**CSE 535: INFORMATION RETRIEVAL**

**PROJECT 3**

**EVALUATION OF IR MODELS**

**NAME:** Akshara Santharam

**UBIT NAME:** aksharas

**PERSON NUMBER:** 50313950

**OVERVIEW**

The goal of the project is to implement various IR models, evaluate the IR system and improve the search results based on the understanding, implementation and evaluation of the models. Given twitter data in three languages – English, German and Russian, 15 sample queries and the corresponding relevance judgements, twitter data must be indexed using Solr and the following three IR models: (i) Language Model (ii) BM25 and (iii) Divergence from Randomness (DFR) Model. The results from these three sets will be evaluated using Trec\_eval program. Based of the evaluation results, an attempt is made to improve the performance in terms of Mean Average Precision (MAP).

**DATASET**

The data to be used is Twitter data in json format, training\_tweet.json. Three languages are included- English(text\_en), German(text\_de) and Russian(text\_ru). The training\_tweet.json file contains approximately 3500 tweets with some fields extracted from raw data. The sample tweet format is as follows:

{

“lang”: ,

“id”: ,

“text\_de”: ,

“text\_en”: ,

“text\_ru”: ,

“tweet\_urls”: [],

“tweet\_hashtags”: []

}

**IMPLEMENTING THE DEFAULT CONFIGURATIONS OF THE IR MODELS**

1. **LANGUAGE MODEL**

The Language Model can be implemented as a global configuration using the following similarity class in the schema.xml file:

**<similarity class=”solr.LMDirichletSimilarityFactory”>**

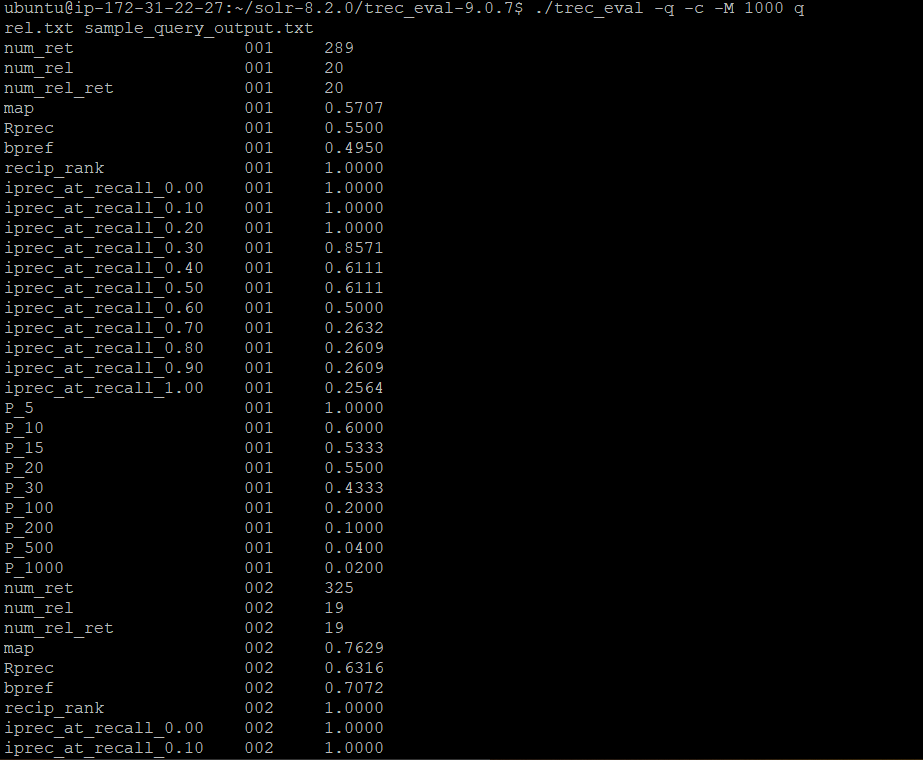
**<float name=”mu”> 2000 </float>**

**</similarity>**

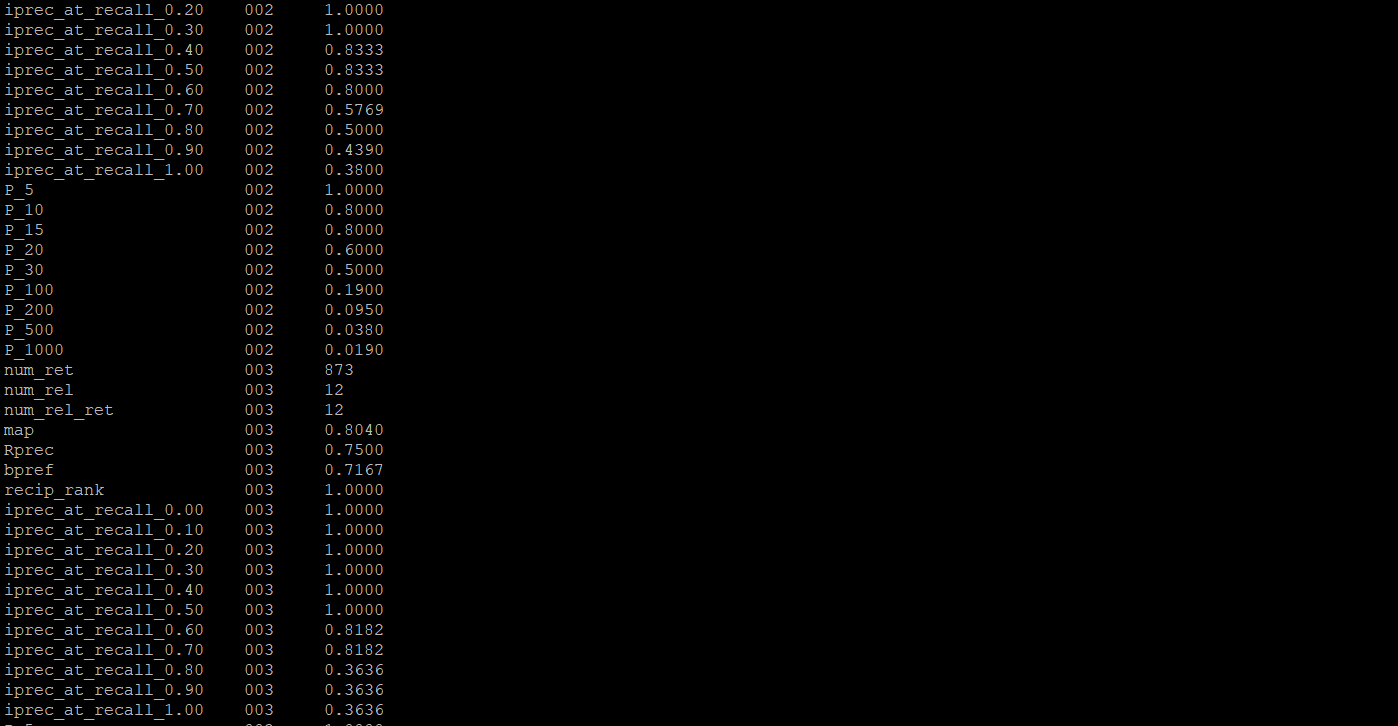
After indexing the training\_tweet.json provided for the configured schema.xml for the core on solr, TREC\_eval is run to evaluate the sample query output file.

