BERT and Sentiment Analysis

Jinjian Tong,

1. Introduction

Sentiment analysis has been an active area in Natural Language Process. With the development of the Internet, services such as social media, e-commerce, and blog where people can share their opinions online pop up. With so many texts collected, it can important for the platform provider or business provider to find out the perceptions of people. Sentiment analysis is such a tool that given a sentence, the program can tell the emotions it contains. In this project, I will conduct research on Sentiment Analysis in Chinese with the state-of-art language model, BERT[1].

2. Related Information

2.1 Sentiment Analysis.

Sentiment analysis is an NLP task that aims to identify, extract, quantify, and study the emotions contained in a word, sentence, or document. There are three main methods for this task including knowledge-based technologies such as emotion dictionary, machine learning such as SVM for traditional machine learning and BERT for deep learning, and hybrid method which combines the previous two. There are several sub-tasks in this topic including the basic task which classifies the emotion polarities, the subject/ objective classification, and aspect-level SA which gives emotions to different parts of the sentence, etc. In this project, I use BERT to approach the basic task of Sentiment analysis.

2.2 BERT

The full name for BERT is Bidirectional Encoder Representations from Transformers. It is a bidirectional language model which is based on the encoder layers from Transformers, and the bidirectional means each word after tokenization will contain information from both left and right. The structure of BERT is simply a stack of encoder layers, but the number of total parameters can be large. Take the base version of BERT as an example, it has 12 layers of the encoder and 110M total parameters. This large number of parameters makes the training computational expensive and will take days of training using TPUs or GPUs.

There are two steps to building a BERT model for a specific NLP task. The first step is pre-training. The aim of pre-training is to train a bidirectional language model on unlabeled text. There are two pre-training tasks. The first one is Masked Language Model (MLM), and the second one is Next Sentence Prediction (NSP).

In MLM, some percentage of the input task is masked with [MASK] tokens, and the model is trained to predict what the masked contents are. In a bidirectional language model, each tokenized word should contain information from both sides. Therefore, masking part of the text, the bidirectional language model should have the ability to predict them. BERT

achieves this bidirectional property through the MLM task. To avoid the mismatch that [MASK] tokens only appear in pre-training and do not exist in the downstream task, the tokens may also be replaced with some random ones.

The purpose of NSP is to enhance the model's understanding of the relationship between two sentences. In this task, BERT is given with a sentence A and a sentence B and is trained to do a binary prediction of whether B is the next sentence of A. However, this task is considered to be problematic. NSP combines the task of topic prediction and coherence prediction. Topic prediction is a relatively easy task and coherence prediction is somewhat overlapped with MLM. Therefore, this task is not really helping, and in some experiments (e.g. RoBERTa), removing the NSP loss matches or slightly improves the performance on the downstream tasks.

After pre-training, the next step is to adapt the bidirectional language model to specific NLP tasks. This process is called fine-tuning. Concatenating the BERT model with output layers and further training with task-specific tasks, we can have an NLP model that has state-of-art performance. Compare to pre-training, this process is relatively computational cheap.

3. Dataset

In this project, we use the datasets provided by HKUST–Xiao-i Joint Laboratory. There are two datasets. The first one is XiEMO1.0 where the emotions are straightforward and relatively easier to classify, and the words used in them are outdated. Additionally, the texts are not well-cleaned. There are meaningless symbols and extra numbers. There are 32674 pieces of Chinese sentences, 25115 pieces with neutral emotion, 3203 positive, and 4356 negative. The longest sentence among them has 16 words. On average, there are 4.34679 words in a sentence, including the punctuations.

The second dataset is XiEMO2.0. Compared to XiEMO1.0, this dataset is more cleaned to use. There are 40000 neutral, 20000 positive, and 20000 negative. The longest one has 238 words, and on average, there are 11.017925 words. This dataset is averagely longer, more balanced, and more updated. However, the emotion polarity in this dataset is also harder to classify. There are no clear words indicating the emotions.

