Fake News Challenge: Stance Detection using Memory Neural Network

Dan Wang
Department of ECE
University of Waterloo
d347wang@uwaterloo.ca

Yuxin Fan
Department of ECE
University of Waterloo
fyuxin@uwaterloo.com

Abstract

Stance detection has been an ongoing fight against fake news. The baseline on FakeNewsChallenge is 75.2%. Based on some previous work on a novel Memory Neural Network, we came up with models achieving score of 78.4%, 3.2% higher than the baseline score. With a dense representation Glove and sparse reprentation TF-IDF as the word representation, we either trained our model using CNN or LSTM or the uniform combination of them as the memory representation. Based on these experiments, we also found that TF-IDF might be an ideal word representation in stance detection tasks when combined with cosine similarity compared to Glove and that CNN might not be a competitive model in these tasks.

1 Introduction

The Internet has become the source of flooding false information including fake news, which has shown its negative effects on the public. People have reported how fake news have affected their daily life and also import decisions like voting. It's very urgent to resist fake news, yet due to the large amount of information, it's impossible to classify such information on the Internet manually. So there have been tons of attempts to complete the task with the power of machine learning. But this is a non-trivial task even for a specialist in linguistics. Therefore, people are currently focusing on classify the texts into 4 categories according to the relevance between the headlines and their news bodies: *unrelated, discuss, agree, disagree*. The first stage of Fake News Challenge (FNC-1) is a competition to encourage contestants work towards such goal.

The multi-classification task would produce 4 kinds of stances with the input as the headlines and bodies of the news. If the body is just non-sense which is not related to the claim (the headline) at all, then it would be classified as *unrelated*; if the claim is aligned to the content of the body, then it's labeled as *agree*; if the claim and the content of the body do not aligned with each other, it would be labeled as *disagree*; the other case would be *discuss*.

As a famous saying, "Garbage in, garbage out", we delicately construct the input of our model. For the word representations, we applied two kinds of word reperesentation: the sparse representation, namely TF-IDF; and the dense word representation, namely the Glove word embedding. These two word reperesentations feed the model either individually or corporately. Our research will show whether they corporate to give better accuracy performance or one of them outperforms the other.

CNNs and LSTMs are the main components in our model whicha are individually limited in their modeling capabilities, and we believe that stance detection performance can be improved by combining these networks in a unified framework. That is to say, instead of training CNN and LSTM individually and combining the three outputs, we combined CNN and LSTM into one unified framework. The unified combination seemed delicate and complex, therefore, we also tried to omit one of the components (e.g. the TF-IDF input layer, the CNN layer or the LSTM layer) as ablation studies.

Our model operates at paragraph level. In addition to the prediction of the stances of the given claim and text pair, inference components in our model also extract snippets from the text body as evidences of the predictions.

The most recent and the highest score on Fake News Challenge is 82%, which is a weighed accuracy, while the third-place winner achieved a score of 80%. In our research and experiments, we achieved 80% of score, and 90.2% of accuracy with Momory Network as the main infrastructure of our model. Note that we did not participate the challenge officially.

In addition to the improvement on stance detection, we are surprised to learn that cosine similarity computed from TF-IDF can be a better word representation on stance detection. Yet this fact is barely reported in other studies. One of a participant of the Fake News Challenge also reported that she found TF-IDF is the main source of performance improvement in the GBRT model[**Chan Ge**].

2 Related Work

2.1 TF-IDF & Cosine Similarity in Stance Detection

[**Matt J. et al**] reported that TF-IDF as one of the two most common ways document representations are often not suitable for document distances due to their frequent near-orthogonality. Another significant drawback of the representation is that it does not capture the distance between individual words.

Compared to dense word representations such as word2vec, many works have reported that TF-IDF is less robust than LSI and LDA, and only has better statistical quality when word term frequency is less than 0.3. ([**A Comparative study of TF-IDF**, **LSI and LDA**]) While in the paper proposed by [**Benjamin Riedel et al.**], TF-IDF and cosine similarity used as the main input features, achieve a very high performance of 81.72 %, the third-place of the competition while the baseline at that time achieved score of 75.2%. Is TF-IDF a strong feature for stance detection or not? We would show the results of our experiment to provide some observations.

