Exploring the Impact of Sentiment Analysis on Current Methods of Fake News Detection

Mudi Yang[1], Lorenzo Flores[1], Himnish Hunma[1], Bernardo Trevisan[1]
[1]Department of Computer Science, Yale University

ABSTRACT

With the rapid proliferation of fake news, optimizing current fake news detection algorithms has become essential to keep up with the pace of misinformation. In this paper, we explore the performance of various fake news detection schemas (FakeBert with Glove, FakeBert with BERT, Dense Neural Net Model with TF-IDF Embeddings, and HuggingFace BERT) and extend their capabilities with BERT based sentiment analysis on the LIAR dataset. We show that sentiment analysis is able to slightly boost the performance of models on assessing the validity of statements and assigning one of six ratings: True, Mostly True, Half True, Barely True, False, Pants on Fire. Our analysis concludes that HuggingFace BERT achieves the best performance across the observed models.

INTRODUCTION

The proliferation of fake articles and misinformation has led to increased interest in the area of fake news detection, which can be condensed to the task of classifying journalistic statements as either true or false. The importance of efficiently detecting fake news lies in the increasingly fast exchange of information. With the advent of new technologies and channels of communication, it has become easier for dangerous sources to propagate falsehoods and exploit internet's loopholes. If unlawful behavior as such is not detected fast enough, an exponential number of people might become victims to it. The motivation to study neural networks' potential to stop the spread of misinformation, therefore, originates from the desire to decrease this effect. Neural networks constitute a promising candidate for the task of fake news detection. That is greatly owing to its ability to isolate important features and to derive inferences from huge amounts of data, one of the main challenges of the task in question.

In this paper, we evaluate the performance of different architectures in classifying statements on LIAR, a benchmark dataset for fake news detection. Additionally, we explore the impact of data augmentation methods and the use of sentiment analysis on the performance of such models. Ultimately, the development of such models can assist ongoing fact checking efforts in combating the spread of misinformation.

METHODOLOGY

Dataset

We used the LIAR dataset, which contains 10,269 training examples, 1,284 testing examples, and 1,267 validation examples. It includes the statement, its subject, the speaker's name, job title, and party affiliation, context in which the statement was made, and counts of true and untrue statements previously made by the speaker. The statements were made between 2007 to 2016, either through social media or in person, usually through a speech or news broadcast. Each statement is classified into one of six labels: True, Mostly True, Half True, Barely True, False, Pants on Fire. These labels were determined by editors from PolitiFact.com, a fact-checking website by the Poynter Institute. The distribution of labels is roughly uniform, with over 2000 statements per label, with the exception of Pants on Fire, which has 1050 statements.

Then, we augmented the data through synonym replacement, which resulted in 30,720 training examples, 3,801 testing examples, and 3,852 validation examples. We further added sentiment analysis as another feature to explore its effect on model performance.

Finally, we explored the performance on the original six categories, and with three classes, False (Pants on Fire, False), Moderately False (Barely True, Half True), and True (Mostly True, True).

Model Architecture

We explored four models, three of which use transfer learning approaches. Speccifically, we used Glove and BERT—pre-trained models that have been trained across large corpuses of books and websites like Wikipedia, to generate embeddings which encode linguistic information.

1. FakeBERT Model with Glove

The team implemented the architecture of FakeBERT (Kaliyar, Goswami, & Narang, 2021), which is a combination of a unidirectional pre-trained word embedding model followed by a 1Dconvolutional-pooling layer network. Statements from LIAR were encoded using Glove embeddings, and capped at a maximum of 100 words, as done in the original model. The network architecture is described in Table 1.

2. FakeBERT Model with BERT Embeddings

The team tweaked the input size of the FakeBERT model to use BERT embeddings, and capped statements at a maximum of 25 words. This increased computational time, as BERT embeddings are longer than that of Glove. The network architecture is described in Table 2.

3. Dense Neural Net Model with TF-IDF Embeddings

The team implemented a dense neural network architecture described in the paper by Thota, Tilak, Ahluwalia, and Lohia (2018). Here, statements were turned into TF-IDF vectors, and passed through the network described in Table 3. The process was repeated using bigrams as inputs to the TF-IDF vectorization process.

4. HuggingFace BERT Model for Sequence Classification

The team implemented a BERT Model for sequence classification, taken directly from the HuggingFace library. HuggingFace allows users to retrain the BERT embeddings directly, which can increase the power of the model.