4. Models

In this project, there are mainly 4 kinds of models. The first kind is BERT with a linear output layer. This model directly uses the pre-defined class from the transformer package: BertForSequenceForSequenceClassification. In this class, BERT is concatenated with a linear layer that uses the 768 hidden dimensions BERT for classification. It generates a logit of three numbers indicating the score for each category. Argmax is used for the final classification. In the first kind of model, three variations of BERT are used here, including Bert-base-chinese, HFL/Chinese-bert-wwm, and HFL/Chinese-roberta-wwm-ext. All of them are pre-trained BERT or RoBERTa (which is similar to BERT) from different sources.

The second kind is BERT with GRU layers and linear output layer. This class mimics the source code of BertForSequenceForSequenceClassification. The only difference is that a bidirectional GRU layer is added here. The GRU layer is defined by the PyTorch class nn.GRU with hidden_dim 256, num_layer 2, and bidirectional True. The pre-trained BERT used here is Bert-base-chinese. The third kind is Bert with LSTM and linear output layer. The structure of this model is similar to the second kind, only using LSTM instead of GRU. Bert-base-chinese is also used here. A Voting classifier is also built. It combines the previous 5 models and votes for the prediction.

5. Result

5.1 XiEMO1.0

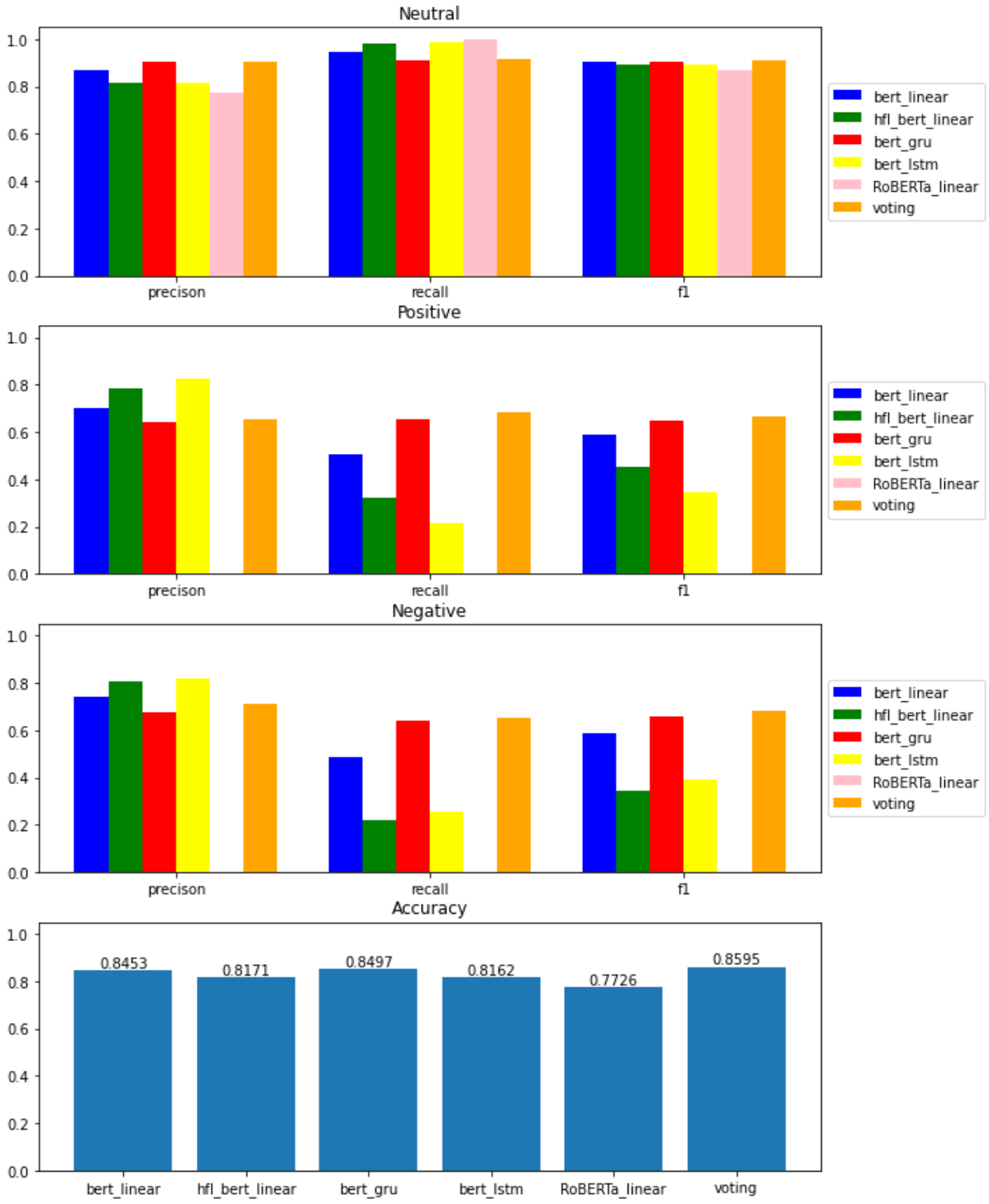

Fig. 1. Models' scores in XiEMO1.0

Fig. 1 shows the score of each model on this dataset. As we can notice, the voting classifier has the best performance among all models. It has the highest accuracy (85.95%) among all

