HW3 Report

kangh

ErrorAnalysis:

For purpose of effectively finding possible causes that make Vector Space Model failing to rank relevant document as the first. I create new function "errorAnalysis" to reorganize matched words in each document. The format is "rel=… '\t' score 't' token '\t' tf '|' token '\t' tf '|'…". The appendix contains results of naiveCosine, tfidfCosine, and okapiBM25.

Firstly I use the document of error analysis on naïveCosine comparing matched words with original document. The following screenshot is the result of scanning over all documents sentence by sentence. I transform raw excel form to FORM1.

types of error \ qid	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	sum
Tense			1	1	1		1	1								1					6
Uppercase, Lowercase					1				1	1							1				4
Each document has some same words	1		1	1								1			1						5
Punctuation		1			1	1	1	1	1	1	1		1	1			1	1	1	1	14
Plural						1															1
Stop words			1			1			1				1	1	1					1	7
LengthBias	1	1		1												1					4
Mismatch vocabulary									1												1
The content of query is paraphrased by document						1						1					1		1		4
Spelling	1										1										2

Types of Error	Frequency	Example	Solution
Tense	6	Query4	StanfordLemmatizer
Upper, Lowercase	4	Query9	StanfordLemmatizer
Each document have same key words	5	Query1	tfidf
Punctuation	14		Cornlp.tokenizer, regular expression
Plural	1	Query6	StanfordLemmatizer
Stop words	7	Query9	dictinoary
Length bias (shorter document may be overestimated, and longer document may be underestimate)	4	Query16	Pivoted normalized document length (Apply to cosine similarity function)
Mismatch vocabulary	1	Query9	NLP, feature selecting model (Hard to implement)
The content of query is paraphrased by document	4	Query13	NLP, feature selecting model (Hard to implement)
Spelling	2	Query11	Spelling correction (maybe Lingpipe)

Form 1

The statements of above examples are as follows:

1. Tense:query4
   qid=4 rel=99 What year did Wilt Chamberlain score 100 points?
   qid=4 rel=1 On March 2, 1962, Wilt Chamberlain scored a record 100 points in a game against the New York Knicks.
   qid=4 rel=0 A 100 point game was a highlight in a career of Wilt Chamberlain
   qid=4 rel=0 Wilt Chamberlain most famous record is the 100 points he scored in the Philadelphia Warriors' 169-147 defeat of the New York Knicks.
2. Upper, Lowercase:query9
   qid=9 rel=99 What was the first spaceship on the moon
   qid=9 rel=1 Luna 2 was the first spacecraft to reach the surface of the Moon.
   qid=9 rel=0 And he was down to about 15 seconds of fuel, after dodging boulders on the moon, when the Eagle landed on July 20, 1969.
   qid=9 rel=0 When the cool-thinking Armstrong realized that the computer navigation would set Eagle, the lunar lander, down among dangerous boulders in a gaping crater, he acted.
   qid=9 rel=0 Eagle was the first manned spacecraft that reached the surface of the moon
3. Each document have same key words:query1
   qid=1 rel=99 Give us the name of the volcano that destroyed the ancient city of Pompeii
   qid=1 rel=1 In A.D. 79, long-dormant Mount Vesuviuserupted, burying in volcanic ash the Roman city of Pompeii; an estimated 20,000 people died.
   qid=1 rel=0 You can see Vesuvius in the background, near ruins of Pompei; its last eruption was in 1944.
   qid=1 rel=0 Vesuvius is located near the ruins of the destroyed city of Pompeii.
   qid=1 rel=0 In 79 A.D., this ancient city was buried in an avalanche of hot ash from Mount Vesuvius.
4. Punctuation： Query 5
   qid=5 rel=99 What river is called China's Sorrow?
   qid=5 rel=1 People of China have mixed feelings about River, which they often call "sorrow of China"
   qid=5 rel=0 Yangtze is longest river in Asia in general and in China, in particular.
   qid=5 rel=0 Yellow river is often called the mother of China
5. Plural:query6
   qid=6 rel=99 Who was the first person to run the mile in less than four minutes
   qid=6 rel=1 RogerBannister was the first to break the four-minute mile barrier.
   qid=6 rel=0 The claim that a 4-minute mile was once thought to be impossible byinformed observers was and is a widely propagated myth created by sportswriters and debunked by Bannister himself in his memoir, The Four Minute Mile (1955).
   qid=6 rel=0 It is hard for humans to run the mile faster than in four minutes
6. Stop words: query9
   qid=9 rel=99 What was the first spaceship on the moon
   qid=9 rel=1 Luna 2 was the first spacecraft to reach the surface of the Moon.
   qid=9 rel=0 And he was down to about 15 seconds of fuel, after dodging boulders on the moon, when the Eagle landed on July 20, 1969.
   qid=9 rel=0 When the cool-thinking Armstrong realized that the computer navigation would set Eagle, the lunar lander, down among dangerous boulders in a gaping crater, he acted.
   qid=9 rel=0 Eagle was the first manned spacecraft that reached the surface of the moon
7. Length bias (shorter document may be overestimated, and longer document may be underestimate)： query16
   qid=16 rel=99 When did Bob Marley die
   qid=16 rel=1 Bob Marley died in 1981 from cancer at age 36.
   qid=16 rel=0 Without proper and timely removal of tumor one can die like Bob Marley or Steve Jobs.

qid=16 rel=0 Bob Marley was a Jamaican reggae singer-songwriter, musician, and guitarist who did achieve international fame.

8. Mismatch vocabulary：query9

qid=9 rel=99 What was the first spaceship on the moon
qid=9 rel=1 Luna 2 was the first spacecraft to reach the surface of the Moon.
qid=9 rel=0 And he was down to about 15 seconds of fuel, after dodging boulders on the moon, when the Eagle landed on July 20, 1969.
qid=9 rel=0 When the cool-thinking Armstrong realized that the computer navigation would set Eagle, the lunar lander, down among dangerous boulders in a gaping crater, he acted.
qid=9 rel=0 Eagle was the first manned spacecraft that reached the surface of the moon

9. The content of query is paraphrased by document：query13

qid=13 rel=99 How deep is Crater Lake?
qid=13 rel=1 Oregon's Crater Lake tops it at 1,932 feet at its greatest depth.
qid=13 rel=0 Crater Lake is a caldera lake in the western United States.
qid=13 rel=0 There are no rivers flowing into or out of Crater Lake

10. Spelling：query11

qid=11 rel=99 Where is Devil's Tower
qid=11 rel=1 Devils Tower can be found in Crook County
qid=11 rel=0 To the west, across the Wyoming border, is the staggeringly beautiful Devil's Tower National Monument.
qid=11 rel=0 Devil's Tower is an igneous intrusion that rises dramatically 1,267 feet (386 m) above the surrounding terrain.
qid=11 rel=0 In 1941, Petzoldt joined other rock climbers to rescue a marooned parachutist who had landed atop Devil's Tower

Architecture:

Task 2 is still based on basic UIMA framework provided by task1. As is shown in figure1, the DocumentVectorAnnotator first read from one document or query each pipeline, and then break down each document by extracting words including its position and term frequency. Next RetrievalEvaluator get data from CAS, and write into files for future process. Finally, if CollectionProcess finishes, it will call MRR calculation module, otherwise, start another new pipeline.

