Fake News Detection

Han Zhang,
Shanghai Jiao Tong University
zhanghan07@sjtu.edu.cn,

Renyuan Lu,
University of Liverpool
sgrlu2@liverpool.ac.uk,

Liuyi Chen,
Huazhong University of Science and Technology
u202113863@hust.edu.cn,

Xintong Zhong,
Shanghai Jiao Tong University
janice-zhong@sjtu.edu.cn.com

Abstract

In this project, with the incentive to construct a language classification model that could differentiate between fake news and real news, we test the different model structures as well as different embedding method to find the best model. The model with best accuracy is BERT(Devlin et al. 2018) structure with BERT tokenizer as embedding layer while our most efficient model that has both high accuracy and uses less parameters is Transformer encoder(Vaswani et al. 2017) model with TF-IDF mebedding method.

Introduction

Fake news detection is one major concern in post-pandemic era. As the technological barrier is way lower than before, everyone can publish we-media contents. Detecting fake news is therefore much more important currently.

This gives rise to our incentives to construct and train the model to automatically detect the fake news.

Related Works

Combining news with facts

In (Vijjali et al. 2020), the authors divide the task into two stages. For the first stage, they train model A to match the news body with the explanations of facts. For the second stage, they switch to another model B, the input of which are the news body as well as the facts. Then model B will judge whether the piece of news is fake or real according to the facts. As what the authors propose is a general architecture, they have tested different kinds of models including neural networks embedding methods and traditional embedding methods for model A and model B. Finally, they claim that using BERT(Devlin et al. 2018) for model A and ALBERT(Lan et al. 2019) for model B can give the best performance.

However, the method generally assumes that all of the news for detection would have corresponding explanations for further analysis, which actually is not practical for the online fake news among we-media. Authors of fake news would not add fact afterwords and some common facts would not appear in any piece of news. Therefore, we want to construct our model without the explanation or fact to the specific news.

An overview of all existing models

In (Gundapu and Mamidi 2021), the authors look back on fake news detection models in three categories machine learning models, deep learning models and transformer-based models, after which they draw a conclusion that the model with the best precision and accuracy rate is an ensemble model they constructed on the basis of three separate transformer-based models.

Building on their conclusion, we generated an idea of integrating different models and means of embedding to compare the outcomes in pursuit of the quickest and the most accurate one. Hence, we cross-matched two sets of models and embedding in the research.

Detecting Fake COVID-19 News with NLP

In (Nistor and Zadobrischi 2022), the authors proposes the use of advanced machine learning methods and natural language processing to identify and combat the prevalence of fake news on social media during the COVID-19 pandemic. The study achieves a success rate of over 90% in detecting fake news on Facebook and proposes the development of a browser extension to limit users' interactions with fake material. However, the authors also acknowledge the need for ongoing research and development efforts to address the constantly evolving nature of fake news.

Overall, the study highlights the importance of a multifaceted approach to combating fake news on social media that includes advanced NLP techniques, machine learning algorithms, and ongoing research and development. The findings and proposed solutions of this study have broader implications for the detection and prevention of fake news across a wide range of topics and contexts.

Background and Methods

Similar to the general classification and entailment tasks in language models, fake news detection task would give the model with a sentence and a ground truth label of whether this piece of news is fake or real when training. No matter it is traditional machine learning model or neural network model, the first step to deal with the text input is to use embedding layers, which could transform the words and sentence into numerical values for the model to learn.

Note that for our dataset, the input will only contain the content without other metadata such as the authors or the fact related to it.

Embedding

In this section, we will introduce some embedding techniques that we use, including GloVe (Brochier, Guille, and Velcin 2019), TF-IDF and BERT tokenizer.

GloVe GloVe (Global Vectors for Word Representation) is an unsupervised learning algorithm for generating word embeddings, which are vector representations of words in a high-dimensional space. These embeddings can capture semantic and syntactic relationships between words and combine the advantages of Word2Vec and count based method.

The basic idea behind GloVe is to learn a low-dimensional vector representation for each word in a corpus, such that the dot product of the word vectors is proportional to the log of the number of times the corresponding words co-occur in a given context. Specifically, the GloVe algorithm aims to minimize the following cost function:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - log(X_{ij}))^2 \quad (1)$$

where V is the size of the vocabulary, X_{ij} is the number of times word i appears in the context of word j, and f(.) is a weighting function that down-weights the contribution of very frequent co-occurrences. The parameters w_i and b_i represent the vector and bias of word i, respectively.

