

Natural Language Processing Assignment #2

text classification

March 30, 2019

Due date: 5/8 11:59 PM. These questions require thought, but do not require long answers. Please be as concise as possible. We encourage students to discuss in groups for assignments. You should final submit a package which consists: a report (Latex) and source code with the necessary annotation. You should put your implementation ideas and observation in the report. And the report needs your name and student number. The code should be put into a fold with three files: `train.py`, `model.py` and `test.py`. We don't supply the source code and you can implement models with Tensorflow or Pytorch. The data for this assignment can be found at <https://cloud.tsinghua.edu.cn/d/794030b1ecc145c2a30c/>. If you have any questions, you can discuss on <https://github.com/thunlp/NLP-THU>.

1 Task Definition

Opinion mining (sometimes known as sentiment analysis or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to the voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine. Industrial circles utilize opinion mining techniques to detect people's preference for further recommendation, such as movie reviews and restaurant reviews. In this assignment, we need to establish a sentiment classification model for the given sentence.

2 Dataset

We supply Stanford Sentiment Treebank datasets for experiments, which include *stanfordSentimentTreebank.zip* and *trainDevTestTrees_PTB.zip*. *stanfordSentimentTreebank.zip* is the raw data and has detail information in *README.txt*. Furthermore, you can utilize the processed version *trainDevTestTrees_PTB.zip* and the following introduction can help you.

For the first sentence in `test.txt`: (2 (3 (3 Effective) (2 but)) (1 (1 too-tepid) (2 biopic))), you can transfer it to the tree structure as shown in Figure 1.

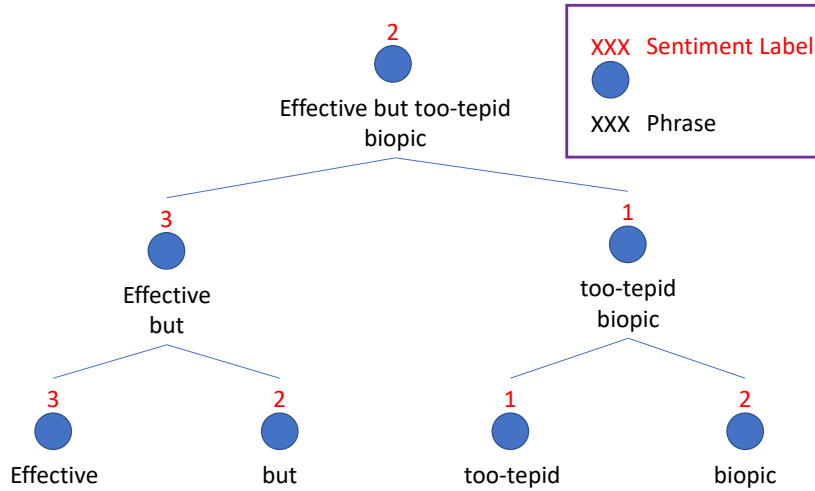


Figure 1: The sentiment tree of the sentence "Effective but too-tepid biopic."

And the sentiment label from 1 to 5 indicate affective states of very negative, negative, neutral, positive, very positive, respectively.

3 Assignment Requirement

In this section, we describe the requirement of sentiment classification system for this assignment.

Basic Version: You should implement a RNN or CNN model for sentence sentiment classification. For example, give a sentence "Effective but too-tepid biopic.", you should predict this sentence sentiment label (As shown in Section 2 the ground truth for this sentence is 2). Then you should use the test set to evaluate your model with accuracy, which means the number of correct predicted examples divides the total number of test set sentences. What's more, you should tune parameters according to following instructions and show the result with different parameter settings.

- Train with or without dropout. (The dropout rate should be set to 0.2)
- Train with different hidden size. (The hidden size should be set to 256 and 512)
- Train with a different number of the hidden layer. (The number of hidden layer should be set to 1 and 3)
- Train with pre-trained word embedding. (We supply GloVe pretrained word embedding with 300-dimension for your experiments and you can explore the model performance with the same dimension without pre-trained word embeddings.)

Compare with the different settings and show the accuracy on the test set in your report.

Improvement Version (Optional): For this dataset, every phrase and word in the sentence is annotated with its sentiment label and you can consider how to utilize phrase and word sentiment labels to improve your model performance based on the basic version.