Information Retrieval

Assignment 2

Dibyendu Roy Chaudhuri
MT19034

Q1) CLI tool

Methodology:

1. For creating an inverted index, I have used a dictionary of lists. In the list first index is used to show the size posting list and the second index is the posting list where indexes of every document are stored in ascending order.
2. I have also maintained another list where document names are stored in ascending order.
3. A frequency vector matrix is also created to perform cosine similarity operations between query and document.
4. Three types of similarity checking algorithm are used-
    a. Jaccard Coefficient
    b. Tf-Idf based document retrieval
    c. Cosine similarity
5. I have used 5 types of formula to calculate Tf-
    a. Binary weight
    b. Raw frequency weight
    c. Term frequency
    d. Log normalization
    e. Double normalization K
6. Zone indexing is also used to give extra attention to the title
    a. The value for g is shifted between (0.6 to 1) to observe the changes in the result.
7. Zone indexing is used for only Tf-Idf based document retrieval and for cosine similarity based document retrieval.
8. Model returns Top K most similar documents.

Preprocessing Step:

1. Tokenize is done on following delimiters-
   ("\s", "-", ".", "@", "t", "\n", ‘"’, ">", ",", "?", ":", "{", "(", "[", ")", "}", "]", "<", "_", "!", "/", "|", "\", "*", "=", "^")
2. Convert whole text into lower case.
3. Lemmatization is used.
4. Pickle library is used to store the intermediate inverted index and vector table.

Assumption:

1. I have taken stop words as valid words.
2. I have assumed the title is not present in the documents. So I just have fetched titles from index files.
3. In the index file, the title also contains the author’s name.

Q2) Edit Distance-based word correction

Methodology:

1. English dictionary words are stored in a list.
2. When a query is given-
   a. For each word, model checks if that word is already present in the list.
   b. If present in the list then, simply move to the next word.
   c. Otherwise use the edit distance string algorithm for every word present in that list.
   d. Return top k words whose operation cost is minimum.

Preprocessing Step:

1. Query is tokenized on following delimiters- ("\s", "-", ".", "@", "t", "\n", ‘"’, ">", ",", "?", ":", "{", "(", "[", ")", "}", "]", "<", "_", "!", "/", "|", "\", "*", "=", "^")
2. Convert whole text into lower case.
3. Lemmatization is used.