University of Toronto

Creative Applications of Natural Language Processing

(ECE 1786)

Fake News Detection

Final Report

Liu Zeyuan

1006626894

Mon Dec 14, 2022

1710 words in total

(Excluding Title Pages and References)

Permission

permission to post video: yes
permission to post final report: yes
permission to post source code: no

1 Introduction

The widespread use of internet-based methods of communication and information dissemination has greatly facilitated modern living. It's undeniable that the internet has improved our quality of life and made it possible to have access to vast amounts of information.

However, because to the prevalence of social media, ordinary people may easily produce and modify this data, and sharing it might have disastrous consequences. Since many people on social media have a tendency to blindly trust what they read or see shared by their friends, it is easy for false tales to go viral and gain a lot of traction. Technology giants Google and Facebook have started beta testing new tools to assist consumers identify and report fake news sites. Nevertheless, Both of the two companies haven't generated mature and convincing methods to automatically detect those fake news. It's necessary to look deep insight into these fields and develop a machine learning model that can handle this task.

My project ”Fake News Detection” aims at deciding whether a given piece of news is convincing or not based on news content and title, providing reference for fake news identification and helping people get rid of the fake news surrounding them.

2 Illustration

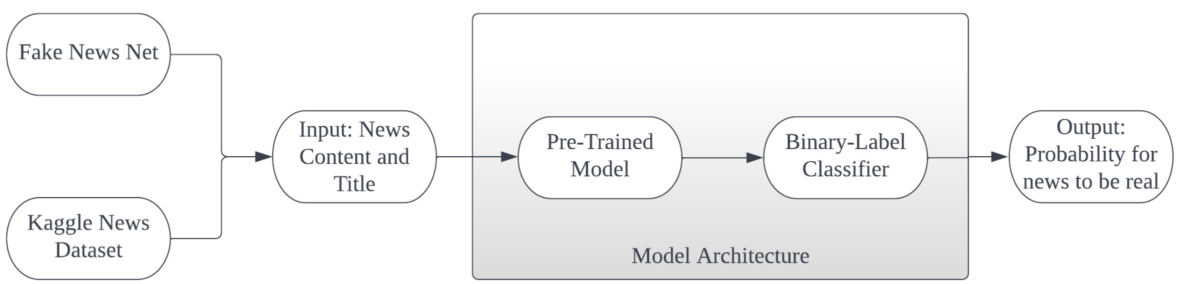


Figure 1: Overall model of Fake News Detection

3 Background & Related Work

In recent years researchers have applied deep learning models to automatically learn effective features for rumor detection. Nearly all methods are text-based supervised fake news detection methods, which take the textual information of news as input to detect fake news. A total of six prominent works from 2019-2020 have brought a huge impact on fake news detection, as shown in Figure 2.