models. Additionally, it has the highest precision and f1 score for neutral and the highest recall and f1 score for positive and negative. The best performance among the first five models is BERT_GRU. It has a similar performance as the voting classifier. It has the highest precision and f1 score for neutral and the highest recall and f1 score for positive and negative among the five models. The second best is BERT_linear. Hfl_BERT_linear and BERT_LSTM have nice performances on neutral cases and on the precision for positive and negative. However, they are scoring low on the recall and f1 for positive and negative. The worst among them is ROBERTa_linear. It has a 1.0 recall for neutral cases, and the reason behind that is that it is classifying all the texts as neutral.

Generally, BERT is not having a sound performance on this dataset. The reason might be that the dataset is not cleaned enough. The meaningless symbols and extra numbers cannot be simply deleted since there are also cases where they are meaningful and needed in some cases. This uncleaned dataset leads to an unideal performance.

5.2 XiEMO2.0

The models are not having an expected performance on the first dataset. Thus, before I conducted research on this dataset, I thought the performance would be low. However, when I applied the best two models in the first example, BERT_linear and BERT_GRU on it, they had relatively high accuracies. BERT_linear achieved 87.01% and BERT_GRU achieved 89.74%. Therefore, I applied all the models (except voting) on XiEMO2.0 and checked their performance. Fig. 2 shows the result