2.2 CNN and LSTM in Stance Detection

In the paper proposed by [**Sahil Chopra et al.**], one of the teams particiting the Fake News Challenge in 2017, they applied an SVM trained on TF-IDF cosine similarity to descern whether a headline-article pairing is *related* or *unrelated*. Then they performed a couple of neural networks architecture built on top of Long-Short-Term-Memory Models (LSTMs) to label the pairing as *agree, disagree* o r*discuss*. Ultimately, their best performing neural network architecture scored 0.8658 according to the FNC-1's performance metric.

[**Yi-Chin Chen et al.**] proposed Convolutional Neural Networks for stance detection and rumor verification which was a solution for SemEval-2017 Task 8. They beated the baseline classifier on different event data with good F1 scores. The best of their submitted runs achieved rank first among all scores on subtask B.

In addition, works with the combination of CNN and LSTM are constructed, too. [**Chinnappa Guggilla et al.**] described a supervised model based on Convolutional Neural Networks and Long-Short-Term-Memory Networks with different types of distributional word embeddings but didn't incorporate any of them. They claimed to have achieved a significant improvement over the state of the art for a binary-class dataset and comparable performance to the state of the art on the other multi-class dataset.

2.3 Memory Network in Stance Detection

[**Sukhbaatar et al.**] presents an end-to-end version of memory networks incorporating convolutional and recurrent neural networks as well as similarity matrices such that the model doesn't train on the intermediate 'supporting facts' strong supervision of which input sentences are the best memory accesses, making it much more realistic. They also have multiple hops (computational steps) per output symbol. This paper is a useful extension of memNN because it removes the strong, unrealistic supervision requirement and still performs pretty competitively. However, it comes up with various model decisions and choices without a clear motivation and without empirical ablation studies. They also didn't apparently use a development set for most of these decisions. In addition, the gap might be even smaller or negative if the current model did not so many extra parameters to tune over, as compared to LSTMs.

According to the above work with CNN and LSTM in text classification and stance detection tasks, those memory networks looked very promising. Therefore, we aim to dig deeper to these models.

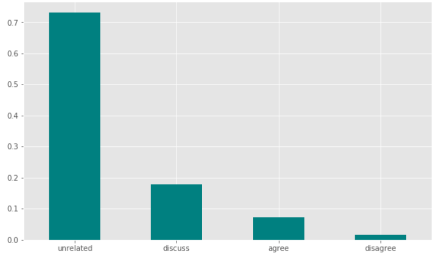

Figure 1: Imbalanced dataset: the ratio of 4 stances

3 Data Preprocess and Feature Engineering

3.1 Data Exploring and Adjusting Techniques

The data set was published in 2017 as part of the first Fake News Challenge. The data is split into different files containing article bodies and headlines individually. A body-headline pair does not necessarily come from the same article. Therefore, out of totally 49972 training instances, there are only 1683 unique body texts and 1648 unique headlines. The bodyheadline pairs are mapped to one of four labels: *unrelated, discuss, agree*, or *disagree*.

Due to the way the dataset was generated, The *unrelated* class was inflated its share of the data set to 73%. The second largest class, *discuss*, is also significantly larger than the remaining classes. With almost 18% of training examples, the *discuss* class is nearly two times larger than the *agree* class with 7%, and eight times greater than the *disagree* class, which gathers less than 2% of the training data.

The the dataset is significantly unbalanced, shown in Figure 3.1.

Thus, one of the most demanding aspects of this task is dealing with the highly imbalanced class distribution.

We tried several simple random-sampling strategies to mitigate the class imbalance, such as undersampling the majority classes, oversampling the minority classes in the mean time, and combining both of the sampling production. The best strategy so far is that all but the majority class (i.e. the three related classes) were oversampled randomly with replacement until their size equaled the size of the majority class. While removing the excess instances from the majority classes should be an optimal way of improving the visibility of the minority class, under-sampling on the majority class turns out to hurt the performance of our models.

