Final Project: Sentiment Analysis **CSCE 633**
Due: 11:59pm on Aug 8, 2024

Instructions for assignment submission
a) Please write a brief report including your experiment details, results and discussion; **submit the pdf file on Canvas**. Also **submit a jupyter notebook** including your code.
b) Make sure TAs can run your code on Google Colab.
c) **You can use any available libraries for this project.**
d) Please start early :)

Project: Transformer for Sentiment Analysis

We will use the Yelp reviews dataset, which comprises hundreds of thousands reviews with stars. For this project, we only use a subset of it for experiments.

In this dataset, there are two fields of interest: 'text' which contains a qualitative review from a user, and 'label' which contains 'stars' (a quantitative ranking ranging from 1 to 5). The connection between qualitative and quantitative feedback is defined here as: a negative review equals one to three stars, and a positive review equals four or five stars. Note that labels might range from 0 to 4 (instead of 1 to 5) - just need a little fix.

Our goal is to implement the powerful Transformer model for sentiment analysis task based on the text reviews and stars.

(a) Data Pre-processing

(1) Load data from Hugging Face Hub. Randomly select 10,000 data points for training and 1,000 data points for validation from the original 'train' dataset, and 2,000 test data points from the original 'test' dataset.

(2) For convenience, now change the created train/val/test data from Hugging Face's dataset to pandas dataframe. Pre-process the data by removing the punctuation marks and stop words, and converting all words to lowercase. Note that the regular expression operations (doc) and nltk library might be useful.

(3) Convert the label into two types by the number of stars: $Positive \geq 4, Negative \leq 3$.

(b) Input Data Preparation The input of the Transformer model should be a fixed-length review sequence where integer numbers represent words. Here you need to build vocabulary for the dataset and pad / truncate the review sequences to the same length.

(c) Transformer Implementation Implement a Transformer model which is composed of an encoder network (i.e., multi-head self-attention layers) and a prediction head mapping the hidden representation of input sequence into the label space (i.e., three classes). Find more details about Transformer in paper. You may need to implement positional embeddings, a vocabulary embedding table, and mask indicators for padded tokens. PyTorch is recommended for model implementation.

(d) Model Training and Finetuning

(1) Train the model with SGD (or Adam/AdamW) optimizer using mini-batch fashion based on the training dataset.

(2) Plot two curves across training process, where the x-axis is the training epochs, and the y-axis is the training loss or validation accuracy.

(3) Finetune the transformer model, and save the best model with the highest validation accuracy.

(e) Test Result Analysis

(1) Load the best model saved above and report the accuracy of the model on the test dataset.

(2) What are the impacts of hyper-parameters, such as the hidden dimension and the number of attention layers, on the Transformer?