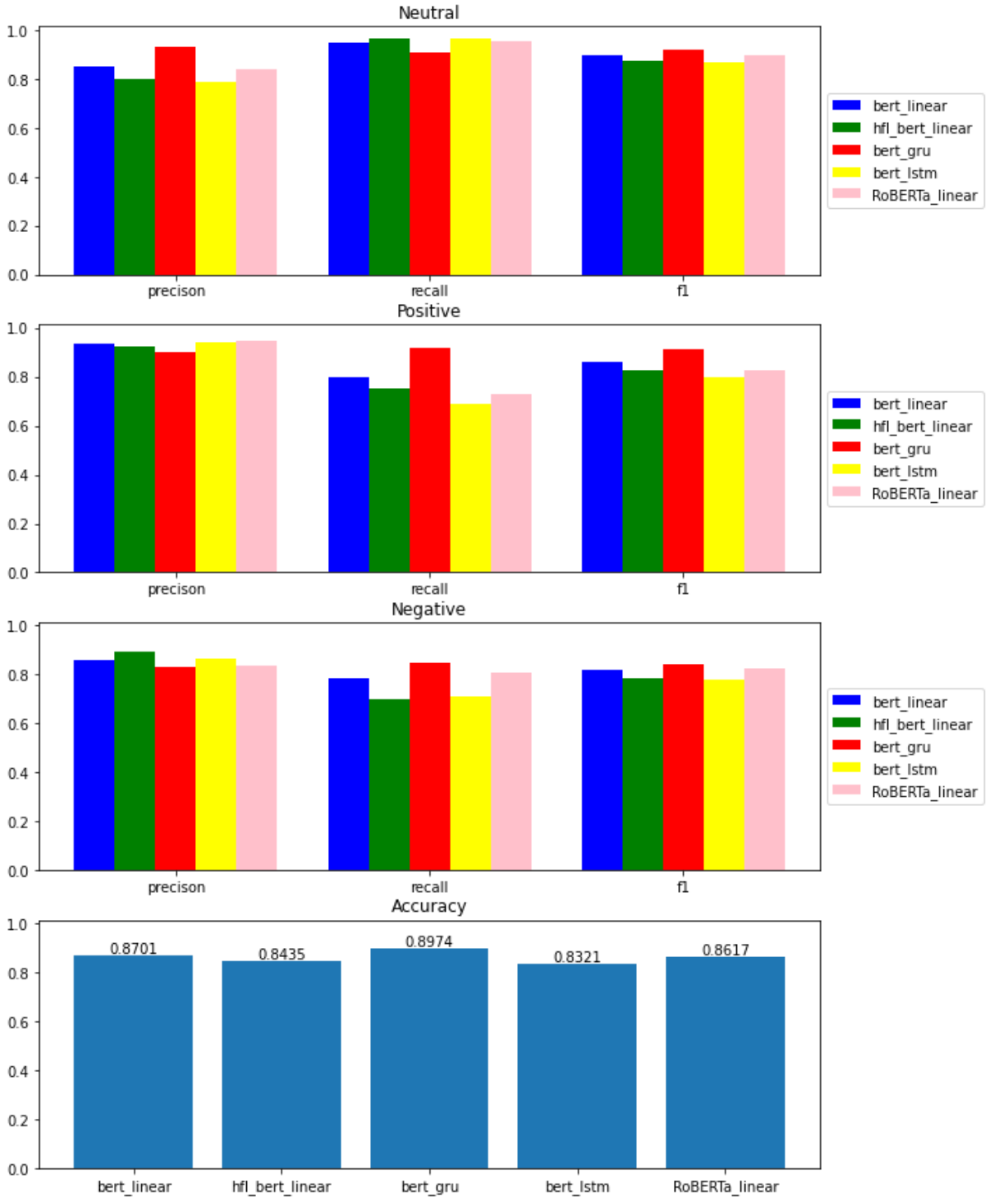

Fig. 2 Models' scores in XiEMO2.0

The models have obviously better performance on XiEMO2.0 than XiEMO1.0. The reason is obvious but I neglected it when choosing which dataset to work on first. As mentioned, XiEMO2.0 are more cleaned. Unlike XiEMO1.0, the data contained in XiEMO2.0 are more ready to use. Despite the fact that it is relatively harder to tell the emotions, it is no problem for BERT.

Just like the last experiment, BERT_GRU is the model with the best performance among the five models. It has the highest accuracy, highest precision for neutral, and the highest recall and f1 for recall and f1 for negative and positive. This is the same case when working on XiEMO1.0. Additionally, BERT_linear is again the second-best model.

Additionally, RoBERTa has a nice performance in this dataset. The accuracy and other scores are similar to BERT_linear.

5.3 BERT_GRU and BERT_linear

In the previous two experiments, BERT_GRU and BERT_linear are the two models which have the best performance. In this section, a deeper analysis of their test results will be conducted.

5.3.1 XiEMO1.0 Among the 6535 total test cases in XiEMO1.0, BERT_GRU failed 982 of them and BERT_linear failed 1011 of them. Fig. 3 shows the failed cases for BERT_GRU. Among the 982 failed cases for BERT_GRU, 453 cases have the identity of neutral, 210 are positive, and 319 are negative, and BERT_GRU's classification result actually matches this proportion pattern. While it is not for BERT_linear. Fig. 4 shows that BERT_linear has a higher possibility to classify the text into a neutral case, which is the main case for XiEMO1.0. In XIEMO1.0, 25115 out of 32674 pieces are neutral.