Other information about the dataset:

- The median number of paragraphs per article is 9;
- The median length of paragraph is 27 words;
- The median length of headlines is 10 words.

3.2 Feature Engineering

This part mainly described how data are preprocessed and what input are fed into the models.

For the preprocessing part, we removed the accents, non-ASCII characters and other special characters which are noises. Also, punctuations, URLs, names which provide no useful information the detection task were removed. Then the article bodies as well as the headlines were tokenized and compiled using a pretrained Glove model downloaded from the internet. For those vocabularies having no tokens in the model, they were mapped to ¡UNK¿.

Articles are trimmed with only the first 9 paragraphs, as is said in the previous subsection, the median number of paragraphs per article is 9. Short articles with paragraph number less than 9 is padded with zeros until they have the same size as the others'. Then each paragraph and each headline are trimmed by preserving only the first 15 words, with those shorter than 15 words padded with zeros. This seem reasonable since the median paragraph length is 27 words while the median length of headlines is 10 words.

In addition to representation with Glove model, we also computed the TF-IDF of the corpus as a sparse representation of the infomation. Cosine similarity computed based on TF-IDF to reflect the similarity between each paragraphs and each headlines, as well as cosine similarity computed based on the two memory representation, CNN and LSTM would be computed as important input for intermediate layers of our models.

4 Model Architecture

Memory network is advanced in the context of learning efficient text representations, document classification, and natural language inference. On the basis of string similarity, it achieves state-of-art according to stance detection problem [**Sukhbaatar**]. The original paper proposes a novel model with memory networks for stance detection, which combines the similarity matrix with inference component to extract textual snippets that are relevant to the input claims.

A memory network is consist of 6 layers: $\{M, I, F, G, O, R\}$, where M memory layer that is a sequence of objects or representation, I is input layer that is used to map input to its representation, F is the inference layer that is used to take an input document d as evidence and a textual statement s as a claim and convert them into their corresponding representations in the input I. F identifies the relevant parts of inputs. After that, G is generalization layer that is used to update the memory and O represents output layer that generates an output from the updated memory, and converts it to a desired response format with R response layer.

4.1 Layers

The implementation and details for each layer are showing as the following points:

Input: maps the input to its representation in the memory, using word embeddings and a sparse TF-IDF representation. We implement this step in previous section.

Dense Representation: the dense word vectors were obtained from the GloVe corpus pre-trained on Twitter data. We map each word in the input sequence to its 100-dimensional GloVe vector, and set the max paragraph length to 15 and the number of paragraph to 9.

```
   dense body (n_samples,
n_paragraphs=9,
max_paragraph_len=15,
embedding_dim=100) dense claim
(n_samples, max_claim_len=15,
embedding_dim=100)
```

Sparse Representation: term frequency-inverse document frequency.

```
   sparse body (n_samples,
n_paragraphs=9, vocab_size) sparse
claim (n_samples, vocab_size)
```

Memory: contains the representations of the corpus, based on representations learned by a CNN and a LSTM. Figure 2-7 show how memory component works.

Inference: identifies the relevant parts of inputs and updates the memory. The structure of inference component is shown in Figure 8.

Output: produces output for each new input

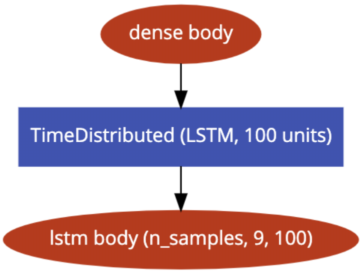
dense body
TimeDistributed (LSTM, 100 units)
lstm body (n_samples, 9, 100)

Figure 2: Passing body to LSTM

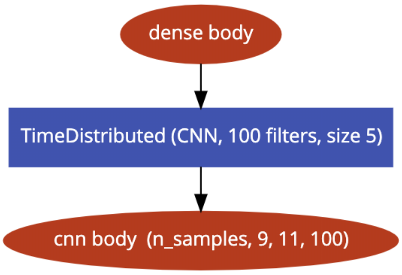
dense body
TimeDistributed (CNN, 100 filters, size 5)
cnn body (n_samples, 9, 11, 100)

