Fake News Stance Detection with Siamese CNN

Yuqi Liu / 301355758
Xinyue Ma / 301297142
Yufei Wang / 301338190

Abstract

As media becomes the main source of important news, the authenticity of those news becomes one of the most important things. The ultimate goal of the Fake News Challenge is meant to identify fake news through computers automatically. In order to achieve this ultimate goal, people need to approach the ideal results step by step. At the first step of this challenge, people need to teach computers to classify whether a headline and the associated body text are related or not. The baseline method provided by the organizers of this challenge is a gradient boosting classifier associated with 10-fold cross-validation. In this project, we propose a deep learning model using the idea Siamese model to check whether a headline and a body text are related. We first extract features from headlines and body texts using one-dimensional CNN respectively. And then we implement an MLP classifier to decide the relationship between a headline and the associated body text. The relationship between a headline and a body text contains agree, disagree, discuss, or unrelated.

1 Introduction

As the media becomes easier to spread the news around the world, only one piece of news can have a sensational response. However, many people are making up stories with an intention to deceive for secondary gain. And fake news can cause a lot of confusion about the facts of current events. Thus, this is one of the most serious challenges faced by the news industry today. Fake News Challenge (Pomerleau and Rao, 2017)is meant to explore the potential ability of artificial intelligence technologies including machine learning and natural language processing to solve the problems about fake news. As deep learning develops rapidly, it is possible to use deep learning models to estimate the stance of a body text from a news article relative to a headline.

So far, it is difficult for AI to directly point out fake news using any existing technologies since the dataset for training such a model is not available and sometimes some fake news is even plausible from human perspectives. Due to the difficulties of directly point out whether a story is real or not, Fake News Challenge proposed the first stage that deep learning can do to combat the fake news problem. The first stage of the Fake News Challenge focuses on the task of stance detection. Stance detection is the task that uses artificial intelligence technologies to estimate the stance of a body text from a news article relative to a headline. The relationship between a headline and a body text includes “agree”, “disagree”, “discuss” and “unrelated”. (the example will be given in the Section 4).

1. **Agrees**: The body text agrees with the headline.
2. **Disagrees**: The body text disagrees with the headline.
3. **Discusses**: The body text discuss the same topic as the headline, but does not take a position
4. **Unrelated**: The body text discusses a different topic than the headline

In this report, we will use combine sentence embeddings and the Siamese model to estimate the stance of a body text from a news article relative to a headline. The Siamese model contains a one-dimensional convolutional neural network model and a multilayer perceptron neural network. One-dimensional CNN is used to extract features from headlines and body texts. MLP model is used to classify the relationship between a headline and the associated body text. These two models will work

together to estimate the stance of a body text from a news article relative to a headline. The details will be introduced in the model part.
The remaining parts of this report are organized as follows: In Section 2, we will introduce some previous work related to Fake News Challenge; in Section 3, we will introduce the details of the neural network we use for fake news stance detection; in Section 4, we will introduce the datasets we use for this task and demonstrate the pre-processing part we do to the datasets; in Section 5, we will demonstrate the details about experiments and the results of experiments; In Section 6, we will present the results of our experiments. In Section 7, we will conclude the results of experiments and analyze them. And we will also discuss future work.