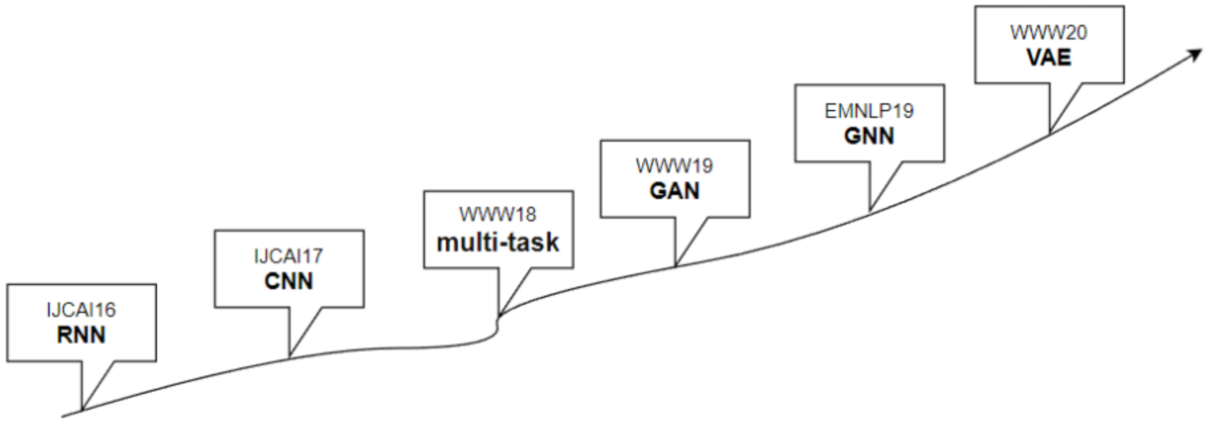


Figure 2: Deep Learning Methods Development to Detect Rumors

Dr. Ma Jing applied deep learning technology to fake news detection for the first time [1]. This method inputs each sentence of the news into the cyclic neural network RNN, LSTM or GRU, uses the hidden layer vector of the cyclic neural network to represent the news information, and inputs the hidden layer information into the classifier to obtain the classification result [1]. Their basic GRU plus embedding unit is shown in Figure 3.

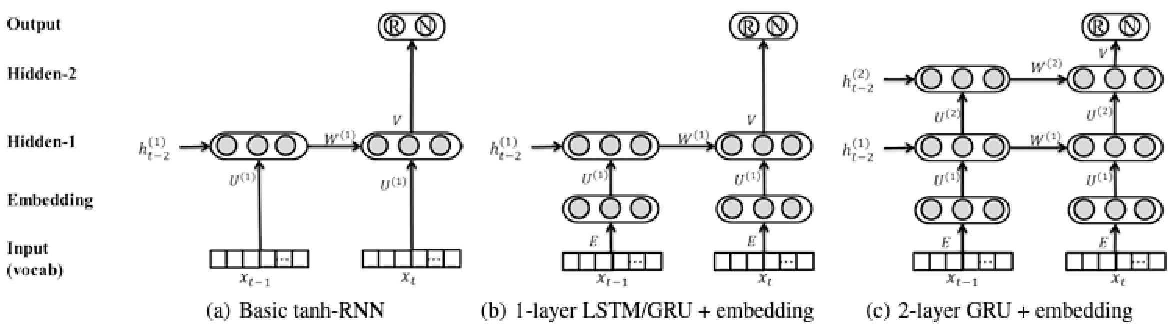


Figure 3: GRU based Rumor Detection Model Unit

Yu et al. used convolutional neural networks to model news articles for the first time in 2017 [2]. The model maps each post of a news event to a vector space, then concatenates each post vector to form a matrix, then uses a convolutional neural network to extract text features, and inputs the obtained embedding vector into the classifier to obtain the final classification result [2]. The model structure unit is shown in Figure 4.

