Project 3: Evaluation of IR Models

Astrid Yvette Gomes
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
UBIT Name: astridyv
UBIT Id:50317492
astridyv@buffalo.edu

Abstract

In this project we have implemented various IR models, evaluated the IR system and improved the search results based on the implementation and the evaluation. The input was twitter data in three languages - English, German and Russian, 15 sample training queries and the corresponding relevance judgements. We implemented the following three IR models: (i) Language Model, (ii) BM25 and (iii) Divergence from Randomness (DFR) Model. The results from these three sets are then evaluated using the Trec_eval program. Based on the evaluation results, we improved the performance in terms of Mean Average Precision (MAP).

Introduction

When a user submits a query to a search engine the first thing it must do is to determine which pages in the index are related to the query and which are not. So a search engine must compute a number of relevance scores using different page elements and weight them all to arrive to a final score. We make use of models to compute a similarity score.

Dataset

We Given is a set of 3440 tweets which have been retrieved. The tweets are in three languages English, German and Russian. Corresponding to the language of the tweet the text_en, text_de and text_ru fields have values of the tweet_data.

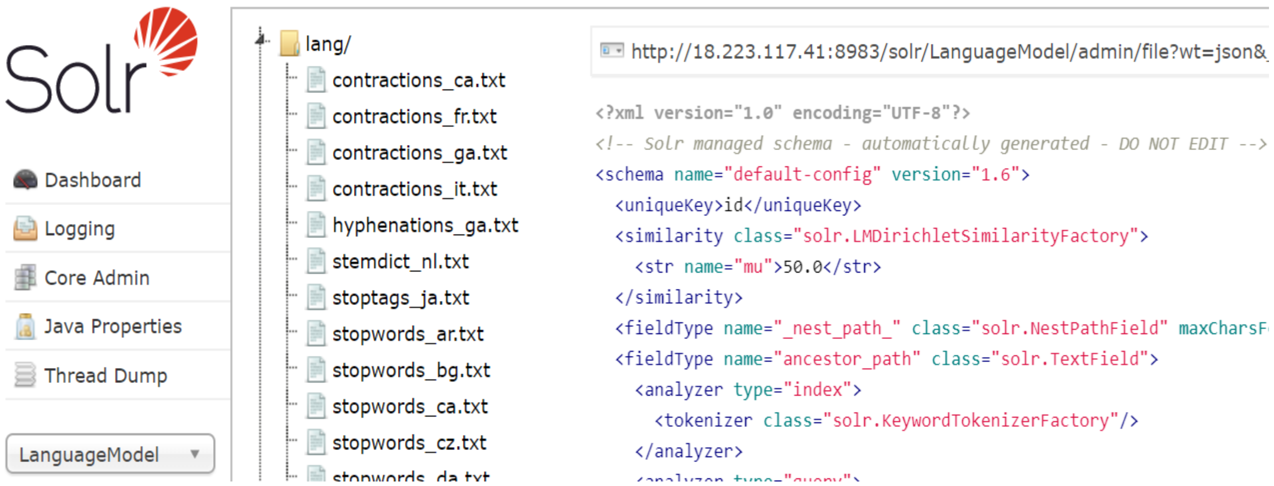
Preprocessing:

The input dataset was loaded into solr. Three different cores were created to implement each IR model. The schema and config file is updated to get best Mean Average Precision (MAP) values We shall now look at the three IR models implemented.

1 Language Model

- **Implementation of Language Model:**

A language model is a probability distribution over sequences of words. The probability of a word depends on the previous n-words. This model is also known as bag of words model.
We have Implemented the language model in solr using the '**LMDirichletSimilarityFactory**' factory. Parameters are mu (float) which is a smoothing parameter. The default value is 2000. Below is a screenshot of its implementation in solr.



```
<?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">50.0</str>
  </similarity>
  <fieldType name="_nest_path_" class="solr.NestPathField" maxCharsF
  <fieldType name="ancestor_path" class="solr.TextField">
    <analyzer type="index">
      <tokenizer class="solr.KeywordTokenizerFactory"/>
    </analyzer>
```