Sentiment Analysis

The team also experimented with adding sentiment analysis score as input to the model. Using the nlp-text-emotion model, which is namely using a pre-trained BERT, was trained to classify each statement as expressing either joy, sadness, fear, anger, or no sentiment (i.e. neutral) using data from dailydialog, emotion-stimulus and isear. A one hot encoding vector was generated per statement to encode the sentiment based on the five classes. The team then concatenated the embedding vector and the sentiment vector, and fed this as input to the classification model. The idea behind this is that using sentiment as an additional feature will provide the model with more information to make a classification, which is expected to improve the model's performance.

Input Tweaking

The team then attempted to augment the data with artificial statements, which were made by replacing words in the original sentences with their synonyms. Deep learning models typically learn better with more data (Feng et al., 2021). To this end, data augmentation is performed to increase the size of the dataset, which is expected to improve the model's performance.

After looking up different libraries enabling NLP input manipulation, we settled on a library called nlpaug. This tool enables manipulations including inserting keyboard typos, antonym conversions, back translations, random word dropping, etc… The manipulation we focused on was Synonym switching. Using tools such as NLTK's Wordnet, the library samples a number of words from the sentence and tries to find their synonyms from Wordnet's synsets.

After feeding a subset of the LIAR dataset to the SynonymAug tool, we found a few issues:

- Certain nouns were switched out which made the augmented sentence meaningless. One example was: "It started when natural gas took off" → "It started when natural accelerator pedal took off"
- Given the political nature of the statements, certain terms have different meanings in common language and political context. Switching out those terms would make the sentence lose its political meaning.
- Certain letters get replaced my numbers, eg: "I'm" → "1' m"

To address the first issue, we created a dictionary of the 1000 most common adjectives and embedded it in the local version of the nlpaug library, and tweaked the augmenting function to only switch out synonyms of adjectives instead of switching out nouns as well. The tweak proved to fix issues from the first bullet point, but also reduced the scope of the augmenter since less words could be changed.

To increase the scope of the modified augmenter, we also created dictionaries of adverbs and verbs which should be safe to switch out in the input. We also increased the lower bound of words to switch out to see what the results would be. While some examples were augmented successfully,

> Eg: "The economic turnaround started at the end of my term." → "The **economical** turnaround **start out** at the **conclusion** of my term."

While some were quite nonsensical:

> Eg: Hillary Clinton agrees with John McCain "by voting to give George Bush the benefit of the doubt on Iran." → Hillary Clinton **fit** in with John McCain "by **vote** to **present** George Bush the **welfare** of the **uncertainty** on Iran."

There are some definite issues with the tenses not being respected during the switching, and after testing a few iterations of the library, we found that changing adjectives and adverbs was more accurate than changing verbs as well. For example, when adjectives, adverbs, and verbs were exchanged:

> Eg. Health care reform legislation is likely to mandate free sex change surgeries. → Health **fear** reform legislation is **probable** to mandate **complimentary** sex **modification** surgeries.

When only adjectives and adverbs were exchanged:

> Eg. Health care reform legislation is likely to mandate free sex change surgeries. → Health care reform legislation is **potential** to mandate **complimentary** sex change surgeries.

Some of these strange changes must have been due to nlpaug being unable to differentiate between verbs and gerundives. We also understood that the lower bound on the number of words being switched out will require some tuning for the best accuracy.
In the process, we also found that due to the probabilistic nature

TABLE 1
Classification Accuracies by Model Using the Original LIAR Dataset

METHOD	TEST ACCURACY (3 CLASSES)	TEST ACCURACY (6 CLASSES)	VALIDATION ACCURACY (3 CLASSES)	VALIDATION ACCURACY (6 CLASSES)	EPOCHS	LR
FakeBERT Model with Glove	38.12%	21.23%	39.40%	19.78%	100	1e-4
FakeBERT Model with BERT	36.62%	22.02%	40.65%	22.89%	100	1e-4
Dense Neural Net Model with Unigram TF-IDF Embeddings	42.14%	22.02%	41.12%	20.09%	10	1e-4
HuggingFace Model with BERT Embedding	**48.77%**	**28.33%**	**48.59%**	**26.71%**	**2**	**5e-5**

TABLE 2
Classification Accuracies by Model Using the Augmented LIAR Dataset

METHOD	TEST ACCURACY (3 CLASSES)	TEST ACCURACY (6 CLASSES)	VALIDATION ACCURACY (3 CLASSES)	VALIDATION ACCURACY (6 CLASSES)	EPOCHS	LR
FakeBERT Model with Glove	38.51%	19.81%	39.39%	20.32%	100	1e-4
FakeBERT Model with BERT	41.19%	22.02%	42.28%	22.58%	30	9e-5
Dense Neural Net Model with Unigram TF-IDF Embeddings	40.41%	20.04%	42.36%	21.49%	10	1e-4
HuggingFace Model with BERT Embedding	**45.14%**	**24.78%**	**43.06%**	**26.40%**	**2**	**5e-5**

of the augmenter, it was possible to feed the same sentence to the augmenter multiple times and obtain different results. Therefore, running the augmenter several times on the LIAR dataset will give us more input for free.

