Fake News Detection with Various Natural Language Processing Models

Jiahe Wang jwa378
Shilin Wang swa285

Abstract

With the development of technology, the spread of fake news has been more common and harmful to our society as a whole. Detecting fake news is very arduous for humans, so it is meaningful to achieve fake news detection automatically through computers. This paper studies and compares seven deep learning models with the same hyper-parameters setting. Our results show that a model that combines the contextualized BERT embeddings with a CNN classifier has the best performance while several other models demonstrate promising potential. We hope that our findings can help facilitate future research on this topic and help the industry to mitigate the harm of fake news.

1 Introduction

The innovation of the internet enables a faster and more direct way for news publishing, sharing, and consuming with less regulations and standards for supervisions. As the internet has become more and more mature, a great number of people now rely on the internet to consume daily news and information, but most people have fallen for fake news as it is hard to determine which news is real and which is false. Also, analyzing and moderating such fake news manually is infeasible due to the vast volume of data on the internet. One practical approach to addressing this problem would be detecting fake news automatically through computers.

In our research, we have tried to address the problem of fake news by exploring and comparing various deep learning classification models and we aim to finding the best performing model according to evaluation metrics. The inputs are short sentences about the news (e.g., Liam Payne Just Dissed Harry Styles' Solo Music) and the output is 0 and 1, where 0 represent the news is real while 1 represent the news is fake.

The rest of the paper is organized as follows. Section 2 lists the prior related works. Section 3 presents the proposed approach. Section 4 describes the datasets we used. Section 5 demonstrates the details about experiments. Section 6 and Section 7 discusses and analyzes the results of our experiments. We also discuss future work and summarize our key findings in Section 8 and 9, respectively.

2 Related Work

Several works based on NLP approaches have been proposed for fake news classification. Perez-Rosas et al.(Perez-Rosas et al., 2018) designed two new datasets specifically for fake news detection by encompassing seven distinct news categories. One dataset is obtained through a combination of crowdsourced and manual annotation methods, whereas the other is gathered directly from the internet. They carried out a thorough experimental analysis to figure out the potential linguistic properties that are uniquely presented in the fake news using their own datasets. They also generated a detector using linear SVM classifier based on the linguistic features including syntactic, lexical, and semantic information and achieved accuracy rates as high as 76 percent.

Miller (Miller, 2017) used FNC-1 dataset provided by The Fake News Challenge to predict whether the headline of the article inferred the content of the article or not. Their network architecture incorporated an attention mechanism and multiple Bidirectional LSTMs. And the highest accuracy of 57 percent was attained by the BiLSTM and Multilayer Perceptron model. Additionally, Davis (Davis, 2017) used the same dataset for detecting fake news, but the highest accuracy they achieved 93 percent. They introduced a bag-of-words model followed by a three-layer multi-layer perceptron, which outperformed the other three neural architecture models in their research.

From recent papers, we found out that much work has been focused on the effects of datasets

on the performance of fake news detection. For example, Qiao et al (Qiao et al., 2020) studied the previous fake news detection papers and proposed that the linguistic characteristics of fake news have been overlooked. For example, they argued that characteristics such as lexical diversity and coordination could be indicators of whether a piece of news is real or fake, as a result, they tagged their data with features from 6 feature groups and improved their model's performance. In addition, Mridha et al (Mridha et al., 2021) compared and studied multiple models' performance on fake news datasets, with an emphasis on the respective advantages and disadvantages of previous authors' work and their models, most notably their computational cost and lack of pre-processing, which we learned to improve during our work. Another relevant discovery they made was that LSTM was successful at extracting significant features from diverse data sources.