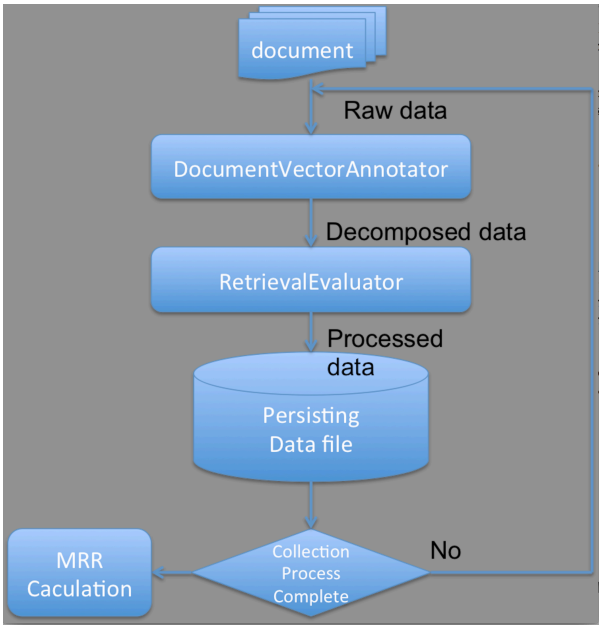
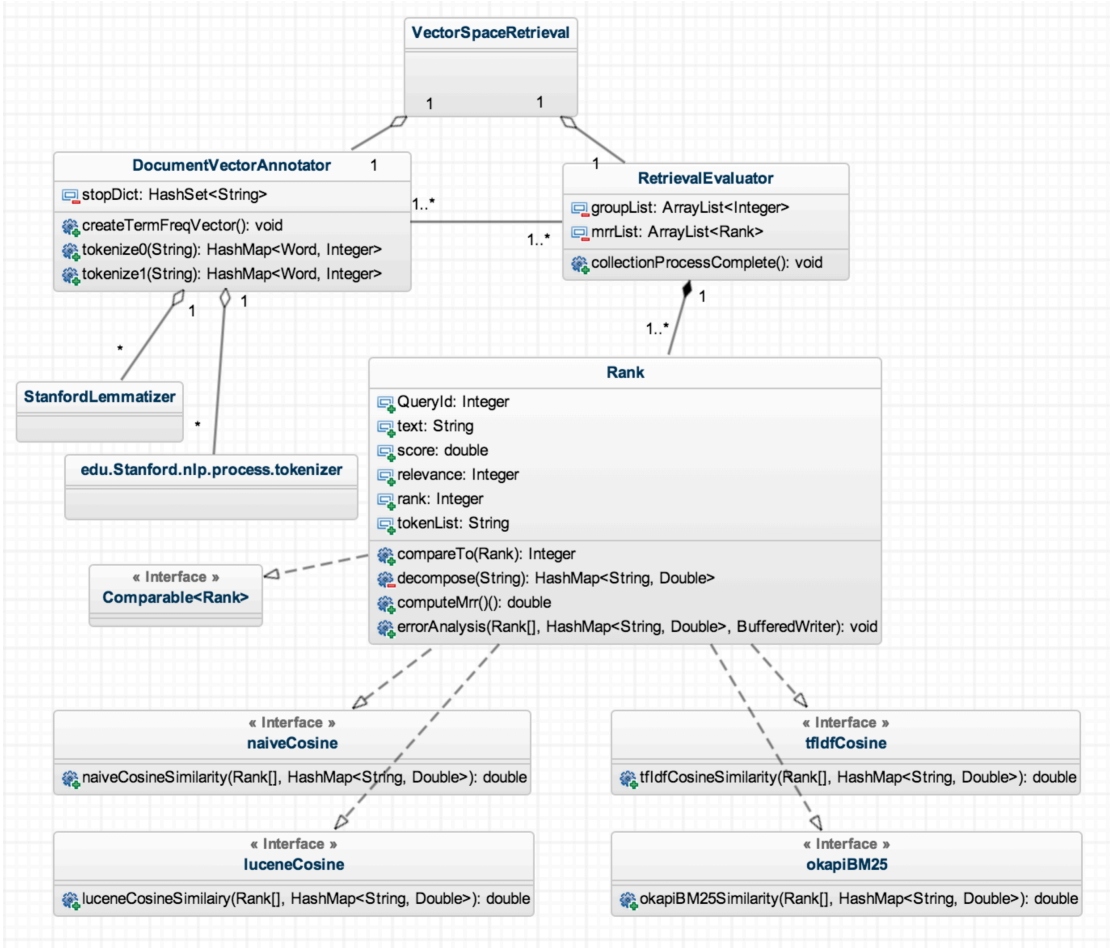


figure1



Here is a class diagram oftask2,VectorSpaceRetrieval, DocumentVectorAnnotator, RetrievalEvaluator are classes provided by task1.

In class DocumentVectorAnnotator, I used provided StanfordLemmatizer for reduce all words to original stem and coreNlptokenizer, which used in HW0 to separate all words from punctuation. Additionally, I build dictionary using given stopword.txt to filter out stopwords in documents.

In class RetrievalEvaluator, I add a new class called Rank; it mainly functions as capsulizing similarity functions and storing variables read from persisting txt file. Class Rank implements interface of Comparable, which provides convenience for future potential sort. Besides function of computeMrr required by taks1, it also implements three interfaces of scoring method: naiveCosine, tfIdfCosine, OkapiBM25 and luceneSimilarity. Moreover, there are extra two helper functions: errorAnalysis and decompose. Function errorAnalysis will compare result and pick out valuable words that can match with words in query and output into a files preparing error analysis and function decompose modifies data structure helping convert string tokenList stored in tokenList to HashMap<String, Double> first

dimension stores each token, and second dimension stores corresponding term frequency of that word.

Tokenization algorithm:

In tasks, I used given regular expression pattern "[\\s+]" to simply split sentence into words by blanks. It actually better than using pattern [" "], because when a thread of blanks appear, the latter one will make mistake on considering blanks as a word. However, it neglects punctuation, uppercase impact and etc. as shown in Form1. So, I decide try to use some existed tokenizer such coreNlp to erase all bad effect. The MRR can be improved from 0.4375 to 0.508.

Better stemming algorithm:

Based on tokenization algorithm, I used given StanfordLemmatizer elevate MRR from 0.505 to 0.5547.

Better or different similarity measures:

Java code is based on tfidfCosine function (the name may be not precise), each score method can be inserted to achieve different scoring effect. The tfidfCosine function is pasted in the appendix.

1. Tf-Idf:

It is more about a thought rather than a method. When most documents have the same words shown in query, it cannot be an effective word for discriminating documents. Thus, if a word is contained by many documents, its importance will be marginalized. Based on tf, term frequency, we multiply another factor called idf, inverse document frequency. In other word, the weight of each word, one hand, is positively relevant to its tf, on the other hand, is negatively relevant to idf.

I choose calculation equation from Wikipedia. It does not use log as smoothing like standard one. Because our dataset is pretty small, there is no need to smooth tf of both query and documents. Actually the practical result is better.

Peek of implementation:

```
383
384  private double tfIdfCosineSimilarity(Map<String, Double> queryVector,
385          Map<String, Double> docVector,  Map<String, Integer> df, int D){
386
387     for (String word : queryVector.keySet()){
388        double idf = D*1.0/((df.get(word) == null ? 0 :df.get(word)) +1);
389          queryVector.put(word,(Math.log(queryVector.get(word)) + 1) * idf);
390     }
391     for (String word : docVector.keySet()){
392       docVector.put(word,(Math.log(docVector.get(word))+1));
393     }
394  return computeCosineSimilarity(queryVector, docVector);
395  }
```

2.Lucene:

The calculation equation is as follows:

$$\frac{\sum d_i \cdot q_i}{\sqrt{\sum d_i^2} \cdot \sqrt{\sum q_i^2}} = \frac{\sum\left(\sqrt{tf} \cdot \left(1+\log\frac{N}{df+1}\right)\right) \cdot \left(qtf \cdot \left(1+\log\frac{N}{df+1}\right)\right)}{\sqrt{\sum\left(\sqrt{tf} \cdot \left(1+\log\frac{N}{df+1}\right)\right)^2} \cdot \sqrt{\sum\left(qtf \cdot \left(1+\log\frac{N}{df+1}\right)\right)^2}}$$

"tf" "idf" "idf"

"doc length normalization" "query length normalization"

(11-442 / 11-642: Search Engines Document Representation)

There are two characteristics comparing to standard tdIdf equations (log smooth):

1.The tf weight is sqrt(tf) instead of log(tf)+1.

2. Use idf2 instead of idf.

There are two effects brought by this modification:

1.Strong reward for terms that are frequent in this document.

2.Strong penalty for terms that are frequent across the corpus.

