Assignment 2: Sequence Learning

Total marks: 20

Submission process: Do this assignment in your preferred notebook (Jupyter, Colab any will do) and submit the notebook after you run all the blocks. Name the notebook using only your student ID-- no other words or characters (this is mandatory and I will deduct marks if you do not do it). You will have one chance to submit, so don't make a mistake. Your submission deadline is January 11, 2024, 11:59pm and no extension will be given whatsoever.
Submission link: https://forms.gle/HBwHE28ieCuSXBzPA You will need to be logged in using your bracu email. Do not include the IMDB file or the GloVe file (they are way too big).

Preamble: Setting up environments for this assignment

Hopefully you still have your environment from Assignment 1 ready. You will need some more libraries on top of that.

1. Install Numpy, Scipy and Scikit-Learn if you have not already. Your python installation should have it, but just make sure they are there.
2. The bonus part of the assignment will require you to build a neural net classifier for the task. For that you will need Tensorflow, Keras or PyTorch. For learning purposes, I prefer Keras. To install Keras (that uses Tensorflow as a backbone), follow this tutorial: https://www.activestate.com/resources/quick-reads/how-to-install-keras-and-tensorflow/ You will need this tutorial up until the "Get a version of Python, pre-compiled with Keras and other popular ML Packages" section.

Useful tutorial: for all the questions, you will find this tutorial extremely useful.
https://realpython.com/python-keras-text-classification/

You can always go online and search for solutions as these questions are really common across the world for teaching NLP, so there's no point stopping you from doing that. What I want you to do is learn why something is done and how you can make it better-- and for that, I can give you the tutorials I like the most. Do not plagiarize-- use these codes to learn.

Question 1: Building a classifier [10]

You have used Pandas for a small part of your assignment 1. Now, we will see what we can do with it. We will use the same dataset (IMDB movie review dataset) that you used in Assignment 1.

a. Load the dataset using Pandas like you did previously. [1]
b. Use a bag-of-words model (known as CountVectorizer in scikit-learn) to convert the reviews into vectors. [1]
c. Split the data into two parts, training and testing. You will train a machine learning model on the training part of the data, and then you will test its

performance on test part of the data. Use an 80-20 split for your data, that is, 80% of the data will be for training and 20% of the data will be for testing. [1]

d. Now, train two machine learning models on the training part of your data. You will use a logistic regression model and a naive bayes model. [5]
e. Compare the accuracies of the models. Use visualization techniques that you like (charts, bar graphs etc.). Which one is better? Explain your answer. [2]

Question 2: Building a classifier (contd.) [5]

Perform the same tasks mentioned in question 1, but now, instead of a bag-of-word vectorizer, you will use a TF-IDF vectorizer. If you have done the bonus part of the previous assignment, you already know how to do it. Run logistic regression and naive bayes models on this as well, and show which one is better between a bag-of-words and a TF-IDF model. Explain your answer.

Question 3: Training a deep learning model [5]

From the previous two questions, you learnt how to build a classical machine learning model. Now, we will build a shallow (single hidden layer) recurrent neural network model to do the same thing. Use the tutorial given in the preamble. Your model will have three layers: one Embedding Layer with output shape 100 which will convert your words to a 100-length vector, one dense layer with an output shape 10, and the final output layer (another dense layer) with output shape 1. Do not use any pooling or any other layer.

Sequential neural nets use word vectors, not sentence vectors that were used in previous two questions. So, you will need to learn how to use **word embeddings.** In the “Using Pretrained Word Embeddings” section of the tutorial, you will learn how to load pretrained GloVe embeddings into an embedding layer. This embedding layer is your input layer, which will feed into your hidden dense layer, and the output of your dense layer will give you a probability whether the review is positive or negative.

Build your model the way you see in the tutorial. Run it for 50 epochs (it may run for a long time depending on your computer). You can play with the hyperparameters like changing the batch size etc. See how well this model performs compared to the other models.