3 Approach

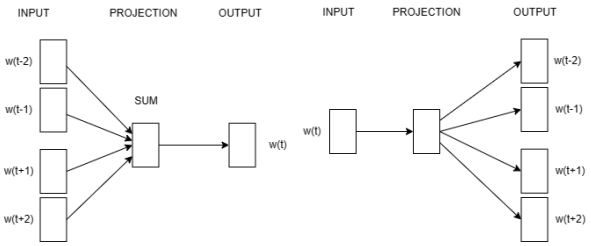
3.1 Word embeddings

In this section, we will compare two popular word embedding techniques: FastText, a static word embedding, and BERT, a contextual word embedding.

3.1.1 FastText Embedding

FastText (Joulin et al., 2016) embedding represents words using character n-grams instead of learning word vectors directly. This allows the embeddings to capture the essence of shorter words and suffixes and prefixes and effectively handle unknown words as well. After representing words with character n-grams, a continuous bag of words (CBOW) or Skip-gram model is employed to learn the embeddings. Specifically, the word representation is learned by using a large window to slide over both sides of context words, so each character n-gram has a corresponding vector representation and words are represented by the summation of these vectors, as depicted in Figure 1 (Mikolov et al., 2013).

The FastText embeddings we used is provided by Facebook's AI Research lab[1], which is pre-trained on a combination of the Wikipedia, UMBC webbase, and statmt.org news datasets. The output dimension of our FastText embeddings is 768.



(a) CBOW (b) Skip-gram

Figure 1: Architecture of the CBOW model and Skip-gram model

3.1.2 BERT Embedding

In this study, the term "BERT embedding" refers to the contextualized output vectors produced by transformer layers of BERT (Devlin et al., 2019), rather than the initial static word embeddings from the embedding layer. Unlike some traditional embeddings methos that assign a fixed representation to each word regardless of context, BERT generates dynamic word representations, which means that the same word can have different embeddings based on its surrounding words in a sentence. The BERT embedding can also handle unknown words effectively by usibng the word-piece tokenization.

We installed the pytorch interface for BERT by Hugging Face[2] and used the pre-trained "bert-base-uncased" to build the word embeddings. The output dimension of our BERT embeddings is 768.

3.2 Deep learning classifier

3.2.1 LSTM

The LSTM (Hochreiter and Schmidhuber, 1997) is a specialized recurrent neural network architecture designed to manage sequence data by maintaining long-term dependencies. In contrast to vanilla RNNs, LSTMs are more adept at learning from longer sequences as they effectively address the vanishing gradient issue by using additional input/output gates and cells. Specifically, the extra additive components and forget gate activations in LSTMs enabling gradients to not diminish so rapidly. LSTMs are commonly employed in scenarios where sequential patterns are crucial, making them a suitable choice for our fake news classification challenge.

In this paper, the model takes learned word embeddings of dimension 768 as input and then feeds the input vector to an LSTM layer (output dimen-

[1]https://github.com/facebookresearch/FastText

[2]https://huggingface.co/BERT-base-uncased

sion: 768) to enhance the context available for the algorithm. This is followed by a fully connected layer (input dimension: 768), a ReLU activation function, a dropout layer, and a final linear layer (Figure 2).

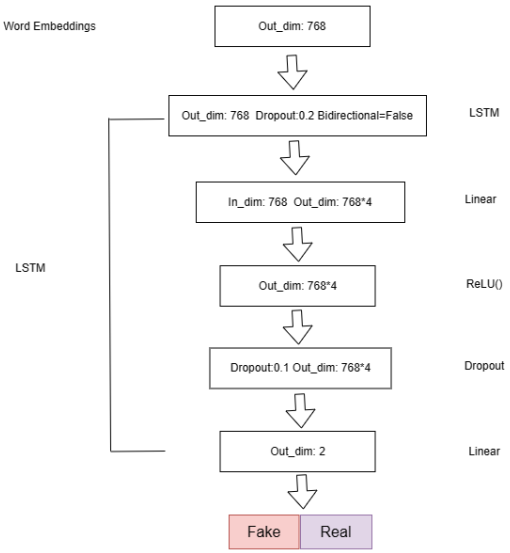


Figure 2: The model architecture of LSTM classifier

3.2.2 BiLSTM

A BiLSTM (Graves and Schmidhuber, 2005) is a sequence processing model comprising two LSTMs: one processing the input in a forward direction and the other in a backward direction. By incorporating both directions, BiLSTMs enhance the network's accessible information, which in turn bolster the algorithm's contextual understanding.