Combined Model

The team combined the best performing neural net with the sentiment analysis model into a single Jupyter Notebook for easier usage. The GitHub code repo can be found here.

Training Procedure and Evaluation

The models were trained using cross entropy loss, and the accuracy was calculated across both test and validation sets from the LIAR dataset. In addition to the cross-entropy loss, we added a penalty L when training the model on the augmented data to penalize outputs wherein the classification probabilities were different for the original and augmented statements.

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}\left|p_{i,j} - \overline{p_j}\right|$$

Where N is the number of statements sharing the same ID, k is the number of classes (either 3 or 6), $p_{i,j}$ is the classification probability for class j on statement i, and $\overline{p}_j$ is the average classification probability across all N statements for class j.

Classification accuracy was used to evaluate the results and compare it to results from previous literature.

RESULTS AND INSIGHTS

We present the results for the classification accuracies achieved by models trained on the original and augmented datasets in Tables 1 and 2 respectively.

On both original and augmented datasets, the HuggingFace BERT model performed better than the other models. Notably, it outperformed the reported classification accuracy achieved by the FakeBERT model when trained solely on the statements on the 6 class task. A possible reason for the improved performance is that the HuggingFace BERT model retrains the embeddings and penalizes for classification loss on the LIAR dataset. This may allow the embeddings to extract aspects of statements that better distinguish between real or fake statements.

Based on the resulting confusion matrices for the HuggingFace BERT model in Figure 1, appears that the more moderate classes (e.g. Moderately False, Barely True) were classified correctly more often, whereas the more extreme classes (e.g. True, Pants on Fire) were often classified incorrectly. This may be because it might be difficult to discern what about a statement would make it absolutely true or absolutely false – thus the model was able to minimize loss by ranking most statements as moderately false.

The dense neural network model with unigram TF-IDF embeddings came second on both tasks. This may be due to the much larger dimensionality of the inputs in comparison to the FakeBERT model, which allows the model to encode for or learn more information. We trained the same model using bigram TF-IDF embeddings, which achieved similar results to the original dataset. Unfortunately, the augmented dataset was too computationally expensive to train.

Across all models, those trained on the augmented datasets achieved lower accuracy than the original models. While the team's intuition for augmenting the dataset was that more data may train the model to recognize patterns in fake and real statements, adding more statements does not seem to have helped the model in terms of performance.

Modified Loss

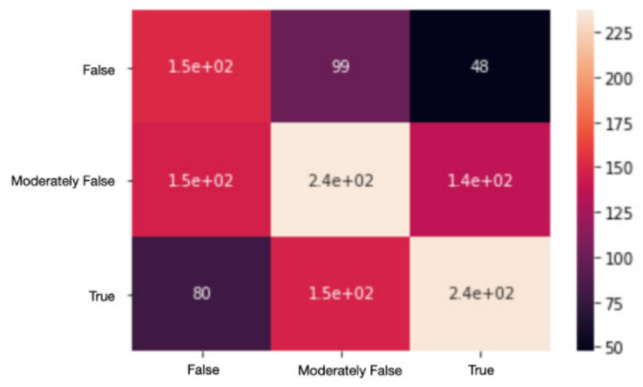
We selected the dense neural net model with unigram TF-IDF embeddings, as this was the best performing model wherein the loss could be easily tweaked. The results are shown in Table 3.

The additional penalty does not seem to improve model performance in terms of classification accuracy, as this model performed relatively similar to the counterpart models trained on the original and augmented datasets.