Peek of implementation:

```
398  private double luceneCosineSimilarity(Map<String, Double> queryVector,
399          Map<String, Double> docVector,  Map<String, Integer> df, int D){
400
401     for (String word : queryVector.keySet()){
402        double idf = (1+Math.log(D*1.0/((df.get(word) == null ? 0 :df.get(word)) +1)));
403          queryVector.put(word,queryVector.get(word) * idf);
404     }
405     for (String word : docVector.keySet()){
406       double idf = (1+Math.log(D*1.0/((df.get(word) == null ? 0 :df.get(word))+1)));
407       docVector.put(word,Math.sqrt(docVector.get(word)) * idf);
408     }
409  return computeCosineSimilarity(queryVector, docVector);
410  }
411
```

3.OkapiBM25:

This score method also incorporates tfIdf thought but different interpretation on Idf value.

$$\mathrm{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$ (Wikipedia)

However, this method may have some drawback that when term appears in whole document corpus more than half, it will contribute negative score to the final document score. This will result to some problem in some situation when 2 almost identical document. If one has some word that existed in query but not contained by the other document, the latter one will unhopefully have lower score than former one.

Peek of implementation:

```
265  private double okaipiBM25(HashMap<String, Double> query, HashMap<String, Integer> df,
266          HashMap<String, Double> tf, int D, double avgd) {
267    double score = 0;
268  //  double k1 = 2; // 1.2-2.0
269    double b = 0.75;
270    double docLength = 0;
271    for (String word : tf.keySet()) {
272      docLength += tf.get(word);
273    }
274    // System.out.println("************");
275    for (String word : query.keySet()) {
276      if (tf.get(word) == null)
277        continue;
278      double idf = Math.log((D - df.get(word) + 0.5) / (df.get(word) + 0.5));
279      // System.out.println(idf + " " + df.get(word) + " " +D + " " + (D - df.get(word) + 0.5) /(
280      // df.get(word) + 0.5));
281      score += idf * (tf.get(word) /* (k1 + 1)*/) / (tf.get(word) + k1 * (1 - b + b * docLength / avgd));
282    }
283    // System.out.println(score);
284    return score;
285  }
```

Improvement:

Notice: Here, I just use result generated by scoring method “dfIdf” In every example, the first query denotes original naiveCosine method, and second query denote improved results. The key words in query marked red.

query1.0: of 2|that 1|volcano 1|ancient 1|destroyed 1|Give 1|name 1|us 1|the 3|Pompeii 1|city 1|0. rel=0 of 2|destroyed 1|the 2|city 1|
1. rel=1 the 1|city 1|of 1|
2. rel=0 the 1|of 1|
3. rel=0 city 1|of 1|ancient 1|
query1.0: volcano 1.0|us 1.0|pompeius 1.0|name 1.0|give 1.0|destroy 1.0|city 1.0|ancient 1.0|
0. rel=0 0.3612 pompeius 1.0|destroy 1.0|city 1.0|
1. rel=0 0.2060 city 1.0|ancient 1.0|
2. rel=1 0.0736 pompeius 1.0|city 1.0|
3. rel=0 0.0000

Analysis：Although the redundant stop words were removed, rank0 document still has more key words than rank2 document.

query2.0: to 1|see 1|Jordan 1|What 1|come 1|the 1|crowd 1|has 1|been 1|ever 1|Michael 1|largest 1|0. rel=0 the 1|Michael 1|largest 1|Jordan 1|crowd 1|
1. rel=1 to 3|the 2|Michael 1|see 1|
2. rel=0 to 1|the 2|
3. rel=0 the 2|Michael 1|largest 1|
query2.0: see 1.0|michael 1.0|largest 1.0|jordan 1.0|ever 1.0|crowd 1.0|come 1.0|
0. rel=0 0.3980 michael 1.0|largest 1.0|jordan 1.0|crowd 1.0|
1. rel=1 0.1745 see 1.0|michael 1.0|jordan 1.0|
2. rel=0 0.1715 michael 1.0|largest 1.0|crowd 1.0|
3. rel=0 0.0000

Analysis：Although uppercase was transformed into lowercase, same problem with example 1 exists

query3.0: of 1|In 1|which 1|a 1|did 1|year 1|purchase 1|Alaska 1|happen? 1|
0. rel=0 of 1|a 1|purchase 1|Alaska 1|
1. rel=0 purchase 1|of 1|Alaska 1|
2. rel=1 Alaska 1|year 1|
query3.0: year 1.0|purchase 1.0|happen 1.0|alaska 1.0|
0. rel=1 0.4433 year 1.0|purchase 1.0|alaska 1.0|
1. rel=0 0.0000 purchase 1.0|alaska 1.0|
2. rel=0 0.0000 purchase 1.0|alaska 1.0|

Analysis：Punctuation are removed, and result is correct

query4.0: Chamberlain 1|points? 1|What 1|score 1|did 1|year 1|Wilt 1|100 1|
0. rel=0 Chamberlain 1|100 1|Wilt 1|
1. rel=1 Chamberlain 1|100 1|Wilt 1|
2. rel=0 Chamberlain 1|100 1|Wilt 1|
query4.0: year 1.0|wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
0. rel=1 0.0771 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
1. rel=0 0.0583 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
2. rel=0 0.0000 wilt 1.0|point 1.0|chamberlain 1.0|100 1.0|

Analysis：Plurals have been transformed into single, and because of length bias, rank0 document has higher cosine score.

query5.0: is 1|China's 1|Sorrow? 1|What 1|called 1|river 1|把时态去掉了
0. rel=0 is 1|called 1|river 1|
1. rel=0 is 1|river 1|
2. rel=1
query5.0: sorrow 1.0|river 1.0|china 1.0|call 1.0|
0. rel=1 0.2733 sorrow 1.0|river 1.0|china 1.0|call 1.0|
1. rel=0 0.1224 river 1.0|china 1.0|call 1.0|
2. rel=0 0.0000 river 1.0|china 1.0|

Analysis：Tense has been removed , and result is correct

query6.0: to 1|person 1|minutes 1|Who 1|than 1|mile 1|the 2|four 1|run 1|first 1|in 1|less 1|was 1|
0. rel=0 to 1|minutes 1|the 1|run 1|in 1|than 1|mile 1|four 1|
1. rel=1 to 1|mile 1|the 2|first 1|was 1|
2. rel=0 to 1|was 2|mile 1|in 1|
query6.0: run 1.0|person 1.0|minute 1.0|mile 1.0|less 1.0|four 1.0|first 1.0|
0. rel=0 0.1890 run 1.0|minute 1.0|mile 1.0|four 1.0|
1. rel=1 0.1858 minute 1.0|mile 1.0|four 1.0|first 1.0|
2. rel=0 0.0000 minute 1.0|mile 1.0|four 1.0|

query7.0: become 1|a 1|What 1|did 1|year 1|Alaska 1|state? 1|
0. rel=0 did 1|Alaska 1|
1. rel=0 a 1|
2. rel=1 Alaska 1|
3. rel=0
query7.0: year 1.0|state 1.0|become 1.0|alaska 1.0|
0. rel=1 0.2126 state 1.0|become 1.0|alaska 1.0|
1. rel=0 0.1497 become 1.0|alaska 1.0|
2. rel=0 0.1021 state 1.0|alaska 1.0|
3. rel=0 0.0366 state 1.0|

Analysis：The result is correct because of removing redundant stopwords, and stemming

method.

query8.0: Tyson 1|bite 1|did 1|ear? 1|Mike 1|Holyfield's 1|When 1|0. rel=0 Tyson 1|Holyfield's 1|
1. rel=1 Tyson 1|Holyfield's 1|
2. rel=0 Tyson 1|
3. rel=0 Tyson 1|
query8.0: tyson 1.0|mike 1.0|holyfield 1.0|ear 1.0|bite 1.0|
0. rel=0 0.2390 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
1. rel=1 0.1806 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
2. rel=0 0.0000 tyson 1.0|holyfield 1.0|
3. rel=0 0.0000 tyson 1.0|holyfield 1.0|

Analysis：Although first two documents has same key words, length bias make rank0 document words score much larger.

query9.0: moon 1|spaceship 1|What 1|on 1|the 2|first 1|was 1|
0. rel=0 moon 1|the 3|first 1|was 1|
1. rel=1 the 3|first 1|was 1|
2. rel=0 the 2|was 1|on 2|
3. rel=0 the 3|
query9.0: spaceship 1.0|moon 1.0|first 1.0|
0. rel=0 0.2828 moon 1.0|first 1.0|
1. rel=1 0.2342 moon 1.0|first 1.0|
2. rel=0 0.0415 moon 1.0|
3. rel=0 0.0000

