Fake News Identification Using Neural Language Models

Andrew L. Mackey
Computer Science and Computer Engineering
University of Arkansas
Fayetteville, Arkansas, USA
almackey@uark.edu

Susan Gauch
Computer Science and Computer Engineering
University of Arkansas
Fayetteville, Arkansas, USA
sgauch@uark.edu

Jacob Fuller
Computer Science and Computer Engineering
University of Arkansas
Fayetteville, Arkansas, USA
jrf022@uark.edu

Caden Williamson
Computer Science and Computer Engineering
University of Arkansas
Fayetteville, Arkansas, USA
cjw059@uark.edu

Kate Pearce
Bentonville High School
Bentonville, Arkansas, USA
kate.g.pearce@gmail.com

***Abstract*—Given the widespread use of social media and other online platforms for sources of new content, there has been an increased interest in the research community to improve existing methods for automating the detection of fake news content. We present several models for automating the detection of fake news content while utilizing current state-of-the-art neural language models. Our work provides an evaluation of the efficiency of different transformer-based neural language models for the fake news detection task. The evaluation shows that the proposed models are able to maintain high accuracy (98.5%) throughout experimentation tasks. We conclude by discussing the effects of the different neural language models.**

***Index Terms*—Fake news classification; misinformation; neural language models; natural language processing**

I. Introduction

A growing number of users obtain news from social media platforms and other online sources. According to results from the Pew Research Center, approximately 68% of American adults obtain their information from social media platforms occasionally [23]. There are multiple reasons for why many users prefer these methods of delivery, among which we will find ease of access and discussion-oriented interactions as possible reasons. The growing dependence of these platforms as sources for news has prompted many organizations to take steps increase the reliability of various postings on their sites.

The low barriers to entry with digital platforms facilitate the introduction of fake, deceptive, misleading, or malicious news content to users with relative ease with widespread societal impacts if left unabated. In one case, a single malicious posting to a social media site caused a significant fluctuation in stock market activity [27]. Another major implication has been in election interference where authors post blatantly falsified information to favor a particular political candidate [28]. It should also be noted that not all news information has sinister underlying motivations. Some sources of online news are published with the intent of misleading the reader for entertainment purposes. The consequence of this scenario is some readers may be inadvertently misled to believe it is real.

The task of classifying fake news is often not a binary decision due to the varying degrees in correctness and underlying motivation or intent of deception. For example, a single statement can be factually incorrect, or otherwise inaccurate, thus prompting a discussion for how the document should be classified. Granular-level techniques could be employed to evaluate the truthfulness of statements on a scale rather than using binary levels. An example of this can be found in how PolitiFact ranks comments on a scale [9]. There are many challenges in this task given how minor changes in wording can lead to differences in how correct a statement is.

In this work, we propose a model that leverages state-of-the-art neural language models to automate the detection of fake news to curtail the spread of misinformation. We present an analytic study of advancements in neural language models and their impact on the fake news detection task. Furthermore, we provide empirical evidence that demonstrates how our proposed model leads to improvements in the fake news detection performance. We discuss the efficiency of our proposed model relative to other models with similar goals.

In Section 2, we begin by reviewing previous work conducted over fake news, neural networks, and neural language models. Section 3 presents information over the fake news detection task, the data set used, and our proposed models. Section 4 details the experiments and results described in this work. Section 5 concludes by discussing the significance of

 ISBN: 978-1-61208-986-7

the experiments.

II. Related Work

In the following, we consider previous work in the areas of fake news detection and word embeddings.

A. Fake News Detection

Fake news has been an active research area for the past several years across multiple disciplines. We first start by defining fake news as content whereby the authors intend to deceive, or otherwise mislead, readers. Content labeled as fake news can be further divided in different categories: satire, hoax, propaganda, and clickbait. Satirical work is fake news content with some purpose of entertainment, generally through sarcasm or other misinformation [9]. Hoaxes are content whereby the author passes deceptive content as truthful, also with the goal of deception for humor. Propaganda are deceptive content with the goal of causing harm to a specific entity or party. Clickbait is content that attracts users through misleading content.

Fake news detection has a variety of tasks that have been reviewed, such as rumor detection [25], spam detection [26], and emotion analysis of news articles [20]. Fake news detection can also involve text, images, or a combination of both. Automated fake news detection methods usually target biases in the linguistic style of writing and word usage [9]. This typically involves the content of the article, user reactions or responses, or source fo the article [6].