2 Prior related work

William and Ferreira (Ferreira and Vlachos, 2016) present a dataset that is derived from a digital journalism project for rumor debunking. This dataset contains 300 rumored claims and 2595 associated news articles. These claims and articles are collected and labeled by journalists. The journalists estimate the veracity of claims and articles. And each article is associated with a headline. The relationship between a headline and an article contains "for", "against" and "observing", where "observing" indicates that there is little information in the article related to the associated claim. The current Fake News Challenge extends the work of William and Ferreira. Instead of using stances including "for", "against" and "observing", the current Fake News Challenge uses "agree", "disagree", "discuss" and "unrelated".
The Fake News Challenge (Pomerleau and Rao, 2017) was proposed in 2017 and there were many teams that tried to increase the accuracy of fake news stance detection. The top-3 teams successfully increased the relative score to over 81. The rank-1 team is called SOLAT in the SWEN (Baird et al., 2017) by Talos Intelligence and their model is based on a weighted average between gradient-boosted decision trees and a deep convolutional neural network. They use 50% of the tree model predictions and 50% of the deep learning model predictions. The rank-2 team is called Athene (Hanselowski et al., 2018) and they use handcrafted features and five MLP models where each MLP has six hidden layers. The rank-3 team is called UCL Machine Reading (Riedel et al., 2017) and they use an MLP with bag-of-words features.

3 Approach

3.1 Baseline

The baseline method is provided by the Fake News Challenge. It uses a gradient boosting classifier with 10-fold validation. It creates features according to the common words in both headlines and body texts. And using those features to train the gradient boosting classifier. The handcrafted features include word/n-gram overlap features and indicator features for polarity and refutation. For the word/n-gram overlap features, the algorithm gets the word-level intersection between a headline and the associated bodies and uses

$$\frac{INTERSECTION(headline, bodytext)}{UNION(headline, bodytext)}$$

as one of the features for this headline-body pair. In addition, the algorithm will also check whether there are "refuting words" including "fake", "deny" and so on as another feature of this headline-body pair and use the "refuting words" to calculate polarity. And the number of a word in the headline appears in the body text is another important feature of this headline-body pair. The gradient boosting classifier uses these features to learn and make predictions about the stances between headlines and body texts.

3.2 Deep learning

3.2.1 Siamese model

Siamese model (Bromley et al., 1993) is usually used to classify whether two images are the same based on the features extracted from the two images. We find that the Siamese model could also be used to identify the relationship between two possible events that have interrelations. So we decide to use the Siamese model to do the fake news stance detection. The Siamese model contains two parts. The first part of this model is a one-dimensional convolutional model. Since natural language processing usually uses one-dimensional embeddings as the input of the neural network, we use a one-dimensional convolutional neural network to extract features from embeddings. The key point to this CNN part is that we extract features from headlines and body texts using CNNs respectively but these two CNNs actually share their weights so that we can make connections between the headlines and the associated body texts. After the CNN

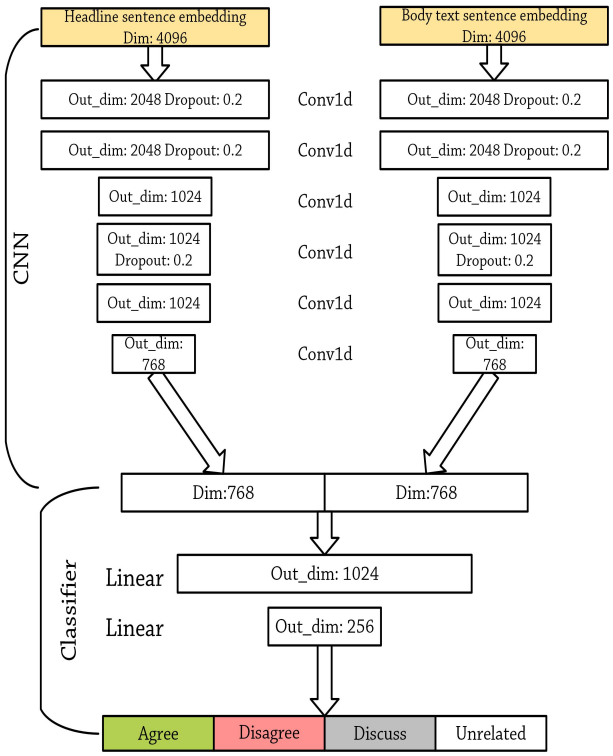
model, we get two 768-dimensional vectors respectively. And the second part is an MLP neural network containing two hidden layers. This MLP neural network is used as the classifier. In this neural network, we simply concatenate two 768-dimensional vectors into one 1536-dimensional vector as the input of the MLP neural network. There are two reasons why we only concatenate these two vectors rather than making some calculations to them. First, although we force them into the sentence embeddings having the same dimension (will discuss this in the next subsection), it is unreasonable to make some calculations to them since we do not need the distance between their features. Second, only using concatenated vectors is easy to manipulate. And after the MLP neural network, the output will be a 4-dimensional vector. Each dimension represents one of the relationships between the headlines and the associated body texts. At last, we use a softmax layer to decide the most possible relationship between a headline and a body text. The details of our model are indicated in the Figure 1.