Analysis：Same reason with query8

query10.0: Who 1|Prize 1|won 1|the 1|1992? 1|Nobel 1|Peace 1|in 1|
0. rel=1 won 1|the 1|Nobel 1|in 1|
1. rel=0 the 1|Nobel 1|in 1|Peace 1|
2. rel=0 Nobel 1|in 1|
3. rel=0 the 1|
query10.0: win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
0. rel=1 0.6907 win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
1. rel=0 0.0338 prize 1.0|peace 1.0|nobel 1.0|
2. rel=0 0.0300 prize 1.0|peace 1.0|nobel 1.0|
3. rel=0 0.0000 prize 1.0|nobel 1.0|

Analysis：Although original rel=1 document rank0, the performance is extremely enhanced comparedto other document score.

query11.0: Where 1|is 1|Devil's 1|Tower 1|
0. rel=0 is 1|Devil's 1|Tower 1|
1. rel=0 is 1|Devil's 1|Tower 1|
2. rel=0 Devil's 1|Tower 1|
3. rel=1 Tower 1|
query11.0: tower 1.0|devil 1.0|
0. rel=1 0.0000 tower 1.0|devil 1.0|
1. rel=0 0.0000 tower 1.0|devil 1.0|
2. rel=0 0.0000 tower 1.0|devil 1.0|
3. rel=0 0.0000 tower 1.0|devil 1.0|

Analysis：Manually choose rel=1 document when ties appears.

query12.0: of 1|is 1|height 1|redwood? 1|What 1|the 2|tallest 1|
0. rel=0 is 1|the 2|tallest 1|
1. rel=0 the 2|tallest 1|
2. rel=1 the 2|
3. rel=0 is 1|the 1|tallest 1|
query12.0: tallest 1.0|redwood 1.0|height 1.0|
0. rel=0 0.2067 tallest 1.0|redwood 1.0|
1. rel=0 0.0467 tallest 1.0|redwood 1.0|
2. rel=0 0.0404 tallest 1.0|redwood 1.0|
3. rel=1 0.0000 redwood 1.0|

Analysis：The result is worse, because lacking key matched words

query13.0: is 1|Crater 1|Lake? 1|How 1|deep 1|

0. rel=0 is 1|Crater 1|
1. rel=0 Crater 1|
2. rel=1 Crater 1|
query13.0: lake 1.0|deep 1.0|crater 1.0|
0. rel=1 0.0000 lake 1.0|crater 1.0|
1. rel=0 0.0000 lake 1.0|crater 1.0|
2. rel=0 0.0000 lake 1.0|crater 1.0|

Analysis：Although the result is correct, it is due to manually choosing rel=1 document when tie appears.

query14.0: Who 1|for 1|Commodores 1|lead 1|singer 1|the 2|was 1|
0. rel=0 Commodores 1|the 2|
1. rel=1 for 1|lead 1|singer 1|was 1|
2. rel=0 the 1|was 1|Commodores 1|
query14.0: singer 1.0|lead 1.0|commodore 1.0|
0. rel=1 0.5164 singer 1.0|lead 1.0|commodore 1.0|
1. rel=0 0.0000 commodore 1.0|
2. rel=0 0.0000 commodore 1.0|

Analysis：The result is correct, one should be noticed that other document score is 0 due to tfidf factorbecause “commodore” appears in all documments.

query15.0: is 1|What 1|on 1|the 1|coldest 1|earth? 1|place 1|
0. rel=0 is 1|the 1|coldest 1|place 1|
1. rel=0 is 1|the 1|coldest 1|place 1|
2. rel=1 coldest 1|
query15.0: place 1.0|earth 1.0|coldest 1.0|
0. rel=1 0.1667 earth 1.0|coldest 1.0|
1. rel=0 0.1458 place 1.0|coldest 1.0|
2. rel=0 0.0940 place 1.0|coldest 1.0|

Analysis：The result is correct, because using tokenize algorithm and tfidf thought, you can see earth in first document is unique but other keywords appears across documents.

query16.0: Bob 1|die 1|did 1|Marley 1|When 1|
0. rel=0 Marley 1|did 1|Bob 1|
1. rel=0 Marley 1|Bob 1|die 1|
2. rel=1 Bob 1|Marley 1|
query16.0: marley 1.0|die 1.0|bob 1.0|
0. rel=1 0.1048 marley 1.0|die 1.0|bob 1.0|
1. rel=0 0.0669 marley 1.0|die 1.0|bob 1.0|
2. rel=0 0.0000 marley 1.0|bob 1.0|

Analysis：Although first two documents has the same key words but length bias make words weight of rank2 sentence smaller.

query17.0: Which 1|is 1|U.S. 1|leading 1|state 1|the 1|corn 1|producer? 1|
0. rel=0 is 1|leading 1|the 1|corn 1|
1. rel=0 state 1|corn 1|U.S. 1|leading 1|
2. rel=1 is 1|the 1|
query17.0: us 1.0|state 1.0|producer 1.0|lead 1.0|corn 1.0|
0. rel=0 0.3191 state 1.0|producer 1.0|lead 1.0|corn 1.0|
1. rel=0 0.2315 us 1.0|state 1.0|lead 1.0|corn 1.0|
2. rel=1 0.0439 producer 1.0|corn 1.0|

Analysis：The rank is still lowest, BTW, us should be the stop words but missing in stopword.txt

query18.0: Where 1|McDonald's 1|built? 1|the 1|first 1|was 1|
0. rel=0 McDonald's 1|the 1|
1. rel=1 the 4|
2. rel=0
query18.0: mcdonald 1.0|first 1.0|build 1.0|
0. rel=1 0.0000 mcdonald 1.0|
1. rel=0 0.0000 mcdonald 1.0|
2. rel=0 0.0000 mcdonald 1.0|

Analysis：When ties appear, we choose document with rel =1.

query19.0: disaster 1|The 1|which 1|Hindenburg 1|Jersey 1|1937 1|took 1|New 1|town? 1|place 1|in 2|
0. rel=0 took 1|The 1|place 1|Hindenburg 2|disaster 1|
1. rel=0 1937 1|in 1|disaster 1|which 1|
2. rel=1 Hindenburg 1|in 1|
query19.0: town 1.0|take 1.0|place 1.0|new 1.0|jersey 1.0|hindenburg 1.0|disaster 1.0|1937 1.0|
0. rel=0 0.2465 take 1.0|place 1.0|hindenburg 1.0|disaster 1.0|
1. rel=0 0.1108 hindenburg 1.0|disaster 1.0|1937 1.0|
2. rel=1 0.0314 hindenburg 1.0|1937 1.0|

Analysis：The performance hasn't been improved, because of lacking key words

query20.0: Keystone 1|is 1|What 1|State? 1|the 1|
0. rel=0 Keystone 2|is 1|the 1|
1. rel=1 is 1|the 1|Keystone 1|
2. rel=0 is 1|
query20.0: state 1.0|keystone 1.0|
0. rel=1 0.3394 state 1.0|keystone 1.0|
1. rel=0 0.1042 keystone 1.0|
2. rel=0 0.0000

Significance Test

I do statistical significance test based on tokenization algorithm and StanfordLemmatizer.

P-value: It can help me know whether difference from different samples will have the value rather than caused by random sample error.

Confidential interval: The 95% confidence of interval around mean of sample. The more narrow of interval, the more close to true mean.

Singleton T test: In this environment, we don't know the average MRR of each method, so we cannot apply this test to these samples.

Paired-T test: It helps test whether the means of paired sample are equal from perspective of distribution.

I use Matlab to do some statistical significance test:

The second experiment is about comparing two different methods whether they have significant difference on MRR.

We assume H that mean of sample, MRR, is no huge difference from different result of scoring approaches. If H is 0, we accept original assumption; else we reject H. The confidence interval is actually 95% confident range in which difference of mean between different method lies.

Scoring method	Mean	Standard Deviation	Confidence Interval (>95%)
naiveCosine	0.6542	0.3270	(0.5011, 0.8072)
tf-IdfCosine	0.7625	0.3050	(0.5060, 0.8107)
luceneCosine	0.6542	0.3270	(0.5011, 0.8072)
okaipiBM25 (k1 = 2, b=0.75)	0.6500	0.3318	(0.4947, 0.8053)

We do singleton t test on each method. The result gives us average value, MRR; standard deviation and 95% confidence Interval of MRR.