Some authors have also studied the effects of social media, disinformation, and political polarization with public policy-making and quality of democracies [22]. Social media has enabled misleading and/or fake news content to propagate throughout social networks with limited restrictions [10]. It has also been demonstrated the users can have problems differentiating between real and fake content [1] [3]. Furthermore, it may also be difficult to label an article as being real or fake due to nuances in the writing [8].

B. Word Embeddings

Word embeddings [29] are neural language models where words are represented as continuous vector-representations in a dimensional space that is typically reduced in size in comparison to other techniques, such as bag of words methods where feature vectors are often of $|V|$ width, or the size of the lexicon. In the work presented in [21], the authors introduced the Continuous Bag-Of-Words (CBOW) and continuous Skip-gram models as methods for learning distributed vector representations that reflect both syntactic and semantic relationships between words in a language. The CBOW model seeks to predict a word based on the context words. The Skip-gram model was developed to find different word representations that can be used for establishing adjacent words for a given document by maximizing the average log probability where c denotes the size of the window and w_t represents the centralized word:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j\leq c, j\neq 0}\log \Pr(w_{t+j} \mid w_t) \tag{1}$$

One major problem identified in the work for the full softmax is its lack of efficiency. The authors in Mikolov et al. [29] thus proposed other techniques that are computationally efficient approximations to the full softmax, such as negative sampling, Noise Contrastive Estimation (NCE), negative sampling, and subsampling.

In subsequent work, the authors in Pennington et al. [5] presented improvements to word embedding representations by constructing a global log-bilinear regression model that combined global matrix factorization and local context windowing techniques. The work from [19] presented bidirectional language models computed over the entire input sentence while jointly maximizing the log likelihood of both the forward and backward directions:

$$\Pr(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} \Pr(t_k \mid t_1, t_2, ..., t_{k-1}) \tag{2}$$

$$\Pr(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} \Pr(t_k \mid t_{k+1}, t_{k+2}, ..., t_N) \tag{3}$$

$$\tag{4}$$

The log likelihood of the forward and backward passes is defined by the following where Θ_x is the token representation parameters, Θ_s is the softmax layer parameters, and $\overrightarrow{\Theta}_{\text{LSTM}}$ and $\overleftarrow{\Theta}_{\text{LSTM}}$ are the parameters for the long short-term (LSTM) layers for the forward and backward directions:

$$\sum_{k=1}^{N}\Big[\log\big(\Pr(t_k \mid t_1, ..., t_{k-1}\ ;\ \Theta_x, \overrightarrow{\Theta}_{\text{LSTM}}, \Theta_s\big) \tag{5}$$

$$+\log\big(\Pr(t_k \mid t_{k+1}, ..., t_N\ ;\ \Theta_x, \overleftarrow{\Theta}_{\text{LSTM}}, \Theta_s\big)\Big] \tag{6}$$

Authors from Devlin et al. [13] presented the Bidirectional Encoder Representations from Transformers (BERT) where their approach focuses on bidirectional pre-training while achieving a fine-tuned representation model to reduce or eliminate dependencies on task-specific architectures. The authors also contend that their work improves the work from [19] as prior work concatenated independently trained forward and backward language models whereas BERT implemented deep bidirectional representations.

DeBERTa, which improves upon BERT and RoBERTa models, uses a disentangled attention mechanism [16]. This allows each word to be represented by utilizing two vectors that encode both the content and position. The attention parameters for the tokens are calculated by using disentangled matrices for both the content and relative positions. In addition, DeBERTa incorporates an enhancement to the mask decoder as a substitute for the output softmax layer that allows it to predict

 ISBN: 978-1-61208-986-7

the masked tokens for the purpose of pretraining. XLNet, an extension of the Transformer-XL model, is an autoregressive method that learns bidirectional contexts through the maximization of expected likelihood over all permutations of the input sequence factorization order [15]. GPT-J is a neural language model with 6 billion parameters that is an open-source alternative to the GPT-3 model [17] [18].

C. Recurrent Neural Networks

LSTM and Gated Recurrent Unit (GRU) networks are both forms of recurrent neural networks that are capable of processing temporal sequences of data. Both networks are capable of managing the vanishing gradient problem that is commonly found when processing long sequences in traditional recurrent neural networks.

