Prompt-based Text Matching Methods for Fake News Stance Detection

Zeyong, Jin
zeyongj@sfu.ca

Zhi, Feng
zhif@sfu.ca

Yuqing, Wu
ywa292@sfu.ca

Abstract

In recent years, the task of text stance detection has become an emerging direction in the field of Natural Language Processing (NLP). Text stance detection means that the machine learning system accepts two pieces of input text, and then output the stance relationship between the two pieces of text, such as agreement, opposition, or neutrality. With the rapid development of large-scale neural network parameter Pre-Training Technology (PTM), stance detection and other related tasks have more powerful computational models. This article is dedicated to using the BERT pre-training model to implement the classifier of stance detection related to fake news recognition. This paper uses the BERT model to process news headlines and news article body and based on fine-tuning the model to continue training the BERT model, and finally detects the stance relationship between news headlines and news article body. In this paper, Fake News Challenge (FNC-1) is selected as the experimental environment, and the BERT model is used to process FNC task data, and an accuracy rate of 90.37% is obtained.

1 Introduction

In recent years, the task of text stance detection has become an emerging direction in the field of Natural Language Processing (NLP). The task of text stance detection also has an in-depth impact on other tasks in the field of NLP, such as analyzing online debates (Walker et al., 2012; Sridhar et al., 2015), and determining the authenticity of rumours on Twitter (Lukasik et al., 2016), or understanding the argument structure of a persuasive article (Stab and Gurevych, 2017).

The main purpose of this project is to use Pre-Trained Natural Language Model to automatically determine the relationship between news headlines and news article body, to quickly identify some fake news or irrelevant news.

The task involved in this project is Fake News Challenge (FNC-1)[1]. The input of FNC-1 includes a piece of news headline and a piece of news content. The news headline is a short text and the news article body is a long text. The output content of the task is the stance relationship between the news headline and the news article body, which are one of `Agree`, `Disagree`, `Discuss`, and `Unrelated` respectively. This paper uses the BERT model to process FNC task data and obtains an accuracy of 90.37%.

Our team's current achievements and unsolved problems are as follows. Up to now, the work progress we have achieved includes the following points:

- Focusing on the tasks we want to solve, more in-depth research on various NLP models and algorithms, and a list of references;
- Targeting For our task data, various analysis methods including exploratory data analysis (EDA) have been carried out to fully understand the input and output forms of the task and various internal attributes of the data;
- Figure out various technical routes for implementing NLP models, especially *Hugging-Face* [2] and *Google Colab* [3] as the main online and open-source implementation method;
- The Fake News classification based on the BERT model has been initially implemented, and good experimental results have been obtained.

At present, the difficulties we encounter include:

- During the investigation, we found many NLP sub-task scenarios, but the understanding of these sub-task scenarios is not yet in place.

[1] http://www.fakenewschallenge.org/

[2] https://huggingface.co/

[3] https://colab.research.google.com/

For example, with our in-depth research, we found that FNC is not only a Text Classification problem but also a Text Matching problem. We have further studied more models, methods and open-source implementations under the subdivision of Text Matching, but we have not fully grasped these contents.

- When using large-scale PTM for training, there is a lack of better computing equipment, and the use of Google Colab is not enough.

2 Related Work

This project plans to use the BERT pre-training model to solve the text matching problem between news headlines and news article body, to achieve position detection between them. Text matching refers to taking two texts as input and predicting their relationship category or relevance score by understanding their respective semantics. The representation-based text matching model focuses on constructing the representation vector of the text and predicts the relationship or correlation score between two paragraphs of text based on the representation vector. From the perspective of research methods, representation-based text matching models can be divided into traditional text matching models, DNN-based text matching models, and PTM-based text matching models. In this project, our choice is **BERT base model (uncased)** (`bert-base-uncased`), which is an uncased pre-trained model on English language using a masked language modelling (MLM) objective.

2.1 Traditional Text Matching Model

Traditional text matching models, such as TF-IDF (Ullman, 2011) and BM25 (Liu and Özsu, 2009), rely on hand-craft defined features to calculate the similarity between features. The Lltent Dirichlet Allocation (LDA) approximate inference algorithm based on Gibbs sampling can map sentences to implicit spaces, and get the latent semantic expression (Blei et al., 2003).