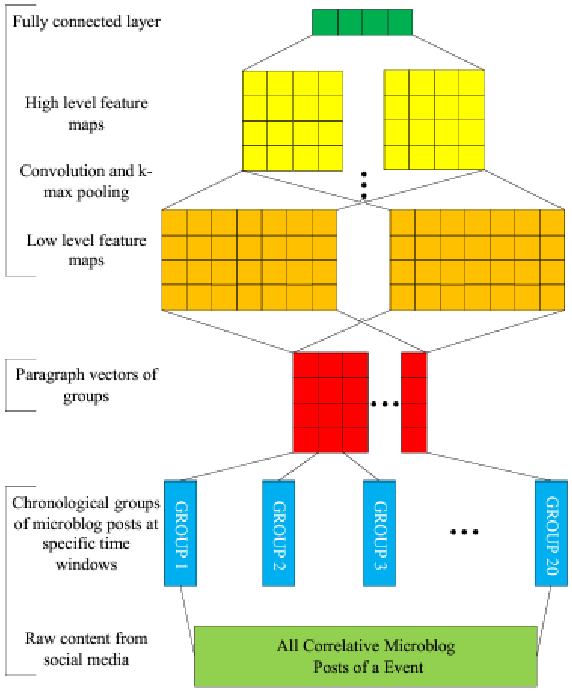
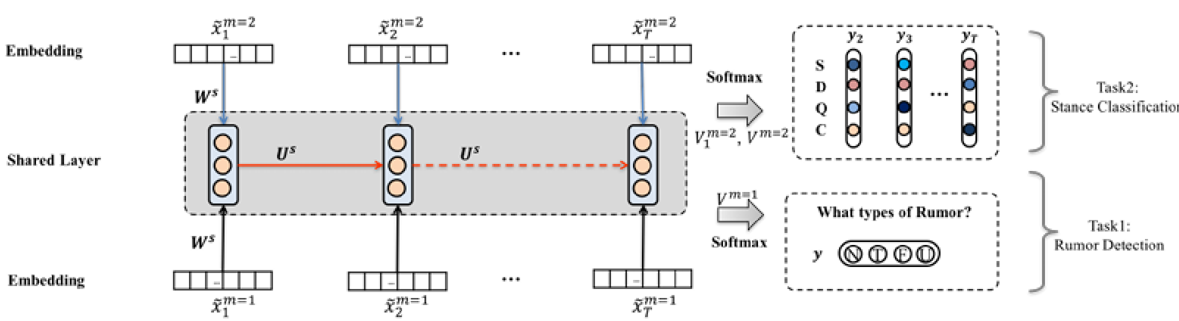


Figure 4: CNN based Model Diagram

Dr. Ma Jing also applied the idea of multi-task to fake news detection for the first time in 2018 [3]. This article combines the fake news detection task and the position classification task into a multi-task model, and uses RNN as the backbone to train the two tasks and achieves good results [3]. The uniform shared-layer structure is shown in Figure 5.



(a) Uniform Shared-Layer Architecture

Figure 5: Multi-Task based Model Unit

Later on, GAN, GNN and VAE is also applied to fake news detection, but due to their complicated model structures, I'll not describe them in detail in this project.

4 Data Processing

The data I have collected and cleaned so far is from Fake News Net, Kaggle Fake News Data.

In Fake News Net, the dataset is split into PolitiFact Dataset and GossipCop Dataset. However, most of the news link is out-of-dated, and removed by the websites. I figured out way to search the link directly from PolitiFact Website and GossipCop Website itself. I noticed that the two websites had their own official account on Twitter, so I established a flask server and send request to Twitter to get the news they published . The response contains the related link.

To get the label and news's information from the link, it is different for PolitFact and GossipCop.

In PolitiFact, journalists and experts in their fields look over the political news and give fact-checking evaluations to say whether a news story is real or not. I use these claims as facts for both fake and real news stories. In the results of PolitiFact's fact-checking evaluation, the source URLs of the web pages that posted the news articles are listed. These URLs can be used to get the news content that the news articles are about. It is down by using newspaper3k package in python. However, some of the source news articles' web pages are taken down and can no longer be found. To solve this problem, firstly, I check to see if the page was archived and automatically get the content from the Wayback Machine; if not, I use Google websearch in an automated way to find news articles that are most related to the actual news.

GossipCop is a website where you can check the facts of entertainment stories from different news sources. GossipCop gives news stories ratings on a scale from 0 to 10 that show how fake or real they are. From what I've seen, almost 90 percent of the stories on GossipCop have scores of less than 5. This is mostly because the point of GossipCop is to show more fake stories. I crawl the news articles on E! Online, which is a well-known and trusted website for entertainment news, to find real entertainment news stories. I think that every article on E! Online is real news. I get all of the fake news stories from GossipCop that have scores of less than 5.

In this way, I get the news' titles, news' texts and news' labels for both PolitiFact and GossipCop. I concatenated two datasets to form the Fake News Net dataset.

For Kaggle Fake News Data, it directly contains the news title, news contents and labels. I simply moved other unrelated columns to form Kaggle dataset.