An LSTM cell contains three major gates that attempt to control how information passes into and out of the cell: input gate i, output gate o, and a forget gate f. Let t represent the time step of a sequence such that $t \in [0...\tau]$. The input gate $\mathbf{i}_t$ at time step t determines what information is relevant and can be added from the previous hidden state $\mathbf{h}_{t-1}$ and the current input $\mathbf{x}_t$. The forget gate $\mathbf{f}_t$ is used to decide which information will be be utilized and which information will be forgotten (or ignored) at time step t. Lastly, the output gate $\mathbf{o}_t$ is used for determining the values of the next hidden state $\mathbf{h}_t$. An LSTM maintains cell state for both the short-term and long-term. The short-term is denoted as h and the long-term is denoted as c, noting that $\mathbf{c}_{t=0} = \begin{bmatrix}0 & 0 & \cdots & 0\end{bmatrix}$ and $\mathbf{h}_{t=0} = \begin{bmatrix}0 & 0 & \cdots & 0\end{bmatrix}$.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \tag{7}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \tag{8}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \tag{9}$$

$$\widetilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \tag{10}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \widetilde{\mathbf{c}}_t \tag{11}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{12}$$

The two activation functions used are the sigmoid activation function $\sigma(x)$ and hyperbolic tangent function $\tanh(x)$. The range of values for both activation functions differ with $\sigma(x) \in [0, 1]$ and $\tanh(x) \in [-1, 1]$. The operator $\odot$ denotes the Hadamard product. The input vector $\mathbf{x}_t$ is defined as $\mathbf{x}_t \in \mathbb{R}^d$ where d denotes the number of input features. The hidden state vector $\mathbf{h}_t$ is defined as $\mathbf{h}_t \in (-1, 1)^h$.

GRU cells are similar to LSTM cells, but with simplifications while maintaining comparable performance. There are several notable differences between GRU and LSTM cells. First, an LSTM maintains two state vectors for c and h; a GRU cell maintains a single vector h for state. In addition, a GRU maintains an update gate $\mathbf{z}_t$ that controls for both the forget gate and input gate, and a reset gate $\mathbf{r}_t$ which is responsible for the short-term memory $\mathbf{h}_t$.

Recurrent layers generally look at the previous and current time steps to produce an output without any intuition about future time steps. In some situations, it may be advantageous to gather context for inputs at a given time step. Consider the situation of word embeddings. A token at time step t may require information from previous and future time steps. A simple solution to this is to implement a *bidirectional recurrent layer* which implements two recurrent layers and finally combine the outputs of each layer at each time step t (generally this is performed through concatenation): one recurrent layer that iterates in the order of $\{t = 0, t = 1, t = 2, ..., t = n\}$, and another recurrent layer that iterates in the reverse order of $\{t = n, t = n-1, t = n-2, ..., t = 1\}$. Bidirectional recurrent layers can be applied to standard recurrent units, LSTM units, or GRUs.

D. Attention

Attention mechanisms were introduced by [24] for the task of neural machine translation. Their work extended the basic encoder-decoder models and allowed for decoders to have an attention mechanism so encoders are not required to encode all information into fixed-length vectors. This attention layer, or *alignment model*, is capable of focusing on features at each time step that are important, which is trained jointly with the encoder-decoder model. The following conditional probability was proposed where s_i represents the hidden state of the RNN for time step i, y_i represent the target word, and c_i represents the context vector:

$$p(y_i \mid y_1, ..., y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \tag{13}$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{14}$$

The context vector c_i is computed from a the sequence of annotations h_i as $\langle h_1, h_2, \cdots, h_{T_x} \rangle$ which is produced by an encoder from a given input sentence. The context vector c_i is constructed as a weighted sum between the annotations h_i and the weights of annotation α_{ij}, where e_{ij} represents the alignment model with scores that measure how well the output aligns with the previous hidden state of the decoder (where a represents a feedforward neural network):

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \qquad \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{15}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{16}$$

It is important to note that a is jointly trained with the model. In addition, the sum of all weights α_{ij} for a given time step will have a sum of 1. Other work has been proposed to use multiplicative attention with other simplifications that improve on concatenative attention.

 ISBN: 978-1-61208-986-7

III. Fake News Detection

A. Dataset

In our experiments, we used the publicly available dataset from [9]. The dataset contains different news articles obtained from seven different fake, or otherwise unreliable, news sites, including The Onion, The Borowitz Report, Clickhole, American News, DC Gazette, The Natural News, and Activist Report. Each of the news articles are labeled as being *satire*, *hoax*, *propaganda*, or *trusted*. The trusted news articles were obtained from [11]. In the trusted news articles data source, the authors constructed an approach to building a supervised reading comprehension dataset with news articles obtained from convolutional neural networks ($n = 90,266$). Our experiments extracted $n = 10,000$ randomly selected news articles from the set of possible $90,266$ different articles available in the CNN dataset.

