Foundations of Natural Language Processing

Peking University, 2025

Assignment 3: Due on May 31, 2025 by 23:59

(TA: thomastao@pku.edu.cn,ziruiwu@pku.edu.cn)

1. Directions

PLEASE read these instructions to ensure you receive full credit on your assignment.

- Submit your work as a zip file through **Course**, which should include one report in PDF (for 2.1 and 2.2), your source code in Python (for 2.1(2) and 2.2), and an expanded BERT tokenizer (for 2.2).
- The report should be generated from the LaTeX template in the attachment but you do not need to submit the LaTeX code.
- Your grade will be based on the contents of the PDF report, the source code, and the new BERT tokenizer. Additional files will be ignored.
- Please carefully read following pages, which may be helpful:
  [1] https://huggingface.co/learn/llm-course/chapter6/6
  [2] https://huggingface.co/docs/transformers/main_classes/tokenizer#transformers.PreTrainedTokenizer.add_tokens
  [3] https://huggingface.co/docs/transformers/main_classes/model#transformers.PreTrainedModel.resize_token_embeddings

LATE SUBMISSION POLICY

- Late submission will have 5% deducted from the final grade for each day late.
- No submission will be accepted after **June 7**, one week after the due date. Your submission time will be based on the time of your last submission to Course.

Therefore, do NOT re-submit after midnight on the due date unless you are confident that the new submission is significantly better to overcompensate for the credit loss of your previous submission. Submission time is **non-negotiable** and will be based on the time you submit your last file to Course. The credits (especially when deducted) will be rounded to the nearest integer.

2. Problem Description

In this assignment, you are required to expand the vocabulary of BERT's tokenizer for a text classification task in the biomedical domain.

2.1 Implement the WordPiece algorithm (55 Credits)

Tasks: Write a report to answer the Question (1) (3) (4), and fulfill the code in Question (2).

(1) Briefly explain the principle of the WordPiece algorithm. You may use formulas, and examples to illustrate. **(25 Credits)**

(2) Implement WordPiece by completing the code in the attachment *(file: wpalg.py)*. **(10 Credits)**

- In the file "wpalg.py", there is only one function "wordpiece", which receives two parameters, "training_corpus" and "vocab_size".
- You can only add code between the two lines containing "#======".

(3) Try to train a tokenizer by WordPiece and tokenize a given sentence. **(10 Credits)**

- Use the "wordpiece" function implemented in Question (2) to obtain your vocabulary, where "training_corpus" is following list and "vocab_size" is 120.

  ["peking university is located in haidian district",
  "computer science is the flagship major of peking university",
  "the school of electronic engineering and computer science enrolls approximately five hundred new students each year"]

- Tokenize this sentence: *nous etudions a l universite de pekin*
- Note that if you have correctly implemented the function "wordpiece", you may get the tokenization result via running "python wpalg.py" in Command.
- You are required to **give the tokenization result in your report**.

(4) Provide an example text that will result in a token sequence with [UNK] when using the tokenizer obtained in Question (3). **(5 Credits)**
Briefly introduce why the tokenizer of Llama does not need [UNK]. **(5 Credits)**

2.2 Expand BERT's tokenizer with WordPiece (45 Credits)

Background: In the previous assignment, you developed a BERT-based classifier for a biomedical task. Biomedical texts often contain uncommon words (such as disease names). It is inefficient to encode these texts by using a tokenizer trained on general-domain corpus. One strategy to address this is to expand the vocabulary of the pre-trained BERT model with relevant biomedical tokens. A richer vocabulary can reduce the need to break down complex biomedical terms into sub-word units, potentially improving training efficiency.

In this assignment, you will perform the following steps:

Step 1: Train a WordPiece tokenizer on the provided biomedical corpus (*file: pubmed_sampled_corpus.jsonline*). You can use the code from **Section 2.1** or utilize relevant readily available toolkits (e.g., Hugging Face's tokenizers library).

Step 2: Select 5000 domain-specific tokens from the tokenizer in Step 1 and add them into the original BERT tokenizer (i.e., bert-base-uncased).

Step 3: Expanding the vocabulary necessitates adjusting the input embedding layer of the BERT model, to accommodate the expanded vocabulary size.

Step 4: Train a new classifier on the HoC dataset using your modified BERT model with the expanded vocabulary.

You should write a report to answer the following questions:

1. In Step 1, how did you train your tokenizer? What parameters did you use? What is the size of your resulting vocabulary? **(10 Credits)**

2. In Step 2, how did you select these new tokens? Explain your selection strategy. Please sample 50 of the new tokens you added and list them in your report. Briefly discuss any interesting characteristics or patterns you observed within this sample. **(5 Credits)**

3. After Step 2, sample three sentences from the HoC dataset in last assignment. Show their tokenized results using original BERT tokenizer and the expanded one. What are the differences between the results from two tokenizers? **(5 Credits)**

Then compare the average length (in tokens) of HoC's training set, using original

BERT tokenizer and using the expanded one. **(5 Credits)**

4. In Step 3, how many new parameters are introduced? How did you initialize the new parameters of the newly introduced parameters? **(5 Credits)**

5. In Step 4, what is the performance of your BERT model with the expanded vocabulary on the HoC dataset? Is the performance higher or lower than that of the original BERT model from your previous assignment? If the performance is lower, what might be the potential reasons? **(15 Credits)**