Paired-T test	(BM25, cosine)	(tf-Idf, cosine)	(lucene, cosine)	(tf-Idf, BM25)
P-value	0.9731	0.0571	1	0.1337
Confidence interval	(-0.2590,0.2506)	(-0.0286, 0.2453)	(-0.0759, 0.0759)	(-0.0936, 0.3186)
H	0	0	0	0

Unfortunately, the result of paired T test does not reach my expectation. In other words, we cannot rule out possibility that the most salient improvement using unnormalized tfIdf could be caused by random sample error.

Interesting findings:

When I compare raw document with processed token list, I found the first example Pompeii hasn't been correctly stemmed by StanfordLemmatizer. If Pompeii is followed by dot '.', it will be reduced to "pompeii." Otherwise, it will turn into "pompeiis". After I solve this little bug, my naiveCosine performance can be improved by 5%.

Although I try many score methods, except naïveCosine all other methods need large number of documents to support meaningful df value. Note that our test dataset is quite small. Luncne put tf into log function, for this small set weaken effect of tf. On the contrary, tfIdf does not use log function, surprisingly get better performance. To sum up, there is no one method can apply to all conditions. We need to choose an adaptable method according to practical dataset.

Citation：

1. Okapi BM25. (2014, September 15). In Wikipedia, The Free Encyclopedia. Retrieved 00:56, October 21, 2014, from http://en.wikipedia.org/w/index.php?title=Okapi_BM25&oldid=625661591

2.Document Representation, 11-442 / 11-642: Search Engines, Jamie Callan Carnegie Mellon University

3. How to Correctly Interpret P Values
Jim Frost 17 April, 2014 http://blog.minitab.com/blog/adventures-in-statistics/how-to-correctly-interpret-p-values

Appendix

```
private void tfidfCosine(Rank r[], HashMap<String, Double> queryVector) throws IOException {
  HashMap<Integer, HashMap<String, Double>> tfidfDict = new HashMap<Integer, HashMap<String, Double>>(
  HashMap<String, Integer> memory = new HashMap<String, Integer>();
  double avgd = 0;
  for (int i = 0; i < r.length; i++) {
    tfidfDict.put(i, decompose(r[i].getTokenList()));
    for (String word : tfidfDict.get(i).keySet()) {
      avgd += tfidfDict.get(i).get(word);
    }
  }
  avgd /= r.length;

  for (int i = 0; i < r.length; i++) {
    // int docLength = 0;
    HashMap<String, Double> docVector = tfidfDict.get(i);

    HashMap<String, Double> docLuceneVector = tfidfDict.get(i);
    /*
     * for (String word : doc.keySet()){ docLength += doc.get(word); }
     */
    for (String word : docVector.keySet()) {
      int df = 0;
      if (!memory.containsKey(word)) {
        for (int j = 0; j < r.length; j++) {
          df += tfidfDict.get(j).containsKey(word) ? 1 : 0;
        }
        memory.put(word, df);
      } else {
        df = memory.get(word);
      }
        if (df == 0) df += 1;
        double idf = Math.log(r.length * 1.0 / df);
        double tf = docVector.get(word);
      //  docVector.put(word, tf * idf);                          //ajust to tfidf(deprecated)
    }
    // double score = ltfIdfCosineSimilarity((Map<String, Double>)queryVector.clone(), (Map<String, Dou
       double score = luceneCosineSimilarity((Map<String, Double>)queryVector.clone(), (Map<String, Dou
    // double score = okaipiBM25(queryVector, memory, docVector, r.length, avgd);  //adjust to okaipiBM
        r[i].setScore(score);
  }
}
```

Report for cosine:

```
query1.0:  volcano 1.0|us 1.0|pompeius 1.0|name 1.0|give 1.0|destroy 1.0|city 1.0|ancient 1.0|
0.  rel=0  0.4009 pompeius 1.0|destroy 1.0|city 1.0|
1.  rel=0  0.2236 city 1.0|ancient 1.0|
2.  rel=1  0.1667 pompeius 1.0|city 1.0|
3.  rel=0  0.0000

query2.0:  see 1.0|michael 1.0|largest 1.0|jordan 1.0|ever 1.0|crowd 1.0|come 1.0|
0.  rel=0  0.4781 michael 1.0|largest 1.0|jordan 1.0|crowd 1.0|
1.  rel=0  0.3273 michael 1.0|largest 1.0|crowd 1.0|
2.  rel=1  0.2417 see 1.0|michael 1.0|jordan 1.0|
3.  rel=0  0.0000

query3.0:  year 1.0|purchase 1.0|happen 1.0|alaska 1.0|
0.  rel=1  0.6708 year 1.0|purchase 1.0|alaska 1.0|
1.  rel=0  0.3536 purchase 1.0|alaska 1.0|
2.  rel=0  0.3162 purchase 1.0|alaska 1.0|

query4.0:  year 1.0|wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
0.  rel=0  0.6172 wilt 1.0|point 1.0|chamberlain 1.0|100 1.0|
1.  rel=1  0.5661 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
2.  rel=0  0.5270 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|

query5.0:  sorrow 1.0|river 1.0|china 1.0|call 1.0|
0.  rel=1  0.6325 sorrow 1.0|river 1.0|china 1.0|call 1.0|
1.  rel=0  0.6124 river 1.0|china 1.0|call 1.0|
2.  rel=0  0.3780 river 1.0|china 1.0|

query6.0:  run 1.0|person 1.0|minute 1.0|mile 1.0|less 1.0|four 1.0|first 1.0|
0.  rel=0  0.5714 run 1.0|minute 1.0|mile 1.0|four 1.0|
1.  rel=1  0.5345 minute 1.0|mile 1.0|four 1.0|first 1.0|
2.  rel=0  0.2535 minute 1.0|mile 1.0|four 1.0|

query7.0:  year 1.0|state 1.0|become 1.0|alaska 1.0|
0.  rel=1  0.5000 state 1.0|become 1.0|alaska 1.0|
1.  rel=0  0.3780 state 1.0|alaska 1.0|
2.  rel=0  0.3333 become 1.0|alaska 1.0|
3.  rel=0  0.1667 state 1.0|

query8.0:  tyson 1.0|mike 1.0|holyfield 1.0|ear 1.0|bite 1.0|
0.  rel=0  0.6761 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
1.  rel=1  0.4781 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
2.  rel=0  0.2236 tyson 1.0|holyfield 1.0|
3.  rel=0  0.1865 tyson 1.0|holyfield 1.0|