Figure 3: Passing body to CNN

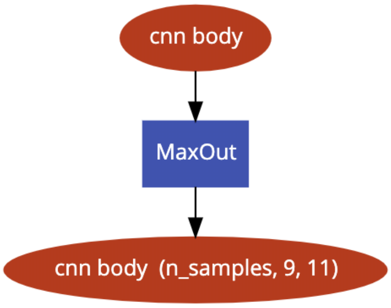

Figure 4: Passing cnn body to MaxOut

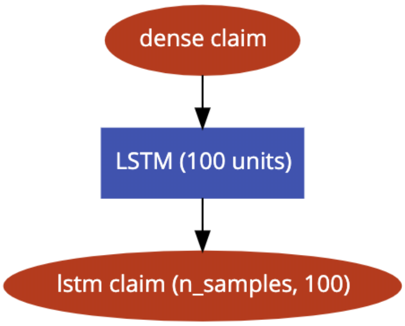

Figure 5: Passing claim to LSTM

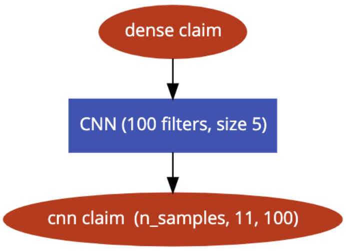

Figure 6: Passing claim to CNN

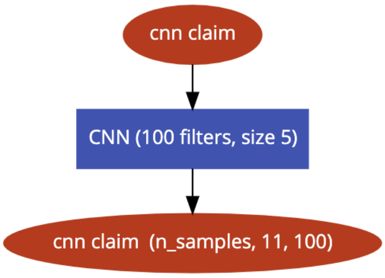

Figure 7: Passing cnn claim to CNN

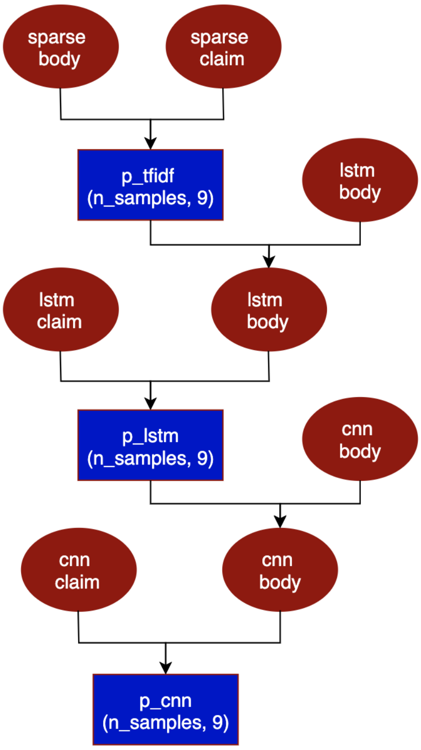

Figure 8: The structure of inference component

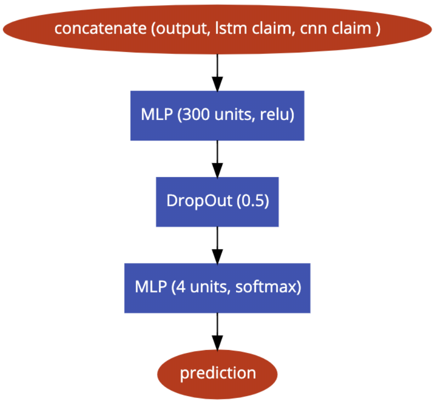

Figure 9: The structure of response component

and current memory state. Concatenates the intermediary states produced by generation layer.

```
   concatenate [ mean(cnn
body), max(p_cnn), mean(p_cnn),
max(p_lstm), mean(p_lstm),
max(p_tfidf), mean(p_tfidf) ]
```

Response: converts the output into a desired response format. Consists of a dense network that predicts the final stance. The structure of inference component is shown in Figure 9.

4.2 Motivation and Ablation Studies

Each CNN and LSTM block captures information about the input representation at different scales [10]. Therefore, we explore if further improvements can be obtained by combining information at multiple scales.