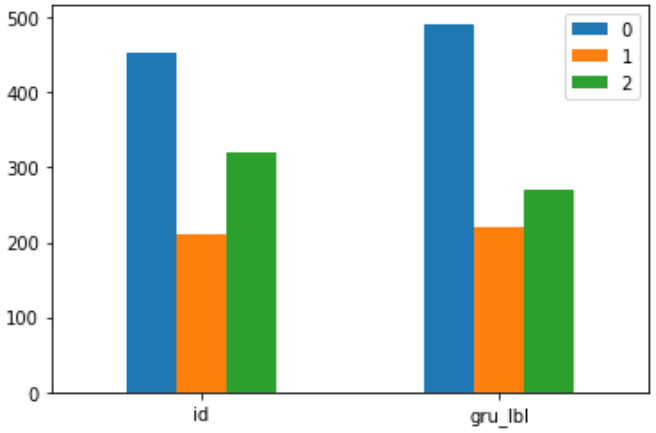

Fig. 3. Failed cases for BERT_GRU

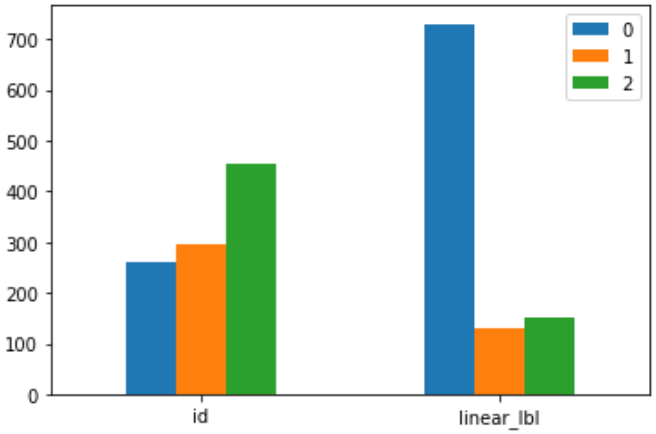

Fig. 4. Failed cases for BERT_linear

Fig. 5, 6, and 7 show the cases where BERT_GRU and BERT_linear have different opinions. BERT_linear always has the most zeros, which means neutral cases. In contrast, BERT_GRU is more general.