Finally, I concatenated two datasets to form my training datasets. Also, I directly link the news titles to news contents to form the final training samples. Then I do some analysis on the training datasets. To see the number of examples in each class, the results is shown in Figure 6.

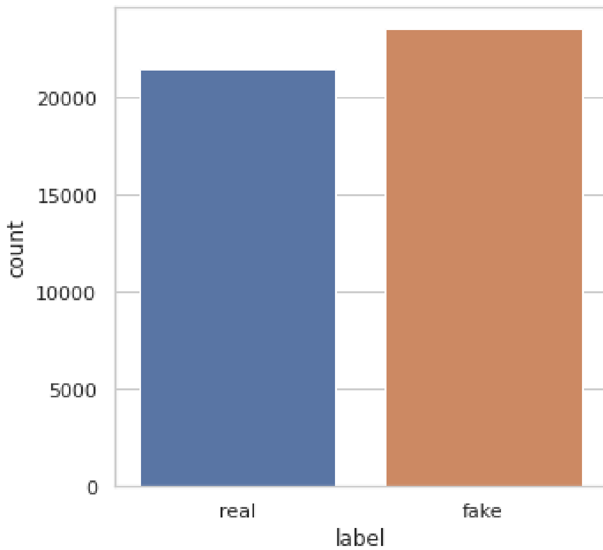


Figure 6: Number of Examples in Two Classes

One training example is shown below in Figure 7:

	text	label
4380	Republican National Committee cuts ties with Moore: PoliticoWASHINGTON (Reuters) - The Republican National Committee is withdrawing support for Alabama Republican Senate candidate Roy Moore after allegations surfaced that he had sexual contact with teenage girls decades ago, Politico reported on Tuesday, citing a senior party official. The move further isolates Moore, who has denied the accusations. Republican leaders have distanced themselves from the candidate and the National Republican Senatorial Committee cut ties with him last week.	1

Figure 7: One Training Example

5 Architecture and Software

The architecture of the model I found that had the best performance is BERT model with classification head.

For specific, I used bert-base-uncased pre-trained model, which had 110M # parameters. By the way, the tokenizer I used was also bert-base-uncased pre-trained tokenizer.

The structure of my model is shown below in Figure 8:

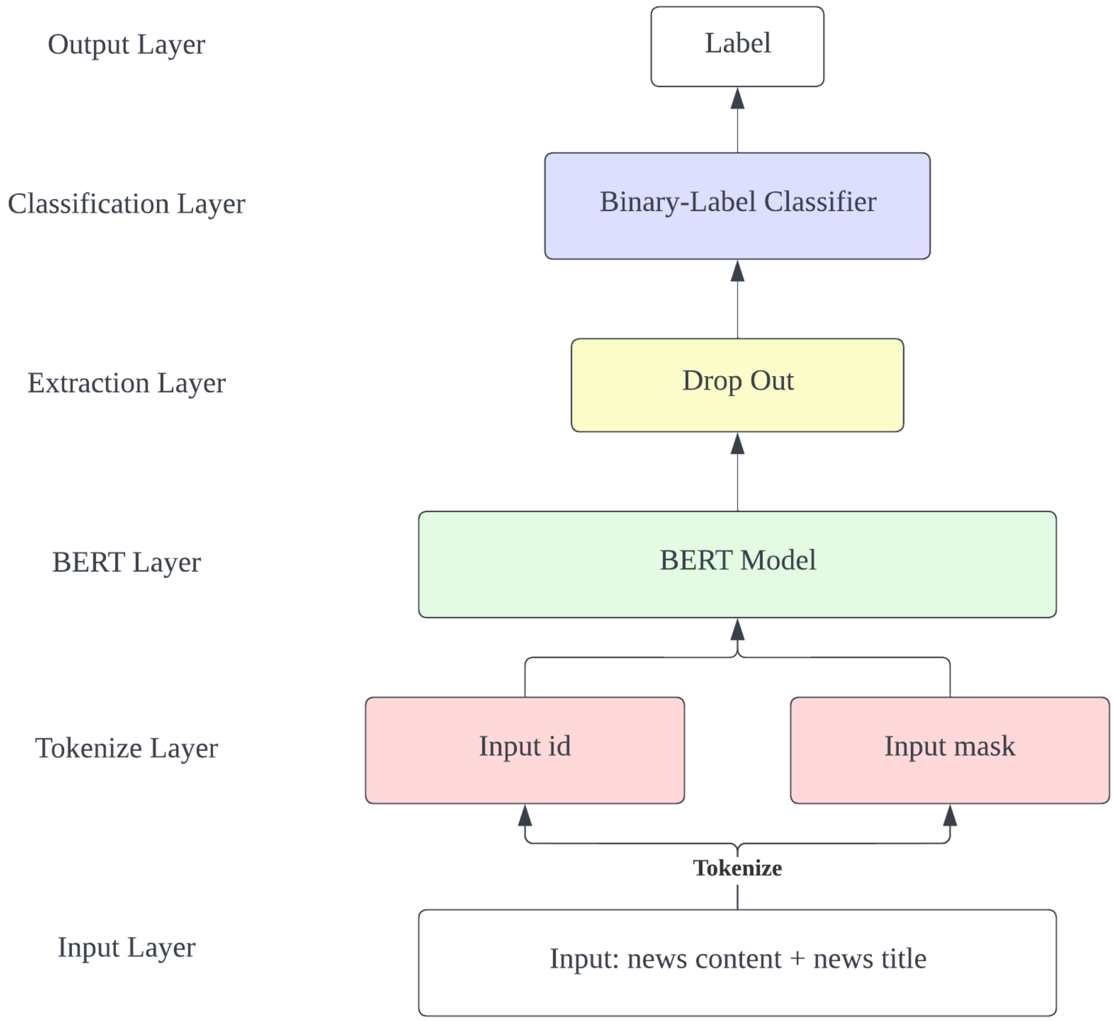


Figure 8: Model Structure

Notice that there are two inputs, one is for input_ids, and the other if for attention_mask, as introduced in assignment 4.

For the Drop Out Layer, I choose a drop of rate 0.5 before entering tanh MLP Layer, and anther drop out layer of rate 0.2 before entering sigmoid MLP Layer.

For the hyperparameter, I chose Adam as optimizer, with learning rate as 1e-05, loss function as binary_crossentropy, epochs as 10, batch_size as 30.

6 Baseline Model

The baseline model I will use is a 2-layer GRU together with word embedding, as stated in [2]. It is a Rumor Detection Milestone Model developed in 2016, and lots of newest fake detection models take this GRU models as their baselines. As a result, I also try to compare my architectures with this model.

The diagram of the model unit is shown in Figure 9 and related equation is shown below.