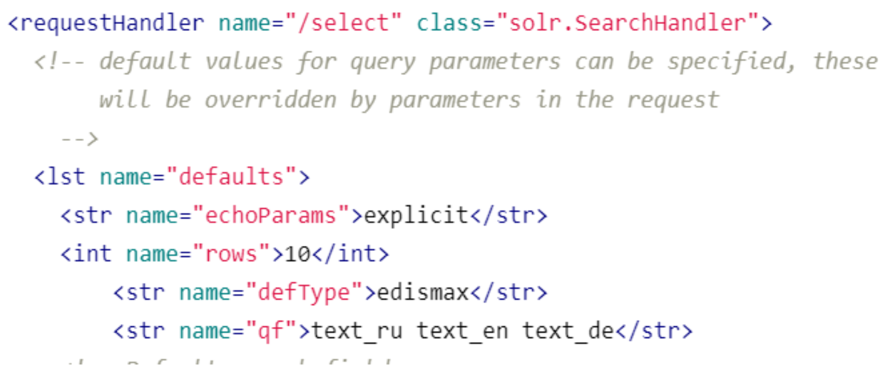
- **Strategies to improve Map score:**

  Since the default mu is given to be mu=2000, we first run for the default value without making any changes in config file. The MAP value is found to be 0.2515. We can then try tweaking the mu parameters to get best possible MAP value. If we decrease mu=1000 the MAP is still 0.2515. We can see there has been no change. Then substituted the mu=1 and the MAP=0.2534
  There was an increase ,but a very slight and almost negligible increase. So to improve the performance of our system we can use query parsers. To increase performance I have used the 'Edismax' query parser with query fields as text_en. text_de, text_ru. Below table shows Mu and MAP values without query parsers.

MU	MAP(No query parser)
2000	0.2515
1000	0.2515
1	0.2534

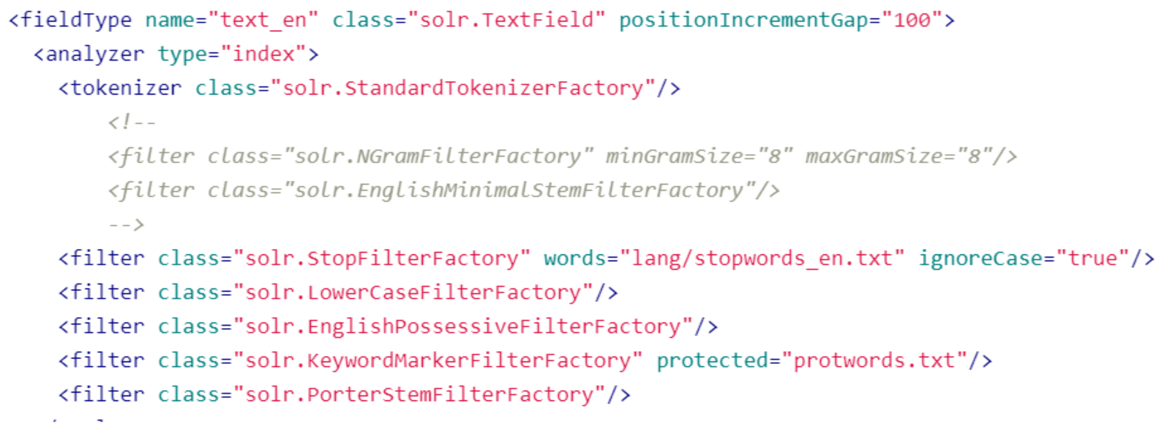
Add the query parser in the config file.

Screenshot of changes in schema config file.



```
<requestHandler name="/select" class="solr.SearchHandler">
  <!-- default values for query parameters can be specified, these
       will be overridden by parameters in the request
    -->
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <int name="rows">10</int>
        <str name="defType">edismax</str>
        <str name="qf">text_ru text_en text_de</str>
```

In managed schema added the EnglishPossessiveFilterfactory and PorterStemFilterFactory to boost performance.