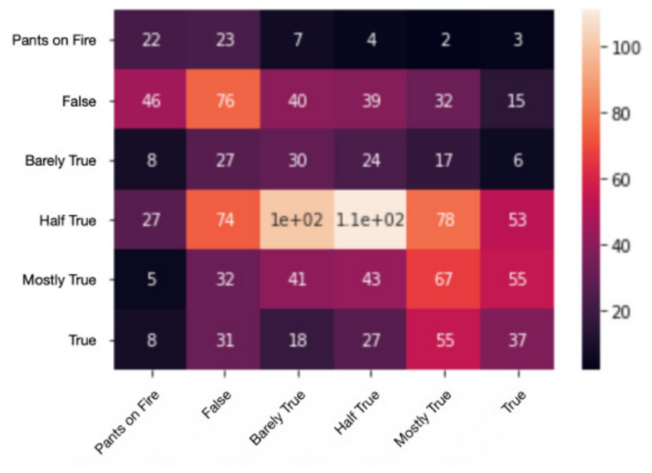
Sentiment Score

Using the dense neural net model with unigram TF-IDF embeddings, the team explored the effect of adding sentiment score to the input. The results are shown in Table 4.

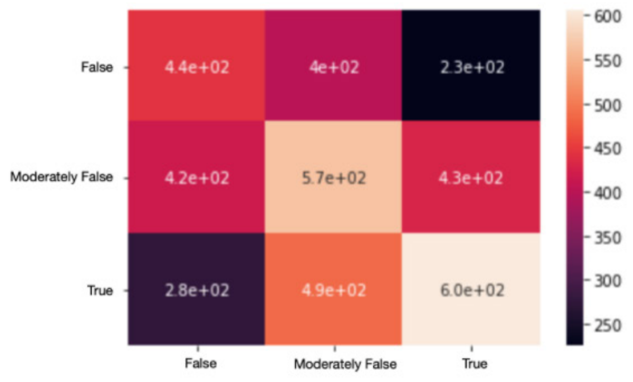
The team found that adding sentiment scores to the original model improved the model's performance on the 6 class task (Test Accuracy: +0.55%, Validation Accuracy: +3.27%), whereas its impact on the 3 class task was not as significant (Test Accuracy: -0.95%, Validation Accuracy: +0.31%). Additionally, the classification accuracy dropped when the team both augmented the dataset and used sentiment score in the training process. Sentiment was classified in



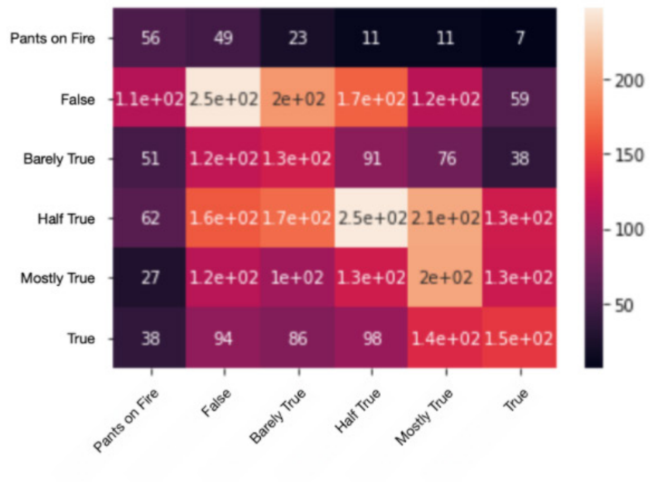
a. 3 Classes, Original Dataset



b. 6 Classes, Original Dataset



c. 3 Classes, Augmented Dataset



d. 6 Classes, Augmented Dataset

FIGURE 1. *Confusion Matrices for HuggingFace BERT Model on Trained Datasets.*

TABLE 3

Classification Accuracies by Model Training Method with the Dense Neural Net Unigram TF-IDF Model

METHOD	TEST ACCURACY (3 CLASSES)	TEST ACCURACY (6 CLASSES)	VALIDATION ACCURACY (3 CLASSES)	VALIDATION ACCURACY (6 CLASSES)	EPOCHS	LR
Dense Neural Net Model with Unigram TF-IDF Embeddings	42.14%	22.02%	41.12%	20.09%	10	1e-4
Dense Neural Net Model with Unigram TF-IDF Embeddings Sentiment Score	40.41%	20.04%	42.36%	21.49%	10	1e-4
Dense Neural Net Model with Unigram TF-IDF Embeddings with Paired Loss	38.43%	22.09%	42.75%	22.11%	10	1e-4

TABLE 4

Classification Accuracies by Model Training Method with the Dense Neural Net Unigram TF-IDF Model Incorporating Sentiment Analysis