The architecture (Figure 3) of the BiLSTM classifier closely resembles the LSTM (Section 3.2.1), with a few differences. Firstly, the parameter "bidirectional" is set to be true in the BiLSTM layer. Secondly, the input dimension of the following fully connected layer is changed to 768 * 2. This modification reflects the bidirectional nature of the BiLSTM layer, which captures information from both forward and backward directions, resulting in a more robust model.

3.2.3 CNN

Another popular method in the neural network field is CNN classification. Partially due to the popularity of its application in image processing, Convolutional Neural Network is very effective in extracting relationships and information from different parts of data, in our case words from texts, and classify based on such information. We hope that our

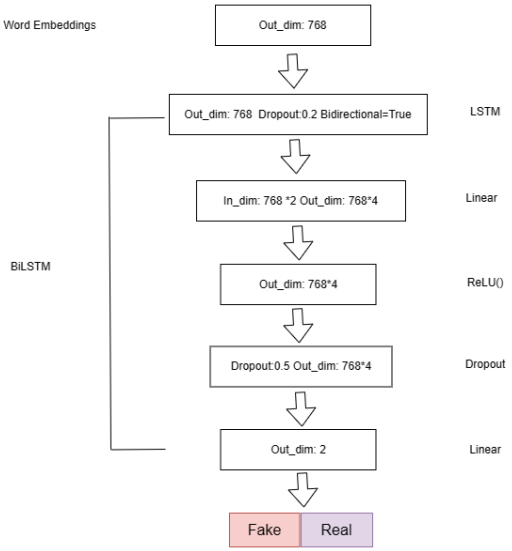


Figure 3: The model architecture of BiLSTM classifier

convolutional network can first capture the information of the texts and then form key discoveries about how to effectively classify fake news. Like the methods mentioned in 3.2.1 and 3.2.2, we plan to use cross-entropy loss to train the network and we would like to find out that if the CNN method can continue its advantage in image classification and apply the same mechanism to text classification.

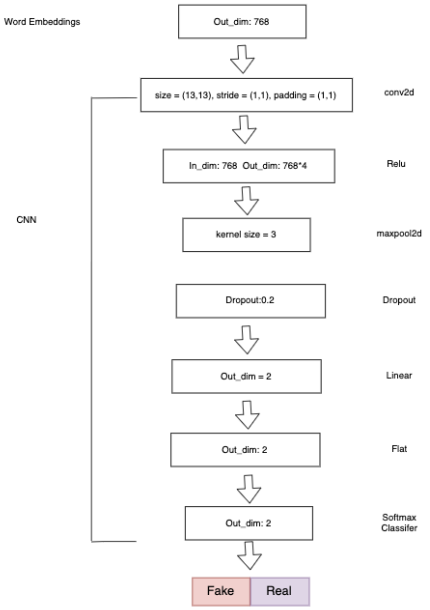


Figure 4: The model architecture of CNN classifier

We followed the similar model architecture(including code) designed by (Mozafari et al., 2019)(Figure 4), which has its special convolution

method and max pooling layers in order to capture the relationships within different texts inside of headline texts. It is also worth mentioning that this structure uses all layers available from BERT because we hope that CNN can perform better when dealing with data with higher complexity.