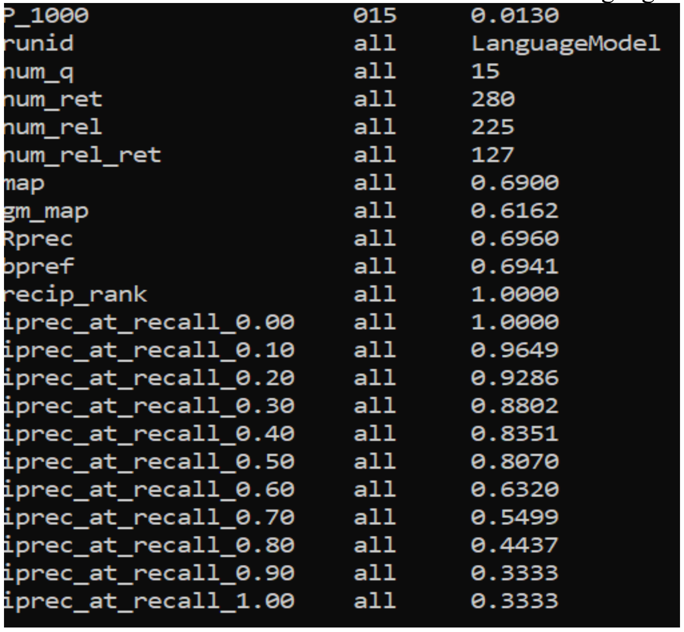
```
<fieldType name="text_en" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
        <!--
        <filter class="solr.NGramFilterFactory" minGramSize="8" maxGramSize="8"/>
        <filter class="solr.EnglishMinimalStemFilterFactory"/>
        -->
    <filter class="solr.StopFilterFactory" words="lang/stopwords_en.txt" ignoreCase="true"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.EnglishPossessiveFilterFactory"/>
    <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
    <filter class="solr.PorterStemFilterFactory"/>
```

Now rerun with changes in schema config and schema file. The mu value was kept at a default of 2000 ,the MAP= 0.6299. This was a huge increase in MAP values. We can vary the Mu parameter and try and get best possible MAP. The various methods which have been used are mentioned in the table below. For the MU=50 we get the MeanAveragePrecision=0.6900

MU	Mean Average Precision(With query parser)
2000	0.6299
1500	0.6312
1000	0.6346
50	0.69

Result:

Below is a screenshot of the best MAP value for LanguageModel

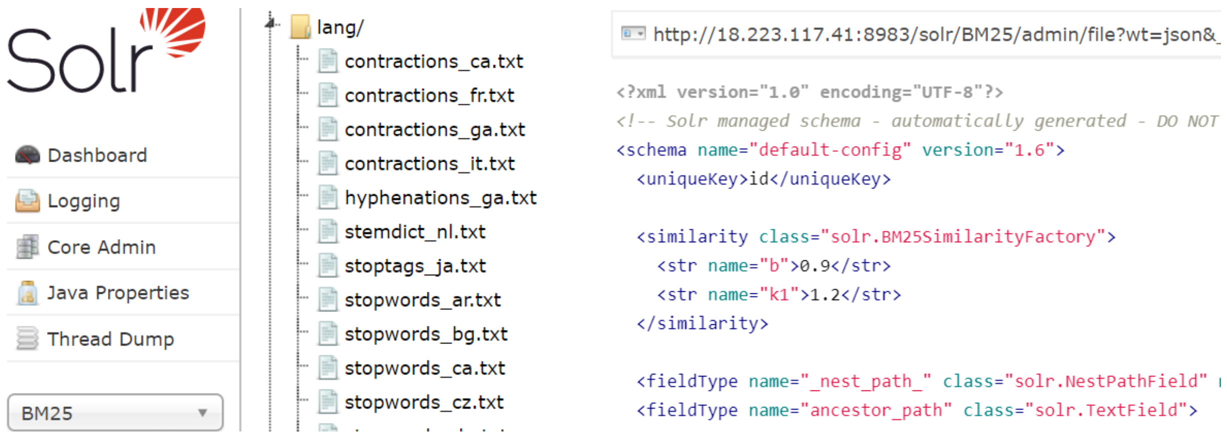


```
P_1000                  015     0.0130
runid                   all     LanguageModel
num_q                   all     15
num_ret                 all     280
num_rel                 all     225
num_rel_ret             all     127
map                     all     0.6900
gm_map                  all     0.6162
Rprec                   all     0.6960
bpref                   all     0.6941
recip_rank              all     1.0000
iprec_at_recall_0.00    all     1.0000
iprec_at_recall_0.10    all     0.9649
iprec_at_recall_0.20    all     0.9286
iprec_at_recall_0.30    all     0.8802
iprec_at_recall_0.40    all     0.8351
iprec_at_recall_0.50    all     0.8070
iprec_at_recall_0.60    all     0.6320
iprec_at_recall_0.70    all     0.5499
iprec_at_recall_0.80    all     0.4437
iprec_at_recall_0.90    all     0.3333
iprec_at_recall_1.00    all     0.3333
```

