IR ASSIGNMENT 1 REPORT

MOHAMMAD KAIF

All code files are properly commented, and GitHub has been used for version control.

Q1) The first question involved preprocessing the input text files and storing the preprocessed results. To achieve the same, I did the following:

lowercase the text: used the .lower() function from the standard library

perform tokenization: used the word_tokenize() function from the nltk library

remove stopwords: used the stopwords corpus from nltk and removed words from the tokenized word list if it was found in the stopwords corpus

remove punctuations: I used the isalnum() function from the standard library in order to get rid of the punctuation marks and have alphanumeric characters in the preprocessed corpus.

remove blank space tokens: used the strip() and split() functions from the standard library in order to remove the blank space tokens.

The os directory was used to list files in a particular directory.

Apart from this, basic file i/o was used.

Q2) I created a dictionary where the key is the word and with each key, it contains a set of files which contain the given word. This is my unigram inverted index where the keys are called the dictionary and with each word is associated a postings list. It searches for the word in the unigram inverted index and retrieves the list of files processing the query from left to right.

Q3) I created a dictionary where the key is the word, with each dict[word], there is a dictionary that contains the file as the key and a value as a list of the index of

occurrences of the word.

You can understand it as follows:

dict[word][file_id] = list of word_index_of_occurence.

After this, i traversed the dictionary to check for the first word of the file and then recursively check for the next few words of the phrase.

In both q2 and q3, the pickle library was store the dump files of unigram inverted index and positional index. Apart from this, basic file io and the os module to read directories was used.