CIS 410/510: Natural Language Processing
Fall 2019

due 11:55pm Tuesday, Oct. 22, 2019

1 Problem 1

[40 points]

[Document Similarity] Suppose our pets have produced two documents:
D1 = [woof woof meow] D2 = [woof woof squeak]

(a) What is the cosine similarity of D1 and D2, not using idf weighting? [15 points]

(b) What is the cosine similarity if idf weighting is used? [15 points]

(c) How would the answer to (b) change if we added a third document:

D3 = [meow squeak]

to the collection? [10 points]

2 Problem 2

[40 points]

[Naive Bayes and smoothing] Do exercises 4.1 and 4.2 in third (on-line) edition of the text (`https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf`) (page 81). Show the intermediate steps in your calculation. Compute using probabilities, not logs of probabilities (so instead of adding logs of probabilities, you multiply probabilities).

[20 points for 4.1] and [20 points for 4.2]

3 Problem 3

[20 points]

[`Word2Vec` with `gensim` and `nltk`] Professor Heidi Kaufman from the Department of English would like to conduct research on the black rebellions in nineteenth-century print culture. Using "rebellion" and "slave" as the keywords, she would like to search for these words in a large collection of articles in the nineteenth century to understand the time and location distributions of such words in the documents.

Prof. Kaufman discussed with Fred, a student in the Department of Computer and Information Science to see if he could help to do this task. Fred suggested that Prof. Kaufman could extend the keyword list to include similar words using the natural language processing technologies. This could help to improve the coverage of the search and provide a better estimations about the rebellions.

In this problem, we will implement Fred's idea using `word2vec`, the word representation/embedding model we learn in class. We will use two popular libraries in `python` (i.e., `gensim` and `nltk`) for this purpose.

In order to complete this problem, please do the following steps. Note that you are allowed to search the Internet for whatever you need to do this problem (i.e., Problem 3).

1. Learn to install `python`, `gensim`, and `nltk` on your machine. `conda` might be helpful.

2. Use the Brown corpus (that can be obtained from `nltk`) to train the `word2vec` model using the `gensim` library (you can choose either the CBOW or Skip-gram model). While there are many resources online that explain how to do this, I find this link very helpful (please acknowledge if you use it in your submission):

   `https://www.kaggle.com/jihyeseo/word2vec-gensim-play-look-for-similar-words`

3. Show the top ten words that are most similar to "rebellion" and "slave" based on the cosine similarity of the word vectors in the `Word2Vec` model you trained in step 2. Please include the cosine similarity for each word you report.

4. Redo step 3, but instead of using the `word2vec` model trained in step 2, use the 300d pre-trained `word2vec` model provided by Google at `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit`. You will need to download this file (about 1.6GB) and load it (using `gensim`) to retrieve the most similar words. More information about this pre-trained model can be found at `https://code.google.com/archive/p/word2vec/` while the Kaggle link above might help to show how to load the vectors. Do you see any difference between the lists of words in steps 3 and 4?