2 BM25 Model

- **Implementation of BM25 Model:**

Implemented the BM25 model in solr using the '**BM25SimilarityFactory**' factory. Parameters are K1 and b where k1 - Controls non-linear term frequency normalization (saturation) and b - controls to what degree document length normalizes the term frequency (tf) values. The value of b can be between 0 and 1 only. The default values of k1 is 1.2 and b=0.75. Below is a screenshot of the implementation of the BM25 model. It can be noted that the default Information retrieval model of a system is the BM25 model.



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO NOT
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>

  <similarity class="solr.BM25SimilarityFactory">
    <str name="b">0.9</str>
    <str name="k1">1.2</str>
  </similarity>

  <fieldType name="_nest_path_" class="solr.NestPathField"
  <fieldType name="ancestor_path" class="solr.TextField">
```

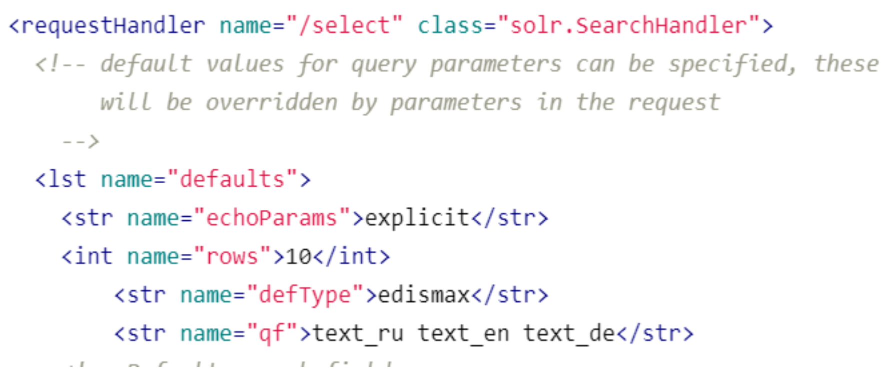
- **Strategies to improve Map score:**

Since the default value for the parameters are b=0.75 and k1=1.2 ,we execute for the default value without making any changes in config file . The MAP value found to be 0.2469. Since b can only vary between 0 and 1 we can try taking both these values for b.
We then tweak the b and k1 parameters and substituted b=1 and k1=2 the MAP value decreased to 0.2461. Then we can substitute b=0 and k1=1 ,and the MAP=0.2374, there was an increase ,but a very slight and almost negligible increase. The below table shows the b and k1 and MAP value when query parsers have not been used. So we can try boosting using query parsers for increasing performance and one of the ways is using the 'Edismax' query parser with query fields as text_en. text_de, text_ru. Below table shows 'b' and 'k1'and MAP values without query parsers.

b	k1	MAP(No Query Parsers)
0.75	1.2	0.2469
1	2	0.2461
0	1	0.2374

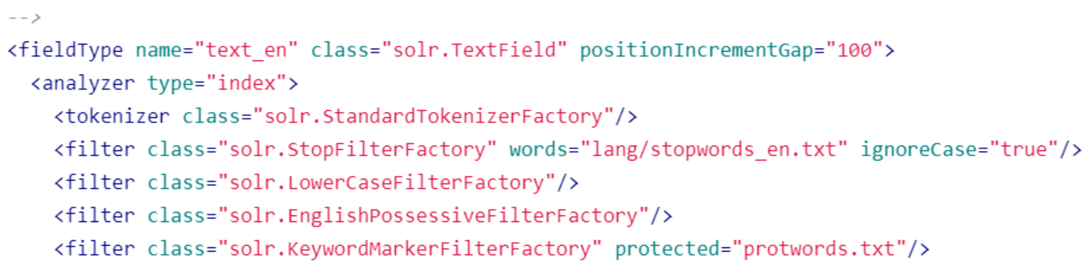
Add the query parser in the config file.

Screenshot of changes in schema config file.



