Evaluation of IR Models

Sanidhya Chopde
Department of Computer Science
University at Buffalo
Buffalo, NY 14260
schopde@buffalo.edu

Abstract

The aim of this project is to implement different IR, models namely BM25 model, Divergence from Randomness (DFR) Model and Language Model, evaluate them and improve the search results based on the understanding of the models.

1 Introduction

In this project, twitter data is given in three languages - English, German and Russian. We need to index the given twitter data using Solr and implement the following three IR models: (i) Language Model, (ii) BM25 and (iii) Divergence from Randomness (DFR) Model. We then will evaluate results from these three sets using the Trec_eval program. Based on the evaluation results, we will try to improve the performance of the models in terms of Mean Average Precision (MAP).

2 Dataset Definition

The data to be used for this project is Twitter data saved in json format, training_tweet.json. Three languages are included - English (text_en), German (text_de) and Russian (text_ru). training_tweet.json: This file contains tweets (approximately 3,500) with some fields extracted from raw data.

3 Implementing IR Models

The IR models here are implemented using the similarity module. A similarity module defines how matching documents are scored. Configuring a custom similarity is considered an expert feature and the built-in similarities are most likely sufficient as is described in similarity. Most existing or custom Similarities have configuration options which can be configured via the index settings.

3.1 Language Model

In the LM Dirichlet similarity there's an option of fine tuning the 'mu' parameter which is by default set to 2000. The scoring formula in the paper assigns negative scores to terms that have fewer occurrences than predicted by the language model, which is illegal to Lucene, so such terms get a score of 0. I have chosen the value of 'mu' as 10 to tune my model.

```
<similarity class="solr.LMDirichletSimilarityFactory">
        <float name="mu">10</float>
</similarity>
```

Figure 1. Similarity module for Language Model

Following are the values I tried for ‘mu’ to tune the model:

mu = 2000			mu = 1500			mu = 10		
map	all	0.6468	map	all	0.6474	map	all	0.7180
gm_map	all	0.5631	gm_map	all	0.5642	gm_map	all	0.6435
Rprec	all	0.6370	Rprec	all	0.6370	Rprec	all	0.7349
bpref	all	0.6798	bpref	all	0.6798	bpref	all	0.7412
recip_rank	all	1.0000	recip_rank	all	1.0000	recip_rank	all	1.0000
iprec_at_recall_0.00	all	1.0000	iprec_at_recall_0.00	all	1.0000	iprec_at_recall_0.00	all	1.0000
iprec_at_recall_0.10	all	0.9733	iprec_at_recall_0.10	all	0.9733	iprec_at_recall_0.10	all	0.9917
iprec_at_recall_0.20	all	0.8923	iprec_at_recall_0.20	all	0.8923	iprec_at_recall_0.20	all	0.9250
iprec_at_recall_0.30	all	0.8042	iprec_at_recall_0.30	all	0.8045	iprec_at_recall_0.30	all	0.8798
iprec_at_recall_0.40	all	0.7240	iprec_at_recall_0.40	all	0.7244	iprec_at_recall_0.40	all	0.8064
iprec_at_recall_0.50	all	0.7194	iprec_at_recall_0.50	all	0.7244	iprec_at_recall_0.50	all	0.7707
iprec_at_recall_0.60	all	0.6139	iprec_at_recall_0.60	all	0.6139	iprec_at_recall_0.60	all	0.7390
iprec_at_recall_0.70	all	0.5445	iprec_at_recall_0.70	all	0.5453	iprec_at_recall_0.70	all	0.6239
iprec_at_recall_0.80	all	0.3831	iprec_at_recall_0.80	all	0.3831	iprec_at_recall_0.80	all	0.4356
iprec_at_recall_0.90	all	0.3534	iprec_at_recall_0.90	all	0.3534	iprec_at_recall_0.90	all	0.3925
iprec_at_recall_1.00	all	0.3233	iprec_at_recall_1.00	all	0.3233	iprec_at_recall_1.00	all	0.3633
P_5	all	0.7600	P_5	all	0.7600	P_5	all	0.8400
P_10	all	0.5667	P_10	all	0.5667	P_10	all	0.6667
P_15	all	0.4622	P_15	all	0.4667	P_15	all	0.5111
P_20	all	0.3800	P_20	all	0.3833	P_20	all	0.4200
P_30	all	0.2733	P_30	all	0.2711	P_30	all	0.2889
P_100	all	0.0873	P_100	all	0.0873	P_100	all	0.0880
P_200	all	0.0437	P_200	all	0.0437	P_200	all	0.0440
P_500	all	0.0175	P_500	all	0.0175	P_500	all	0.0176
P_1000	all	0.0087	P_1000	all	0.0087	P_1000	all	0.0088

Figure 2: Comparison of various mu values

By reducing the value of ‘mu’ the MAP value seemed to be increasing and hence I tried various values of mu starting at 2000 and kept on reducing it till I got a satisfactory result.

