COMP5423 NATURAL LANGUAGE PROCESSING

Lab1 Homework: Sentiment Analysis on Movie Reviews

Due date: 11:59pm, 25/2/2025 (Tuesday)

Group Size: Individual

Homework Objective

"There's a thin line between likably old-fashioned and fuddy-duddy, and The Count of Monte Cristo ... never quite settles on either side."

The Rotten Tomatoes movie review dataset is a corpus of movie reviews used for sentiment analysis. This assignment presents a chance to benchmark your sentiment-analysis ideas on the Rotten Tomatoes dataset. You are asked to label phrases on a scale of five values: negative, somewhat negative, neutral, somewhat positive, positive. Obstacles like sentence negation, sarcasm, terseness, language ambiguity, and many others make this task very challenging.

Evaluation

Submissions are evaluated on classification accuracy (the percent of labels that are predicted correctly) for every parsed phrase. The sentiment labels are:

0 - negative; 1 - somewhat negative; 2 - neutral; 3 - somewhat positive; 4 - positive

Dataset Description

The dataset is comprised of tab-separated files with phrases from the Rotten Tomatoes dataset. The train/test split has been preserved for the purposes of benchmarking, but the sentences have been shuffled from their original order. Each Sentence has been parsed into many phrases by the Stanford parser. Each phrase has a PhraseId. Each sentence has a SentenceId. Phrases that are repeated (such as short/common words) are only included once in the data.

- train.tsv contains the phrases and their associated sentiment labels. We have additionally provided a SentenceId so that you can track which phrases belong to a single sentence.
- test.tsv contains just phrases. You must assign a sentiment label to each phrase.

Requirements

(1) Process the text input.
(2) Design your method to classify the sentiments.
(3) Compare and analyze models' performance (e.g., the classification accuracy on the validation set) in various settings.

(4) Compare and analyze different models' performance.
(5) Participate the Kaggle competition.

Submission and Grading Scheme

Pack all files in one zipped file with the clear name for each file and submit to Blackboard.

1. (40%) The Source Code of Your Program (including basic command line UI system) 20 scores
   (1) A readme file describes the structure of your program and how to run it. 10 scores
   (2) Add annotations in the front of each (Python) file to specify its usage. 5 scores
   (3) Clear comments of your code. 5 scores

2. (40%) Report
   (1) How you process the text input. 15 scores
   (2) How you classify the emotions. 15 scores
   (3) The results on the test set. 5 scores
   (4) The UI system. Describe its function and its workflow. Some screenshots are necessary. (online systems are not necessary) 5 scores

3. (20%) Predicted result on the test set and a screenshot of your ranking under Leaderboard along with your name in the competition.

Submit a "csv" file named test_prediction.csv, where each line is the predicted label of the corresponding text in the test_data.txt. We provide a submission "csv" file that you can refer to (sample_submission.csv).

Remarks*:

1. A good accuracy on the test set **don't guarantee** a high score of this project, your design and a clear description of your design are also valuable.
2. You are strongly recommended to implement the system in **Python**.

Free GPU resources:

- Baidu PaddlePaddle: courese registeration tutorial usage
- Google Colab: tutorial usage
- Kaggle Kernel: tutorial usage