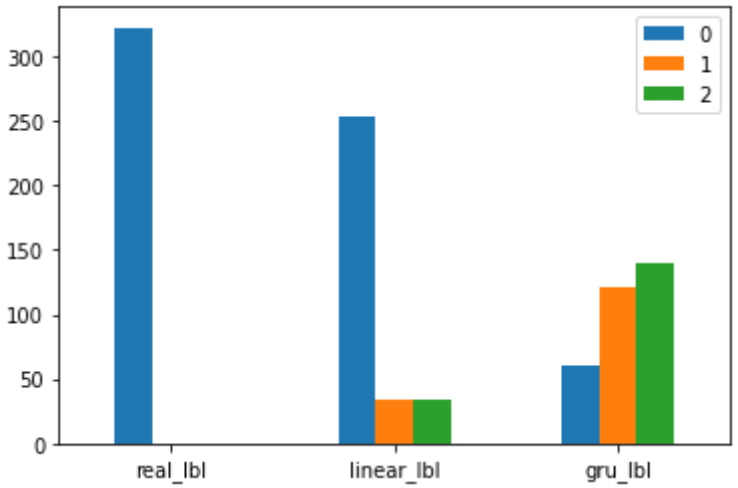

Fig. 5. Where BERT_GRU and BERT_linear are different and the real_lbl is 0

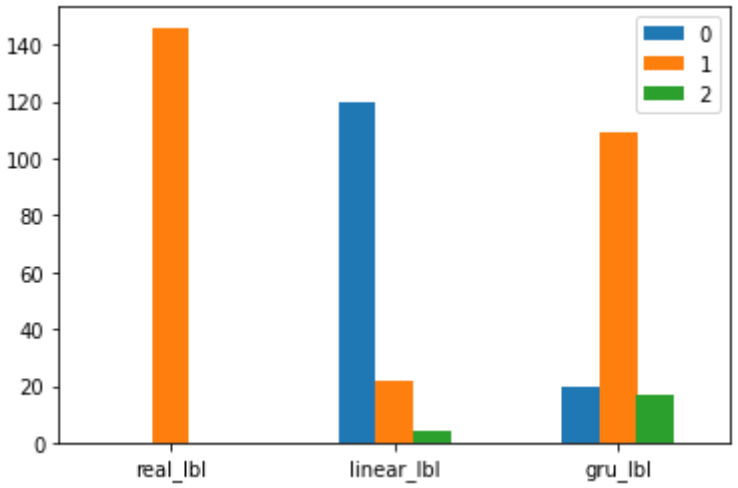

Fig. 6. Where BERT_GRU and BERT_linear are different and the real_lbl is 1

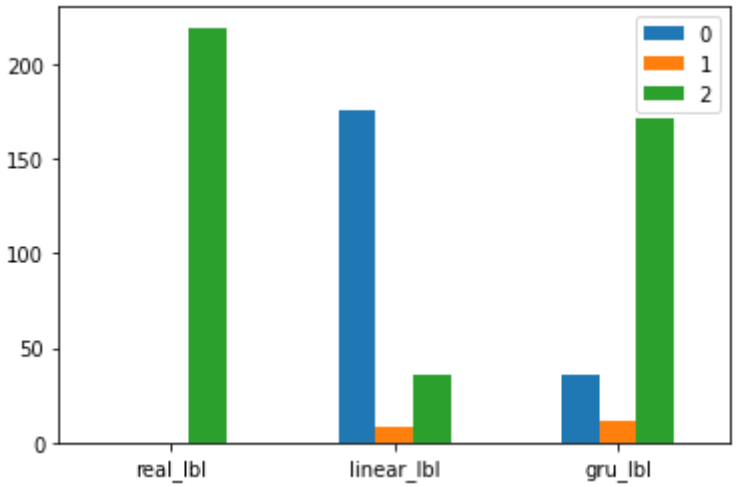

Fig. 7. Where BERT_GRU and BERT_linear are different and the real_lbl is 2

5.3.2 XiEMO 2.0 Similar process is conducted on the second dataset. Among the 16000 test cases, BERT_GRU failed 1642 of them and BERT_linear failed 2079. Surprisingly, Fig. 8 to Fig. 12 are showing the same pattern as the last experiment. It turns out that both datasets are having a similar effect on the model training. In both experiments, BERT_linear has a higher probability to classify the text as neutral, and BERT_GRU tends to be general, which makes BERT_GRU have a better performance.