2.2 DNN-based Text Matching Model

The DNN-based Text Matching Model uses Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN) to automatically capture text features, and obtain the representation vectors of two pieces of text respectively. Deep structured semantic models (DSSM) (Huang et al., 2013) use the same deep neural network to calculate their respective representation vectors for two texts. Architecture-I (ACRI) (Hu et al., 2014) model uses CNN to fuse the semantic information of adjacent words in a sentence. IBM Watson Lab proposed a hybrid neural network structure combining CNN and LSTM networks (Tan et al., 2016). Tree-structured LSTM networks (Tree-LSTM) (Tai et al., 2015) model extends LSTM networks to tree topologies. The hierarchical encoding model (HEM) (Lu et al., 2020), through the hierarchical encoding module and hierarchical matching mechanism makes full use of the semantic features of the text to capture multi-view interactive information for text matching.

2.3 PTM-Based Text Matching Model

To solve the problem of polysemous words, scholars have proposed pre-training models such as ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and BERT (Devlin et al., 2018). ELMo uses a two-layer bidirectional long and short-term memory network to dynamically adjust word semantics using context information; Compared with ELMo, GPT uses Transformer (Vaswani et al., 2017) as a feature extractor, which avoids the shortcomings of ordinary RNN that cannot be calculated in parallel, but GPT's one-way language model only uses the above information, while ignoring the following information. The BERT model is similar to ELMo and GPT. It uses Transformer as the feature extractor, uses a two-way language model, takes into account the advantages of both ELMo and GPT, and is universal for various downstream tasks. Based on BERT, RoBERTa (Liu et al., 2019) uses dynamic masking mechanism instead of static masking mechanism and removing NSP (next sentence prediction) and other methods, which improves the way of BERT pre-training. The above-mentioned pre-training model can replace the representation vector of a sentence encoded in a network structure such as a CNN or an RNN in the previous model. For example, SBERT (sentence-BERT) (Reimers and Gurevych, 2019) obtained the semantic representation of two sentences separately based on BERT, and achieved good results.

3 Approach

This project plans to use the BERT pre-training model to solve the text matching problem between news headlines and news article bodies. The specific implementation method is to preprocess and splice the news headlines and news article body, and then combine the text of the news headlines and news article body into the BERT model, and fine-tune the BERT parameters on this basis.

3.1 Assumption

In the preliminary experimental stage, we tried to use the Sentence-BERT model to calculate the coding of each news headline and the coding of the news content, and calculate the similarity score between news headlines and the news article bodies. The results of preliminary experiments show that the coding similarity between most news headlines and news content is close to 1.0, which means that it is difficult to achieve the final task based on Sentence-BERT alone. The same meaning or the opposite meaning may only have a few word differences, so it is difficult to solve it through vocabulary-level pre-training coding.

In the era of NLP based on pre-training, the parameters of the pre-trained network are difficult to obtain, and it is also hard to optimize large-scale parameters, so the cost of modifying the data format is far less than modifying the algorithm or modifying the parameters. Instead of changing the algorithm on a large scale and changing the parameters of the algorithm, it is better to actively modify the format of the input data, which is especially important for individual researchers who lack funds and experimental equipment. This is also a more practical and feasible way for individual researchers to use the open-source pre-training models of large enterprises.

3.2 Tokenization

Before we use the pre-trained model, it is necessary to use the tokenizer associated with the pre-trained model. The tokenizer will split the text we provide in the same way as the pre-trained corpus, and it will use the same corresponding tags for indexing. The BERT model is difficult to handle texts with more than 512 characters, so long texts will be truncated at 512 characters, and the length of short texts will be automatically padded to 512. The tokenizer of the BERT model converts the text string into three encodings, namely `input-ids`, `attention-mask` and `token-type-ids`.

Among them, `input-ids` is the vocabulary index corresponding to each token in the sentence. The `attention-mask` array represents the part that the model needs to pay attention to, where 1 represents a meaningful token, and 0 represents the part that is completed by a blank token. The `token-type-ids` array represents which tokens are from the first sentence and which tokens are from the following sentences, which is very meaningful for tasks related to text matching.