In order to optimize this cost function, GloVe uses stochastic gradient descent (SGD) to update the word vectors and biases in an iterative manner. The resulting word embeddings can be used for various natural language processing tasks, such as text classification, sentiment analysis, and machine translation.

TF-IDF TF-IDF (Term Frequency - Inverse Document Frequency) is a heuristic embedding method utilizing the inductive bias of language that the more frequent a word appeared in a document, the more it can represent the document. Also, it takes the information entropy into consideration as the more frequency a word appeared overall, the more likely that it does not have a concrete meaning, therefore should have a lower capability to represent any documents. The equation to define the TF-IDF is as follows:

$$tfidf(d,t) = tf(d,t) * idf(t) \quad (2)$$

$$tf(d,t) = \frac{\#\text{ of } t}{\text{total number of words in } d} \quad (3)$$

$$idf(t) = log(\frac{\text{total number of documents}}{\#\text{ of } d \text{ having } t}) \quad (4)$$

where d is current document and t is current word. For a larger TF-IDF score, the word should often appear in the current document but seldom appear in other document.

Note that TF-IDF would give a sparse vector for each sentence (document), where the dimension of vector is the number of words in the whole corpus.

BERT Tokenizer The BERT tokenizer is the embedding layer of BERT, which is pretrained in a large text dataset. This tokenizer essentially is a dictionary or a matrix that maps each unique words to a vector of specific dimension, which is 512 for the base BERT model.

Experiment Settings

Data Formatting

Our dataset mainly comes from the work of (Patwa et al. 2020). The training set has 6420 entries of (content, label) pair while the testing set has 2140 entries of data pairs. The example of (content, label) pair is listed below:

Content: The CDC currently reports 99031 deaths. In general the discrepancies in death counts between different sources are small and explicable. The death toll stands at roughly 100000 people today.
Lable: Real

The distribution of length of each sentence in the training and testing set is as follows:

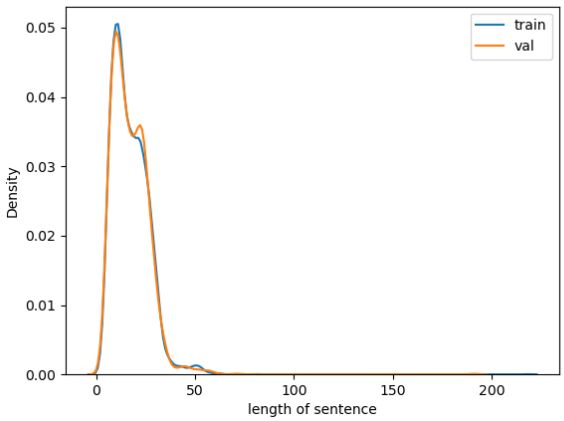

Figure 1: Distribution of sentence length

According to 1, most of the samples have the length smaller than 200 (after data preprocessing), so we will use 200 as maximum sequence length in the experiment if needed.

Possible feature engineering of the dataset includes: extraction of the author, separation of numbers with the major part. Here we do not do such feature engineering mainly because two reasons:

- Not all entries of news have its author listed in the content.
- Although there could be number in word format and number in Arabic numeral format, they do not affect the final result much.
- We want to make a general model which does not have certain requirements on the author or other metadata about the content.

Data Preprocessing

Still, we need to do some data cleaning on the original data to extract the useful information. However, this process does not rely on any expert knowledge of the data. Our preprocessing method includes:

- Subtract the website information from the content.
- Extract the alphabetic and numerical content and subtract other symbols such as ”&”, comma and period.
- Use lower case for each word.
- Split each word and use a list to represent the original data (for certain embedding methods)

We drop the website information as we believe that it does not contribute to the credibility of the news, instead, the website information only functions as a name, which is used to distinguish.

We also extract the alphabetic and numerical as other symbol is designed for human to read but would not give significant impact on the meaning of the sentence. Same as for the lower case.

Baseline

As part of the prepration of the experiment, we first reproduce the result of (Patwa et al. 2020). We use TF-IDF embedding and traditional machine learning model called SVM (support Vector Machine) as our baseline model. The model performance is quite satisfactory: The model accuracy is

	Fake (label)	Real (label)
Fake (prediction)	963	57
Real (prediction)	83	1037

Table 1: Confusion matrix of SVM using TF-IDF

93.45% in testing set.

