**1. Contribution**

1. My job in our group is to use Google Bert to classify the text and complete a pipeline.

2. Use various word processing methods for specific text to improve the final f1 score (0.986)

3. I arrange member’s work within the group and meeting time. Three rehearsals were completed through communication with members before the final pre.

4. Use my own server to build a python environment for team members

5. Check out the relevant information and code about Glove on the Internet and use it as a demo for other members.

**2. Idea**

1. Specific text optimization for specific texts, such as fake\_or\_real\_news.csv. Because Bert learns each sentence and the relationship between the sentences, using regular expressions to process non-sentences in the article (such as #hashtag) is necessary.

2. By checking out the relevant information, I learned that EDA (Easiest Data Augmentation) can improve the classification in the case of small text, so I tried two methods of EDA, one is Random Insertion, the other is Synonym Replacement)

3. Use Google Bert as a method of text classification. Google Bert is a cutting-edge approach to do NLP tasks with excellent and efficient classification and I use the Bert model with the softmax layer to complete text classification.

**3. Something I want to share**

The initial idea of our group was to use Google Bert to generate vectors for downstream text classification tasks. However, through code testing, the output includes word vectors and sentence vectors with high dimensions and huge time consumption. Therefore, extracting vectors separately from the Bert model is a bit hard. If the vector can be extracted, the output also includes the word vector and the sentence vector, which is difficult to handle. Thus, we used the method mentioned in the article- adding softmax layer to finish the text. Then, I begin my work to use a completed Bert pipeline to finish the job.

At the beginning, I spent a lot of time collecting various information about Bert. Because Bert is based on the existing NLP model, I started with the history of NLP and learned the principles of multiple models, from GloVe to LSTM and ELMo, and finally to Bert. I finally got a glimpse of how Bert works and wrote a blog to my URL: <https://nave.work> .

In fact, I used the original data to feed the model and got a 0.98 f1 score, so how to optimize on this basis becomes a problem. I observed a lot of content that was not a sentence by observing the text, including the URL and #hashtag, so I used regular expressions to remove specific content to improve the f1 score, and finally reached the score of 0.986.

In the final training process, due to the tight time and heavy tasks, I rented 1080ti to complete all the tasks. Of course, I am satisfied with the final result. This is based on the powerful per-training model of Google Bert. Anyway, I gained a lot of knowledge through this project and learned a lot of debugging skills, I love NLP!! (Finally attach my Contribution chart:

A screenshot of a cell phone

Description automatically generated

and Github repo: <https://github.com/NavePnow/Google-BERT-on-fake_or_real-news-dataset#5-part4-reference>