We trained four models, ranked by their structure complexity :

- Model 1: Glove + TF-IDF + CNN + LSTM
- Model 2: Glove + CNN + LSTM
- Model 3: Glove + TF-IDF + LSTM
- Model 4: Glove + CNN

We investigate the model performance without passing the output of the CNN layer into both LSTM layers (Model 3).

Our experiments conducted to understand the impact of TF-IDF in stance detection task: Model 1 VS Model 2. We find that joining TF-IDF provides percent of relative accuracy improvement compared to Model 1, the most complex model.

And excluding input of CNN features gives an additional percent of relative improvement.

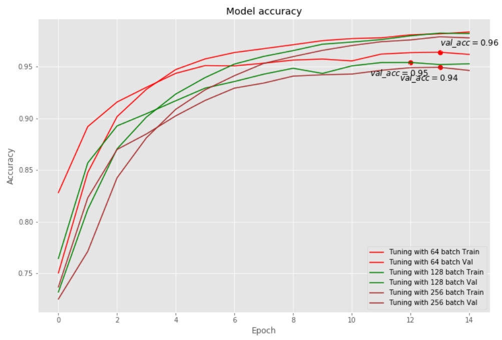

Figure 10: Tuning model with 3 batch size

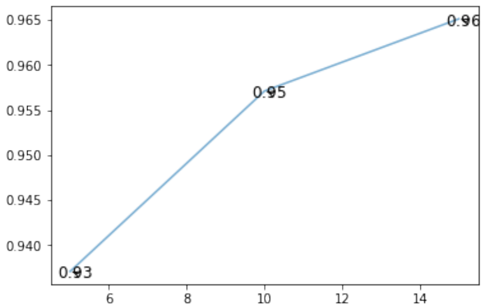

Figure 11: Tuning model with 3 epoch

5 Experiments

We make experiments on Model 2 and tune it with two hyper-parameters: epoch and batch size. We apply three different epoch values into this model: 64, 128, 256; and three different batch size: 5, 10, 15. For time-distributed LSTM layer has 100 units and tanh activation and for time-distributed CNN layer with 100 feature maps of width 5 and ReLU activation. In our experiment, we evaluate this model based on the accuracy.

Tuning batch size: the batch size is a hyper-parameter that defines the number of samples to work through before updating the internal model parameters. The tuning result is shown in Figure 10. The three model achieves similar result around 95%, which means changing batch size would not improve accuracy on validation. Therefore, we chose 128 as batch size in the final model.

Tuning epoch: the number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset and the accuracy in three models is shown in Figure 11.

Model	Accuracy
CNN+Glove	69.1%
CNN+LSTM+Glove	76.7%
LSTM+Glove+TF-IDF	**90.5%**
Memory Neural Network	83.23%

Table 1: Accuracy of the 4 models

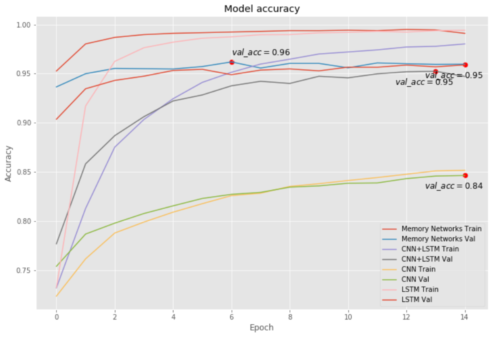

Figure 12: Accuracy of the 4 models

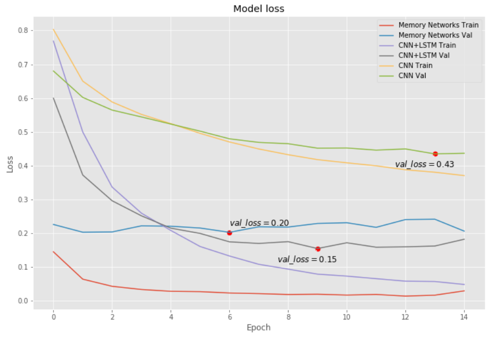

Figure 13: Loss of the 4 models

6 Results

6.1 Accuracy & Loss

We evaluated our models on competition dataset of the Fake News Challenge. We utilize accuracy as one of evaluation methods, which is the number of correctly classified examples divided by the total number of examples. The accuracy of the four models is shown in Figure 12 while the loss of the four is shown as Figure 13. In addition, we provide the accuracy table of the stated four models in Table 1.

