ELEC6910Y - Topics in Deep NLP
Assignment 2

The Hong Kong University of Science and Technology, Spring 2023

Deadline: Friday, March 31st 2023, 11:59pm HKT

Submission details

You need to submit a zipfile named "ELEC6910Y_Assignment2_STUDENT_ID.zip" directly to Canvas containing:

- Commented code to run the assignment (either python files of Jupyter Notebook)
- Written report (pdf)

The provided code and report must be commented, clean and runnable without any modifications (avoid any absolute paths from your local work directory). Any code that cannot be ran will be dismissed and not counted towards the grade.

For this assignment, the written report containing all the figures, plots and result tables should come in a separate pdf file.

Introduction

In this assignment, you will use two different transformer models to perform text classification. First using an encoder-only model, BERT [1], for a binary classification task to predict whether a given article headline is clickbait or not. Then, you will use a decoder-only model, GPT-2 [2], for a multi-label classification task to predict the category of a given news article description. While decoder-only model are often used for text generation tasks, and encoder-only model for a variety of natural language understanding tasks, you will see that both architecture can be leveraged to achieve strong performance on text classification.

If you do not have access to any GPUs, you can use Google Colab https://colab.research.google.com/. The free version gives access to one K80 GPU which is enough to run this assignment. You are free to use any Python library of your choice (pytorch, tensorflow, keras etc...). However, it is strongly advised to use the *transformers* library from Huggingface (compatible with most ML libraries), at least to load the necessary models.

If you are unfamiliar with Huggingface, you can look up a tutorial on how to load and finetune a pre-trained transformer model using their libraries: https://huggingface.co/docs/transformers/training

1 Clickbait Title Classification (50%)

In the first part of this assignment, you are required to do binary text classification on a dataset composed of article headlines taken from various news websites. You will train a transformer model to classify whether a given article headline is "clickbait" or not. You can find the data for this assignment already split into train/test/valid split in the folder "./data/clickbait".

Definition "Clickbait" (Oxford Dictionary): Content whose main purpose is to attract attention and encourage visitors to click on a link to a particular web page. Ex: "What every student needs to know!", "You will not believe how incredibly efficient those 5 tips are..."

1.1 Preliminary Data Analysis (10%)

1. In terms of linguistics, what makes a headline/title 'clickbait' to a human reader? Find some examples that illustrates your claim in the dataset and report them. (You can browse through the dataset yourself to get an idea. Are there any lexical patterns for example?) (5%)
2. Compute and plot the distribution of the length of clickbait titles vs the length of non-clickbait titles (using all three data splits). What do you observe? Show the plot in the report. (5%)

1.2 Using BERT for Text Classification (40%)

The intrinsic features that you have observed in question 1.1 are part of what is captured by language models during finetuning. In this part, you are required to train and evaluate BERT, an encoder-only model presented in the course.

Use the following model configuration: **bert-base-uncased (110M parameters)**

Note: You are NOT asked to implement the model from scratch. Instead, you should load the pre-trained model from a library of your choice (transformers from Huggingface is advised), and finetune it directly.

1. Provide the dataloaders, training and evaluation scripts (10%)
2. Report 4 evaluation metrics on the test set: Accuracy, Precision, Recall and Macro-F1. You are required to achieve a Macro-F1 higher than 95%. (10%)
3. Report at least 3 different hyperparameter settings along with the evaluation results on the test split for each of the settings. (5%)
4. Plot the training loss and validation loss curves. (5%)
5. Following evaluation, report 3 false negatives samples (positive that got wrongly predicted as negative) and 3 false positives samples (negative that got wrongly predicted as positive) from the test set. For each of these samples, report the predicted probability for both classes. Comment on possible reasons why each of these samples were misclassified by the model. (10%)

2 News Category Classification (50%)

In the second part, you will do multi-label text classification on a subset of the AG News Dataset [3]. This is a 4-label classification task with the following classes: (0) World, (1) Sports, (2) Business, (3) Sci/Tech. The dataset is composed of titles and description fields of various news articles. The original paper achieves an error rate of 9.51 using CNNs back in 2016.

The dataset is provided in the folder "./data/ag_news". In this part, you are required to train and evaluate GPT-2, a decoder-only model seen in the course. As you know, models such as GPT-2 are commonly used for text generation tasks, but you will see here that they can also be leveraged for classification.

Use the following model configuration: **gpt2 (117M parameters)**

1. Provide the dataloaders, training and evaluation scripts (10%)
2. Report two evaluation metrics on the test set: Error rate and Macro-F1. You are required to reach an error rate lower than 10%. (10%)
3. Report at least 3 different hyperparameter settings along with the evaluation results on the test split for each of the settings. (5%)

4. Plot the training loss and validation loss curves (5%)
5. Report 3 misclassified samples from the test split. For each of these samples, provide the predicted probability for each of the 4 class. Comment on the possible reasons why those samples are misclassified. (10%)
6. Implement a function called *interactive*. Given *(model, tokenizer)*, this script should interact with the user as follows: (10%)
   - Prompt the user to type some text.
   - Process the text with the finetuned model and tokenizer.
   - Reply with the predicted probability for each of the 4 classes, and the final classification decision.
   - Loop back to first point.

   You are free to customize this function as you see fit as long as the four points above are respected!

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[3] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.