
2) b - Controls to what degree document length normalizes values.

Class -> public class BM25Similarity extends Similarity

```
<similarity class="solr.BM25SimilarityFactory">
    <float name="k1">1.2</float>
  <float name="b">0.75</float>
</similarity>
```

```
runid                   all     BM_25
num_q                   all     20
num_ret                 all     381
num_rel                 all     305
num_rel_ret             all     171
map                     all     0.6815
```

Schema.xml MAP value

2) Vector Space Model –

1) The VSM is an algebraic model used for Information Retrieval. It represent natural language document in a formal manner by the use of vectors in a multidimensional space.
2) We created a core called VSM_core to implement VSM model. This relevance model is implemented by adding solr.ClassicSimilarity factory in similarity subclass in the schema.xml file of VSM_core.
The following shows the implementation of Vector Space Model:

```
<similarity class="solr.ClassicSimilarityFactory">
```

```
runid                   all     VSM
num_q                   all     20
num_ret                 all     381
num_rel                 all     305
num_rel_ret             all     175
map                     all     0.6830
```

Schema.xml MAP value

3) DFR MODEL-

1) We created DFR_core to implement DFR model.
2) In order to use the DFR similarity in Solr, we add the solr.DFRSimilarityFactory class in our schema.xml file. An implementation of the DFR similarity is as follows:

Class -> public class DFRSimilarity extends SimilarityBase

```
<similarity class="solr.DFRSimilarityFactory">
  <str name="basicModel">G</str>
  <str name="afterEffect">B</str>
  <str name="normalization">H2</str>
  <float name="c">7</float>
 </similarity>
```

Schema.xml

```
runid           all     VSM
num_q           all     20
num_ret         all     380
num_rel         all     305
num_rel_ret     all     174
map             all     0.6886
```

MAP value

Observation –

	BM25	DFR	VSM
MAP values	0.6815	0.6886	0.6830

Improving MAP values.

I) Changing tuning Parameters of Similarity class.

Reason: To check if any parameters other than the default, improves the map value.

1) BM 25 model –

BM25 has two parameters: k1 and b.
Maximum and minimum scores for BM25similarity parameters are k1 [1.2, 2] and b[0.5,0.8]. Changing the values of k1 and b will change the scores.

A) We changed Tuning parameters of the BM model to k1=1.3 and b= 0.76

```
<similarity class="solr.BM25SimilarityFactory">
     <float name="k1">1.3</float>
   <float name="b">0.76</float>
</similarity>
```

Observation-
MAP value = 0.6824

B) We changed Tuning parameters of the BM model to k1=1.2 and b= 0.50

```
<similarity class="solr.BM25SimilarityFactory">
<float name="k1">1.2</float></similarity>
<float name="b">0.5</float>
</similarity>
```

Observations-
MAP value = 0.6820

2)DFRmodel-

DFR model has three parameters: BasicModel,AfterEffect,Normalization
BasicModel has values: Be,G,P,D,In,Ine,I(F)
AfterEffect has values: L,B
Normalization has values:H1,H2,H3,Z
We changed the tuning parameters of DFR model as follows.

A)

P: Poisson approximation of the binomial model,
L: Laplace's law of succession,
H2: Term frequency density inversely related to length

```
<similarity class="solr.DFRSimilarityFactory">
   <str name="basicModel">P</str>
 <str name="afterEffect">L</str>
 <str name="normalization">H2</str>
 <float name="c">7</float></similarity>
```

Observation –
MAP value - 0.6888

B)
I(F): Inverse term frequency,
B: Ratio of two Bernoulli processes ,
Z: Term frequency normalization provided by a Zipfian relation

```
<similarity class="solr.DFRSimilarityFactory">
   <str name="basicModel">I(F)</str>
 <str name="afterEffect">B</str>
 <str name="normalization">Z</str>
 <float name="c">7</float></similarity>
```

Observation –
MAP value - 0.6819

3)VSM Model –

VSM model has no parameters.