3.2.2 Sentence embeddings

Usually, word embeddings are used as the input of natural language processing since there are lots of word embedding datasets and word embeddings are easy to use with only losing the meaning between words. However, we decide to use sentence embeddings as the input of our model. There are two reasons. First, since we use a one-dimension convolutional neural network, we need to input multiple headline-body pairs at the same time to ensure the productivity of our model. It is hard to input multiple headline-body pairs at the same time since headlines and body texts have different lengths. If we do not force them into sentence embeddings that have the same lengths, we would have to input headline-body pairs one-by-one, which is clearly not an appropriate and efficient way to do this task. Second, since we are going to explore the relationship between headlines and body texts, the most important thing is the relationship between sentences.

For encoding sentences into sentence embeddings, we use a pre-trained model named InferSent (Conneau et al., 2017). InferSent is a sentence embeddings method that provides semantic representations for English sentences. It is trained on natural language inference data. InferSent is trained on fastText word embeddings. It encodes sentences to 4096-dimensional vectors based on the word embeddings of the words in the sentences.

Figure 1: Details of the model used for stance detection



4 Data

The data provided is (headline, body, stance) instances, where the stance is one of unrelated, discuss, agree, disagree. The training dataset contains 49972 instances and 1683 body texts and the test dataset contains 25413 instances and 904 body texts. The bodies contain the body text of articles with corresponding IDs. The IDs are used to make pairs between headlines and bodies. The instances contain the labeled stances for pairs of article headlines and article bodies. The example is provided in the Table 1.

4.1 Train and validation split

The original train dataset is shown in the Table 2. For the purpose of training a good model, we decide to split the train dataset into train dataset and validation dataset. The numbers of stances for each category after splitting are shown in the Table 3.

Table 1: Examples of the relationship between a body text and four headlines

Body text	
Over the weekend there was a rumor flying around that someone actually stole the Batmobile off the set of Batman V Superman: Dawn of Justice. How cold something the big and bulky get stolen off a secure set, but stay hidden away in a major city like Detroit? While I may not know the insurance coverage for this, it seems that the stolen Batmobile was nothing more than a publicity stunt. And to add some more fun to this, director Zack Snyder tweeted a photo of the culprit. Hint: Snyder is currently in a twitter war with this movie. Hit the jump to check it out. So there you have it, it was nothing more than a publicity stunt created by social media. It also a great way to market both films, but more importantly we get to see more of the Batmobile. Now it's Bad Robot or Disney's move. The rivalry all started back last July when Snyder tweeted a shot of Superman (Henry Cavill) donning a Sith robe while wielding a red lightsaber. The director added the superjedi to rub some salt on the wound. Of course then Bad Robot responded with the following tweet. They even called out the Batman V Superman: Dawn Of Justice director with "The C3Ped Crusader":	
Headlines	
Agree	BATMOBILE NOT STOLEN, MTV CONTEST PRIZE REMAINS UNCLAIMED.
Disagree	Batmobile Stolen From "Batman v Superman: Dawn of Justice" Set, Zack Snyder Knows Who Did It.
Discuss	UPDATE: BATMAN v SUPERMAN Batmobile Reportedly Stolen By (Shocker!) Detroit Locals.
Unrelated	Michelle Obama's face blurred by Saudi state television.