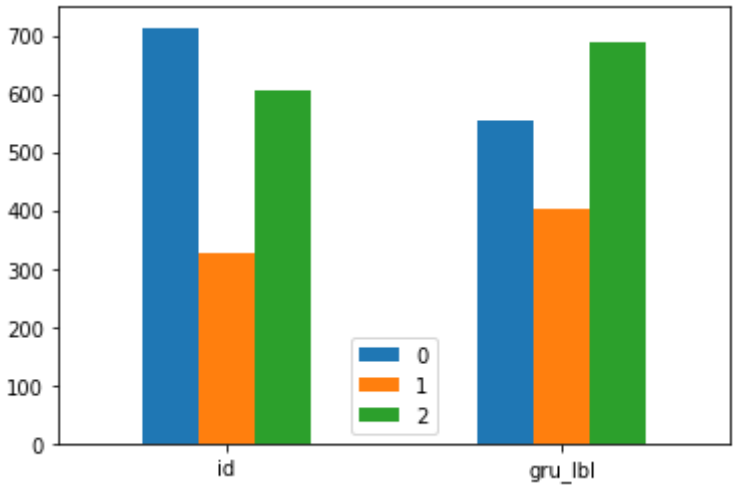

Fig. 8. Failed cases for BERT_GRU

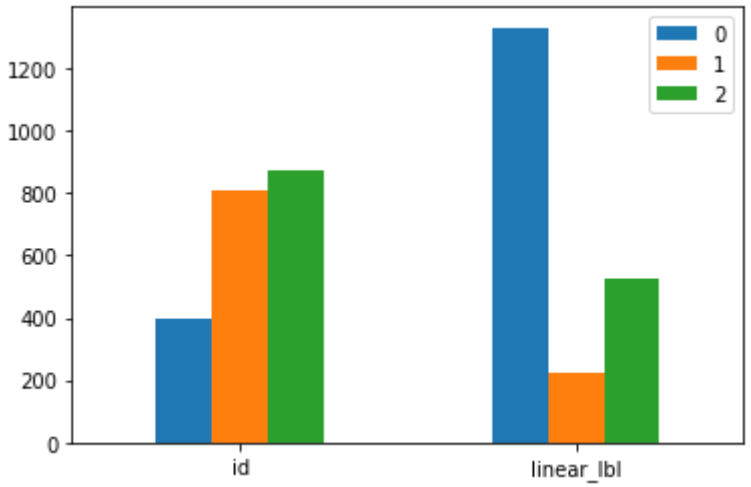

Fig. 9. Failed cases for BERT_linear

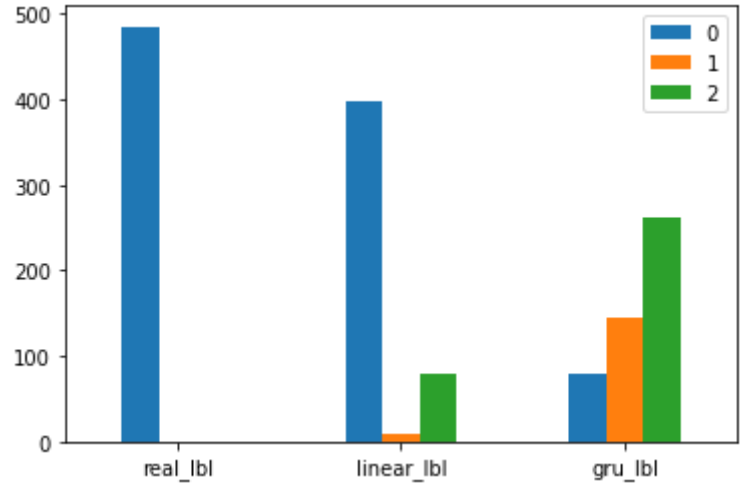

Fig. 10. Where BERT_GRU and BERT_linear are different and the real_lbl is 0

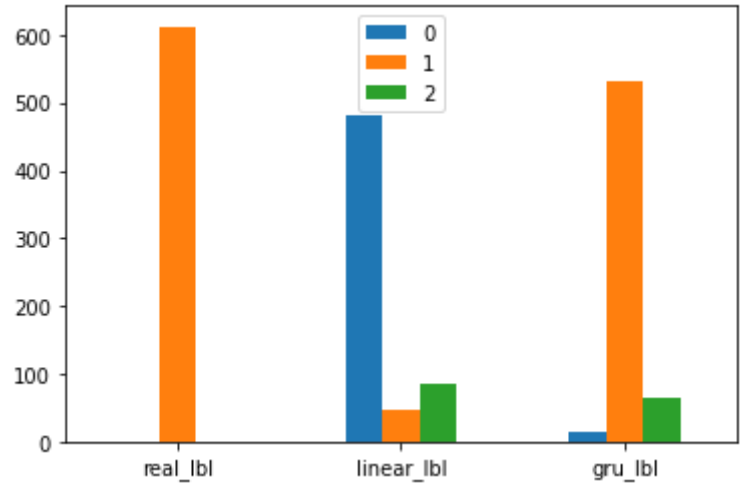

Fig. 11. Where BERT_GRU and BERT_linear are different and the real_lbl is 1

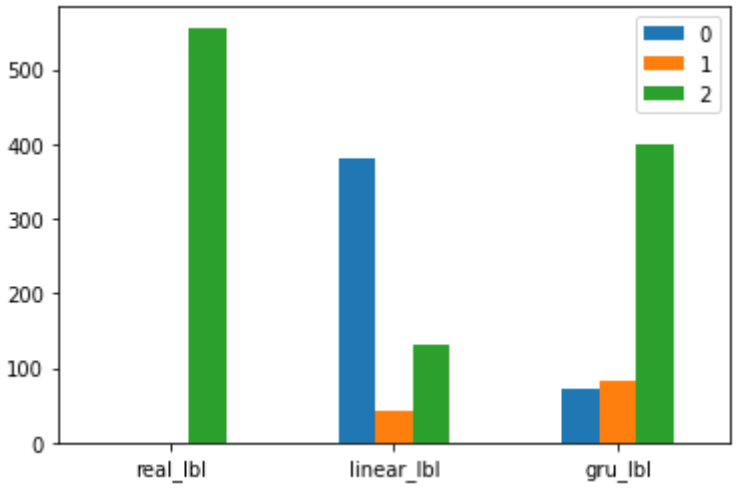

Fig. 12. Where BERT_GRU and BERT_linear are different and the real_lbl is 2

6. Future Work

There are many future works remaining for this project. First is the further tuning for models. Despite that much time has been spent on tuning the five models, it is possible to improve their performance more. One difficulty encountered in the model training is that some methods were used to improve the performance such as get_lienar_schedule_with_warm_up with a warm-up ratio of 0.1. However, this operation actually decreases the models' accuracy on the test dataset, and it is worth learning where I went wrong. In addition to that, hyperparameter tuning can be done on BERT_GRU and BERT_LSTM. The hidden dimension and number of layers of BERT_GRU and BERT_LSTM are randomly chosen and never changed during the experiment. Since BERT_GRU is having a sound performance on both datasets, and BERT_linear is doing a good job in XiEMO2.0, it is worth trying to see if modifying those parameters will increase the performance or not.

The second remaining thing is that, although it is not mentioned, the project has a second goal to achieve, that is finding the emotion words which lead to the emotion polarity. One naive thinking is that the emotional value of each word can be calculated, and use the largest among them or add them together to get the final result. However, this training will involve marking the emotional value for each word in the dataset, either manually or automatically. Also, this solution has a problem approaching the emotional phrases. Further work is needed for this solution.

