CSI 5360 Natural Language Processing

Spring 2018

Programming Homework 2

Due date: 3/2 (Fri) 11:59pm

The goal of this homework is to apply the TD-IDF model you developed last time for information retrieval and measure the performance of such algorithm.

You will be given a set of text documents and queries. Your task is to retrieve the document based on the cosine similarity of the query and the set of documents. For each query you will be given the results of the query for comparison's sake. Thus you will need to evaluate the result of the queries with the given result.

Implementation

You should implement the following function

Retrieve(Corpus, q, rank, minsim)

- *Corpus* is a CorpusReader_TFIDF object
- *q* is the string that forms the query
- *rank* is a keyword parameter, if present, the function should return the top <rank> documents from the query. If rank is negative, then the output should be ranked. (e.g. -15 will returned the top 15 documents ranked by cosine similarity [from most similar to least similar]
- *minsim* is a keyword parameter, if present, the function should return all documents that the have cosine similarity of at least *minsim*
  (If both *rank* and *minsim* are presence, than both condition are to be satisfied)
  (If neither parameters are mentioned, the function do nothing and return an empty list)

The function is to return the list of documents (based on the index in the Corpus).

Task

Your goal is to use the function implemented to answer the question, "How should I set up our tfidf model [various tf, idf calculation methods; stemming; stopwords etc.] such that it will perform the best for information retrieval?]

You will be given the following

- Text documents: A set of text documents, each stored as a separate file in a directory. You can assume nothing else stores in that directory.
- Query file: A text file that contains a list of queries. Each line start with a number (query number), which is used to reference the query in other files. The rest of the line contains a list of keywords for the query.
- Result file: A text file that contains the correct result of a query. Each line will start with a number (query number) and then the file name of a file. This means that the file should be retrieved by the query. Anything on the line after the filename is to be ignored.

Notice that a query may retrieve multiple documents, and a document may be retrieved by multiple queries.

- Configuration file: A text file that store the information above, plus the information about the query:
    - The first line store the name of the directory which store the text documents
    - The second line store the name of the query file
    - The third line store the name of the result file
    - The fourth line has one of the following format
        - r <number>: the algorithm should return the top <number> documents.
        - s < number>: the algorithm should return all documents with similarity of at least <number>

Any extra words on each line of the configuration file is ignored.

Your task is to write a program that read in the configuration file (you should check if the file then based on the information on the file, read the documents, create the various TFIDF models, and run the queries and obtain results.

You are to apply various preprocessing to create different corpus for retrieval purposes. You should implement the following:

Corpus	Stemming	Remove stop words	tf	idf
1	N	N	raw	inverse
2	N	N	log	smoothed
3	Y	Y	raw	inverse
4	Y	Y	raw	smoothed
5	Y	Y	log	inverse
6	Y	Y	log	smoothed

Output

Your program should output the following:

1. For each corpus above, and for each query, return the recall and precision for that query. Each (corpus query, recall, precision) should be one line. You should print corpus number, query number, recall and precision in such order. Recall and precision should be to 4 decimal places. Output results for all corpus and query pairs first then go to 2

2. For each corpus, output the overall average recall and precision, which is the average recall and precision for each query.

3. You should also calculate “overall recall” / “overall precision”, which is calculated by treating by counting all queries together as one.

The following table illustrates points 2 and 3.

Assume we have 3 queries q1, q2, and q3. The second column listed the documents that SHOULD be retrieved. Assume that when we run the query on the corpus, we retrieved a different set of documents, listed in column 3. We list the recall and precision for each query in column 4 and 5, and we list the overall precision and recall for the last column

Query	Correct Answer	Retrieved by query	Recall	Precision
q1	3 5 7	1 3 5 8	2/3	2/4
q2	1 4 8 10 11	1 4 9 12	2/5	2/4
q3	4 8 9	4 7 11 12	1/3	1/4
Overall			(2+2+1)/(3+5+3)	(2+2+1)/(4+4+4)

For step 2 and 3, you should have one line for each corpus, it should be a list of five numbers in the order of (corpus number, average recall, average precision, overall recall, and overall precision). Once again, all precision and recall is to 4 decimal places.

Report

You should also have a report to compare the performance of the various procedures. You should plot a graph of MAP (Mean average Precision) using overall recall and precision for various methods and use them for the basis of comparison.

You should also have a page in the report to discuss whether stemming and/or removing stop words is the right strategy. You should any other measures you feel appropriate to justify your argument. Your report should be about 2-3 page long (although there is no limit either way).