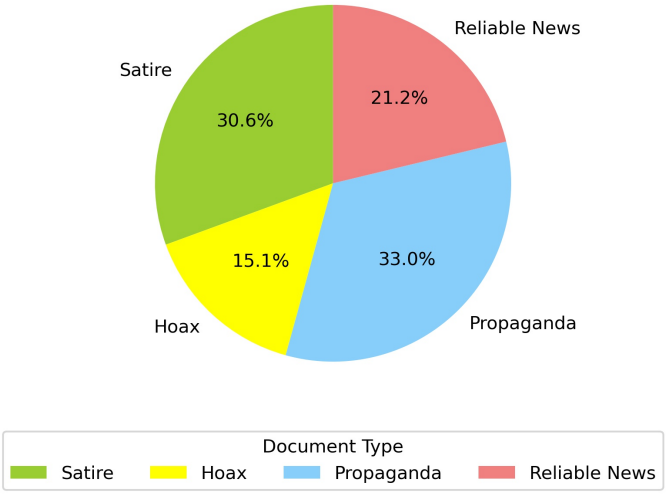

Fig. 1. The distribution of the dataset used for classification is presented by the news article type.

TABLE I
NEWS ARTICLES WITH NUMBER OF DOCUMENTS, AVERAGE DOCUMENT LENGTHS, AND MEDIAN DOCUMENT LENGTHS

Doc. Type	# of Docs	Avg. Tokens	Med. Tokens
Satire	$13,942$	206 ± 177	105
Hoax	$6,892$	141 ± 122	109
Propaganda	$15,061$	587 ± 808	458
Trusted	$9,681$	428 ± 205	401

Table I summarizes the type of news articles, document frequencies, mean document lengths and standard deviations, and median document lengths. News articles from the propaganda class have a higher average number of tokens than other classes. When considering the robustness of the statistical measures to control for outliers, the median of the propaganda class is marginally higher than the trusted class. All data is preprocessed using standard natural language preprocessing techniques, including downcasing, stopword removal, tokenization, etc. We utilize the NLTK toolkit [1] for computational linguistic analysis. The overall distribution of the data can be seen in Figure I.

B. Models

Our experiments evaluate different neural network models and word embeddings for the fake news detection task. We construct five baseline models that are developed with an embedding layer, which is a trainable dense vector that can be used to represent each unique word in the lexicon. The first model contains a single embedding layer, two LSTM layers, an attention layer, and a classification segment comprised of two dense layers, and an output layer with a softmax activation function having a number of neurons corresponding to the number of output classes. Dropout is also added to mitigate the possible situation of overfitting. Attention layers are used to determine which parts of the data have greater importance than other parts through a separate alignment model. This alignment model is trained jointly with the other parts of the network.

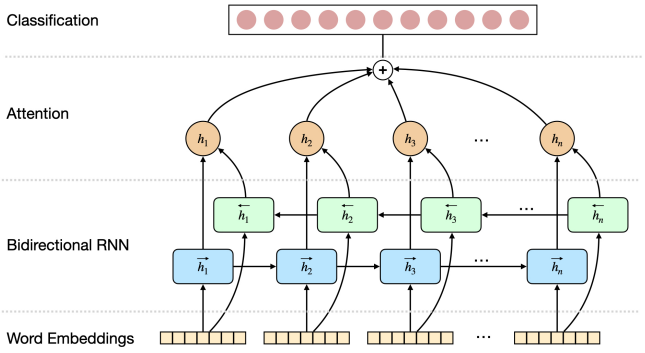

Fig. 2. The proposed recurrent neural network model for our experiments uses a bidirectional network with an attention layer.

The next several models are constructed from the original model with modifications to the sequence processing layers. The second model uses bidirectional LSTM) layers so that both the forward and backward sequences are concatenated and learned during the training phase (see Figure 2). The third model considers the use of GRU in place of LSTM layers. The forth model is constructed from bidirectional GRU layers.