7. Conclusion

In this project, sentiment analysis on two Chinese datasets is conducted. Five models are used on them, and the model with the best performance is BERT_GRU. Through this project, knowledge about BERT, neural network training, and data visualization is gained.

Appendix: minutes for four meetings

First meeting

time: Feb 10 at 3.30 pm

place: Zoom

attending: Jinjian Tong, Prof. Lin

content:

1. Go over the whole process of the individual project
2. Things to do: do some initial work on BERT, demo the result in the next meeting

Second meeting

time: March 4 at 5.15 pm

attending: Jinjian Tong, Prof. Lin

content:

1. Demonstrate the two models built, using the source code of BERT and PyTorch
2. Discuss possible ways to improve the performance
3. Things to do: improve the performance of models

Third meeting

time: April 8 at 5.00 pm

attending: Jinjian Tong, Prof. Lin

content:

1. Demonstrate the improved model, and do the data visualization
2. Discuss how to solve the emotional words problem
3. Things to do: Further tuning the models, increase the accuracy, and do a summary of the individual project in the next meeting

Fourth meeting

time: May 13 at 4.00 pm

attending: Jinjian Tong, Prof. Lin
content:

1. Presentation of the semester
2. Discuss what contents can be further explained in the report.
3. Things to do: report

Reference:

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding arXiv:1810.04805

[2] RoBERTa: A Robustly Optimized BERT Pretraining Approach arXiv:1907.11692