METHOD	TEST ACCURACY (3 CLASSES)	TEST ACCURACY (6 CLASSES)	VALIDATION ACCURACY (3 CLASSES)	VALIDATION ACCURACY (6 CLASSES)	EPOCHS	LR
Dense Neural Net Model with Unigram TF-IDF Embeddings	42.14%	22.02%	41.12%	20.09%	10	1e-4
Dense Neural Net Model with Unigram TF-IDF Embeddings Sentiment Score	41.19%	22.57%	41.43%	23.36%	20	3e-4
Dense Neural Net Model with Unigram TF-IDF Embeddings with Paired Loss	37.33%	21.46%	37.77%	22.35%	20	3e-4

5 levels, giving a high dimensional context to the statements. Considering the 6-class classification problem; there is a mapping from a 5-dimension feature space to a 6-dimension solution space. In the 3-class classification, there is a mapping from a 5-dimensional feature space to a 3-dimensional solution space. Assuming sentiment score gives accurate context about the veracity of a statement, in the latter problem, we are mapping to a lower dimensional solution space and possibly incurring misclassifications due to smoothing. This could explain the drop in accuracy for the 3-class classification problem.

Comparison to Previous Literature

The networks that the team explored achieved very similar accuracy to those from previous literature that made classifications solely using the text. We present them in Table 5.

Previous work often incorporated the other features of the LIAR dataset in prediction, such as the subject, speaker, job, party affiliation (Democrat or Republican), and history – which is a list of the previous statements within the dataset made by the same speaker, and the truth scores of those statements. The BERT-Based Mental Model (Ding et al., 2020) for example concatenated the claim and history, and achieved 45.9% and 49.3% accuracy on the test and validation accuracies respectively. The increase in accuracy makes sense, as the credibility of previous statements made by a speaker likely has an impact on the credibility of his or her other statements. A limitation of incorporating these features however is that they are not readily available for real-time fake news detection, as these features are encoded manually.

We also note that the LIAR dataset appears to be more difficult to crack in comparison to other fake news datasets. The Fake-News dataset for example, which consists of text drawn from the period around the 2016 US presidential election, consists of two classes – true and false, in comparison to the LIAR dataset which further distinguishes between six levels of truthfulness. The FakeBERT model (Kaliyar et al., 2021) achieved a 98.90% classification accuracy on the Fake-News dataset, while this model architecture achieved relatively low classification accuracy when applied on the LIAR dataset. This tells us that the ability for models to perform fake news detection is also largely affected by the dataset.

TABLE 5

Comparison of Results with Previous Literature

PAPER	DESCRIPTION	TEST ACC	VALIDATION ACC
Wang et al., 2017	Word2Vec CNN Model	27.0%	26.0%
Ding et al., 2020	Hybrid BERT Model	27.3%	26.71%
Ours	HuggingFace BERT Model	28.33%	23.36%
Ours	Dense Neural Net Model with Unigram TF-IDF Embeddings Sentiment Score	22.57%	23.36%

The concentration of the Fake-News dataset on statements made during the election campaign may have resulted in more polarizing statements and stronger language that models may have picked up on, in comparison to the LIAR dataset which consisted of statements made in various contexts.

CONCLUSION

In this paper, the team compared the performance of different neural network architectures in classifying fake news, and the impact of data augmentation, the usage of a new loss function, and the addition of sentiment score on the performance of the models. Overall, the HuggingFace BERT model achieved the highest classification accuracy (Test: 28.33%, Valid: 26.71%), which is attributed to the fact that the model retrained the BERT embeddings, while the other architectures simply loaded pre-trained embeddings. Across the board, the networks had difficulty in classifying the more extreme categories (i.e. True, Pants on Fire), which demonstrates how difficult it can be for models to distinguish between levels of truthfulness in statements, and attests to the difficulty of the LIAR dataset in general.

Moreover, the team found that adding a sentiment analysis dimension led to better model performance. It is possible that providing the sentiment classifications helped extract features of the text that were not captured by the embeddings. Unfortunately, input augmentation did not improve the models' performance, which could be due to the limitations of nlpaug as a data augmentation tool. Certain augmented sentences were nonsensical, and adopted synonyms changed the meaning of the sentence in a political context. This may have possibly diluted the underlying features that make a statement true or not. Similarly, the modified loss had little effect on model performance.

We have also seen that models in general have a lower performance on LIAR compared to other datasets such as Fake-News. In fact, previous literature achieved relatively high classification accuracies using traditional machine learning models on the Fake-News dataset, whereas even deep learning models were unable to attain similar accuracies for the LIAR dataset. This speaks to how polarity, tone, and word choice affect model performance on the task of fake news classification.