3.2.4 BERT

BERT (Devlin et al., 2019) is currently one of the most powerful NLP models, capable of handling a wide range of tasks such as text classification, semantic similarity, and question answering. It has revolutionized the field by taking extensive left and right contexts of a given word into account, which allows BERT to provide a profound understanding of language and context. BERT is built on the transformer architecture and utilizes the attention mechanism to capture relationships between words in a sentence. Since BERT is pre-trained on massive text corpora, it can be easily fine-tuned for specific NLP tasks and achieving remarkable performance at the same time.

We used the pre-trained BERT base classification model[3], which begins with an embedding layer that aims to convert input tokens into continuous representations. This is followed by 12 Transformer layers, which are responsible for capturing the context and relationships within the text. After the 12 Transformer layers, a fully connected dense layer with softmax activation is added for classification(Figure 5).

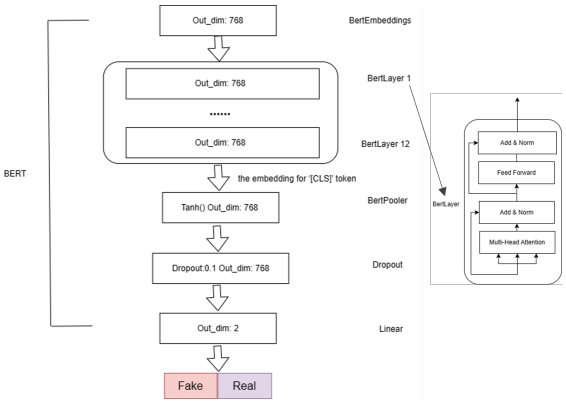


Figure 5: The model architecture of BERT

3.3 Loss Function

As our task is a binary classification problem, the training loss function used in all methods is cross-entropy. And the cross-entropy can be computed by taking the probabilities of events in P and Q as follows (Koppert-Anislmova):

$$H(P, Q) = -\operatorname{sum} x \text{ in } XP(x) * \log(Q(x))$$

4 Data

By researching online and referencing to multiple sources, we utilized and combined multiple data sources. We started with Fake News Net (Shu), a curated list of news data containing real news and fake news titles verified politifact and gossipcop, and combined it with the Fake News Dataset (Perez-Rosas et al., 2018), a crowd-sourced dataset containing both legitimate news and fake news data. All the the data mentioned above have news texts related information, which will be the input for our models, and the "y" column which indicates whether the title belongs to legitimate news backed by facts or fake news.

Upon data oversampling, we gained that the classes of our dataset, true news or fake news, are distributed evenly as shown in Figure 6.

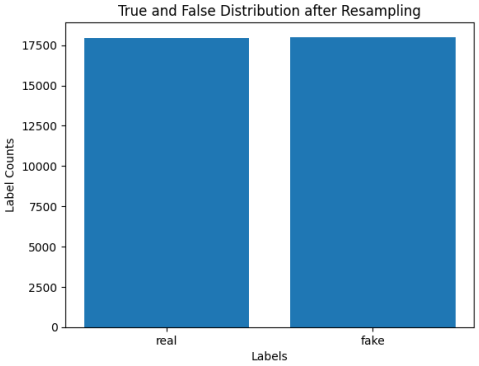


Figure 6: Balanced datasets after resampling

Furthermore, we also looked into the texts inside of these news headlines, and it became conspicuous that inside of a large portion of headlines consisted of celebrity names, and we are curious about whether certain celebrities are more prone to be related to fake news compared with other celebrities. Therefore, here is a graph of 7 celebrities whose names appear most frequently inside of our data,

Label	News
0: Real	Tesco to pay £129m fine over accounting scandal.
1: Fake	Breaking News: Snapchat to purchase Twitter for 255 billion.
0: Real	Just another day in the Kardashian-Jenner world.
1: Fake	Liam Payne Just Dissed Harry Styles' Solo Music
0: Real	Woman arrested three times as she tries to see President Trump
1: Fake	FBI investigates computer link between Trump and Clinton

Table 1: Examples of the input and output

[3]https://huggingface.co/BERT-base-uncased

and we can directly find out that the proportion of real news and fake news differ significantly for different celebrities.