3.3 BERT Model

The BERT model needs to accept the above 3 text encodings as input and output various related outputs including hidden states and attention. When experimenting with Huggingface's open source models, we can choose `BaseModelOutput`, `BaseModelOutputWithPooling`, etc [4]. The difference between these models is whether to perform *Pooling* or *Cross Attention* calculations after the output model. Assuming that the maximum length of the three token input is 512, then the length of the `last-hidden-state` of the BERT output model is also 512, and the `pooler-output` is also 512.

3.4 Output Layers

The result of the BERT output needs to be input into the *Bidirectional-Encoder* first. If the length of the `last-hidden-state` of the BERT output model and the length of the pooler-output are 512, then the output shape of the Bidirectional-Encoder is 512 × 512. The output of the Bidirectional layer is input into *Global Average Pooling* and *Global Max Pooling* respectively, and each of the above pooling layers will output an output result with a length of 512. Concatenating the outputs of the above two pooling layers will result in a vector with a length of 1024, and then drop out this vector. The vector after dropout can be output to the fully connected network, and the number of output neurons of the fully connected network is the number of types of classification problems.

4 DataSet and Evaluation

The task involved in this project is the Fake News Challenge (FNC-1). The goal of the Fake News Challenge is to explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem.

4.1 Data

For FNC-1 we have chosen the task of estimating the stance of a body text from a news article relative

[4] https://huggingface.co/transformers

to a headline. Specifically, the body text may agree, disagree, discuss or be unrelated to the headline.

4.1.1 Dataset

- Training Set: Pairs of headline and body text with the appropriate class label for each, for example `[headline, body text, label]`.
- Testing Set: Pairs of headline and body text without class labels used to evaluate systems, for example: `[headline, body text]`.

4.1.2 Input

A headline and a body text - either from the same news article or from two different articles. An Example of a headline is *"Robert Plant Ripped up $800M Led Zeppelin Reunion Contract"*, an Example snippet from body texts is *"... Led Zeppelin's Robert Plant turned down £500 MILLION to reform supergroup..."*.

In the experimental part, we need to clean and merge the original data and encode it into a vector form that can be processed by BERT.

4.1.3 Output

Classify the stance of the body text relative to the claim made in the headline into one of four categories:

- Agrees: The body text agrees with the headline, for example: *"... Led Zeppelin's Robert Plant turned down £500 MILLION to reform supergroup... "*
- Disagrees: The body text disagrees with the headline, for example: *"... No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together..."*
- Discusses: The body text discusses the same topic as the headline, but does not take a position, for example: *"... Robert Plant reportedly tore up an $800 million Led Zeppelin reunion deal..."*
- Unrelated: The body text discusses a different topic than the headline, for example: *"... Richard Branson's Virgin Galactic is set to launch SpaceShipTwo today..."*

4.2 Metrice

In this project, we use 6 evaluation metrics to measure the models used in this paper and some traditional natural language processing methods. These metrics are Accuracy, Precision, F1, Micro-average ROC, Marco-average ROC, and FNC, where the first 5 metrics are very common machine learning metrics and the FNC is specific to the FNC task.

FNC is a weighted, two-level scoring system:

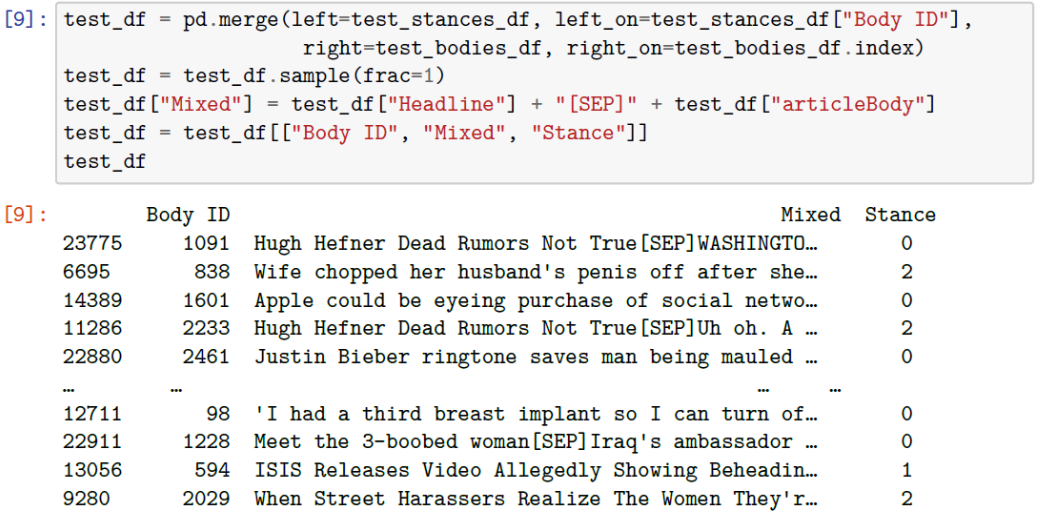
- Level 1: Classify headline and body text as related or unrelated 25% score weighting.
- Level 2: Classify related pairs as agrees, disagrees, or discusses 75% score weighting.