```

```
query9.0:  spaceship 1.0|moon 1.0|first 1.0|
0.  rel=1  0.4364 moon 1.0|first 1.0|
1.  rel=0  0.4364 moon 1.0|first 1.0|
2.  rel=0  0.1741 moon 1.0|
3.  rel=0  0.0000

query10.0:  win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
0.  rel=1  0.9129 win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
1.  rel=0  0.3721 prize 1.0|peace 1.0|nobel 1.0|
2.  rel=0  0.3464 prize 1.0|peace 1.0|nobel 1.0|
3.  rel=0  0.2582 prize 1.0|nobel 1.0|

query11.0:  tower 1.0|devil 1.0|
0.  rel=1  0.5774 tower 1.0|devil 1.0|
1.  rel=0  0.4472 tower 1.0|devil 1.0|
2.  rel=0  0.4082 tower 1.0|devil 1.0|
3.  rel=0  0.3651 tower 1.0|devil 1.0|

query12.0:  tallest 1.0|redwood 1.0|height 1.0|
0.  rel=0  0.5164 tallest 1.0|redwood 1.0|
1.  rel=0  0.3333 tallest 1.0|redwood 1.0|
2.  rel=0  0.3086 tallest 1.0|redwood 1.0|
3.  rel=1  0.1543 redwood 1.0|

query13.0:  lake 1.0|deep 1.0|crater 1.0|
0.  rel=0  0.5774 lake 1.0|crater 1.0|
1.  rel=0  0.4714 lake 1.0|crater 1.0|
2.  rel=1  0.3849 lake 1.0|crater 1.0|

query14.0:  singer 1.0|lead 1.0|commodore 1.0|
0.  rel=1  0.7071 singer 1.0|lead 1.0|commodore 1.0|
1.  rel=0  0.2582 commodore 1.0|
2.  rel=0  0.2182 commodore 1.0|

query15.0:  place 1.0|earth 1.0|coldest 1.0|
0.  rel=0  0.5774 place 1.0|coldest 1.0|
1.  rel=0  0.4364 place 1.0|coldest 1.0|
2.  rel=1  0.3203 earth 1.0|coldest 1.0|

query16.0:  marley 1.0|die 1.0|bob 1.0|
0.  rel=1  0.6547 marley 1.0|die 1.0|bob 1.0|
1.  rel=0  0.4804 marley 1.0|die 1.0|bob 1.0|
2.  rel=0  0.3482 marley 1.0|bob 1.0|
~~
query16.0:  marley 1.0|die 1.0|bob 1.0|
0.  rel=1  0.6547 marley 1.0|die 1.0|bob 1.0|
1.  rel=0  0.4804 marley 1.0|die 1.0|bob 1.0|
2.  rel=0  0.3482 marley 1.0|bob 1.0|

query17.0:  us 1.0|state 1.0|producer 1.0|lead 1.0|corn 1.0|
0.  rel=0  0.7303 state 1.0|producer 1.0|lead 1.0|corn 1.0|
1.  rel=0  0.4781 us 1.0|state 1.0|lead 1.0|corn 1.0|
2.  rel=1  0.2236 producer 1.0|corn 1.0|

query18.0:  mcdonald 1.0|first 1.0|build 1.0|
0.  rel=0  0.1925 mcdonald 1.0|
1.  rel=0  0.1925 mcdonald 1.0|
2.  rel=1  0.1204 mcdonald 1.0|

query19.0:  town 1.0|take 1.0|place 1.0|new 1.0|jersey 1.0|hindenburg 1.0|disaster 1.0|1937 1.0|
0.  rel=0  0.3536 take 1.0|place 1.0|hindenburg 1.0|disaster 1.0|
1.  rel=0  0.3354 hindenburg 1.0|disaster 1.0|1937 1.0|
2.  rel=1  0.1581 hindenburg 1.0|1937 1.0|

query20.0:  state 1.0|keystone 1.0|
0.  rel=1  0.4714 state 1.0|keystone 1.0|
1.  rel=0  0.2500 keystone 1.0|
2.  rel=0  0.0000
```

Report for tfIdf:

```
tfIdf errorAnalysis report
query1.0:  volcano 1.0|us 1.0|pompeius 1.0|name 1.0|give 1.0|destroy 1.0|city 1.0|ancient 1.0|
0.  rel=0  0.3612 pompeius 1.0|destroy 1.0|city 1.0|
1.  rel=0  0.2060 city 1.0|ancient 1.0|
2.  rel=1  0.0736 pompeius 1.0|city 1.0|
3.  rel=0  0.0000

query2.0:  see 1.0|michael 1.0|largest 1.0|jordan 1.0|ever 1.0|crowd 1.0|come 1.0|
0.  rel=0  0.3980 michael 1.0|largest 1.0|jordan 1.0|crowd 1.0|
1.  rel=1  0.1745 see 1.0|michael 1.0|jordan 1.0|
2.  rel=0  0.1715 michael 1.0|largest 1.0|crowd 1.0|
3.  rel=0  0.0000

query3.0:  year 1.0|purchase 1.0|happen 1.0|alaska 1.0|
0.  rel=1  0.4433 year 1.0|purchase 1.0|alaska 1.0|
1.  rel=0  0.0000 purchase 1.0|alaska 1.0|
2.  rel=0  0.0000 purchase 1.0|alaska 1.0|

query4.0:  year 1.0|wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
0.  rel=1  0.0771 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
1.  rel=0  0.0583 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
2.  rel=0  0.0000 wilt 1.0|point 1.0|chamberlain 1.0|100 1.0|

query5.0:  sorrow 1.0|river 1.0|china 1.0|call 1.0|
0.  rel=1  0.2733 sorrow 1.0|river 1.0|china 1.0|call 1.0|
1.  rel=0  0.1224 river 1.0|china 1.0|call 1.0|
2.  rel=0  0.0000 river 1.0|china 1.0|

query6.0:  run 1.0|person 1.0|minute 1.0|mile 1.0|less 1.0|four 1.0|first 1.0|
0.  rel=0  0.1890 run 1.0|minute 1.0|mile 1.0|four 1.0|
1.  rel=1  0.1858 minute 1.0|mile 1.0|four 1.0|first 1.0|
2.  rel=0  0.0000 minute 1.0|mile 1.0|four 1.0|

query7.0:  year 1.0|state 1.0|become 1.0|alaska 1.0|
0.  rel=1  0.2126 state 1.0|become 1.0|alaska 1.0|
1.  rel=0  0.1497 become 1.0|alaska 1.0|
2.  rel=0  0.1021 state 1.0|alaska 1.0|
3.  rel=0  0.0366 state 1.0|

query8.0:  tyson 1.0|mike 1.0|holyfield 1.0|ear 1.0|bite 1.0|
0.  rel=0  0.2390 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
1.  rel=1  0.1806 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
2.  rel=0  0.0000 tyson 1.0|holyfield 1.0|
3.  rel=0  0.0000 tyson 1.0|holyfield 1.0|
```

```
 84
 85 query16.0:  marley 1.0|die 1.0|bob 1.0|
 86 0.  rel=1  0.1048 marley 1.0|die 1.0|bob 1.0|
 87 1.  rel=0  0.0669 marley 1.0|die 1.0|bob 1.0|
 88 2.  rel=0  0.0000 marley 1.0|bob 1.0|
 89
 90 query17.0:  us 1.0|state 1.0|producer 1.0|lead 1.0|corn 1.0|
 91 0.  rel=0  0.3191 state 1.0|producer 1.0|lead 1.0|corn 1.0|
 92 1.  rel=0  0.2315 us 1.0|state 1.0|lead 1.0|corn 1.0|
 93 2.  rel=1  0.0439 producer 1.0|corn 1.0|
 94
 95 query18.0:  mcdonald 1.0|first 1.0|build 1.0|
 96 0.  rel=1  0.0000 mcdonald 1.0|
 97 1.  rel=0  0.0000 mcdonald 1.0|
 98 2.  rel=0  0.0000 mcdonald 1.0|
 99
100 query19.0:  town 1.0|take 1.0|place 1.0|new 1.0|jersey 1.0|hindenburg 1.0|disaster 1.0|1937 1.0|
101 0.  rel=0  0.2465 take 1.0|place 1.0|hindenburg 1.0|disaster 1.0|
102 1.  rel=0  0.1108 hindenburg 1.0|disaster 1.0|1937 1.0|
103 2.  rel=1  0.0314 hindenburg 1.0|1937 1.0|
104
105 query20.0:  state 1.0|keystone 1.0|
106 0.  rel=1  0.3394 state 1.0|keystone 1.0|
107 1.  rel=0  0.1042 keystone 1.0|
108 2.  rel=0  0.0000
109
110
```

```
query16.0:  marley 1.0|die 1.0|bob 1.0|
0.  rel=1  0.1048 marley 1.0|die 1.0|bob 1.0|
1.  rel=0  0.0669 marley 1.0|die 1.0|bob 1.0|
2.  rel=0  0.0000 marley 1.0|bob 1.0|

query17.0:  us 1.0|state 1.0|producer 1.0|lead 1.0|corn 1.0|
0.  rel=0  0.3191 state 1.0|producer 1.0|lead 1.0|corn 1.0|
1.  rel=0  0.2315 us 1.0|state 1.0|lead 1.0|corn 1.0|
2.  rel=1  0.0439 producer 1.0|corn 1.0|

query18.0:  mcdonald 1.0|first 1.0|build 1.0|
0.  rel=1  0.0000 mcdonald 1.0|
1.  rel=0  0.0000 mcdonald 1.0|
2.  rel=0  0.0000 mcdonald 1.0|

query19.0:  town 1.0|take 1.0|place 1.0|new 1.0|jersey 1.0|hindenburg 1.0|disaster 1.0|1937 1.0|
0.  rel=0  0.2465 take 1.0|place 1.0|hindenburg 1.0|disaster 1.0|
1.  rel=0  0.1108 hindenburg 1.0|disaster 1.0|1937 1.0|
2.  rel=1  0.0314 hindenburg 1.0|1937 1.0|

query20.0:  state 1.0|keystone 1.0|
0.  rel=1  0.3394 state 1.0|keystone 1.0|
1.  rel=0  0.1042 keystone 1.0|
2.  rel=0  0.0000


query9.0:  spaceship 1.0|moon 1.0|first 1.0|
0.  rel=0  0.2828 moon 1.0|first 1.0|
1.  rel=1  0.2342 moon 1.0|first 1.0|
2.  rel=0  0.0415 moon 1.0|
3.  rel=0  0.0000

query10.0:  win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
0.  rel=1  0.6907 win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
1.  rel=0  0.0338 prize 1.0|peace 1.0|nobel 1.0|
2.  rel=0  0.0300 prize 1.0|peace 1.0|nobel 1.0|
3.  rel=0  0.0000 prize 1.0|nobel 1.0|

query11.0:  tower 1.0|devil 1.0|
0.  rel=1  0.0000 tower 1.0|devil 1.0|
1.  rel=0  0.0000 tower 1.0|devil 1.0|
2.  rel=0  0.0000 tower 1.0|devil 1.0|
3.  rel=0  0.0000 tower 1.0|devil 1.0|

query12.0:  tallest 1.0|redwood 1.0|height 1.0|
0.  rel=0  0.2067 tallest 1.0|redwood 1.0|
1.  rel=0  0.0467 tallest 1.0|redwood 1.0|
2.  rel=0  0.0404 tallest 1.0|redwood 1.0|
3.  rel=1  0.0000 redwood 1.0|

query13.0:  lake 1.0|deep 1.0|crater 1.0|
0.  rel=1  0.0000 lake 1.0|crater 1.0|
1.  rel=0  0.0000 lake 1.0|crater 1.0|
2.  rel=0  0.0000 lake 1.0|crater 1.0|

query14.0:  singer 1.0|lead 1.0|commodore 1.0|
0.  rel=1  0.5164 singer 1.0|lead 1.0|commodore 1.0|
1.  rel=0  0.0000 commodore 1.0|
2.  rel=0  0.0000 commodore 1.0|

query15.0:  place 1.0|earth 1.0|coldest 1.0|
0.  rel=1  0.1667 earth 1.0|coldest 1.0|
1.  rel=0  0.1458 place 1.0|coldest 1.0|
2.  rel=0  0.0940 place 1.0|coldest 1.0|

query16.0:  marley 1.0|die 1.0|bob 1.0|
0.  rel=1  0.1048 marley 1.0|die 1.0|bob 1.0|
1.  rel=0  0.0669 marley 1.0|die 1.0|bob 1.0|
2.  rel=0  0.0000 marley 1.0|bob 1.0|
```