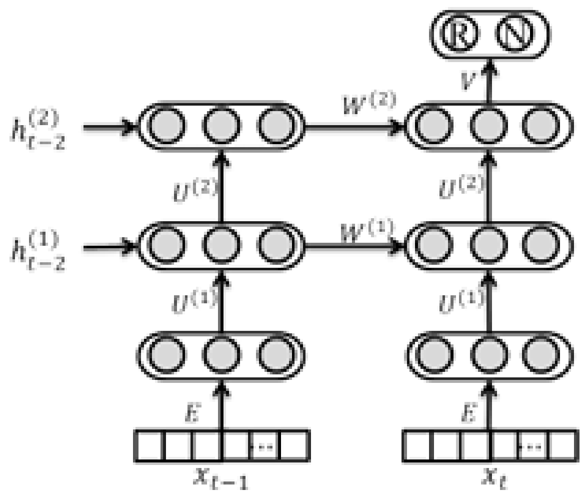


Figure 9: 2-Layer GRU plus Embedding Model Diagram

$$
\begin{aligned}
x_e &= x_t E \\
z_t^{(1)} &= \sigma\left(x_e U_z^{(1)} + h_{t-1}^{(1)} W_z^{(1)}\right) \\
r_t^{(1)} &= \sigma\left(x_e U_r^{(1)} + h_{t-1}^{(1)} W_r^{(1)}\right) \\
\tilde{h}_t^{(1)} &= \tanh\left(x_e U_h^{(1)} + \left(h_{t-1}^{(1)} \cdot r_t^{(1)}\right) W_h^{(1)}\right) \\
h_t^{(1)} &= \left(1 - z_t^{(1)}\right) \cdot h_{t-1}^{(1)} + z_t^{(1)} \cdot \tilde{h}_t^{(1)} \\
z_t^{(2)} &= \sigma\left(h_t^{(1)} U_z^{(2)} + h_{t-1}^{(2)} W_z^{(2)}\right) \\
r_t^{(2)} &= \sigma\left(h_t^{(1)} U_r^{(2)} + h_{t-1}^{(2)} W_r^{(2)}\right) \\
\tilde{h}_t^{(2)} &= \tanh\left(h_t^{(1)} U_h^{(2)} + \left(h_{t-1}^{(2)} \cdot r_t^{(2)}\right) W_h^{(2)}\right) \\
h_t^{(2)} &= \left(1 - z_t^{(2)}\right) \cdot h_{t-1}^{(2)} + z_t^{(2)} \cdot \tilde{h}_t^{(2)}
\end{aligned}
$$

For the hyperparameter, AdaGrad algorithm is going to be used for parameter update. I will set the vocabulary size K as 5000, the embedding size as 100, the size of the hidden units as 100 and the learning rate as 0.5.

7 Quantitative Results

Surprisingly, my model do extremely well. The test accuracy and F1-score of my model and baseline model are shown below in Figure 10.

	BERT Model			GRU Model		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Fake (0)	0.96	0.91	0.93	0.82	0.75	0.78
Real (1)	0.93	0.97	0.95	0.78	0.84	0.81
Accuracy			0.94			0.80
Macro Avg	0.94	0.94	0.94	0.80	0.80	0.80
Weighted Avg	0.94	0.94	0.94	0.80	0.80	0.80

Figure 10: Quantitative Analysis of BERT Model and Baseline Model

The accuracy of my model is far more accurate than the baseline model, with both F1-score and Accuracy greater than Baseline Model.

8 Qualitative Results

Both failure examples and success examples are given in the figure below.