The FNC-1 organizers propose the hierarchical evaluation metric FNC, which first awards 0.25 points if a document is correctly classified as related (i.e., s $\in$ `AGR`; `DSG`; `DSC`) or `UNR` to a given headline. If it is related, 0.75 additional points are assigned if the model correctly classifies the document-headline pair as `AGR`; `DSG` and `DSC`. The goal of this weighting schema is to balance out a large number of unrelated instances.

4.3 Baselines

To compare the degree of merit of the models, we constructed 4 classifiers using traditional natural language processing and machine learning algorithms. In all-natural language processing tasks, text strings first need to be converted into some form of vector representation, whether it is deep learning such as RNN, BERT, etc., or traditional natural language processing algorithms such as Naïve Bayes and HMM. In this paper, we use the encoders `TfidfVectorizer` and `CountVectorizer` provided by the machine learning algorithm library `scikit-learn`, which encode the text into a numerical sparse matrix. In performing the vector representation, we also remove the stopwords from the text and use both *1-gram* and *2-gram* for a richer encoding.

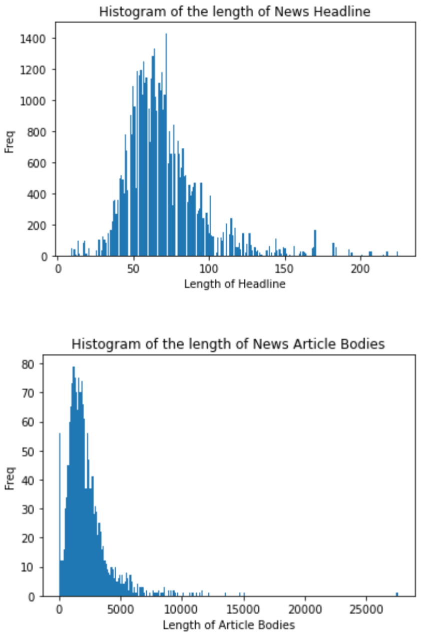
We chose the Bernoulli Naive Bayes algorithm (`BernoulliNB`) and the logistic regression (`LogisticRegression`) algorithm as classifiers, which are combined with the previous two encodings to form a total of four classifiers. We will use these are four classifiers to compare the BERT-based Fake News Stance Detection method.



```
[9]: test_df = pd.merge(left=test_stances_df, left_on=test_stances_df["Body ID"],
                          right=test_bodies_df, right_on=test_bodies_df.index)
     test_df = test_df.sample(frac=1)
     test_df["Mixed"] = test_df["Headline"] + "[SEP]" + test_df["articleBody"]
     test_df = test_df[["Body ID", "Mixed", "Stance"]]
     test_df
```

```
[9]:        Body ID                                              Mixed  Stance
     23775     1091  Hugh Hefner Dead Rumors Not True[SEP]WASHINGTO…       0
     6695       838  Wife chopped her husband's penis off after she…       2
     14389     1601  Apple could be eyeing purchase of social netwo…       0
     11286     2233  Hugh Hefner Dead Rumors Not True[SEP]Uh oh. A …       2
     22880     2461  Justin Bieber ringtone saves man being mauled …       0
     …          …                                                 …      …
     12711       98  'I had a third breast implant so I can turn of…       0
     22911     1228  Meet the 3-boobed woman[SEP]Iraq's ambassador …       0
     13056      594  ISIS Releases Video Allegedly Showing Beheadin…       1
     9280      2029  When Street Harassers Realize The Women They'r…       2
```

Figure 1: The processed data, where the Mixed column is the combined content of news headlines and news content, and the Stance column is the Stance encoded by integer mapping.



