UBIT NAME: neeleshb UBIT NO: 50314928

IR PROJECT REPORT

Project report on Evaluation of IR Models

Implementation:

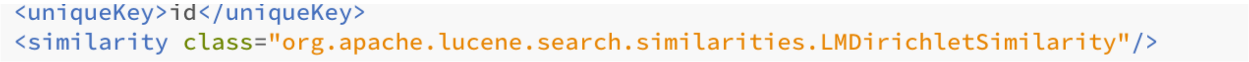
We have implemented 3 IR models, namely – Language model, BM25 and Divergence from Randomness model.

BM25:

We have implemented these 3 models in a similar fashion. By tweaking the schema that was created for every specific core, we achieve in implementing the model. We just have to add the <Similarity> tag in the schema.xml file to implement a certain model. In case of BM25 we don't need to add or insert anything into its schema since Solr implements BM25 model by default. BM25 and TF*IDF sit at the core of the ranking function. They comprise what Lucene calls the "field weight". Field weight measures how much matched text is about a search term. The only reason for the difference here is its derivation from probabilistic information retrieval. Lucene makes one change to BM25's regular IDF. BM25's IDF has the potential for giving negative scores for terms with very high document frequency. So IDF in Lucene's BM25 does this one amazing trick to solve this problem. They add 1 to the value, before taking the log, which makes it impossible to compute a negative value.

Language Model:

There are two types of language models that we can implement in the schema. Let's first talk about how we implemented it. We just had to add a similarity tag in the schema.xml file for the language model in the 'conf' file.



```
<uniqueKey>id</uniqueKey>
<similarity class="org.apache.lucene.search.similarities.LMDirichletSimilarity"/>
```

I have used LMDirichlet similarity model to implement the language model. Probability distribution over strings of text

– how likely is a given string (observation) in a given "language"

– for example, consider probability for the following four strings

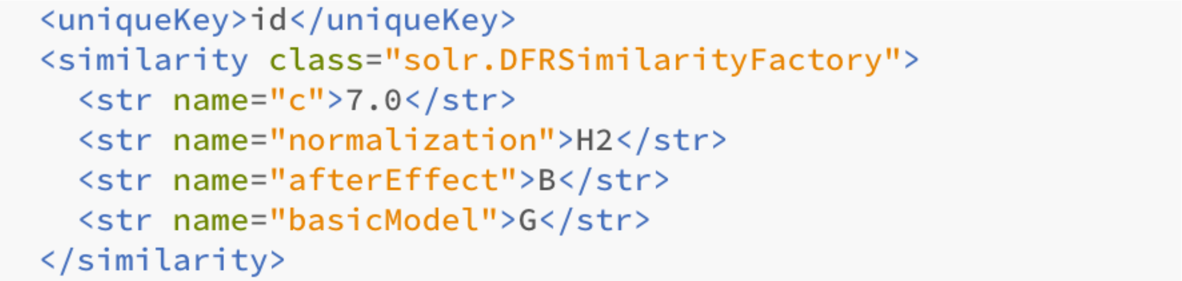
p1 = P("a quick brown dog")
p2 = P("dog quick a brown")
p3 = P("быстрая brown dog")
p4 = P("быстрая собака")
– English: p1 > p2 > p3 > p4
depends on what "language" we are modeling

– In most of IR, assume that p1 == p2

Estimate probabilities of certain "events" in the text

Divergence from Randomness:

DFR model is not implemented like LM model. We have to also define the first normalization and second normalization along with the basic model. The basic model that we used is specified as 'G' in the xml file which is Geometric approximation of Bose-Einstein and for the first normalization we have to add a tag after effect using the Bernoulli model and for the second normalization we have used the 'H2' model which term frequency density inversely related to length.



```
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.DFRSimilarityFactory">
    <str name="c">7.0</str>
    <str name="normalization">H2</str>
    <str name="afterEffect">B</str>
    <str name="basicModel">G</str>
  </similarity>
```

In the field of information retrieval, divergence from randomness, one of the very first models, is one type of probabilistic model. It is basically used to test the amount of information carried in the documents. It is based on Harter's 2-Poisson indexing-model. The 2-Poisson model has a hypothesis that the level of the documents is related to a set of documents which contains words occur relatively greater than the rest of the documents.It is not really a ‘model’, but a framework for weighting terms using probabilistic methods, and it has a special relationship for Term weighting based on notion of eliteness.

Term weights are being treated as the standard of whether a specific word is in that set or not. Term weights are computed by measuring the divergence between a term distribution produced by a random process and the actual term distribution. (Sourec: Wikipedia)