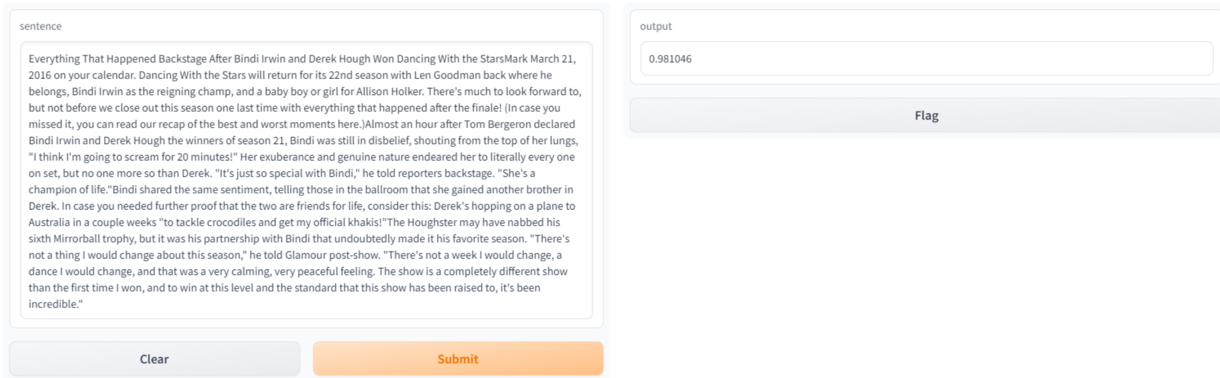


Figure 11: Failure Example

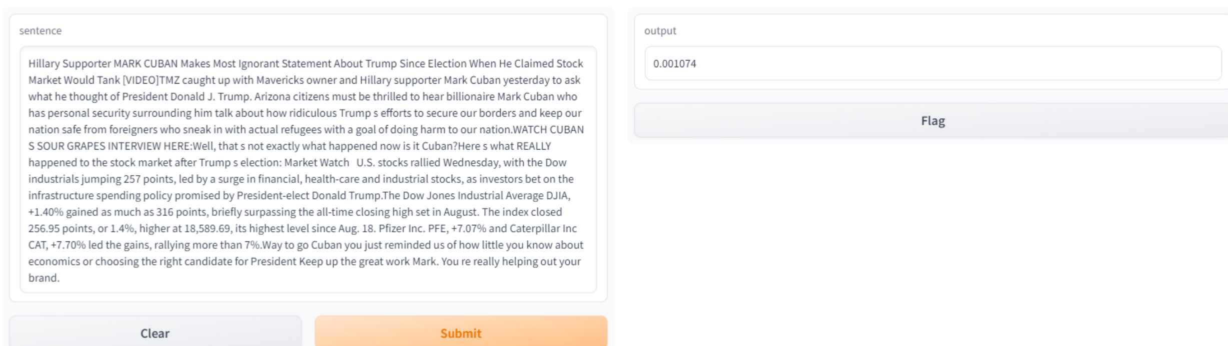


Figure 12: Success Example

To figure out the reason, it's a good idea to look at how the model is trying to detect fake news. Looking through all the data, I notice that most of news that include references from other people have a high probability of being regarded as real news. So the model may mostly take this factor into account and make this a justification standard.

Taking the above two examples, the failure example quotes what many others have said, and the model regard this piece of news having a high probability of being real. However, the actual label of this news is false. For the success example, the news doesn't include any references or data, so the model detects it as fake news, which match the label got from GossipCop.

9 Discussions and Learning

1. The overall performance of the model is extremely good. And I kind of get an insight into how the model detect the news is fake or not. It probably take whether the news have enough references or data into consideration.

2. In data processing part, the number of news in Kaggle dataset is about 2 times the Fake News Net

Dataset. However, I've already investigated Gossipcop and PolitiFact back to 2014. The reason why the number of Fake News Dataset Data is not enough is that the comments in GossipCop and PolitiFact doesn't always directly say that the news is fake or real in the paragraph, actually, some of the authors say "it's not recognized by public", which makes me difficult to get the 0/1 label from the comments by writing simple code. Therefore, lots of news that can't be decided as fake or real are skipped. There may be some more effective methods to label the data.

3. The real news may become fake news by just simply replacing the person's name, which means I can produce my own fake news dataset by revising some people's name in real dataset. However, in this case, since my model mostly take references and data into consideration, my model can't recognize the revised news as fake news. And this problem may take far more time for me to investigate.

10 Individual Contributions

Since I am a group of my own, all works are done by me. The details are listed below.
Collect Dataset – Zeyuan Liu
Data Cleaning – Zeyuan Liu
Baseline Achievement – Zeyuan Liu
Architecture Design – Zeyuan Liu
Model Training – Zeyuan Liu
Model Analysis – Zeyuan Liu
Model Optimization – Zeyuan Liu
Report Writing & Slides Making – Zeyuan Liu

References

[1] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, p. 3818–3824, AAAI Press, 2016.

[2] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 3901–3907, 2017.

[3] J. Ma, W. Gao, and K.-F. Wong, "Detect rumor and stance jointly by neural multi-task learning," in Companion Proceedings of the The Web Conference 2018, WWW '18, (Republic and Canton of Geneva, CHE), p. 585–593, International World Wide Web Conferences Steering Committee, 2018.