5 Experiments

In this section, we will describe some details of processing the raw data and how to load the BERT pre-trained model and encoder, and discuss the results of the BERT model, and finally, compare the results of the BERT model with the four Baselines mentioned before.

5.1 Data Preprocess

The raw data of the FNC project is divided into two files, `bodies.csv` and `stance.csv`, in which the news content is stored in the bodies file and the headlines and corresponding categories are stored in the stance file, and the data rows of these two files are associated with each other by a unique `Body ID`. Before fine-tuning the pre-training model, we need to use `Body ID` to associate the content of these two files, which can be achieved by using only the basic `left joint`. Because the BERT pre-training model uses some prescribed separators for symbol escaping, we need to add a `[SEP]` symbol between the news headline and news content when merging the news headline and news content. To facilitate the calculation of the model, we also need to prepare some conversion tools to convert Stance represented in string form to integer form, and to convert Stance represented in integer form to string form again.

5.2 Pre-trained Model Configuration

This paper uses the PyTorch-based HuggingFace open-source BERT pre-training model as the basis. The BERT model chosen is *bert-base-uncased* and the maximum text length supported by the model is 512, with all text content normalized to lowercase. The encoding tool used to pre-process the input text is *BertTokenizerFast*, and the BERT class used for text matching is *BertForSequenceClassification*.

For initial use, we need to use the initial pre-trained parameter files which downloaded from the remote repo, including: *special_tokens_map.json*, *tokenizer_config.json vocab.txt*, *config.json*, *pytorch_model.bin*, *tokenizer.json*. When the training is finished, we will save the updated data file in

Google Drive. For subsequent use, we only need to import these training data directly from Google Colab Pro to use them.

Parameter Name	Parameter Value
num_train_epochs	3
per_device_train_batch_size	16
per_device_eval_batch_size	64
warmup_steps	500
weight_decay	0.01
logging_steps	100

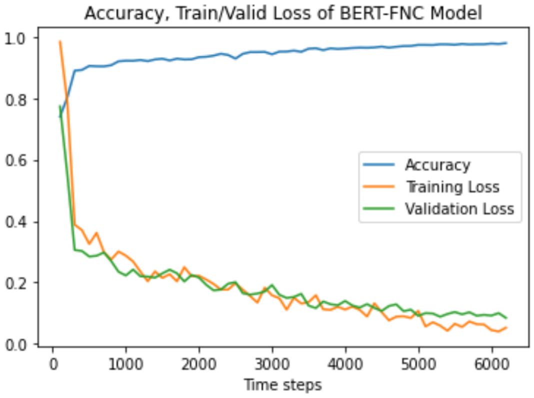


Figure 2: Accuracy, Training Loss and Validation Loss

Figure 2 shows the changing trend of Accuracy, Training Loss, and Validation Loss during the fine-tuning of the BERT model. It can be seen that the BERT model has completed convergence within three epochs and achieved good results.

5.3 Result Analysis

This paper implements a fake news classification model based on BERT and compares the results of the BERT model with the four baseline algorithms mentioned above. Figure 3 (shown on the following page) illustrates the ROC curves of the four reference models and the corresponding AUC values. The closer the ROC curve is in the upper left corner, the better the effect of the model. The greater the area under the ROC curve (that is, the AUC value), the better the model. From the perspective of the ROC curve and AUC value, the Logistics Regression model coded with TF-IDF has the best effect. Its Micro-Average AUC and Macro-Average AUC both exceed 90%. Figure 4 (shown on the following page) shows the ROC curve and the corresponding AUC value of the classification model based on the BERT pre-training model. We can see that the ROC curve of our BERT model has been already very close to the coordinate point in the upper left corner, and its Macro-Average AUC has reached 97%, while Micro-Average AUC reached 99%. The BERT model has a much better effect than the four basic models.