Results and Discussions

As for our experiment part, we test different models including neural network based linear model and Transformer based Transformer encoder model. We also test different embedding layers motioned in the Embedding section, including GloVe, TF-IDF and BERT tokenizer.

GloVe+Transformer

As Transformer models currently outperform other neural network models in the language tasks, we would like to use Transformer as our backbone and add a linear classification head as well as a softmax function before outputting the result. However, as TF-IDF gives a vector of 16k dimensions (total number of unique words in the corpus), which would be too large for the Transformer model dimensions. If we split the input into series, then it may potentially undermine the structure and give the model another kind of illusion. So here we use GloVe embedding and transform each word into a vector of dimension 50. For the efficiency, we constrict the length of the sentence to be 200 and cut the rest. The result is: The hyperparameters are listed under the Fig.3, we can

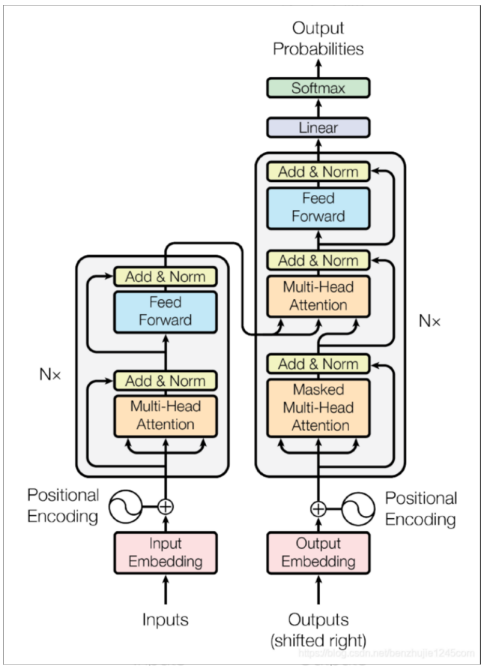

Figure 2: Transformer structure (Vaswani et al. 2017)

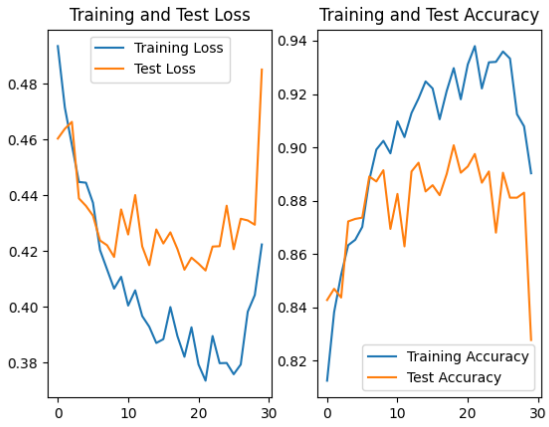

Figure 3: GloVe+Transformer. We use 6 Transformer encoder layers with a linear classification head in the end. Transformer sequence length is 200, dimension is 50, dropout rate is 0. Learning rate is 1e-5.

find that the model only gives an accuracy of 0.9014 in the testing set, which is even worse than the traditional machine learning method using TF-IDF embedding. On top of that, the accuracy goes down sharply in the end. We conjecture that the learning rate is not the optimal and the model is not robust.

TF-IDF+Linear

Then we infer that the TF-IDF embedding may potentially be better than the GloVe embedding in this problem. As a

reference, we construct a simple two-layer linear neural network model and use TF-IDF embedding.

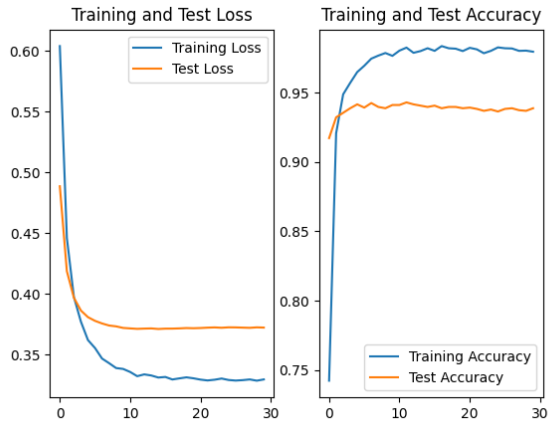

Figure 4: TF-IDF+Linear. We use two layers of linear projections, the first one project the input of 16k dimensions to 10 hidden dimensions and the second one project the 10 hidden dimensions to 2-dimension output. We add a ReLU as activation function between two layers. The dropout rate is 0 and the learning rate is 1e-5.

