Natural Language Processing, Spring 2023, Assignment 3

Instructor: Behrooz Mansouri (behrooz.mansouri@maine.edu)

Due: March 22, 2023

Notes for submission:

1. Submit your file(s) with the correct naming as: NLP_Assignment3_StudentName

2. Two files should be uploaded, one .zip file having all the codes (directory is zipped, and it is named codes) and one .pdf file. There would be a penalty for uploading wrong formatted files. Any other formatting will be ignored

3. Codes should be well-structured with comments to run

4. Codes should be available on your GitHub Repo. Failure to have codes publicly available, results in a 20% reduction in your grade. Make sure to include the GitHub link in your PDF file.

5. Any assumptions made by students should be explicitly mentioned in the submitted document

6. Answers to the questions should be easy to detect. For your codes, name the .ipynb files according to the question numbers (e.g., Question1.ipynb). In your document, use the question number and just write your answer. You should not have the question itself in your document. (e.g., Question 1. Each question…)

Consider the Posts_law.xml file from the Law stack exchange (as in lab 3).

Question 1 (40%)**: Finding Similar Questions**

On Stack Exchanges, questions can have duplicates, meaning that there is an almost identical question already asked before. The Internet Archive provides this information. For this assignment, this information is already processed and is available for you in the file "**duplicate_questions.tsv**". The first column is the question id for a target question, and the other columns are the ids of similar questions.

Our goal is to use fastText embedding to find the most similar question for a given question. FastText is an n-gram embedding model, providing embeddings for words. To get the embedding for a post, you can simply average the embeddings of its words. This code is

provided for you in the file "**Finding Similar Questions.ipynb**". You will train a skip-gram fastText model with your parameters of choice. You will train the model once all the questions (both title and body) and answer.

It is strongly recommended that you save your model by modifying the existing code so that re-using it is easy for you. After your model is trained, for each question in the test set, find the most similar question. You will do this experiment twice, with the same model, once just using the titles, and another time using just the bodies.

After finding similar questions, you will calculate the average P@1 value. P@1 is 1 for a test question if its most similar question is identified correctly, otherwise is 0. Note that some questions might have more than one correct similar question; any of them detected as similar will lead to P@1 of 1.

Discuss your results based on P@1 values, and describe if using titles and bodies provided different or similar results.

Grading: 10% coding 30% detailed analysis

Question 2: (50%): You are about to repeat the previous experiment, but this time you are not using cosine similarity. Each question is represented with a vector size of |d|, where |d| is the dimensionality you used in the fastText model. You will train a Feedforward neural network for a binary classification that predicts whether two questions are similar or not. For training, you have to build pairs of positive and negative examples. In the files, you already have positive samples. You can choose other random questions that are not in the positive sample, as negative pairs. After generating the initial samples, you will divide the data into 80-10-10 splits for training, validation, and test. Note that the test set is never seen anywhere in the training process.

Design a network of your choice. After training your model, provide a diagram of training-validation loss and discuss at which epoch you saved your model for testing. Then discuss the result of your model on test sample by reporting the average P@1, providing examples for true positive, true negative, false positive, and false negative.

Grading: 20% coding 30% detailed analysis.

Question 3 (10%)**:** What are the differences between Skipgram and Continuous bag of words approaches? Describe the advantages and disadvantages of each.