Table 1 (shown on the following page) compares the performance metrics in the BERT model with the four basic models. The BERT model exceeds 90% in the five indicators of Accuracy, Precision, F1, Micro-Average ROC, and Marco-Average ROC. BERT's FNC Score reached 10068.25 points, which is the only model with a score of more than 10, 000. In other words, from the point of view of accuracy, precision, F1, and the FNC score, the effect of the BERT model is much better than that of the other four baselines models.

The reason why the BERT model can get such good results is that its coding principle and internal structure are unique. The TF-IDF and Count Encoding methods are essentially a reflection of the Bag of Words model, that is, this processing method does not consider the text characteristics of the natural language context, nor does it consider the semantic relationship between individual words. The coding mode of the BERT model maps each word to a built-in dictionary, and the built-in dictionary of BERT contains word embedding information for each word, including synonyms, antonyms, and some grammatical changes. The BERT model is a two-way complex neural network that takes into account a lot of contextual information, which cannot be represented by the traditional Bag of Words model. This is also the reason why the BERT model can perform so well in this project.

6 Limitations

There are still gaps in this project, which means that our team must do more research in the future. The first limit is that we have not incorporated the FNC score into the loss function of the BERT pre-training model. Although the FNC Score obtained by the BERT model we implemented has exceeded the watershed of 10,000 points, the ratio of this score to the total score is only 86.41%, while the other scores have reached more than 90%. This shows that our BERT works great in the traditional machine learning evaluation system, but there is no special design for the FNC task. Our understanding of the BERT model and the natural language processing paradigm of deep learning is not deep