```
<requestHandler name="/select" class="solr.SearchHandler">
  <!-- default values for query parameters can be specified, these
       will be overridden by parameters in the request
    -->
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <int name="rows">10</int>
        <str name="defType">edismax</str>
        <str name="qf">text_ru text_en text_de</str>
```

In managed schema added the EnglishPossessiveFilterfactory and PorterStemFilterFactory to boost performance.



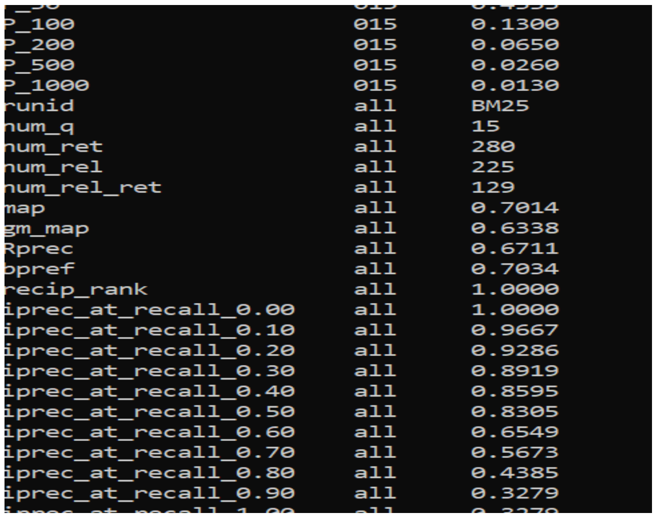
```
-->
<fieldType name="text_en" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory" words="lang/stopwords_en.txt" ignoreCase="true"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.EnglishPossessiveFilterFactory"/>
    <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
```

Once we have made changes in the config file and schema we can execute and calculate the MAP results obtained from Trec. The default values of the system is b=0.75 and k1=1.2, we calculate the MAP= 0.6935. This was a huge increase in MAP values. We can then tweak the b and k1 parameter and try for various values to get the best MAP. Some of the values are listed in the table below. Since the values of b range from 0 to 1, we can try taking these values. We see the MAP increase when we increase b and k1, but on increasing the k1 further and decreasing b the MAP decreases. Now lets keep k1 constant and increase b, we get the MAP=0.7014. The below table specifies these results.

b	k1	MAP(With query parser)
0.75	1.2	0.6935
1	1.9	0.6973
0	1	0.6866
0.9	1.2	0.7014

Results:

Below is a screenshot of the best MAP value for BM25

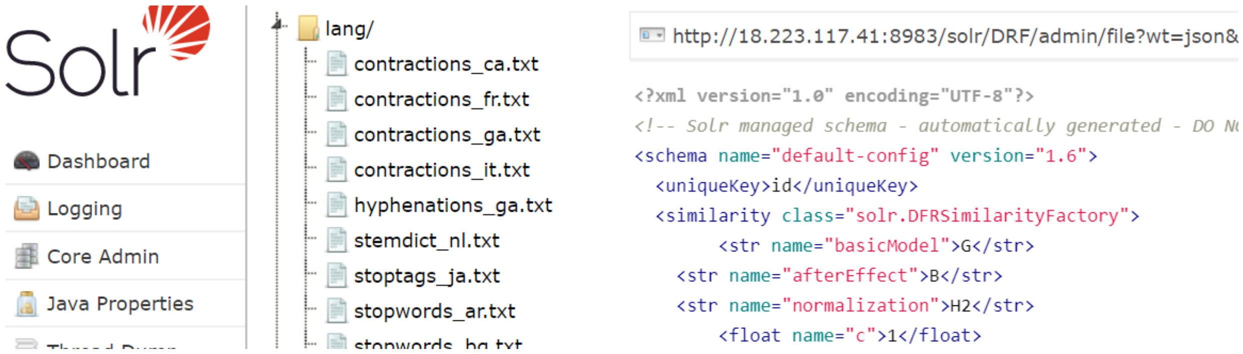


