CS 165B – Machine Learning, Spring 2021

Machine Problem #3
Due June 6 by 11:59pm PST

Note: This assignment should be completed individually. You are not supposed to copy code directly from anywhere else, and we will review the similarity of your submissions.

1. Task and Dataset:

In this assignment, you need to write a Python3 program to solve the dialogue sentiment classification problem, which aims to identify the sentiment (negative, neutral, positive) of each turn utterance in the dialogue. From machine learning perspective, it can be viewed as multiclass single-label text classification task where each sentiment represents a class (label) of a given text. This task is important since it can be used in some real problems such as analyzing user satisafisatin of dialogue system. The dialogues in this task is collected and labeled from the TV show of Friends.

2. Implementation:

You can find the code framework and data in the *mp3_starter_package* in the resources of Piazza.

2.1. About The Input and output

You are supposed to implement the *run_train_test* function.
The inputs of the this function:

- training_data: this is the training data in the form of a list of dictionary, where each dictionary is a sample of a single turn in the dialogue. Each dictionary has the following items:
    - "text": the utterance of the speaker.
    - "label": the sentiment label. 0 (negative), 1 (neutral) or 2 (positive).
    - "speaker": the name of the speaker.
    - "dialogue_id": the id of the dialogue.
    - "utterance_id": the order of this utterance within the dialogue.
- testing_data: this is same to the training_data with "label" item removed.

The output for the function:

- testing_labels: the labels for the testing data in the form of a list of integer.

2.2. How to evaluate your code locally

To evaluate your code in local machine, run the following command:
 # python3 evaluate.py
which will output the f1-score on the development dataset for each class. We will give grades based on the average f1-score.

2.3. How to convert the text into features

There are two general ways to convert the text into features. The first is to use TF-IDF related features. You can use *TfidfVectorizer* provided by sklearn to achieve this. For more information about how to use this, please refer the official document: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Another way requires more background in the NLP field, which convert the text into embeddings using Word2Vec or other language models. If you are interested in this direction, there is a guidance about word embedding: https://towardsdatascience.com/a-beginners-guide-to-word-embedding-with-gensim-word2vec-model-5970fa56cc92. You can also find other resources online. After you have the word embedding, you can try some popular recurrent models given a sequence of word embeddings such as LSTM.

2.4. What modules can be used

I have installed pandas, numpy, nltk, sklearn, pytorch, scipy, tensorflow, keras on the Gradescope. Please try use these tools as you can. If you need to use other tools, please let me know. Note that there is no GPU on the Gradescope, so please make sure your program can run successfully on a single-CPU machine without GPU resources.

3. Submission

3.1. How to submit

To submit your code: upload **a single file mp3.py** (multiple files are not accepted) in the Gradescope, which will show your average f1-score and your grade on the testing data. Note that the testing data on Gradescope is different from the local development dataset that you have.

4. Grading

3.1. Basic grade

I have run a simple baseline which get around 50 average f1-score on the testing data, and I will use this as the threshold. If your f1-score x is higher than 50, then you will get 85+ (x - 50). For example, if you get 55 f1-score, then your score will be 85+(55-50) = 90. Note that the maximum score will be 100. If your f1-score x is lower than 50, then you will get 85 - 0.5 * (50-x)*(50 - x), i.e., the lower your f1-score is than the threshold, the more points you will get by improving the performance.

3.2. Extra credit

Besides the above basic score, we are going to have a leaderboard and give extra credits to students with top performances for this assignment. There are pre-submission tournament and the post-submission tournament.

For the pre-submission tournament, any student who leads the leaderboard for one day will get 2% extra credit. This pre-submission tournament will last from May 27 to June 6. We will count the winner at the end of the day.

For the post-submission tournament, top-3 performers will get extra credit 50%, 25% and 25%. We will count the top-3 performers immediately after the deadline (June 6, 11.59PM PST). Any submissions after the deadline will no be counted for the post-submission tournament.

Your final score is basic_score*(1+extra_credit). You can check the leaderboard on Gradescope about the performance of others, and you don’t have to use to real name for leaderboard.