**./trec\_eval -q -c -M 1000 qrel.txt sample\_query\_output.txt**

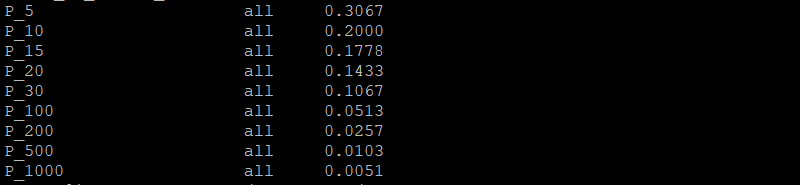
The above command will give the number of common evaluation measure results. The screenshot for the above is as shown below:



**Figure 1**



**Figure 2**

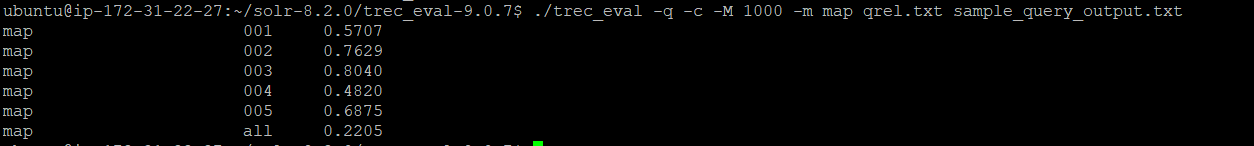


**Figure 3**

-m option can be used to specify the measure we prefer. This command will give the map measure result for each query followed by overall performance.

**./trec\_eval -q -c -M 1000 -m map qrel.txt sample\_query\_output.txt**

The screenshot for the above is as shown below:



**Figure 4**

1. **BM25**

We can implement BM25 model using the following Similarity class in the schema.xml

**<similarity class=”solr.BM25SimilarityFactory”>**

**<str name=”k1”>1.2</str>**

**<str name=”b”>0.7</str>**

**</similarity>**

After indexing the training\_tweet.json provided for the configured schema.xml for the core on solr, TREC\_eval is run to evaluate the sample query output file.

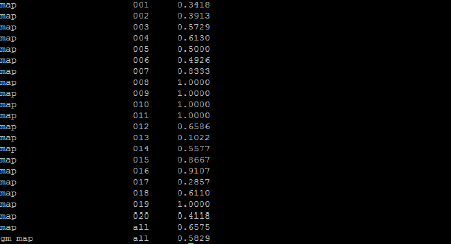
**./trec\_eval -q -c -M 1000 qrel.txt sample\_query\_output.txt**

The above command will give the number of common evaluation measure results.

-m option can be used to specify the measure we prefer. This command will give the map measure result for each query followed by overall performance.

**./trec\_eval -q -c -M 1000 -m map qrel.txt sample\_query\_output.txt**

The screenshot for the above is as shown below:



**Figure 5**

1. **DIVERGENCE FROM RANDOMNESS (DFR)**

We can implement DFR model using the following Similarity class in the schema.xml

**<similarity class=”solr.DFRSimilarityFactory”>**

**<str name=”basicModel”>G</str>**

**<str name=”afterEffect”>B</str>**

**<str name=”normalization”>H2</str>**

**</similarity>**

After indexing the training\_tweet.json provided for the configured schema.xml for the core on solr, TREC\_eval is run to evaluate the sample query output file.

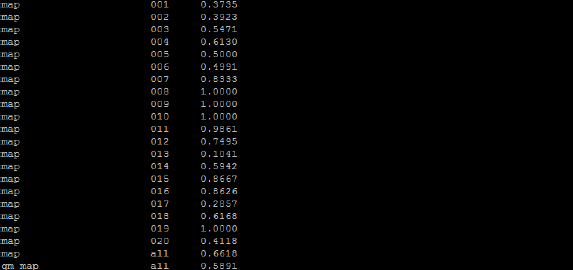
**./trec\_eval -q -c -M 1000 qrel.txt sample\_query\_output.txt**

The above command will give the number of common evaluation measure results.

-m option can be used to specify the measure we prefer. This command will give the map measure result for each query followed by overall performance.

**./trec\_eval -q -c -M 1000 -m map qrel.txt sample\_query\_output.txt**

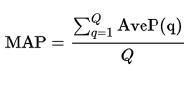
The screenshot for the above is as shown below:



**Figure 6**

**OPTIMIZING THE MODELS**

For optimizing the models, use Mean Average Precision (MAP). Mean average precision for a set of queries is the mean of the average precision scores for each query.



Where Q is the number of queries.