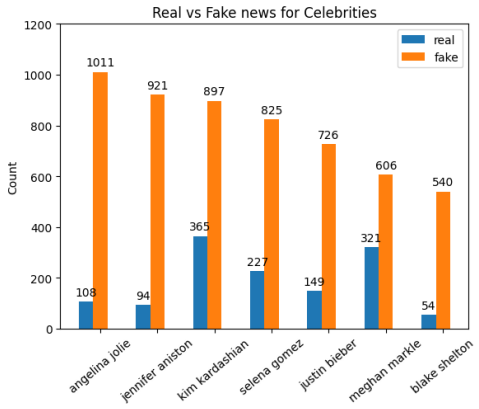


Figure 7: Fake news ratio for celebrities

5 Experiments

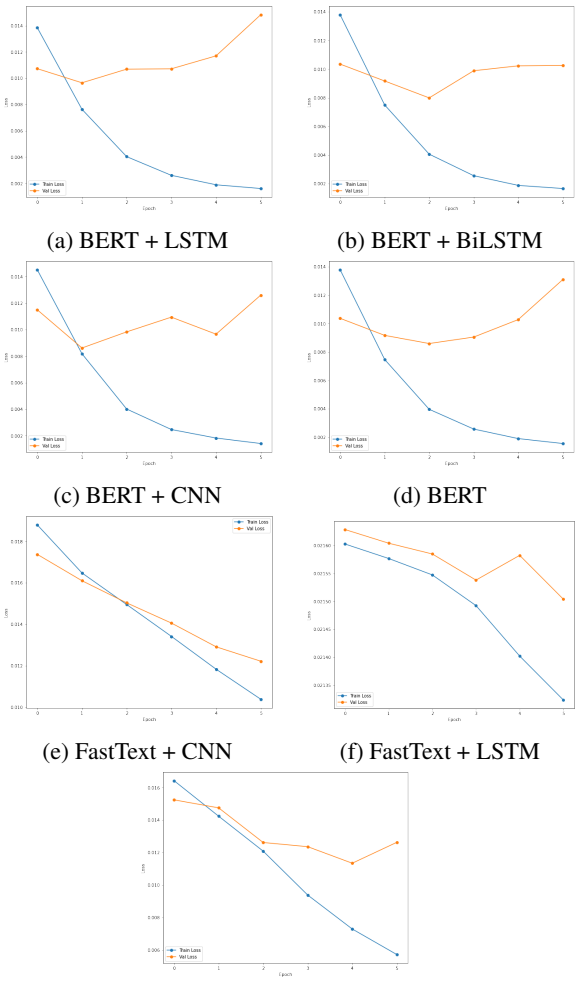
5.1 Evaluation

As for the quantitative metrics to compare models' performance, there are mainly four evaluation matrices we plan to use: accuracy rate, F1 measure, precision, and recall.

Accuracy is a basic evaluation metric that measures how many data points are predicted correctly. It provides a straightforward assessment of a model's performance. The F1 measure is another typical measure used to evaluate classification performance, where the ranges from 1 to 0, and the bigger the value the better the performance for that particular class. Besides, we will also refer to the precision and recall values for each class and each model. Precision calculates the ratio of the number of correctly predicted by the classifier for a class and the total number of samples predicted by the classifier for that class, whereas recall is the ratio of the number of correctly predicted by the classifier for a class and the actual number of samples for that class. These two metrics should give us an accurate overview of how these models perform for each of class.

5.2 Implementation

In order to have a fair comparison of all models performances, we conducted the experiments under the same configurations of hyper-parameters as outlined below(Table 2).



