Assignment 3
Language Modeling with Transformer Basics

STAT8021: Big Data Analytics (Spring 2025)
STAT8307: Natural Language Processing and Text Analysis (Spring 2025)

DUE: **April 18, 2025, Friday, 11:59 PM**

Goal

The main goal of this assignment is for you to get familiar with Transformers. You will implement a Transformer model for text-level classification.

Note that you should use Python 3.8+ and PyTorch to finish this assignment. If you need GPUs, we suggest you to use Google Colab (https://colab.research.google.com/).

Submission

Please submit the following **two files** to Moodle for grading:

- A PDF report of your answers to all the questions.
- Your Python file: `hf_practice.ipynb`.

Total score of this assignment is **100 points**. Please write your procedures in the PDF report if you do not completely finish the code.

Transformer Basics

You will finetune a pre-trained Transformer model from hugging face[1]. The dataset in this part is part of AG News[2]. You will do a text classification task to predict the news type (0 for World news, 1 for Sports news, 2 for Business news, 3 for Sci/Tech news). Please write your code of this part in `hf_practice.ipynb`.

Q1 Finetune the `DistilBertForSequenceClassification`[3] model provided by the hugging face community to predict the type of a given news. **[TOTAL: 100 points]**

(a) The inputs to the transformer model should be tensors. You can use `DistilBertTokenizerFast`[4] to preprocess the `small_ag_news_dataset` we provided in `hf_practice.ipynb`, and then set the dataset as `torch` format. Print the processed first 3 samples in the train set. **[30 points]**

(b) To finetune the model, first you should load the train and test data into `Dataloader`, then define a transformer model with pre-trained weights form `DistilBertForSequenceClassification` and set the number of prediction classes as 4, finally finetune the model and evaluate the performance on the test set. **You are free to tune hyperparameters, including learning rate, batch size, epochs, etc.**. Print the training and testing accuracy for each epoch. **[30 points]**

(c) Test your finetuned model on a small external dataset `chatgpt_generated_new` we provided. Print the predictions and see if the predictions match your human judgement. **[20 points]**

[1] https://huggingface.co/
[2] https://huggingface.co/datasets/ag_news
[3] https://huggingface.co/docs/transformers/v4.27.2/en/model_doc/distilbert
[4] https://huggingface.co/docs/transformers/v4.27.2/en/model_doc/distilbert

(d) You can choose one other pre-trained transformer model from hugging face community to finetinue. Print the training and testing accuracy for each epoch and compare the performance and efficiency of the models you have finetuned in your report. **[20 points]**

Hint: If you cannot find an appropriate model from hugging face community, you can try `RobertaTokenizer` and `RobertaForSequenceClassification`[5].

To receive full credit of Q1(b) and Q1(d), you need to get at least 85% **accuracy on the validation set**[6]**.** Even if you are not able to get this part fully working, write up and document as much as you can so we can give appropriate partial credit.

[5] https://huggingface.co/docs/transformers/model_doc/roberta

[6] Our reference implementation can achieve 99.0% accuracy with the default hyperparameters with DistilBertForSequenceClassification.