Assignment 4

Submitted by

Swati Verma

MT19073

Question 1

Assumptions:

1. No metadata has been removed from any of the documents.

Preprocessing steps:

1. Words in all the documents are converted to lowercase.
2. Punctuations have been removed from all the documents.
3. Numbers are converted into words.
4. Expansion such as don't to do not has not been handled because the library was not working on my system.
5. Stopwords have been removed from all the documents using nltk library.
6. To convert surface words to root word lemmatization is used instead of stemming because lemmatization gives meaningful words.
7. All the above steps are done both for documents and query which is given by the user.

Methodology:

1. A vector is made corresponding to each document. To implement this, a dictionary has been created where key represents the documents and values represent another dictionary of vocabulary size where keys are all the terms in the vocabulary and values are tf-idf values corresponding to each term for each document.
2. After this query is taken from the user and a query vector has been created which contains term and tf-idf value pair for all the vocabulary terms.
3. At last, cosine similarity is calculated directly from the tf-idf values present for each term in document vector and query vector respectively. Users are asked to enter the value of k and top k relevant documents are then retrieved on the basis of cosine similarity.

Question 2

Assumptions:

1. No metadata has been removed from any of the documents.
2. Users will enter the correct names of the document to mark them as relevant.
3. The relevant documents entered by the user belongs to the files corresponding to the ground truth documents only
4. After each iteration relevant documents are shown as 'document name *'.

Methodology:

1. A vector is made corresponding to each document. To implement this, a dictionary has been created where key represents the documents and values represent another dictionary of vocabulary size where keys are all the terms in the vocabulary and values are tf-idf values corresponding to each term for each document.
2. After this query is taken from the user and a query vector is created which contains term and tf-idf value pair for all the vocabulary terms.
3. Cosine similarity is then calculated directly from the tf-idf values present for each term in the document vector and query vector respectively. Users are asked to enter the value of k and top k relevant documents are then retrieved on the basis of cosine similarity.
4. From the top k retrieved documents, the user is asked to enter the value of 'p' so that p%k documents can be marked as relevant. User is then asked to enter 'p' relevant documents.Here we are assuming that the documents entered by the user belong to the files in ground truth provided to us. Documents marked by users as relevant are taken as one rel_list and remaining (k- rel documents) are taken as non relevant documents and then using rocchio's formula a new query vector is calculated with $\alpha = 1$, $\beta = 0.7$, and $\gamma = 0.25$. After that cosine similarity is calculated between the new query vector and all the documents and again top k new documents are retrieved.
5. Taking ground truth documents as a relevant document set, precision, recall and mean average precision is calculated at each point when a document is retrieved from the top k

new document set. And a graph is plotted between precision and recall. The curve obtained looks like sawtooth.

6. For the second iteration again the user is asked to enter ‘p’ relevant documents. Documents which are marked relevant in the previous iteration are not taken in the non-relevant document set.
7. Step 4, 5 and 6 is repeated again four times to obtain four different sets of relevant documents so that we can get better precision.
8. A graph is plotted between four iterations and map values obtained after every iteration.
9. At the end t-Distributed Stochastic Neighbor Embedding or t-SNE is used to represent the distribution of all the query vectors obtained after every iteration. t-SNE is used for dimensionality reduction of the multidimensional data so that we can visualize our data in a better way.