

IR - Assignment 2 Part-2 (Ranked retrieval)

Deadline: 14/09/2018

In this assignment, you will implement ranked retrieval using two simple ranking techniques:

1. Tf-idf based ranking (cosine similarity)
2. Unigram language model based ranking

You need to retrieve the documents based on the following approach:

Inverted Index way: Create an inverted index. This index will help you obtain the list of documents which have at least one query term present. Then only those documents will need to be processed, because the other documents are not relevant to the query anyway.

For each query-document pair, you also have a relevance judgement (that you manually ranked in part-1). Using these relevance scores, calculate the NDCG score for the document set returned by your search engine.

Dataset Description:

1. query.txt contains total 82 queries, which has 2 columns query id and query.
2. alldocs.rar contains documents file named with doc id. Each document has set of sentences.
3. output.txt contains 50 relevant documents (doc id) for each query

Link: https://drive.google.com/file/d/1LOW6HJE_Y7lftHg58Zqccq70c_X78bh5/view?usp=sharing

Part1:

1. Represent each query and document as Inc.ltc tf-idf vector where the corpus will be all the documents in alldocs.rar merged. Use their cosine similarity values to rank the documents.
2. For each query first retrieve top 50 documents using your code. Report precision, recall, f-measure for each query (in a table format) as well as the average.
3. Report the average time of retrieval for Inverted index way

Part2:

1. Compute the $P(q|d)$ for each document. The documents with a higher probability of generating the query ,i.e., higher value of $P(q|d)$ will be ranked higher.
2. For each query first retrieve top 50 documents using your code. Report precision, recall, f-measure for each query (in a table format) as well as the average.
3. Report the average time of retrieval for Inverted index way.

Deliverables:

Submit a tar.gz file on moodle containing the following:

|-src
|-results

1. Your src folder containing your python code.
2. Under results there should be text files for the following information:
 - a. A table containing the NDCG score for each query (as shown below).
 - b. A table containing the precision, recall, score for each query .
 - c. Time taken for Naive search and Indexed search

Name your file as A2_<RollNumber>.tar.gz

Query	TF-IDF NDCG score	Language Model (NDCG score)
Q1	0.00001	0.003
Q2	0.02	0.01

Reference:

- NDCG is Normalised Discounted Cumulative Gain. It is used for measuring the quality of retrieved results. You can use the scikit-learn library function for computing this. (You will have to understand NDCG first however, in order to be able to use that :))
- Go through the chapter-11 of Manning for understanding “Unigram language model based ranking”