HW3: Sentiment Analysis

[TA] Cheng-Tsung Lee (李承璁)

In this homework, we will explore the fascinating field of **sentiment analysis** using deep learning techniques. Specifically, we will focus on **multi-class classification**, where the goal is to predict each sentence in posts from social media as belonging to the label.

Dataset

The **Social Media Dataset** consists of tens of thousands of sentences in posts from social media, annotated with sentiment labels (i.e. positive, neutral, and negative). We will work with this dataset to train and evaluate our classification models.

Tasks

1. **Data Preparation:**
   - ✧ Load and preprocess the Social Media Dataset.
   - ✧ Split the data into training, validation, and testing sets.
   - ✧ Preprocess the datasets to obtain better results.
2. **Model Architecture:**
   - ✧ Design customized network for this homework.
   - ✧ The following architecture might help:
     1. Convolution Neural Network (CNN)
     2. Recurrent Neural Network (RNN)
     3. Transformer
3. **Training and Evaluation:**
   - ✧ Train the model using appropriate loss function.
   - ✧ Monitor training process and validate on the validation set.
   - ✧ Evaluate the model's performance using accuracy.
4. **Inference and Visualization:**
   - ✧ Apply the trained model to unseen sentences from the testing set.
   - ✧ Calculate the average accuracy throughout the testing set.
   - ✧ Visualize the outcome of both training and testing sets.

Learning Objectives

By completing this homework, we aim to:

- ✓ Gain hands-on experience with sentiment analysis tasks.
- ✓ Understand the challenges of natural language processing.
- ✓ Learn to interpret classification results and assess model performance.

Important Date:

1. Start Date: 2025/11/4 (Tue.) 16:00
2. Source code / checkpoint / experiment report submission deadline: 2025/11/24 (Mon.) 23:59
   No late submission will be accepted afterwards.

Submission Form:

1. Please zip all files in one and named "HW3_{student_id}.zip" when uploading to E3 platform.

```
HW3_{student_id}/
├── main.py
├── README.md
├── requirements.txt
└── {student_id}.pdf
```

2. We will also have submission on "Codabench". Please zip all checkpoint and model code files in one when uploading. Make sure the organization's name is your student ID, and the account is linked to the email provided on the class participant list. Each account can submit the answer 2 times every 24 hours.

```
best_model/
├── checkpoint/
│   ├── config.json
│   ├── model.safetensors
│   ├── special_tokens_map.json
│   ├── tokenizer_config.json
│   └── tokenizer.json
└── model.py
```

3. Codabench link: https://nycubasic.duckdns.org/competitions/6/

Requirements:

1. Design your own data processing method.
2. Construct any model architecture on your own.
3. The model weights should be **less than 500M (parameters)**.
4. Plot each epoch's **training phase** and **validating phase** to observe how they change.
5. **Pytorch, pandas, numpy, scikit-learn, matplotlib, regex, tqdm** libraries are allowed.
6. Do not use "PyTorch Lightning" to help.

Implementation Details:

Dataset Files

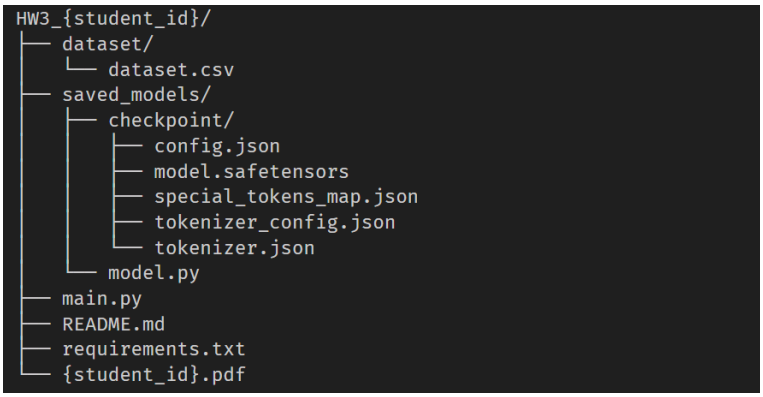
The dataset includes sentences annotated with sentiment labels.

dataset.csv: This file includes 60,001 csv rows, where there are a header and 60,000 rows of data represented as follows:

- id – the index of data
- text – sentence in string format
- label – sentiment label in string format

File Structure

Please organize your project files using the following directory structure that separates data, code, and other resources.