(a) BERT + LSTM (b) BERT + BiLSTM

(c) BERT + CNN (d) BERT

(e) FastText + CNN (f) FastText + LSTM

(g) FastText + BiLSTM

Figure 8: Learning curve (loss) against epochs

Epochs	Learning rate	Batch size	Adam ϵ	Adam β_1	Adam β_2	Adam bias correction	Dropout	Weight decay	Random seeds
6	5e-5	32	1e-6	0.9	0.999	True	0.2	0.01	1

Table 2: Summary of hyper-parameters

Model	Accuracy	Precision	Recall	F1
FastText+LSTM	0.5124	0.6007	0.5016	0.3438
FastText+CNN	0.8266	0.8275	0.8271	0.8265
FastText+BiLSTM	0.8526	0.8546	0.8517	0.8521
BERT+LSTM	0.9022	0.9044	0.9015	0.9019
BERT	0.9072	0.9093	0.9065	0.9070
BERT+BiLSTM	0.9097	0.9111	0.9091	0.9095
BERT+CNN	0.9109	0.9118	0.9105	0.9108

Table 3: Summary of evaluation results

6 Results

As observed from Figure 8, we got an overview about the learning curve (loss) against epochs on training and validation data. The x-axis represents the epoch number and the y-axis indicates the loss value. It is worth noting that for most of the models the training loss, which is the blue line, would decrease sharply for the first few epochs and then become slowly become smaller and smaller. Such a curve is much expected when training a model, but the models with FastText embeddings' training loss curve still decreases fast after the first few epochs and still demonstrated a downward trend at the end. Given more data maybe the training curve will finally slow down, and this might indicate that for our specific topic, FastText embedding requires more training data in order to gain better performance. As for the orange line, the validation loss, most of the models have validation loss increased near the end of the training, indicating potential overfitting after epochs except the FastText CNN model, which still has room of more training with more data.

Moreover, we have trained the model and compared them by the following metrics: accuracy, precision, recall, F1 score(Table 3). All the performances below are based on models predicting results on test sets.

Although the performance of FastText + LSTM is significantly behind other models, this occurs mainly because the model did not converge after the training phase had finished. However, in our previous effort in order to maximize each model's performance, we were able to get a test accuracy of around 0.87 across the quantitative metrics under hyper-parameter settings of epochs = 30, learning rate = 0.01, batch size = 64, and other settings the same as the ones in Table 2. Such performance is considered to be more than adequate for the model to qualify for the fake news detection task, and therefore this model combination still has potential even though it performs badly while other models have adequate scores under this setting.

For other models, we managed to get huge improvements compared with FastText + LSTM, and their balanced matrix indicates that their predictions are reliable with acceptable false positive and false negative results. Besides the worst-performing model of FastText + LSTM, our best-performing model is the BERT + CNN model, which has scores slightly higher than the BERT + BiLSTM model's scores.

In addition, the BERT coupled classifiers are all better than their FastText counterparts, and the gap between the worst-performing BERT model and the best-performing FastText model is still significant. Therefore, we can conclude that in our given hyper-parameter settings and datasets, the BERT embeddings yields better results than the FastText embeddings. It is also worth noticing that BERT + BiLSTM and BERT + CNN tend to perform slightly better than BERT + LSTM and FastText + BiLSTM and the vanilla BERT model as well.

7 Analysis

In this section, we concentrated on error analysis, specifically discussing instances where the mentioned models fall short in performing the fake news detection task. Since the FastText + LSTM model does not converge with the current hyper-parameter settings, we didn't include the detailed qualitative analysis of this model.