Conclusion –

By tuning parameters MAP value increases for BM25 and DFR model.

	BM25	DFR	VSM
MAP values	0.6824	0.6888	--

II) Stop Words filter removal-

Reason -Stop word filter are by default present in schema.xml. Removing the stop filter means that we would consider stop words as query terms which may increase recall. Removing stopwords is an important approach because some stopwords play a crucial role in meaning of query. Ex. Flight from Buffalo to Dubai. In this case removing the stopword filter will increase the rank of relevant document and hence will result an increase in map values.

```
<!--<filter class="solr.StopFilterFactory"
        ignoreCase="true"
        words="lang/stopwords_en.txt"
        /> -->
```

Observations-

	BM25	DFR	VSM
MAP values	0.6815	0.6889	0.6828

Conclusion –
After removing stop word filter MAP value is same for BM25 model.MAP value increases for VSM model and reduces for DFR model.

III) Stem Filter removal -

Reason - When we use stemming filter not all the words can be stemmed to their right root word. Also, stemming can't relate words which have different forms based on grammatical constructs like is, am, be, all represent the same root verb, be. But stemming can't prune them to the common form. The word better should be resolved to good, but stemmers would fail to do that. Hence use of stemmers can give the false positive.

MAP value.

text_en	`solr.PorterStemFilterFactory`
text_ru	`solr.SnowballPorterFilterFactory`
text_de	`solr.GermanLightStemFilterFactory`

Implementing in solr –

Comment the filter class in schema.xml that applies stemmer.

```
<!--<filter class="solr.PorterStemFilterFactory"/> -->
```

Observations –

	BM25	DFR	VSM
MAP values	0.6242	0.6259	0.6022

Conclusion -
Because of removing stemmer filter, recall reduces. Hence MAP value is reduced in all the three models.

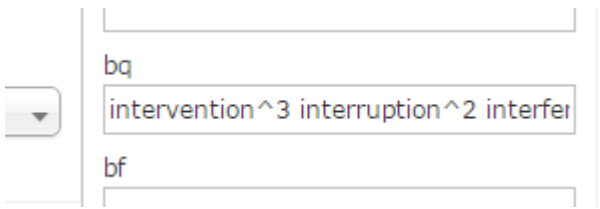
IV) Adding Synonyms-

Reason – Adding synonyms results into query expansion. Hence solr will retrieve more relevant tweets and tweet with synonym will get higher score. Therefore MAP value will go up.
Additionally we boosted the query in solr by giving more weight to original query term than synonym.

Implementation in Solr-
1) We add synonyms in the synonyms.txt file in conf folder of a core.
Ex.

```
intervention,interruption,inteference
crisis,disaster,calamity
```

Synonyms.txt



solr UI

Observation –

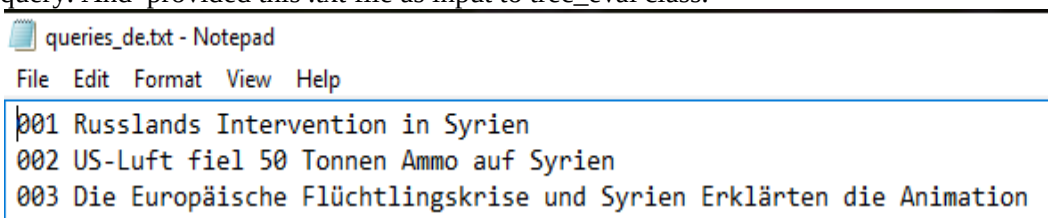
	BM25	DFR	VSM
MAP values	0.6830	0.6892	0.6835