6.2 Weighted scores score report according to the matrice of Fake News Challenge

Weighted scores is a official evaluation method provided by fnc-1. This is a weighted, two-level

Model	Weighted Accuracy
CNN+Glove	39.9%
CNN+LSTM+Glove	%
LSTM+Glove+TF-IDF	**78.4%**
Memory Neural Network	74.0%

Table 2: Weighted score or score achieved under FNC matrices

Model	F1 score
CNN+Glove	25.7%
CNN+LSTM+Glove	%
LSTM+Glove+TF-IDF	**51.05%**
Memory Neural Network	50.1%

Table 3: F1 score of the 4 models

scoring scheme, which is applied to each test example. If the example is from the unrelated class and the model correctly predicts it, the score is incremented by 0.25; otherwise, if the example is related and the model predicts agree, disagree, or discuss, the score is incremented by 0.25. Then the score is normalized by dividing it by the total number of test examples. The result of weighted score is shown in Table 2.

6.3 F1-score

Among all above metrics, we found that our simplified model with the main components as TF-IDF+Glove+LSTM achieved the highest performance. Compared to the most complex model, which is added with CNN layers, the simplified Model 3 is 4.4% higher in weighted accuracy (i.e. the score achieved according the Fake News Challenge metrics), nearly 1% higher in F1 score, and about 7 % higher in accuracy. The same trend is found in the comparison between CNN+Glove and CNN+Glove+LSTM. The model with LSTM outperforms than the one without LSTM in almost every stated aspect. To this point, we can claim that CNN is not better than LSTM when dealing with this task. On the contrast, CNN appeared to hurt the model performance.

In addition, we can find some other interesting facts. To the best of our knowledge, TF-IDF is claimed to be uncompetitive compared to word embeddings, stated as in section *related work*. However, we could observe from the performance report of CNN+LSTM+Glove versus CNN+LSTM+Glove+TF-IDF that without the assistance of TF-IDF, the model performance on stance detection drop dowm dramatically. Due to the limit of time, we didn't do experiments on omitting Glove as the word representation, leaving only TF-IDF (and cosine similarity) as the main input of the models.

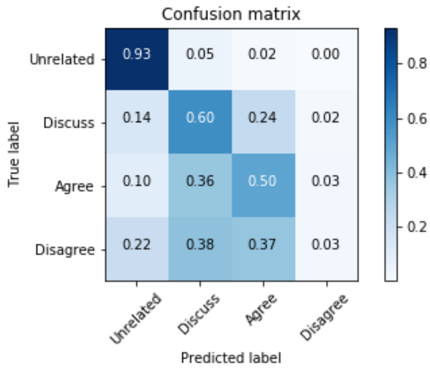

Figure 14: Confusion matrix of Model 1: Glove + TF-IDF + CNN + LSTM

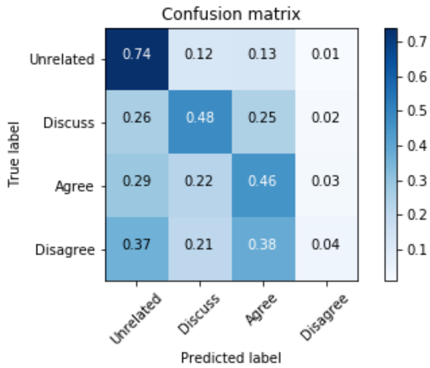

Figure 15: Confusion matrix of Model 2: Glove + CNN + LSTM

6.4 Confusion matrix

The confusion matrix of prediction of the four models is shown in 14,15,16 and 17.

We found that our simplified model, Model 3, i.e. Glove+TF-IDF+LSTM, had improved the accuracy of three out of four classes, i.e. *unrelated, discuss, agree*. Only the *disagree* remained low. It's a pity to find that all these 4 models seemed insensitive to the *disagree* labeled headline-body pairs.

