CSC 495/693: Natural Language Processing, Spring 2023 Assignment 1

Assigned Tuesday January 31, due Wednesday February 15. Max points: 100.

In this assignment, you will practice text prepossessing techniques and train your own the word vectors.

Programming Requirements

You do not have to implement the algorithms but are expected to understand and know how to use them. You have been provided with 1000 movie comments as the dataset "comments1k.zip".

- Please use Python3, not Python2.
- Define a function for each question.

1. Text Prepossessing (40 points)

Given a NLP dataset, we would like to first analyze it and prepare it for downstream applications through text prepossessing techniques. Use Spacy, NLTK or other related python libraries to finish the following tasks.

1) Split comments into sentences and report the average number of sentences per comment.
2) Do tokenization for the dataset and report the average number of tokens per comment.
3) Without considering punctuation, how many words are in each comment on average?
4) Choose any necessary text prepossessing techniques for the dataset to generate a corpus for word embeddings and explain your reasons.

2. Word Embedding (60 points)

Word embeddings are a form of word representation that bridges the human understanding of language to that of a machine using vectors. These vectors capture hidden information about a language, like word analogies or semantic. Use Gensim, GloVe or other related python libraries to finish the following tasks.

1) Train word embeddings vectors using word2vec method with CBOW model (vector_size=100, window=5, min_count=1). Report your computer's parameters (or Google Colab) and the time used for training.
2) Train word embeddings vectors using GloVe method. (vector_size=100, window_size=5, vocab_min_count=1)
3) Evaluate the model trained in 1) and 2) by analyzing the 10 most similar words to each of the following word: *"movie", "music", "woman", "Christmas*". Which model's output looks more meaningful?
4) Train word embeddings vectors using word2vec method with CBOW model and set the vector size as 1, 10, 100 separately. Evaluate the three embedding models. Which one has the best result? Explain.
5) Suppose we use word2vec to train word vectors with window size as 3. Given a sentence "*Very good drama*", it will be transferred to the training set with instances X and corresponding labels Y. If we choose skip-gram model, what are X and Y in the training set? If we choose CBOW model, what are X and Y in the training set?

Writeup

Prepare a writeup on your experiments by using any of the following template:

- ACM (https://www.acm.org/publications/proceedings-template/)
- IEEE (https://www.ieee.org/conferences/publishing/templates.html)

Write down any further insights or observations you made while implementing and running the program. Especially interesting insights may be awarded extra points. You may also receive extra points for well-written code with clear comments and runs efficiently. Conversely, poorly written, or not following the ACM/IEEE format, or hard to understand and inefficient code will lose points.

What to turn in

You will turn in:

1. Your writeup, and
2. Your source code. You may include a readme if needed (e.g. if you wish to bring anything to my attention). Please ensure your code is well documented. **I will not be able to spend a lot of time debugging your code if it crashes during our testing.**

To turn in your code and writeup, use Canvas. Prepare a zip file with all your files and name it <yourname>_prog1.zip. **This zip file should only contain your writeup, source code and readme (if needed) and not executables/object files/data files/unmodified code/anything else, and must be timestamped by the due date to avoid a late penalty.**