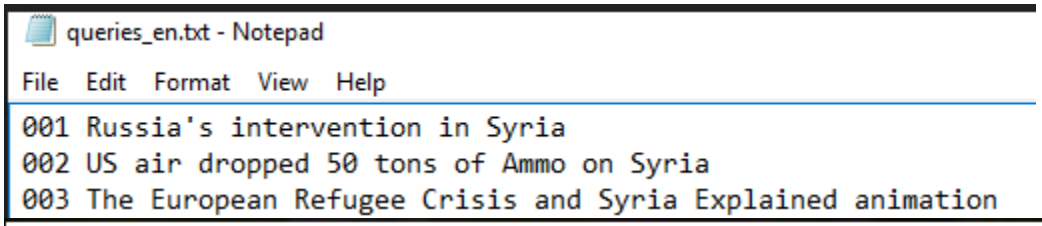
Conclusion –
MAP value increases for all the three models after using Synonyms.

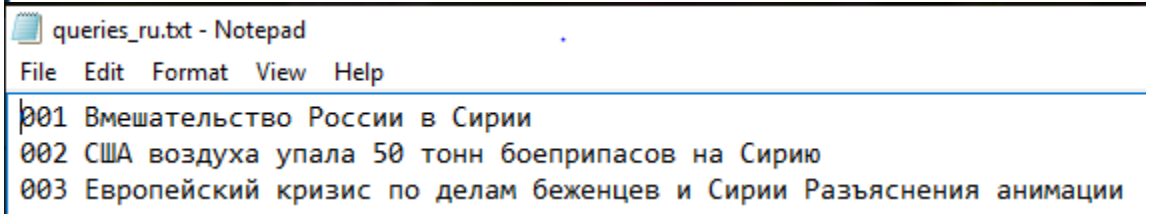
V) Cross language query processing –

Reason - Translating each query in all three languages will expand our search domain in all three language tweets. Ex. It's possible that "Russia's intervention in Syria" has relevant tweets in Russian language. Thus this will increase the score of relevant tweets in all languages and hence MAP value will improve.

Implementation in Solr-
We translated given 20 queries in all three languages. So now we have each query in three different languages. We ran python script for all queries in all three languages and fetched top 20 results for each query. And provided this .txt file as input to trec_eval class.







Observation –

	BM25	DFR	VSM
MAP values	0.6906	0.7013	0.6798

Conclusion –
MAP value should always increases for cross query processing. But as we are considering only top 20 documents, MAP value increases for only DFR model.

VI) Query Boosting –

A)Field Boosting-

In Solr UI under edismax, we can add field in **qf**, which we want to boost. If solr finds query terms in that field that tweet gets boosted.
Ex.
For the query "#HumanRights" we can keep df (Default field) as tweet_hashtags. Then tweets having HumanRights in their tweet_hashtags gets higher score.

B)Term Boosting –

If we want to boost the particular terms from query we put those terms in **bq**. We can also assign weights to those terms which are more important in the query.
Ex. For the query "Human rights of refugees" then refugee is an important term in query then if we set bq = "refugees" then documents having refugee gets higher score.

Observation –
After booting MAP values should ideally increase but we have restriction of using only 20 documents. So there is no considerable increase in MAP value.

SUMMARY –

	BM25	DFR	VSM
DEFAULT	0.6815	0.6886	0.6830
TUNING SIMILARITY CLASS PARAMETERS	0.6824	0.6888	
REMOVING STOPWORD FILTER	0.6815	0.6889	0.6828
REMOVING STEM FILTER	0.6242	0.6259	0.6022
ADDING SYNONYMNS	0.6830	0.6892	0.6835
CROSS LANGUAGE QUERY PROCESSING	0.6906	0.7013	0.6798

Conclusion –

BM25 Model:
By using tuning parameters 1.3 and 0.76, Stopword filter, stem filter, Synonyms , MAP value increases.
DFR Model:
By using Cross Language Query Processing, Stem filter, synonyms and removing Stopword filter, MAP value increases for tuning parameters P,L,H2.
VSM:
By using default Similarity class, Stopword filter, Stem Filter and adding synonym, increases MAP value for VSM.

Reference-

https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html
https://wiki.apache.org/solr/SolrRelevancyFAQ
https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/BM25Similarity.html