7 Conclusion

We performed our model training on the Fake News Challenge datasets using 4 different memory networks and the best one outperformed the baseline model on FakeNewsChallenge.org by

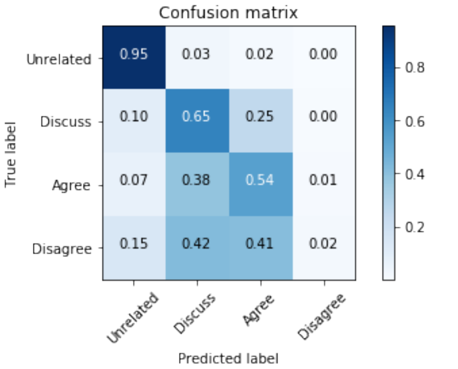

Figure 16: Confusion matrix of Model 3: Glove + TF-IDF + LSTM

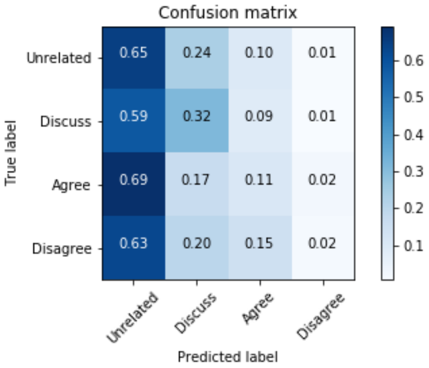

Figure 17: Confusion matrix of Model 4: Glove + CNN

3.2% of weighted acurray (i.e. score). As is shown in the results, the CNN+Glove model performed the worst, with only 69.1% of accuracy, while the 3rd model adapted from the most complex model with CNN eliminated performed the best with 90.5% of accuracy. It might indicate that CNN is incompetitive in this task. The CNN+LSTM+Glove model achieved 76.7%, and is 10.4% less than the first model. Considering that in the meantime,the LSTM+Glove+TF-IDF model which features TF-IDF hit the best score, it might indicate that the more general word representation, TF-IDF might be more helpful in this task.

As we know that word embedding representation contains the similarity information of words while TF-IDF doesn't, however, similarity matrix computed from TF-IDF might cancel out the diadvantage in a better way.

Further work will be finding out why TF-IDF sparked in alike scenes on other datasets.

Appendices

We train models on the Google Colab `https://drive.google.com/open?id=16_1ODR06wwAspvcus7s3cKR2WMXVvR84`. The source files is located in the main folder and the data files is in sub-folder. According to submission files, we place them in the sub-folder as well.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

Matt J. Kusner Yu Sun Nicholas I. Kolkin Kilian Q. Weinberger. 2015. From Word Embeddings To Document Distances. 2013. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1393–1398.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In Proceedings of NIPS, Montreal, Canada, pages 24402448.

Chan Ge. 2019. A Feature-enhanced Gradient Boosting Classifier-based Stance Detector for Fake News Challenge Task. Not publised.

Wen Zhang, Taketoshi Yoshida, Xijin Tang 2011. A Comparative study of TF-IDF, LSI and LDA *Expert Systems with Applications: An International Journal archive Volume 38 Issue 3, March, 2011 Pages 2758-2765.* Pergamon Press, Inc. Tarrytown, NY, USA

Benjamin Riedel and Isabelle Augenstein and Georgios P. Spithourakis and Sebastian Riedel 2017. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task *CoRR, volume abs/1707.03264.*

Sahil Chopra, Saachi Jain, John Merriman Sholar 2017. Towards Automatic Identification of Fake News: Headline-Article Stance Detection with LSTM Attention Models *Fake News Challenge 2017*

Yi-Chin Chen, Zhao-Yang Liu, Hung-Yu Kao 2017. IKM at SemEval-2017 Task 8: Convolutional Neural Networks for stance detection and rumor verification

Chinnappa Guggilla, Tristan Miller, Iryna Gurevych 2016. CNN- and LSTM-based Claim Classification in Online User Comments *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, page 2740-2751* Osaka, Japan