dataset/

This directory contains data related to the **Social Media Dataset**. The dataset files (train, val, test.csv) are likely stored within this subdirectory. We access and manipulate the dataset from here.

saved_models/

This directory is where you would store trained model checkpoints. After training, the best-performing model are saved here for later use. Please save the trained model weights and tokenizer. As for "model.py", it contains model config and architecture, which are useful when running inference.

data.ipynb

This file contains code for handling the dataset. It includes data loading, preprocessing, and other dataset-related functions.

main.py

This file typically contains:

1. Utility functions used across the project. These functions might include visualization tools, metrics computations, or other common tasks.
2. Code for training the models. It includes functions related to model training and optimization.
3. Model evaluation. It includes functions to assess model performance on validation.

README.md

This file explains how to reproduce your final result. It probably includes the setting of virtual environment and training hyperparameters.

requirements.txt

This file lists all the packages (or libraries) used in the Python files.

{student_id}.pdf

The report of this project in required format.

Accuracy

The accuracy counts the number of correctly classified instances out of the total instances and is defined as the ratio of correct predictions to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Report Format: (strict spec.)

The report should strictly follow the structure of the topics or there will be a penalty (-10 pts) on the report score; however, you can freely modify the subtopics. Note that the report must be fewer than 5 pages excluding the cover page (if there is one), in font size 12 and in A4 format or there will be a penalty (report score * 0.8).

1. **Introduction/Objective (10%)**
   - **Objective**

     Clearly state the purpose, background information of this project and outline the main objectives you aimed to achieve.
2. **Implementation Details (30%)**
   - **Data Preprocessing:**

     Describe how you process the data (e.g. data cleaning, data augmentation, train-validation split, etc.)
   - **Model Architecture:**

     Describe the architecture of the model(s) used, highlighting any unique or relevant layers, configurations, and connections.

     What makes your model unique? (Required)
   - **Training Pipeline:**

     The details of any training skills you used in this project. (e.g. hyperparameters, learning rate scheduler, learning rate selection, gradient clipping, batch/layer normalization, early stop, etc.)
3. **Experiment Result (30%)**
   - **Loss and Accuracy:**

     Plot the loss and accuracy history across epochs.
   - **Confusion Matrix:**

Plot and analyze the confusion matrix to show model performance on different classes.

4. **Discussion (30%)**
   - **Analysis of Results:**
     Discuss the experiment result as shown in Section 3.
   - **Additional Observations:**
     Share any other observations that could contribute to a deeper understanding of the experiment.
5. **Extra (10%) (optional)**
   - **Model Comparisons:**
     Compare the performance of different model architectures used, if applicable.
   - **Additional Experimentation:**
     Describe any other experiments conducted beyond the initial requirements, including results and discussion.

Grading Policy:

Homework score = Experimental results (50%) + Report (50%)

Experimental results (50%) (Note that result depends on private test set)

- [+10 pts]: accuracy @Top10%
- [90 pts]: accuracy $\geq$ 84%
- [80 pts]: accuracy $\geq$ 80%
- [70 pts]: accuracy $\geq$ 75%
- [0 pts]: accuracy $<$ 75%

Report (50%) (As mentioned in report format)

If the zip filename or the report has a format error, it will be a penalty (-10 pts).

Please only train & validate your model on the training & validating dataset split by the provided dataset. If the TA found your model trained on the private test set or any other additional datasets through retraining your model, you'll get 0 pts in this homework.

Please ensure the TA can execute the code and the model weight you uploaded. If the TA fails to do so, you'll get 0 pts in this homework.

If you're found faking your model predicting results, you'll get 0 pts in this homework. Also, we will take disciplinary action under school regulations.

Plagiarism is strictly prohibited. If you're found guilty of plagiarizing another's code or results, you and the person/people you plagiarized with will get 0 pts in this homework. Also, we will take disciplinary action under school regulations.

Always cite the resources you applied in your report. Please cite properly to avoid your homework results being considered plagiarism.

If any issues are identified in your submitted files for this lab, the TA will reach out to you via email. Please monitor your inbox closely. Suppose the TA has not received your response within three days of sending the email. In that case, it will be considered that you have forfeited the opportunity to provide further clarification and have accepted the corresponding penalty or consequences.

Contact & Information:

- Please post your question on the E3 forum.
- [TA] Cheng-Tsung Lee (李承璁): ctlee.ee14@nycu.edu.tw
- [TA hours] 11:00-13:00 Tue. ED-716 (Please make an appointment by email first)