The model achieves an accuracy of 93.29% and is much more robust. Despite that we can switch to other activation fucntions such as Tanh, add dropout rate and adjust the hidden dimensions to increase the accuracy further, we do not perform hyperparameter optimization is this part as the current model already proves that the TF-IDF embedding is more effective than the GloVe in this problem.

TF-IDF+Transformer

Inspired by the previous experiment, we would like to combine the TF-IDF embedding with Transformer encoder blocks to achieve a better performance. Still, we need another linear projection layer before the encoder layers to project the input of 16k dimension to an appropriate dimension that the encoder could deal with. Here we project the input to dimension of $l * d$, where l is the sequence length of the encoders and d is the model dimensions.

The confusion matrix is: Here we perform hyperparame-

	Fake (label)	Real (label)
Fake (prediction)	1039	39
Real (prediction)	81	981

Table 2: Confusion matrix of Transformer encoder using TF-IDF embedding

ter optimization by grid search of learning rate, model dimensions and dropout rate. The hyperprarameter of current best model is listed in the caption of Fig.5. The model achieves an accuracy of 94.39% in the testing set, which is about 1% higher than the baseline model.

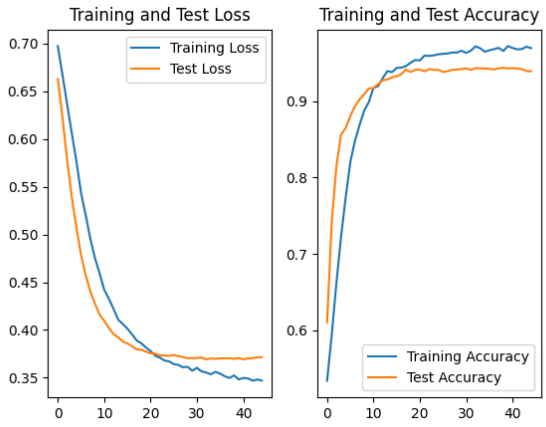

Figure 5: TF-IDF+Transformer. Same as before, we use 6 Transformer encoder layers with classification head and additionally a linear projection layer in the beginning. The sequence length is 32. The model dimension is 256. The dropout rate for encoder is 0.5 and the dropout rate for linear projection is 0.25.The learning rate is 2e-6.

However, the model is relatively large compared to the baseline model, it has total parameters of 67.62M and needs 3.11s to run on the whole testing set.

For the practical purpose, we propose another smaller model, which is still based on Transformer block and TF-IDF embedding but use less parameters by adjusting the hyperparameters.

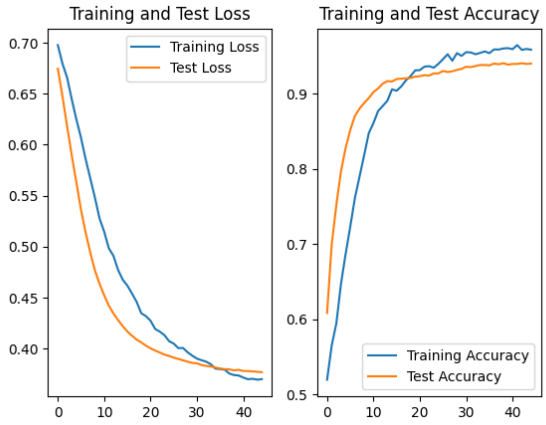

Figure 6: TF-IDF+Transformer. Compared with former model, the sequence length is 16. The model dimension is 128. The dropout rate for encoder is 0.5 and the dropout rate for linear projection is 0.25.The learning rate is 4e-6.

The confusion matrix is: The model accuracy is 94.21% in testing set. As we can conclude, this smaller model does not decrease in accuracy a lot but uses significantly less pa-

	Fake (label)	Real (label)
Fake (prediction)	1064	68
Real (prediction)	56	952

Table 3: Confusion matrix of Transformer encoder using TF-IDF embedding with less parameters

rameters. It only uses 16.91M parameters and needs 2.54s to run on the whole testing set.

BERT

Besides the GloVe and TF-IDF embedding, we could also use BERT tokenizer as our embedding layer. For simplicity, we will directly use the BERT structure and load pretrained model. To get the prediction of the classification result, we add a classification head after the BERT structure, which maps the 768 dimensions output of the first token to 2-dimension prediction result.