Report for okapiBM25:

```
query1.0:  volcano 1.0|us 1.0|pompeius 1.0|name 1.0|give 1.0|destroy 1.0|city 1.0|ancient 1.0|
0.  rel=0  0.0000 city 1.0|ancient 1.0|
1.  rel=0  0.0000 pompeius 1.0|destroy 1.0|city 1.0|
2.  rel=0  0.0000
3.  rel=1  -0.2445 pompeius 1.0|city 1.0|

query2.0:  see 1.0|michael 1.0|largest 1.0|jordan 1.0|ever 1.0|crowd 1.0|come 1.0|
0.  rel=1  0.0000 see 1.0|michael 1.0|jordan 1.0|
1.  rel=0  0.0000
2.  rel=0  -0.3365 michael 1.0|largest 1.0|crowd 1.0|
3.  rel=0  -0.3628 michael 1.0|largest 1.0|jordan 1.0|crowd 1.0|

query3.0:  year 1.0|purchase 1.0|happen 1.0|alaska 1.0|
0.  rel=0  -1.2603 purchase 1.0|alaska 1.0|
1.  rel=0  -1.4124 purchase 1.0|alaska 1.0|
2.  rel=1  -1.4983 year 1.0|purchase 1.0|alaska 1.0|

query4.0:  year 1.0|wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
0.  rel=0  -2.7068 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
1.  rel=1  -2.9147 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
2.  rel=0  -3.5542 wilt 1.0|point 1.0|chamberlain 1.0|100 1.0|

query5.0:  sorrow 1.0|river 1.0|china 1.0|call 1.0|
0.  rel=1  -1.2603 sorrow 1.0|river 1.0|china 1.0|call 1.0|
1.  rel=0  -1.5031 river 1.0|china 1.0|
2.  rel=0  -1.8172 river 1.0|china 1.0|call 1.0|

query6.0:  run 1.0|person 1.0|minute 1.0|mile 1.0|less 1.0|four 1.0|first 1.0|
0.  rel=0  -1.6168 minute 1.0|mile 1.0|four 1.0|
1.  rel=1  -2.3168 minute 1.0|mile 1.0|four 1.0|first 1.0|
2.  rel=0  -2.4324 run 1.0|minute 1.0|mile 1.0|four 1.0|

query7.0:  year 1.0|state 1.0|become 1.0|alaska 1.0|
0.  rel=0  -0.3053 become 1.0|alaska 1.0|
1.  rel=0  -0.3053 state 1.0|
2.  rel=1  -0.6107 state 1.0|become 1.0|alaska 1.0|
3.  rel=0  -0.6847 state 1.0|alaska 1.0|

query8.0:  tyson 1.0|mike 1.0|holyfield 1.0|ear 1.0|bite 1.0|
0.  rel=0  -1.3001 tyson 1.0|holyfield 1.0|
1.  rel=0  -1.5779 tyson 1.0|holyfield 1.0|
2.  rel=1  -1.6805 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
3.  rel=0  -2.1755 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
```

```
query9.0:  spaceship 1.0|moon 1.0|first 1.0|
0.  rel=0  0.0000
1.  rel=0  -0.3033 moon 1.0|
2.  rel=1  -0.3691 moon 1.0|first 1.0|
3.  rel=0  -0.3691 moon 1.0|first 1.0|

query10.0:  win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
0.  rel=0  -1.5948 prize 1.0|nobel 1.0|
1.  rel=1  -1.6970 win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
2.  rel=0  -1.6974 prize 1.0|peace 1.0|nobel 1.0|
3.  rel=0  -1.8287 prize 1.0|peace 1.0|nobel 1.0|

query11.0:  tower 1.0|devil 1.0|
0.  rel=0  -1.3715 tower 1.0|devil 1.0|
1.  rel=0  -1.5429 tower 1.0|devil 1.0|
2.  rel=0  -1.6830 tower 1.0|devil 1.0|
3.  rel=1  -2.0567 tower 1.0|devil 1.0|

query12.0:  tallest 1.0|redwood 1.0|height 1.0|
0.  rel=1  -0.7296 redwood 1.0|
1.  rel=0  -1.0109 tallest 1.0|redwood 1.0|
2.  rel=0  -1.0932 tallest 1.0|redwood 1.0|
3.  rel=0  -1.5286 tallest 1.0|redwood 1.0|

query13.0:  lake 1.0|deep 1.0|crater 1.0|
0.  rel=1  -1.2024 lake 1.0|crater 1.0|
1.  rel=0  -1.4782 lake 1.0|crater 1.0|
2.  rel=0  -1.7450 lake 1.0|crater 1.0|

query14.0:  singer 1.0|lead 1.0|commodore 1.0|
0.  rel=1  -0.3423 singer 1.0|lead 1.0|commodore 1.0|
1.  rel=0  -0.6681 commodore 1.0|
2.  rel=0  -0.7823 commodore 1.0|

query15.0:  place 1.0|earth 1.0|coldest 1.0|
0.  rel=1  -0.4104 earth 1.0|coldest 1.0|
1.  rel=0  -0.9670 place 1.0|coldest 1.0|
2.  rel=0  -1.1911 place 1.0|coldest 1.0|

query16.0:  marley 1.0|die 1.0|bob 1.0|
0.  rel=0  -1.3988 marley 1.0|bob 1.0|
1.  rel=0  -1.4535 marley 1.0|die 1.0|bob 1.0|
2.  rel=1  -1.9236 marley 1.0|die 1.0|bob 1.0|
```

```
query17.0:  us 1.0|state 1.0|producer 1.0|lead 1.0|corn 1.0|
0.  rel=1  -0.7862 producer 1.0|corn 1.0|
1.  rel=0  -0.8435 us 1.0|state 1.0|lead 1.0|corn 1.0|
2.  rel=0  -1.6865 state 1.0|producer 1.0|lead 1.0|corn 1.0|

query18.0:  mcdonald 1.0|first 1.0|build 1.0|
0.  rel=1  -0.5450 mcdonald 1.0|
1.  rel=0  -0.8593 mcdonald 1.0|
2.  rel=0  -0.8593 mcdonald 1.0|

query19.0:  town 1.0|take 1.0|place 1.0|new 1.0|jersey 1.0|hindenburg 1.0|disaster 1.0|1937 1.0|
0.  rel=0  -0.5208 take 1.0|place 1.0|hindenburg 1.0|disaster 1.0|
1.  rel=1  -0.7956 hindenburg 1.0|1937 1.0|
2.  rel=0  -1.3151 hindenburg 1.0|disaster 1.0|1937 1.0|

query20.0:  state 1.0|keystone 1.0|
0.  rel=1  0.0000 state 1.0|keystone 1.0|
1.  rel=0  0.0000
2.  rel=0  -0.1772 keystone 1.0|
```