```
P_100                   015     0.1300
P_200                   015     0.0650
P_500                   015     0.0260
P_1000                  015     0.0130
runid                   all     BM25
num_q                   all     15
num_ret                 all     280
num_rel                 all     225
num_rel_ret             all     129
map                     all     0.7014
gm_map                  all     0.6338
Rprec                   all     0.6711
bpref                   all     0.7034
recip_rank              all     1.0000
iprec_at_recall_0.00    all     1.0000
iprec_at_recall_0.10    all     0.9667
iprec_at_recall_0.20    all     0.9286
iprec_at_recall_0.30    all     0.8919
iprec_at_recall_0.40    all     0.8595
iprec_at_recall_0.50    all     0.8305
iprec_at_recall_0.60    all     0.6549
iprec_at_recall_0.70    all     0.5673
iprec_at_recall_0.80    all     0.4385
iprec_at_recall_0.90    all     0.3279
```

3 Divergence from Randomness (DFR) Model

- **Implementation of DFR Model:**

Implemented the DFR model in solr using the '**DFRSimilarityFactory**' factory. Parameters are the basic model , afterEffect and Normalization. The basic mode is the basic model of information content. The aftereffect is First normalization of information gain and normalization is Second length normalization. The parameter c (float) is hyper-parameter that controls the term frequency normalization with respect to the document length. The default is 1



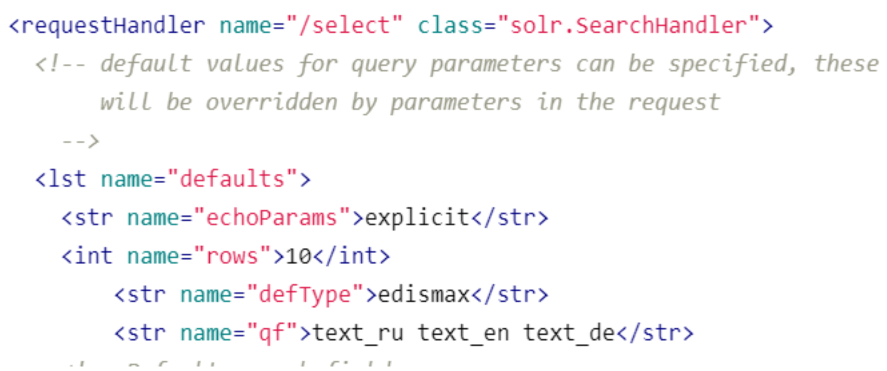
http://18.223.117.41:8983/solr/DRF/admin/file?wt=json&

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO N
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.DFRSimilarityFactory">
        <str name="basicModel">G</str>
    <str name="afterEffect">B</str>
    <str name="normalization">H2</str>
        <float name="c">1</float>
```

- **Strategies to improve Map score:**

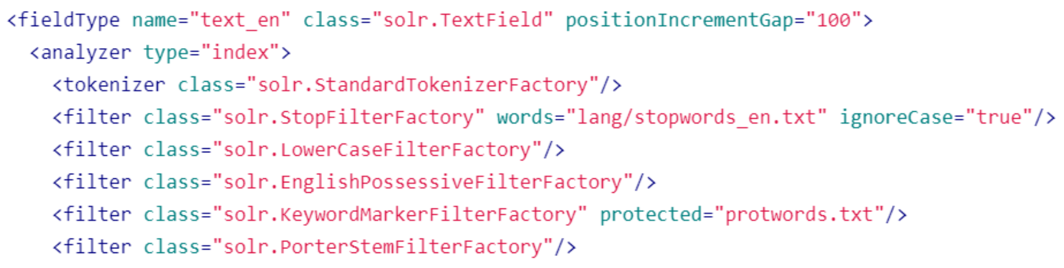
If we execute for the default values without changing solr config file.,the MAP =0.2511. To increase the performance of the system we can add query parsers.

