Spring 2025

EDS 6397 – Natural Language Processing

Assignment #5 – Sentence Pair Classification using Transformer-Based Models

Use of artificial intelligence assistant such as ChatGPT in developing the code for this assignment is allowed. You might still choose not to use AI assistant. However, ***the report needs to be completely written by you****. Use of grammar correction tools is disallowed.*

Assignment Overview

The objective of this assignment is to familiarize you with sentence pair classification applications. More specifically, you will finetune a pretrained RoBERTa model for entailment recognition, and another pretrained RoBERTa model for paraphrase detection.

Note: You must use the roberta-base model from huggingface for this assignment. Make sure you use RobertaTokenizer and RobertaForSequenceClassification.

Assignment Description

Sentence pair classification is a form of supervised learning in which the input is a pair of sentences and the output is a label indicating whether a specific relationship exists between them. In this assignment, you will:

- Finetune two models for paraphrase detection.
- Finetune two models for entailment recognition

Input Data

The input data for both tasks should be accessed via Huggingface (specifically through datasets). The dataset you will use for paraphrase detection as well as the one you will use for entailment detection are both part of original GLUE (General Language Understanding Evaluation) benchmark. https://gluebenchmark.com/

These datasets are pre-processed for LLM input; therefore, you do not need to perform any pre-processing.

Task 1: Paraphrase Detection using RoBERTa

Dataset: MRPC (Microsoft Research Paraphrase Corpus)

Access: Load using Hugging Face datasets.

Splits: train, validation, and test

Labels: Indicate whether the second sentence is a paraphrase of the first.

What to Do:

1. Freeze all weights except for the classification head.
   - Fine-tune the model for up to **20** epochs (minimum **10**).
   - Feel free to implement early stopping. Experiment with learning rate, batch size, and other hyperparameters to improve precision, recall, accuracy, and loss.
2. Unfreeze all weights (default behavior in Hugging Face pretrained models).
   - This time, you're fine-tuning all parameters in RoBERTa and the classification head.
   - Limit training to at most **5** epochs (but no fewer than **3**), due to the increased training cost.
   - Consider using LoRA only if training for each epoch takes longer than one hour. Make sure to explain well why you had to use LoRA, and how you decided on number of parameters.

Task 2: Entailment Detection using RoBERTa:

Dataset: RTE (Recognizing Textual Entailment)

Access: Load using Hugging Face datasets.

Splits: Only **train** and **validation** are usable (the **test** set has no labels). Split train dataset 80-20 and use the 20 for per-epoch metrics during training. Use the Validation dataset for testing the finetuned model. Note that the Validation dataset has 277 pairs of sentences. This should be reflected in your analysis for this task in the report.

Labels: Indicate whether the second sentence is a paraphrase of the first.

What to Do:

1. Freeze all weights except for the classification head.
   - Fine-tune the model for up to **20** epochs (minimum **10**).
   - Feel free to implement early stopping. Experiment with learning rate, batch size, and other hyperparameters to improve precision, recall, accuracy, and loss.
2. Unfreeze all weights (default behavior in Hugging Face pretrained models).
   - This time, you're fine-tuning all parameters in RoBERTa and the classification head.
   - Limit training to at most **5** epochs (but no fewer than **3**), due to the increased training cost.
   - Consider using LoRA only if training for each epoch takes longer than one hour. Make sure to explain well why you had to use LoRA, and how you decided on number of parameters.

The report

Your report should be up to four pages and must include the following sections (failure to do so will result in a deduction of points):

- For each task, you have two finetuned models. Mention the number of trained parameters for each.
- Performance metrics on unseen data should be clearly mentioned and discussed (accuracy, precision, recall, F1 score).
- You need to also mention the per class precision, recall, and F-1 scores.
- Justify your choice of hyper parameter used for all 4 models.
- Describe the finetuning process and discuss any challenges encountered for each task. Did you have to finetune using LoRA technique? If yes, how did the number of finetuned parameters change compared to continued pretraining?
- Include your assessment of the best model for each task. How can the results be even better?
- Include any additional insights you gained from this assignment.

What to submit:

1- Your Python Jupyter Notebooks should include plenty of comments and explanatory cells to ensure that your code is easy to read and understand. The Notebooks should also contain the results of each cell executed by you. They should be named as follows: **[Last_name]_HW5_Task1.ipynb** and **[Last_name]_HW5_Task2.ipynb**. You can instead submit a Google Colab file. Use a similar naming convention.

2- A report (3-4 pages) **in PDF format** that briefly **discusses** what you did for this assignment as well as the results.

Late submissions will be penalized at a rate of 1 point per hour.

We round up the time to the nearest hour.

Make sure you click on submit so your assignment is "Turned In". Uploading the files doesn't mean you turned the assignment in.

Due Date: April 30, 2025 by 11:59 PM (submit through Teams)