It would be interesting to test if data augmentation works with more polarized data. In addition, future work could explore methods for ensuring that data augmentation maintains the underlying truthfulness or meaning of the statements and use other datasets that contain statements outside the political sphere, to explore how the performance of such models generalize.

REFERENCES

Bahad, P., Saxena, P., Kamal, R. (2019). Fake News Detection using Bi-directional LSTM Recurrent Neural Network. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2020.01.072.

Davis, R. (2017). Fake News , Real Consequences : Recruiting Neural Networks for the Fight Against Fake News. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761239.pdf.

Ding, J., Hu, Y., & Chang, H. (2020). BERT-Based Mental Model, a Better Fake News Detector. In: Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence, pp 396–400

Feng, S.Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. A Survey of Data augmentation approaches for NLP. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp 968–988

Kaliyar, R.K., Goswami, A. & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications*, 2021. https://doi.org/10.1007/s11042-020-10183-2.

Nasir, J. A., Khan, O. S., Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights*. https://doi.org/10.1016/j.jjimei.2020.100007.

Riedel, B, Augenstein, I., Spithourakis, G. P., Riedel, S. (2018). A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. arXiv:1707.03264.

Thota, A., Tilak, P., Ahluwalia, S. & Lohia, N. (2018). Fake News Detection: A Deep Learning Approach. *SMU Data Science Review*, 1(3). https://scholar.smu.edu/datasciencereview/vol1/iss3/10

Wang, W. Y. (2017). Liar, liar pants on fire: A new benchmark dataset for fake news detection. In: Proceedings of the 55th annual meeting of the association for computational linguistics (vol 2: short Papers), pp 422–426

APPENDIX

TABLE 1
Architecture for FakeBERT Model (Glove Embeddings)

	INPUT SIZE	OUTPUT SIZE
Embedding	100	100 x 100
Conv1D, Kernel Size: 3, Stride: 1	100 x 100	98 x 128
Conv1D, Kernel Size: 4, Stride: 1	100 x 100	97 x 128
Conv1D, Kernel Size: 5, Stride: 1	100 x 100	96 x 128
Maxpool, Kernel Size: 5, Stride: 5	98 x 128	19 x 128
Maxpool, Kernel Size: 5, Stride: 5	97 x 128	19 x 128
Maxpool, Kernel Size: 5, Stride: 5	96 x 128	19 x 128
Concatenate	3 x 19 x 128	57 x 128
Conv1D, Kernel Size: 5, Stride: 1	57 x 128	53 x 128
Maxpool, Kernel Size: 5, Stride: 5	53 x 128	10 x 128
Conv1D, Kernel Size: 5, Stride: 1	10 x 128	6 x 128
Maxpool, Kernel Size: 5, Stride: 5	6 x 128	1 x 128
Flatten	1 x 128	128
Dense	128	32
Dense	32	3

TABLE 2
Architecture for FakeBERT Model (BERT Embeddings)

	INPUT SIZE	OUTPUT SIZE
Conv1D, Kernel Size: 3, Stride: 1	25 x 768	32 x 766
Conv1D, Kernel Size: 4, Stride: 1	25 x 768	32 x 765
Conv1D, Kernel Size: 5, Stride: 1	25 x 768	32 x 764
Maxpool, Kernel Size: 5, Stride: 5	32 x 766	32 x 153
Maxpool, Kernel Size: 5, Stride: 5	32 x 765	32 x 153
Maxpool, Kernel Size: 5, Stride: 5	32 x 764	32 x 152
Concatenate	(32 x 153), (32 x 153), (32 x 152)	32 x 458
Conv1D, Kernel Size: 5, Stride: 1	32 x 458	32 x 454
Maxpool, Kernel Size: 5, Stride: 5	32 x 454	32 x 90
Conv1D, Kernel Size: 5, Stride: 1	32 x 90	32 x 86
Maxpool, Kernel Size: 5, Stride: 5	32 x 86	32 x 17
Flatten	32 x 17	544
Dense	544	32
Dense	32	3

TABLE 3
Architecture for Dense Neural Net Model (TF-IDF Embeddings)

	INPUT SIZE	OUTPUT SIZE
Dense	11915 (Unigram), 80883 (Bigrams)	1024
Dropout (Probability: 0.2)		
Dense	1024	256
Dropout (Probability: 0.2)		
Dense	256	64
Dropout (Probability: 0.2)		
Dense	64	16
Dropout (Probability: 0.2)		
Dense	16	3
Dropout (Probability: 0.2)		