7.1 Quantitative Analysis

The analysis of the errors for all models utilized in the experiment began by examining the label-level error rates, with in-depth performance data presented in Figure 9. As observed, the labels causing the higher number of errors across all models are the real ones, indicating that the models are more proficient in identifying fake news than real news. Several factors can explain this outcome. Firstly, fake news may display more distinguishable pat-

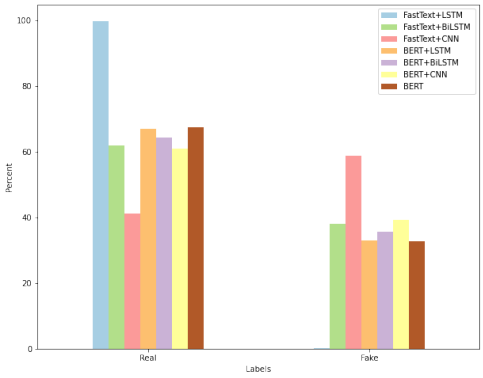


Figure 9: Label-wise error distribution

Model	Average Length
FastText+LSTM	370.11
BERT+LSTM	38.25
BERT	25.60
BERT+CNN	22.55
FastText+CNN	16.81
FastText+BiLSTM	13.43
BERT+BiLSTM	12.72

Table 4: Average length of misclassified fake news texts

terns, whereas real labels could have more nuanced characteristics that are harder for the model to detect. Secondly, real news tends to be more distinctive and may contain more complex expressions and richer emotions, which makes the classification more more difficult compared to fake news. Thirdly, the model may have experienced overfitting issues with the training data, particularly in regards to patterns associated with fake labels. We did not have sufficient time to thoroughly investigate this issue; however, addressing it will be part of our future work.

Considering the purpose of this research, we delved deeper to examine errors that falsely detected fake news. We computed the average length of the misclassified fake news texts and the details are provided in Table 4. The average length varies across the models indicating that each model may have its strengths and weaknesses when detecting fake news with different lengths. Specifically, the BiLSTM classifiers (average lengths of 13 and 12 for Fasttext and BERRT coupled, respectively) seem to have advantages with relatively longer texts. This could be due to the bidirectional LSTM layers' ability to capture essential features in longer texts effectively. However, the LSTM classifiers have the highest average length of misclassified texts, which implies that this model might struggle more with longer texts. While the occurrence of extreme values in the FastText + LSTM model can be attributed to the model not converging with the current hyperparameter settings. And on average, the CNN and BERT models have misclassified text lengths of approximately 23 words. This results highlights the importance of selecting the most suitable model architecture and hyperparams for the specific task and dataset.

	Examples
1	Get a grip teenagers! You've got a few GCSEs not a role on Netflix, says LIZ JONES
2	The Bachelorette: Becca 'Frustrated' That Tia Still Has Feelings for Colton
3	Meghan Markle: Six reasons why she divides people — even Americans
4	Jim Carrey: Hollywood Elites 'Eat Whole Babies' For Christmas

Table 5: Only BERT coupled classifiers detect fake news correctly

7.2 Qualitative Analysis

Table 5 describes the errors made up by FastText coupled classifiers on fake news correctly detected by BERT coupled classifiers. In the presented examples, the fake news contain informal language, multiple semantics, subtle cues, and colloquial expressions that might be hard to captured by the FastText models. For instance, the phase "get a grip" in the example 1 is a colloquialism that may not be universally understood or may be interpreted differently depending on the reader's cultural backgrounds; the example 2 uses the word 'Frustrated' indicates a level of emotion ; the mention of "divides people" in example 3 can evoke strong reactions from readers. And the last example also uses provocative language and an outrageous claim to create shock value. Since the FastText embeddings rely heavily on sequences of words or characters, but the informal language and multiple semantics can disrupt these sequences, making it harder for the model to capture meaningful patterns and relationships. In contrast, having the contextual word embeddings from BERT, the classifiers can better discern the semantics or contextual differences, and then correctly identifying these news as fake.