Table 2: Numbers of headline-body pairs in the original train dataset

Number of headline-body pairs in the dataset				
	Agree	Disagree	Discuss	Unrelated
Training dataset	3678	840	8909	36545
Test dataset	2600	697	4484	18349

Table 3: Numbers of headline-body pairs in the dataset after splitting

Number of headline-body pairs in the dataset				
	Agree	Disagree	Discuss	Unrelated
Training dataset	2916	678	7109	29647
Validation dataset	762	162	1800	6898
Test dataset	2600	697	4484	18349

4.2 Problems about the dataset

The information of FNC-1 dataset					
	Agree	Disagree	Discuss	Unrelated	Total
# Instances	6278	1537	13393	54894	76102
Test dataset	8.25%	2.02%	17.6%	72.1%	

From the Table 2 and Table 3, we can see that the number of unrelated stances is much higher than other stances. Even we treat "agree", "disagree" and "discuss" as one category "related", the number of "unrelated" stances is still as three times as the number of "related" stances. This is an unbalanced dataset. There are several ways to address an unbalanced dataset including Collecting more data and resampling the dataset. First, it is impossible for us to collect headline-body pairs with such a huge amount and label them one by one. Second, for resampling the dataset, neither add copies of instances from the under-represented class nor delete instances from the over-represented class is a reasonable way to deal with since over-sampling would make the model learn those same "related" stances again and again and under-sampling would decrease the size of the dataset. After checking some previous work, we decide to keep the dataset unchanged.

5 Experiments

5.1 Evaluation

5.1.1 Score

The evaluation score is based on a weighted, two level scoring system indicated in the Fake News Challenge (Pomerleau and Rao, 2017).

1. **Level 1**: Classify headline and body text as related or unrelated 25% score weighting/

2. **Level 2**: Classify related pairs as agrees, disagrees, or discusses 75% score weighting

$$Score_1 = Accuracy_{Related,Unrelated}$$

$$Score_2 = Accuracy_{Agree,Disagree,Discuss}$$

$$Score_{FNC} = 0.25 * Score_1 + 0.75 * Score_2$$

Score1 represents the number of correct "related/unrelated" labels. Score2 represents the number of correct "agree", "disagree" and "discuss" labels.
The evaluation indicates that if the model correctly classifies a headline-body pair as "related" or "unrelated" then the score is incremented by 0.25. If the model correctly classifies a headline-body pair as "related" and the actual stance of the headline-body pair is one of "agree", "disagree" and "discuss", then the score is also incremented by 0.25. If the model correctly classifies a headline-body pair as one of "agree", "disagree" and "discuss", then the score is incremented by 0.75 instead of 0.25.
It is easier to classify whether a headline is related to a body text, so it gets less weight. And for classifying as "agree", "disagree" and "discuss", it is more difficult given the amount of the stances that contains those three stances compared to the amount of the "unrelated" stances.

5.1.2 Relative score

The relative score is calculated by

$$\frac{Score}{PerfectScore}$$

, where Perfect Score represents the score we can get when all the labels are correct.

5.2 Implementation

5.2.1 Code we use

We use the code provided by Fake News Challenge to read the provided datasets and to preprocess data including splitting train and validation datasets and also the code to calculate the score (Pomerleau and Rao, 2017).
We use the code to create the InferSent model and use it to generate sentence embeddings (Conneau et al., 2017).

5.2.2 Code we implement

We implement the Siamese model including a one-dimensional CNN model named "FakeNewsCNN" and the MLP classifier named "FakeNewsClassifier" in the notebook. We also implement the code for training our model, testing our model, and displaying the plots by ourselves.

5.3 Methods

After splitting the train dataset into train dataset and validation dataset, we run the baseline method provided by Fake News Challenge on the train dataset and calculate the score on the validation dataset and test dataset. Then, we use the same train dataset to train our model and use the same validation dataset to choose the model that has the best performance, which means that the model has been fully trained and is not overfitting. We will compare the performance of our model with the performance of the baseline method.