Report forLucene:

```
query1.0:  volcano 1.0|us 1.0|pompeius 1.0|name 1.0|give 1.0|destroy 1.0|city 1.0|ancient 1.0|
0.  rel=0  0.2701 pompeius 1.0|destroy 1.0|city 1.0|
1.  rel=0  0.1611 city 1.0|ancient 1.0|
2.  rel=1  0.0691 pompeius 1.0|city 1.0|
3.  rel=0  0.0000

query2.0:  see 1.0|michael 1.0|largest 1.0|jordan 1.0|ever 1.0|crowd 1.0|come 1.0|
0.  rel=0  0.3340 michael 1.0|largest 1.0|jordan 1.0|crowd 1.0|
1.  rel=0  0.1888 michael 1.0|largest 1.0|crowd 1.0|
2.  rel=1  0.1494 see 1.0|michael 1.0|jordan 1.0|
3.  rel=0  0.0000

query3.0:  year 1.0|purchase 1.0|happen 1.0|alaska 1.0|
0.  rel=1  0.4922 year 1.0|purchase 1.0|alaska 1.0|
1.  rel=0  0.1040 purchase 1.0|alaska 1.0|
2.  rel=0  0.0968 purchase 1.0|alaska 1.0|

query4.0:  year 1.0|wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
0.  rel=1  0.2974 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|
1.  rel=0  0.2818 wilt 1.0|point 1.0|chamberlain 1.0|100 1.0|
2.  rel=0  0.2557 wilt 1.0|score 1.0|point 1.0|chamberlain 1.0|100 1.0|

query5.0:  sorrow 1.0|river 1.0|china 1.0|call 1.0|
0.  rel=1  0.5181 sorrow 1.0|river 1.0|china 1.0|call 1.0|
1.  rel=0  0.3822 river 1.0|china 1.0|call 1.0|
2.  rel=0  0.1539 river 1.0|china 1.0|

query6.0:  run 1.0|person 1.0|minute 1.0|mile 1.0|less 1.0|four 1.0|first 1.0|
0.  rel=0  0.3015 run 1.0|minute 1.0|mile 1.0|four 1.0|
1.  rel=1  0.2650 minute 1.0|mile 1.0|four 1.0|first 1.0|
2.  rel=0  0.0637 minute 1.0|mile 1.0|four 1.0|

query7.0:  year 1.0|state 1.0|become 1.0|alaska 1.0|
0.  rel=1  0.2837 state 1.0|become 1.0|alaska 1.0|
1.  rel=0  0.1961 become 1.0|alaska 1.0|
2.  rel=0  0.1719 state 1.0|alaska 1.0|
3.  rel=0  0.0668 state 1.0|

query8.0:  tyson 1.0|mike 1.0|holyfield 1.0|ear 1.0|bite 1.0|
0.  rel=0  0.3906 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
1.  rel=1  0.2820 tyson 1.0|holyfield 1.0|ear 1.0|bite 1.0|
2.  rel=0  0.0643 tyson 1.0|holyfield 1.0|
3.  rel=0  0.0511 tyson 1.0|holyfield 1.0|

```

```
query9.0:  spaceship 1.0|moon 1.0|first 1.0|
0.  rel=0  0.2712 moon 1.0|first 1.0|
1.  rel=1  0.2516 moon 1.0|first 1.0|
2.  rel=0  0.0671 moon 1.0|
3.  rel=0  0.0000

query10.0:  win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
0.  rel=1  0.9640 win 1.0|prize 1.0|peace 1.0|nobel 1.0|1992 1.0|
1.  rel=0  0.1530 prize 1.0|peace 1.0|nobel 1.0|
2.  rel=0  0.1386 prize 1.0|peace 1.0|nobel 1.0|
3.  rel=0  0.0898 prize 1.0|nobel 1.0|

query11.0:  tower 1.0|devil 1.0|
0.  rel=1  0.3086 tower 1.0|devil 1.0|
1.  rel=0  0.2236 tower 1.0|devil 1.0|
2.  rel=0  0.2010 tower 1.0|devil 1.0|
3.  rel=0  0.1771 tower 1.0|devil 1.0|

query12.0:  tallest 1.0|redwood 1.0|height 1.0|
0.  rel=0  0.2691 tallest 1.0|redwood 1.0|
1.  rel=0  0.1219 tallest 1.0|redwood 1.0|
2.  rel=0  0.1082 tallest 1.0|redwood 1.0|
3.  rel=1  0.0324 redwood 1.0|

query13.0:  lake 1.0|deep 1.0|crater 1.0|
0.  rel=0  0.1956 lake 1.0|crater 1.0|
1.  rel=0  0.1460 lake 1.0|crater 1.0|
2.  rel=1  0.1132 lake 1.0|crater 1.0|

query14.0:  singer 1.0|lead 1.0|commodore 1.0|
0.  rel=1  0.6552 singer 1.0|lead 1.0|commodore 1.0|
1.  rel=0  0.0829 commodore 1.0|
2.  rel=0  0.0684 commodore 1.0|

query15.0:  place 1.0|earth 1.0|coldest 1.0|
0.  rel=0  0.3457 place 1.0|coldest 1.0|
1.  rel=0  0.2394 place 1.0|coldest 1.0|
2.  rel=1  0.1923 earth 1.0|coldest 1.0|

query16.0:  marley 1.0|die 1.0|bob 1.0|
0.  rel=1  0.4508 marley 1.0|die 1.0|bob 1.0|
1.  rel=0  0.3042 marley 1.0|die 1.0|bob 1.0|
2.  rel=0  0.1649 marley 1.0|bob 1.0|

query17.0:  us 1.0|state 1.0|producer 1.0|lead 1.0|corn 1.0|
0.  rel=0  0.4566 state 1.0|producer 1.0|lead 1.0|corn 1.0|
1.  rel=0  0.3889 us 1.0|state 1.0|lead 1.0|corn 1.0|
2.  rel=1  0.0728 producer 1.0|corn 1.0|

query18.0:  mcdonald 1.0|first 1.0|build 1.0|
0.  rel=0  0.0472 mcdonald 1.0|
1.  rel=0  0.0425 mcdonald 1.0|
2.  rel=1  0.0260 mcdonald 1.0|

query19.0:  town 1.0|take 1.0|place 1.0|new 1.0|jersey 1.0|hindenburg 1.0|disaster 1.0|1937 1.0|
0.  rel=0  0.2403 take 1.0|place 1.0|hindenburg 1.0|disaster 1.0|
1.  rel=0  0.1491 hindenburg 1.0|disaster 1.0|1937 1.0|
2.  rel=1  0.0475 hindenburg 1.0|1937 1.0|

query20.0:  state 1.0|keystone 1.0|
0.  rel=1  0.4208 state 1.0|keystone 1.0|
1.  rel=0  0.1558 keystone 1.0|
2.  rel=0  0.0000

```