3.2 BM25 Model

To implement BM25 model a TF/IDF based similarity module is used that has built-in tf normalization and is supposed to work better for short fields. This similarity has the following options:

k1: Controls non-linear term frequency normalization. The default value is 1.2.
b: Controls to what degree document length normalizes tf values. The default value is 0.75.

I have used the following k1 & b values to tune this model:

```
<similarity class="solr.BM25SimilarityFactory">
    <float name="k1">1.32</float>
    <float name="b">0.77</float>
  </similarity>
```

Figure 3. Similarity module for BM25 Model

Following are the k1 and b1 model values I tried to tune the model:

k1: 1.9, b: 0.97			k1: 1.56, b: 0.81			k1: 1.32, b: 0.77		
map	all	0.7194	map	all	0.7207	map	all	0.7213
gm_map	all	0.6465	gm_map	all	0.6480	gm_map	all	0.6493
Rprec	all	0.7203	Rprec	all	0.7094	Rprec	all	0.7060
bpref	all	0.7458	bpref	all	0.7459	bpref	all	0.7455
recip_rank	all	1.0000	recip_rank	all	1.0000	recip_rank	all	1.0000
iprec_at_recall_0.00	all	1.0000	iprec_at_recall_0.00	all	1.0000	iprec_at_recall_0.00	all	1.0000
iprec_at_recall_0.10	all	1.0000	iprec_at_recall_0.10	all	1.0000	iprec_at_recall_0.10	all	1.0000
iprec_at_recall_0.20	all	0.9250	iprec_at_recall_0.20	all	0.9250	iprec_at_recall_0.20	all	0.9250
iprec_at_recall_0.30	all	0.8837	iprec_at_recall_0.30	all	0.8833	iprec_at_recall_0.30	all	0.8833
iprec_at_recall_0.40	all	0.7942	iprec_at_recall_0.40	all	0.7917	iprec_at_recall_0.40	all	0.8085
iprec_at_recall_0.50	all	0.7892	iprec_at_recall_0.50	all	0.7880	iprec_at_recall_0.50	all	0.7873
iprec_at_recall_0.60	all	0.7329	iprec_at_recall_0.60	all	0.7373	iprec_at_recall_0.60	all	0.7294
iprec_at_recall_0.70	all	0.5851	iprec_at_recall_0.70	all	0.6228	iprec_at_recall_0.70	all	0.6216
iprec_at_recall_0.80	all	0.4381	iprec_at_recall_0.80	all	0.4381	iprec_at_recall_0.80	all	0.4381
iprec_at_recall_0.90	all	0.3978	iprec_at_recall_0.90	all	0.3978	iprec_at_recall_0.90	all	0.3978
iprec_at_recall_1.00	all	0.3686	iprec_at_recall_1.00	all	0.3686	iprec_at_recall_1.00	all	0.3686
P_5	all	0.8667	P_5	all	0.8667	P_5	all	0.8667
P_10	all	0.6733	P_10	all	0.6733	P_10	all	0.6800
P_15	all	0.5244	P_15	all	0.5244	P_15	all	0.5156
P_20	all	0.4133	P_20	all	0.4133	P_20	all	0.4067
P_30	all	0.2844	P_30	all	0.2867	P_30	all	0.2889
P_100	all	0.0873	P_100	all	0.0880	P_100	all	0.0887
P_200	all	0.0437	P_200	all	0.0440	P_200	all	0.0443
P_500	all	0.0175	P_500	all	0.0176	P_500	all	0.0177
P_1000	all	0.0087	P_1000	all	0.0088	P_1000	all	0.0089

Figure 4: Comparison of various k1 & b values

By reducing the values of k1 & b the MAP values seemed to be increasing and hence I tried with several values of k1 & b gradually reducing them to tune the model.

3.3 Divergence from Randomness (DFR) Model

To implement DFR model a similarity module that implements the divergence from randomness framework has been used. This similarity has the following options:

basic_model: possible values g, if, ine and in.
lambda: possible values df and ttf.
normalization: same as in DFR similarity.