6 Results & Analysis

The following confusion matrices have actual labels on the left and predicted labels on the top.

6.1 Baseline method

From the Table 4 and 5, we can see that the relative scores on the validation dataset and test dataset are over 75%. This means that features extracted from the headlines and body texts are representative and during the process of extracting features, we barely lose information.

Table 4: Performance of the baseline method on the validation dataset

	agree	disagree	discuss	unrelated
agree	115	8	557	82
disagree	16	3	128	15
discuss	60	3	1530	207
unrelated	5	1	96	6796

Score: 3540.0 out of 4448.5 (79.57738563560751%)

Table 5: Performance of the baseline method on the test dataset

	agree	disagree	discuss	unrelated
agree	167	11	1439	286
disagree	36	7	416	238
discuss	228	14	3546	676
unrelated	9	5	364	17971

Score: 8748.75 out of 11651.25 (75.0885098165433%)

6.2 Deep Learning: Siamese neural network

6.2.1 Training process

Figure 2: Running loss on the train and validation dataset

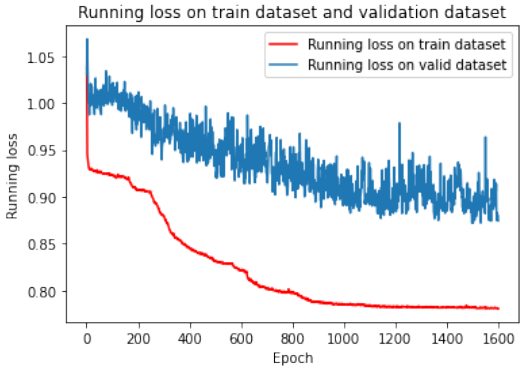
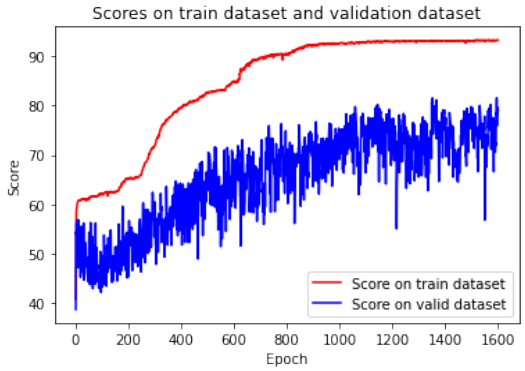


Figure 3: Scores on the train and validation dataset



From the two figures (Figure 2 and 3) indicating the running loss and scores on both the train dataset and validation dataset, we can see that the running loss decreases as the number of epochs increases, and the score (calculated as we indicated in the evaluation part) increases as the number of epochs increases. The two plots can help us select the best model that can have the best performance. Since although after 1000 epochs, the running loss and scores change a few, we still can see that the running loss has a downward trend and the scores have an upward trend. Thus, we decide to use the model which has been trained for 1600 epochs.

6.2.2 Training result

As the training running loss decreases, the relative score on the train dataset increase over 90% (shown in the Table 6), and the relative score on the validation dataset is around 79% (shown in the Table 7), which is close to the relative score on the validation dataset using the baseline method. We expect a similar or higher test score according to the performance on the validation dataset.