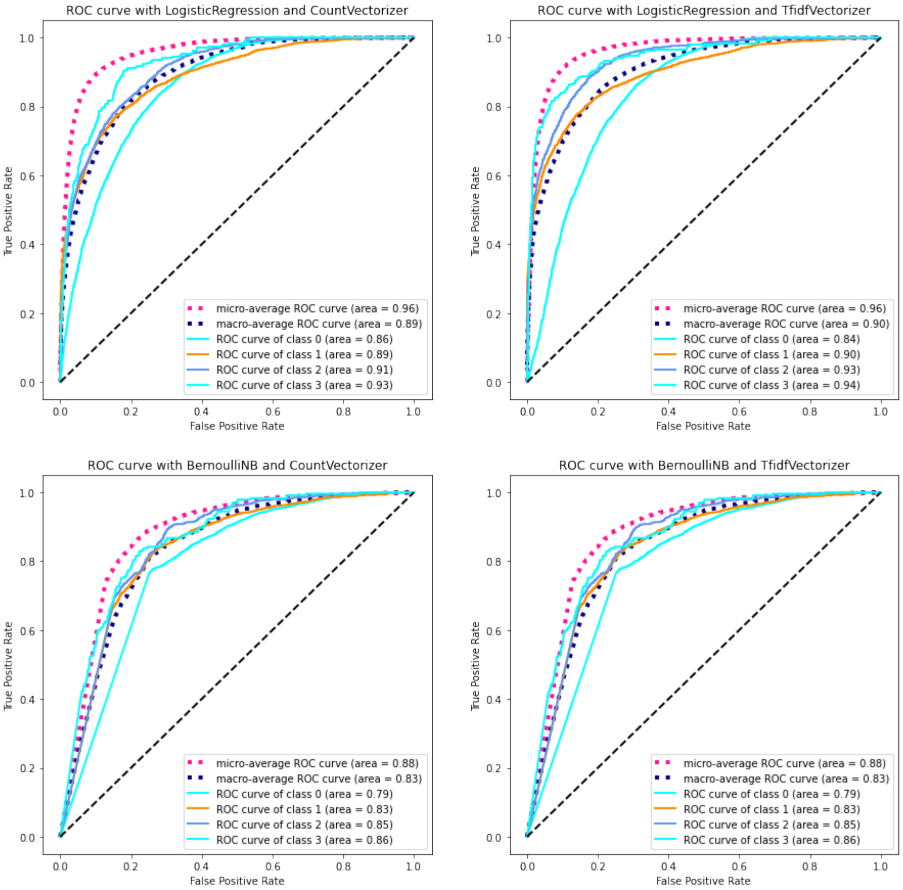


Figure 3: ROC-AUC of baselines

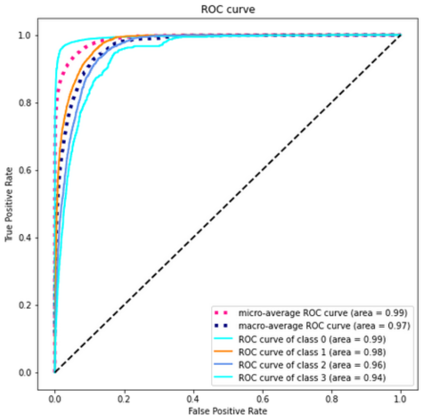


Figure 4: ROC-AUC of BERT

	Accuracy	Precision	F1	Micro-ROC	Marco-ROC	# FNC	% FNC
LR-Count	71.27%	59.20%	62.58%	96.00%	89.00%	4815.5	41.33%
LR-TFIDF	59.00%	57.94%	58.30%	96.00%	90.00%	5026.75	43.14%
NB-Count	66.51%	56.59%	60.90%	87.00%	84.00%	4834.75	41.49%
NB-TFIDF	66.51%	56.59%	60.90%	87.00%	84.00%	4834.75	41.49%
BERT	**90.37%**	**90.60%**	**90.43%**	**99.00%**	**97.00%**	**10068.25**	**86.41%**

Table 1: Performance Metrices

enough, so we did not flexibly modify the optimization function of the BERT model. We need to do that more in future research.

The second limitation is that we have not studied the PROMPT mechanism behind the BERT model further and have not fully exploited the potential of the PROMPT mechanism. The PROMPT mechanism will certainly become the most important new development in the NLP area in the coming years, so we need to continue to explore this direction.

7 Conclusion

This article is dedicated to using the BERT pre-training model to implement the classifier of stance detection related to fake news recognition. This paper uses the BERT model to process news headlines and news article body and based on fine-tuning the model to continue training the BERT model, and finally detects the stance relationship between news headlines and news article body. This paper implements a fake news classification model based on BERT and compares the results of the BERT model with the four baseline algorithms. BERT model is used to process FNC task data, and accuracy of 90.37% is obtained.

There are still some limitations in this project due to our shortages of related knowledge. It is a promising field to explore other behaviours or methods which can be involved in fake news detection in future. Therefore, further research is strongly needed, and we will keep on doing researches in this field when time permits.

8 Acknowledgments

During the project, we had several meetings with Prof. Angel Chang of the School of Computing Science, Simon Fraser University. We would like to thank Prof. Chang for giving us beneficial suggestions on determining the topic, the preparation of the presentation and the arrangements of the source file, etc. We also thank the teaching team for providing feedback on our abstract, milestone paper and presentation.

9 Contributions

First of all, the group worked beautifully independent of the final product. There was good communication among all members, everyone in the group was consulted and participated and the group interactions were respectful. Then we will present the distribution of tasks.

9.1 Zeyong Jin

This group member made the following contributions to the group.

1. Researched the BERT model.
2. Sorted out the principle part of BERT.
3. Designed the pre-training code of BERT.
4. Finished the presentation and the report.

9.2 Yuqing Wu

This group member made the following contributions to the group.

1. Realized the baseline design of TF-IDF.
2. Realized the baseline design of Count Vector.
3. Implemented all the comparative experiments.
4. Finished the presentation and the report.

9.3 Zhi Feng

This group member made the following contributions to the group.

1. Designed the experimental process.
2. Included data pre-processing, comparison and visualization.
3. Implemented the evaluation code and drew all evaluation scores.
4. Finished the presentation and the report.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Ling Liu and M Tamer Özsu. 2009. *Encyclopedia of database systems*, volume 6. Springer New York, NY, USA:.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Wenpeng Lu, Xu Zhang, Huimin Lu, and Fangfang Li. 2020. Deep hierarchical encoding model for sentence semantic matching. *Journal of Visual Communication and Image Representation*, 71:102794.

Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–398.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 116–125.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 464–473.

Jeffrey Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 592–596.