I have used the following basic_model, lambda & normalization values to tune this model:

```
<similarity class="solr.DFRSimilarityFactory">
  <str name="basicModel">G</str>
  <str name="afterEffect">B</str>
  <str name="normalization">H2</str>
  <float name="c">3</float>
</similarity>
```

Figure 5. Similarity module for DFR Model

Following are the values of the options that I tried to tune this model:

basicModel: G, aftereffect: B, normalization: H2, value: 7	basicModel: G, aftereffect: B, normalization: H2, value: 5	basicModel: G, aftereffect: B, normalization: H2, value: 3
map all 0.7136	map all 0.7209	map all 0.7230
gm_map all 0.6376	gm_map all 0.6467	gm_map all 0.6493
Rprec all 0.7131	Rprec all 0.7131	Rprec all 0.7129
bpref all 0.7430	bpref all 0.7486	bpref all 0.7514
recip_rank all 1.0000	recip_rank all 1.0000	recip_rank all 1.0000
iprec_at_recall_0.00 all 1.0000	iprec_at_recall_0.00 all 1.0000	iprec_at_recall_0.00 all 1.0000
iprec_at_recall_0.10 all 0.9762	iprec_at_recall_0.10 all 0.9800	iprec_at_recall_0.10 all 0.9917
iprec_at_recall_0.20 all 0.9200	iprec_at_recall_0.20 all 0.9222	iprec_at_recall_0.20 all 0.9238
iprec_at_recall_0.30 all 0.8705	iprec_at_recall_0.30 all 0.8705	iprec_at_recall_0.30 all 0.8821
iprec_at_recall_0.40 all 0.8088	iprec_at_recall_0.40 all 0.8149	iprec_at_recall_0.40 all 0.8051
iprec_at_recall_0.50 all 0.7763	iprec_at_recall_0.50 all 0.7888	iprec_at_recall_0.50 all 0.7854
iprec_at_recall_0.60 all 0.7300	iprec_at_recall_0.60 all 0.7323	iprec_at_recall_0.60 all 0.7389
iprec_at_recall_0.70 all 0.5949	iprec_at_recall_0.70 all 0.6252	iprec_at_recall_0.70 all 0.6286
iprec_at_recall_0.80 all 0.4410	iprec_at_recall_0.80 all 0.4410	iprec_at_recall_0.80 all 0.4410
iprec_at_recall_0.90 all 0.3978	iprec_at_recall_0.90 all 0.3978	iprec_at_recall_0.90 all 0.3978
iprec_at_recall_1.00 all 0.3686	iprec_at_recall_1.00 all 0.3686	iprec_at_recall_1.00 all 0.3686
P_5 all 0.8267	P_5 all 0.8400	P_5 all 0.8400
P_10 all 0.6467	P_10 all 0.6600	P_10 all 0.6667
P_15 all 0.5111	P_15 all 0.5156	P_15 all 0.5244
P_20 all 0.4200	P_20 all 0.4200	P_20 all 0.4167
P_30 all 0.2889	P_30 all 0.2889	P_30 all 0.2933
P_100 all 0.0880	P_100 all 0.0893	P_100 all 0.0893
P_200 all 0.0440	P_200 all 0.0447	P_200 all 0.0447
P_500 all 0.0176	P_500 all 0.0179	P_500 all 0.0179
P_1000 all 0.0088	P_1000 all 0.0089	P_1000 all 0.0089

Figure 6: Comparison of various values for DFR model tuning

By reducing the values, I noticed that the MAP score was increasing and hence I chose a lesser value every time till I reached a satisfactory result for the MAP score.

4 Conclusion

In this project we first indexed the twitter data on solr and then implemented three IR models namely, Language model, DFR model and BM25 model. We implemented these models using different similarity modules and tried various values in the options of these modules to reach an optimal score for MAP. This project demonstrated how various IR models work differently on a given query and how they affect the similarity between a given query and a document.

Acknowledgements

I am extremely grateful to Professor Rohini Srihari for teaching all the necessary concepts related to various IR models. I would also like to thank all the TA’s of the course for helping me at every step during the course of this project.

References

[1] Professor Rohini Srihari's lecture slides and lecture recordings.
[2] Similarity module (https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-similarity.html)
[3] Trec_eval (http://www.rafaelglater.com/en/post/learn-how-to-use-trec_eval-to-evaluate-your-information-retrieval-system)