Table 6: Performance of the deep learning method on the train dataset

	agree	disagree	discuss	unrelated
agree	1770	99	736	311
disagree	125	380	104	69
discuss	19	0	6708	382
unrelated	6	0	36	29605

Score: 16530.0 out of 18114.75 (91.2516043555666%)

Table 7: Performance of the deep learning method on the validation dataset

	agree	disagree	discuss	unrelated
agree	325	20	261	156
disagree	33	55	44	30
discuss	116	12	1356	316
unrelated	60	16	203	6619

Score: 3512.25 out of 4448.5 (78.95357985837923%)

6.2.3 Test result

Table 8: Performance of the deep learning method on the test dataset

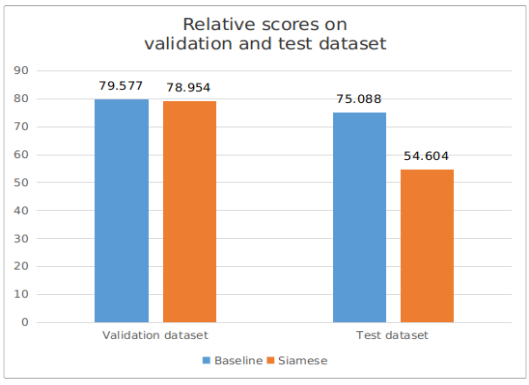
	agree	disagree	discuss	unrelated
agree	268	65	275	1277
disagree	94	14	110	479
discuss	303	38	1670	2453
unrelated	451	128	1087	16683

Score: 6362.0 out of 11651.25 (54.60358330651218%)

6.3 Comparison with baseline method

6.3.1 Analysis of relative scores

Figure 4: Relative scores on the validation and test dataset



From the Figure 4, the result of our model is much lower than the relative score on the validation dataset using the baseline method. After analyzing both the dataset and our method, We propose the

following reasons:
1. Unbalanced dataset
Due to the unbalanced dataset, it is hard to learn about the stances including “agree”, “disagree” and “discuss”. And these three stances have higher values.
2. Different text bodies
Since we use sentence embeddings, we use a headline or a body text as a whole as the input of the neural network model. However, the body texts used in the train dataset and the body texts used in the validation dataset are different. So it is reasonable that the relative scores on the test dataset are different since the baseline method uses word-level information as the features and our method uses sentences as a whole as the features.
3. Sentence embeddings
Using sentence embeddings can help us improve the efficiency of our model but at the same time using sentence embeddings means that we only consider the whole paragraph instead of the words in the headlines or body texts. However, in such a task that some words can be extremely important such as “deny” and “fake”, using sentence embeddings will make the features lose the information of those important words.

6.4 Analysis of score1 and score2

The definitions of score1 and score2 are indicated in the Evalution part.
As the result is not what we expect, we further analyze the score1 and score2 on the validation dataset and test dataset using the baseline method and the Siamese method.
From the Figure 5, we can see that on the validation dataset, both the number of correct “related/unrelated” labels and the number of correct “agree”, “disagree” and “discuss” labels using the Siamese method are similar to those using the baseline method, even the score2 using the Siamese method is higher than the score2 using the baseline method. This means that actually our model has a better performance to identify “agree”, “disagree” and “discuss”.
But from the Figure 6, we can see that on the test dataset, both score1 and score2 using the Siamese method are much lower than the score1 and score2 using the baseline method.
From the analysis about score1 and score2, we confirm that the different body texts in the train dataset and test dataset are the main reason that our model fails to have the same or better performance than the performance of the baseline method. And using sentence embeddings is another reason why we have worse performance on the test dataset. Both reasons result in such a result. Using sentence embeddings means that the neural network would focus on the sentences as a whole and would lose word-level information. After changing the dataset, the sentence-level information may not be suitable for the test dataset. Unlike our method, the baseline method uses word-level features, so the performance would not be affected too much due to the change of the dataset. And an unbalanced dataset would not affect too much in this kind of task.

