Tutorial 3

March 2020

Deliverable: Please solve the following questions. Provide a PDF for Q1 and Q2 with the detailed answeres of the questions. For Question 3 provide a ipython code with a text file reporting results as required. Submit a single ZIP in moodle with all these.

1 Q1

Consider three documents - D_1, D_2, D_3:

- D_1: "Natural language processing is becoming important since soon we will begin talking to our computers."
- D_2: "If computers understand natural language they will become much simpler to use."
- D_3: "Speech recognition is the first step to build computers like us."

Answer the following with respect to the above set of 3 documents after text normalization (stop word removal and lemmatization) has been done on all 3 documents.

(A) What is the vocabulary V?

(B) What are the number of bigrams and trigrams in D_2?

(C) What will be the BoW document vector for document D_3 if we are using a tf (term-frequency) based document vector?

(D) Suppose you have the following two 4-dimensional word vectors for two words w_1 and w_2 respectively: w_1 = (0.2, 0.1, 0.3, 0.4) and w_2 = (0.3, 0, 0.2, 0.5) What is the cosine similarity between w1 and w2? Are the words w1 and w2 similar or dissimilar?

(E) Device the word embedding of "Natural", "Language" based on previous three context words and "tf-idf" as weighting functions.

2 Q2

You are given a corpus C, with d documents and a vocabulary V . The corpus is represented as a matrix of C with a size $d \times |V|$. Each document d_i is a $1 \times |V|$ vector such that d_{ij} represents the number of times the word j appears in document i.

(A) Using the matrix C, write the expression to obtain a word to word co-occurrence matrix W for words in V . Entry w_{ij} implies how often the words w_i and w_j co-occur. The diagonal entries can be set to zero after obtaining W.

(B) Calculate the all pairs Dice coefficient and cosine similarity for the words 'bank,river and shore'. The vectors for the words should be obtained from the matrix W. Matrix C is given below.

	bank'	'fast'	'flow'	'mud'	'river'	'shore'	'tree'	'water'
Doc1	1	0	0	0	1	1	0	1
Doc2	0	1	1	0	1	0	1	1
Doc3	1	0	1	0	0	0	0	1
Doc4	1	0	0	0	1	1	0	1
Doc5	0	0	0	1	1	0	1	1

3 Q3

Coding Assignment: Dataset: All necessary data is available at:
`https://drive.google.com/open?id=1Pvc9MBMc2fFO2vTB4BtgaYs4YhW_Pb0-`
The folder Assignment1 contains query.txt, output.txt, alldocs.rar.

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

You need to do the following:

1. : Write a code to calculate the word representation for documents and query phrases using (word X doc) metric and "tf-idf" weighting function
2. compute the relevant documents corresponding to a query based on following score Score

$$(Q, D) = W_Q.W_D^T$$

   Here W_Q is the word representation matrix for query and W_D is the word representation matrix for document.

3. select the top 50 documents based on the score for each query
4. Report the jaccard coefficient between the obtained results and the provided output.txt