Add the query parser in the config file.
Screenshot of changes in schema config file.



```
<requestHandler name="/select" class="solr.SearchHandler">
  <!-- default values for query parameters can be specified, these
       will be overridden by parameters in the request
    -->
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <int name="rows">10</int>
        <str name="defType">edismax</str>
        <str name="qf">text_ru text_en text_de</str>
```

In managed schema added the EnglishPossessiveFilterfactory and PorterStemFilterFactory to boost performance.

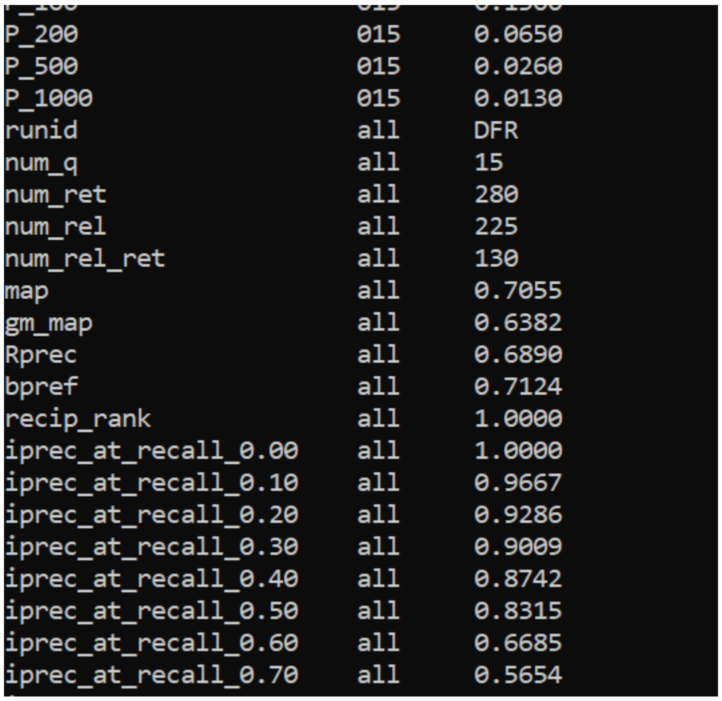


```
<fieldType name="text_en" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory" words="lang/stopwords_en.txt" ignoreCase="true"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.EnglishPossessiveFilterFactory"/>
    <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
    <filter class="solr.PorterStemFilterFactory"/>
```

We can now execute and check the performance of the system with the query parsers. Let us try and improve the Mean Average Precision by changing the parameter ‘c’ value. Let us take it to be 7 we calculate from the TREC evaluation results the MAP=0.7025. Let us increase it to c=9 we get a decrease MAP value 0.7022. Let us decrease the value of c to get better MAP. We see from below table we get the best results for c=1 which is the default value of the system.

c	MAP
7	0.7025
9	0.7022
2	0.7047
1	0.7055

Result:
Below is a screenshot of the best MAP value for DFR



```
P_200                   015     0.0650
P_500                   015     0.0260
P_1000                  015     0.0130
runid                   all     DFR
num_q                   all     15
num_ret                 all     280
num_rel                 all     225
num_rel_ret             all     130
map                     all     0.7055
gm_map                  all     0.6382
Rprec                   all     0.6890
bpref                   all     0.7124
recip_rank              all     1.0000
iprec_at_recall_0.00    all     1.0000
iprec_at_recall_0.10    all     0.9667
iprec_at_recall_0.20    all     0.9286
iprec_at_recall_0.30    all     0.9009
iprec_at_recall_0.40    all     0.8742
iprec_at_recall_0.50    all     0.8315
iprec_at_recall_0.60    all     0.6685
iprec_at_recall_0.70    all     0.5654
```

Conclusion

We have thus successfully implemented these three IR models the Language Model, BM25 and the Divergence from Randomness Model and created a system returned the best matched results. To calculate the Mean Average Precision(MAP) we have successfully used TREC to evaluate the results.

References:

https://moz.com/blog/determining-relevance-how-similarity-is-scored