Table 6 refers to fake news that correctly detected by FastText coupled classifiers, but instead led to errors for BERT coupled classifiers. In detail. these news are all objective descriptions without strong emotional mechanisms and overtly sensa-

	Examples
1	Duchess Meghan and Prince Harry Are Planning a U.S. Tour: Details!
2	Brad Pitt Angelina Jolie Gave $8 Mil to Charity
3	Gwen Stefani gifts Blake Shelton a flagpole for his birthday
4	Taylor Swift and Boyfriend Joe Alwyn Are 'Very Much in Love'

Table 6: Only FastText coupled classifier detect fake news correctly

tional language. which may pose a challenge for BERT coupled classifiers in detecting their fake status as the model may be more focused on detecting patterns in more complex or emotionally charged text. Also, the news are relatively straightforward and focus on specific subjects, such as celebrities or popular TV shows, which could appear in both real and fake news. This may make it difficult for BERT to distinguish between fake and real news based solely on these elements. In contrast, FastText might be more capable of recognizing such patterns using character-level n-grams.

8 Limitations

There are some limitations and areas for future work that can be considered. First, our inputs are mainly short sentences, so it didn't have any additional context or information about the whole news' contents. For some situations. the current input alone does not provide enough data to accurately identify fake news. Thus, it would be beneficial to combine our current inputs with relevant information from the corresponding full articles in order to enhance the accuracy of fake news detection. Second, the models may be facing the overfitting problems according to Figure 7. To address this issue, future work should focus on implementing techniques such as cross-validation and regularization to improve the model's ability to generalize with unseen data and the capability of making accurate predictions based on limited information as well. Besides, the languages of all current datasets are English. Conducting analogous investigations on other languages such as Chinese or French would also be a valuable endeavor in the future.

9 Conclusion

This paper introduces a comparative study of various deep learning classification models for the fake news binary classification task. Specifically, we studied both FassText word embeddings(the static word embeddings) and BERT embeddings (contextual word embeddings) and four deep learning classifiers including LSTM, BiLSTM, CNN, and BERT as well. We observed that the BERT + CNN is the best performance model as an overall suggestion according to the evaluation results on test data and the BERT coupled classifiers are all better than their FastText counterparts. And the BiLSTM classifiers seem to have advantages in analyzing relatively longer news compared to other classifiers based on our dataset. Furthermore, we also noticed that the contextual word embeddings from BERT can better help the classifiers deal with the news containing informal language, multiple semantics, subtle cues, and colloquial expressions, which would be hard to be captured by the FastText embeddings.

10 Contributions

Shilin Wang:

- Implemented FastText + LSTM, BERT + LSTM models.
- Wrote the Introduction, Related Work(partly), Approach(including Word embeddings, Loss Function and partly of Depp learning classifier) sections of the report.
- Wrote the Introduction, Analysis, Limitations, and Conclusion sections of the project notebook.

Jiahe Wang:

- Implemented FastText + CNN, FastText + BiLSTM, BERT + BiLSTM, BERT and BERT + CNN models
- Wrote the Related Work(partly), Data, Experiments, and Results Analysis, Limitations, and Conclusion sections of the report.
- Wrote the Data, Approach, Implementation, and Results sections of the project notebook.

References

Richard I. A. Davis. 2017. Fake news, real consequences: Recruiting neural networks for the fight against fake news.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, 1810.04805.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610. IJCNN 2005.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification.

Inara Koppert-Anislmova. Cross-entropy loss in ml.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *ArXiv*, 1309.4168.

Kurt T. Miller. 2017. Fake news headline classification using neural networks with attention.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media. In *International Workshop on Complex Networks & Their Applications*.

M. F. Mridha, Ashfia Jannat Keya, Md. Abdul Hamid, Muhammad Mostafa Monowar, and Md. Saifur Rahman. 2021. A comprehensive review on fake news detection with deep learning. *IEEE Access*, 9:156151–156170.

Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news.

Yu Qiao, Daniel Wiechmann, and Elma Kerz. 2020. A language-based approach to fake news detection through interpretable features and BRNN. In *Proceedings of the 3rd International Workshop on Rumours and Deception in Social Media (RDSM)*, pages 14–31, Barcelona, Spain (Online). Association for Computational Linguistics.

Kai Shu. Fakenewsnet.