Information Retrieval Project - Task 1

Group 2

October 21, 2021

1 Group Details

- Adhikansh Singh - 17CS30002
- Ankit Bagde - 17CS30009
- Kousshik Raj M - 17CS30022
- Shivam Kumar Jha - 17CS30033

2 Approach

The following is a brief description of the approach taken to solve the problems.

- **Task 1A:** We iterate through each document in the data folder and extract the actual content from it. Afterwards, we convert everything to lower case, remove punctuation marks and then tokenize them. After tokenizing, we remove the stop words and then lemmatize them (assuming that the words are nouns). Then we index these tokens. For each token, we store the postings as a list of (document, frequency) pair. Finally, we sort the posting list for each term, so that merges can be carried out efficiently.
- **Task 1B:** We extract the query Id and the query title for all the queries in the query file. Similar to how we parsed the documents, we parse these queries, except that we convert the tokens back to string and then store them.
- **Task 1C:** We load the inverted index as well as the queries. For each query, we split them into tokens, and sort them according to increasing document frequency. We then sequentially merge them one by one linearly. The final resulting documents after all merges are the results, and we store them.

3 Assumptions

- A proper lemmatizing needs a PoS tag as well, but here we just use the default mode (i.e. noun).
- In the inverted index, instead of only storing the document Ids for each term, we also save the number of occurrences in that document, which will help us calculate the term frequencies in the next task.
- We are storing the queries as string instead of tokens in the queries file.

4 Software Requirements

- Python version - 3.7
- Python libraries used - *os*, *sys*, *re*, *pickle*, *nltk*
- Libraries that need installation - *nltk* (with *stopwords* and *punkt* datasets)

4.1 Running Time

- **Task 1A:-** Indexing takes approximately **1 hour**
- **Task 1B:-** Parsing the queries takes approximately **2 seconds**
- **Task 1C:-** Retrieving the documents for all queries takes approximately **5 seconds**