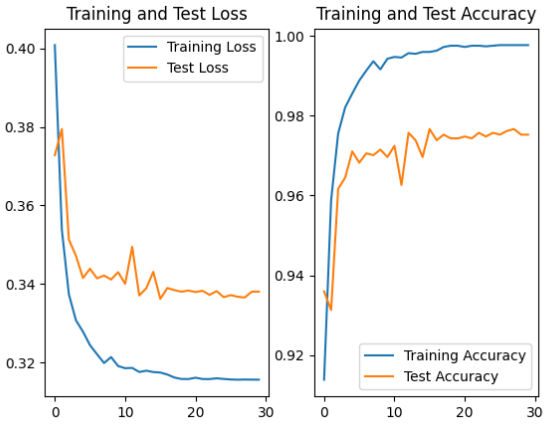

Figure 7: BERT tokenizer+BERT. Base BERT model with pretrained parameters. The dropout rate is 0.2. The learning rate is 1e-5.

The confusion matrix is: The model achieves an accuracy

	Fake (label)	Real (label)
Fake (prediction)	1104	34
Real (prediction)	16	986

Table 4: Confusion matrix of BERT model with BERT tokenizer embedding

of 97.66% in the testing set, which is a remarkable jump. However, we also notice that the model uses 109.53M parameters and needs 15.27s to run in the whole testing set.

Collection of models

According to the Table 5, although the BERT structure with BERT tokenizer embedding method gives the best accuracy, it uses significantly more parameters than the other model.

	Test accuracy	Parameters
TF-IDF+SVM (Baseline)	93.45%	-
GloVe+Transformer	90.14%	-
TF-IDF+Linear	93.29%	-
TF-IDF+Transformer, base	94.39%	67.62M
TF-IDF+Transformer, small	94.21%	16.91M
BERT tokenizer+BERT	97.66%	109.53M

Table 5: Collection of all models

If we take both the performance and the model complexity into consideration, the Transformer small model with TF-IDF embedding gives the most practical and effective result.

Conclusion

To conclude, our incentive is to construct a neural network model that could accurately differentiate between the fake news and the real news. We mainly test different embedding methods and different network structures for the better performance. We also balance between the model performance and the model complexity to propose the most efficient model which uses Transformer encoders as network backbones and TF-IDF as embedding method.

Admittedly, our work is still limited in the dataset and model structures. We only use a training set of 6420 entries and testing set of 2140 entries, which is considerably small compared to other language models. On top of that, we do not perform cross validation in the dataset, which may potentially lead to overfitting and bias. As for the model structures, the combinations of different embedding methods and model structure we test are limited. And more language models could be further experimented on.

Contributions

- **Han Zhang**: Responsible for the experiment part. Construct the Transformer model, BERT model and test the different embedding layers for the better performance. Analyze the model performance as well as the loss curve to come up with the better model structure and adjust the hyperparameters for the training.
- **Renyuan Lu**: Had some feasibility discussions before the project started. Participated in the conduct of the experiments. Completed the collection of relevant materials.
- **Liuyi Chen**: Involved in the experimenting process in terms of comparing different model performance. Responsible for crafting the prototype of the group's research presentation through extensive reading beforehand.
- **Xintong Zhong**:Participated in the preliminary discussions, found some relevant test data, and organized the presentation content.

Acknowledgments

Special thanks to Prof.Jin, who gives the research topics and related information for the problem.

We also appreciate work of (Patwa et al. 2020) for their online open dataset and permit of using it.

References

Brochier, R.; Guille, A.; and Velcin, J. 2019. Global vectors for node representations. *CoRR* abs/1902.11004.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Gundapu, S., and Mamidi, R. 2021. Transformer based automatic COVID-19 fake news detection system. *CoRR* abs/2101.00180.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR* abs/1909.11942.

Nistor, A., and Zadobrischi, E. 2022. The influence of fake news on social media: analysis and verification of web content during the covid-19 pandemic by advanced machine learning methods and natural language processing. *Sustainability* 14(17):10466.

Patwa, P.; Sharma, S.; PYKL, S.; Guptha, V.; Kumari, G.; Akhtar, M. S.; Ekbal, A.; Das, A.; and Chakraborty, T. 2020. Fighting an infodemic: COVID-19 fake news dataset. *CoRR* abs/2011.03327.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *CoRR* abs/1706.03762.

Vijjali, R.; Potluri, P.; Kumar, S.; and Teki, S. 2020. Two stage transformer model for COVID-19 fake news detection and fact checking. *CoRR* abs/2011.13253.