Figure 5: Score1 and Score2 on the validation dataset

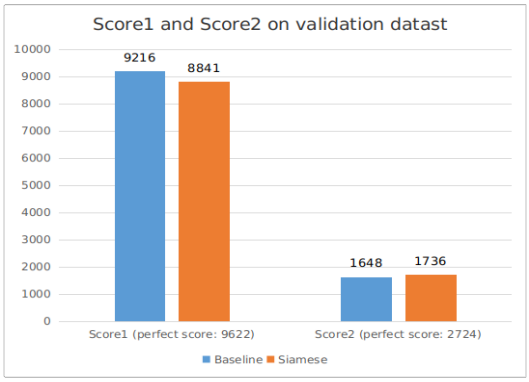
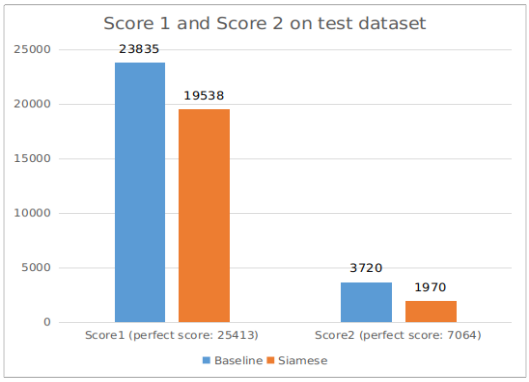


Figure 6: Score1 and Score2 on the test dataset



7 Conclusion

In natural language processing, although the semantics are important, the word-level features can also have huge impacts on the task we are going to achieve. For fake news stance detection, although it looks like to identify the relationship among sentences since we are dealing with the stances between headlines and body texts, the word-level information is still important since there are

words like “deny” and “fake” that can determine the stances between headlines and body texts with only one word. And the number of words in the headline that appears in the body text is another important feature since we can decide how close the headline and the body text are related. Instead of using word embeddings, using sentence embeddings can improve the efficiency of our model. However, the model improves efficiency at the expense of flexibility and accuracy. Thus, it is hard for our model to keep high accuracy after changing the body texts used as the input of our model.

Although our method has worse performance on the test dataset than the performance of using the baseline method, we think our method is still an available model for fake news stance detection since the performance on the validation dataset is still acceptable. We believe that using word embeddings will give us better results since we keep the word-level information and using word embeddings can resist the change of headlines and body texts.

8 Future work

Our next task will be using word embeddings to improve the performance of our model in fake news instance detection. Using word embeddings is a good idea to avoid losing important information of specific words, but the problem is how to build the available data loader as the input of CNN. We will find a way not only to improve the efficiency of our model but also to avoid losing word-level information.

9 Division of work

Yuqi Liu

- Found a sentence embeddings approach named InferSent to encode sentences into sentence embeddings.
- Designed the MLP model in the Siamese model for classifying the stances between headlines and body texts.
- Tuned hyperparameters so that the model can have a proper training process.
- Recorded the results of the baseline method and the Siamese model and write them in the report.
- Collected papers related to stance detection and the Siamese model.
- Analyzed the reasons why our model fails to get better performance.
- Wrote the remaining parts of the report with others.

Xinyue Ma

- Designed the CNN model in the Siamese model for extracting features from sentence embeddings.
- Build the baseline method in the notebook so that it can output the results of the baseline method.
- Tuned hyperparameters so that the model can have a proper training process.
- Record the results of the baseline method and the Siamese model and write them in the report.
- Wrote the introduction part in the report.
- Analyzed the reasons why our model fails to get better performance.
- Wrote the remaining parts of the report with others.

Yufei Wang

- Designed the CNN model in the Siamese model for extracting features from sentence embeddings.
- Tuned hyperparameters so that the model can have a proper training process.
- Tried self-attention mechanism to get sentence embeddings.
- Drew the figures in the report.
- Wrote the approach part in the report.
- Analyzed the reasons why our model fails to get better performance.
- Wrote the remaining parts of the report with others.

References

Sean Baird, Doug Sibley, and Yuxi Pan. 2017. Talos targets disinformation with fake news challenge victory.

Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a “siamese” time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1168.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018. A retrospective analysis of the fake news challenge stance detection task. *arXiv preprint arXiv:1806.05180*.

Dean Pomerleau and Delip Rao. 2017. the fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news.

Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*.