The final model is constructed by using a convolutional neural network (CNN) on the sequences of data. We implement a stack of $1D$-convolutional layers, batch normalization, and ReLU activation layers. The batch normalization layer centers and scales the activation vectors from the hidden layers of the current batch by using the mean and variance. We implement dropout in the model to prevent overfitting. A global max pooling layer is added, which returns the maximum value

[1] https://www.nltk.org/

 ISBN: 978-1-61208-986-7

for each feature channel. Finally, a set of fully connected layers are added followed by a layer with a softmax activation function with the number of units corresponding to the number of $\hat{\mathbf{y}}$ target classes such that $\hat{y} \in \{\text{fake}, \text{real}\}$.

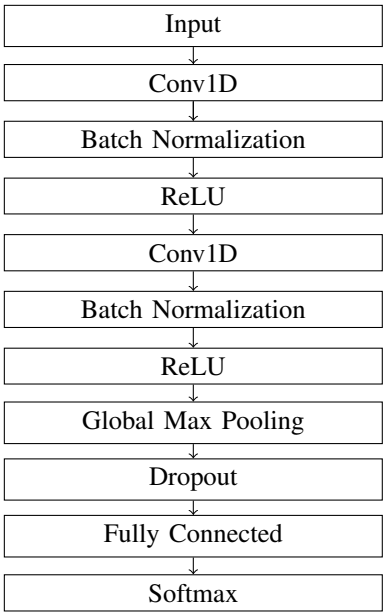

Fig. 3. 1-D convolutional neural network (CNN) model

Each of the documents were converted to sequences of word embeddings. For our baseline model, we assigned a unique integer to each token and implemented a standard word embedding layer with 128 dimensions. To evaluate the performance of various word embedding methods, we compare our baseline model to word embeddings produced by BERT, DeBERTa, XLNet, and GPT-J. The BERT embeddings contained a total of $L = 12$ hidden layers with a hidden size of $H = 128$ and $A = 2$ attention heads. We used the V2-XLarge DeBERTa pre-trained model with a $128,000$ vocabulary, 710 million parameters, hidden size of $H = 1536$, and $L = 24$ hidden layers. XLNet contains $L = 12$ hidden layers, hidden size of $H = 768$, and $A = 12$ attention heads.

For all models, the batch size was set to 64 and we implemented early stopping criteria to limit potential overfitting. We utilize the Adam optimization algorithm and a categorical cross-entropy loss function for this multi-class classification task. A learning rate of 0.001 was used. Each experiment was conducted from training, testing, and validation splits of sizes 0.7, 0.2, and 0.1, respectively. For each of the performance metrics reported, the mean of each experiment is reported from 10 random shuffles of the data.

In our experimentation tasks, we evaluated multiple neural network models, including LSTM, GRU, bidirectional recurrent neural networks (both LSTM and GRU), and 1D convolutional neural networks. Each document is represented in the training set as a set of fixed-length word embeddings formed by utilizing one of the techniques described previously. To ensure that each document contains the same number of features, we padded all documents that where the number of tokens is less than specified sequence length ($|d| < we_{\text{size}}$). Similarly, any document that exceeds the word embedding length ($|d| > we_{\text{size}}$) is right-truncated.

IV. Evaluation

Having presented models for the task of identifying fake news, we present our evaluations of the models using the data described in earlier sections. Our hypothesis is that the fake news detection task can be improved by leveraging state-of-the-art neural language model representations and improved neural network architectures.

Our first task is to establish which neural language model representations work best for the task of automating fake news detection. For this requirement, we used the model architecture as presented previously with LSTM layers, an attention layer, and a classification segment. We compared embedding layers, BERT, XLNet, DeBERTA, and GPT-J for each of the documents in the training and testing sets. Based on our experiments, the BERT neural language model outperforms the other methods with a mean accuracy of 98.1%. While other methods have additional model parameters, attention mechanisms, and other improvements to their models, they do not seem to outperform BERT for this specific task. In addition, the training time for BERT was faster in the amount of training time required when compared to the others. We conclude that BERT achieves both the best results and fastest training time.

TABLE II
Comparison of Neural Language Models

Method	Accuracy	Precision	Recall	F1
WE+LSTM	0.978	0.979	0.977	0.978
BERT+LSTM	**0.981**	**0.981**	**0.981**	**0.981**
XLNet+LSTM	0.956	0.956	0.956	0.956
DeBERTa+LSTM	0.960	0.960	0.959	0.960
GPTJ+LSTM	0.971	0.971	0.971	0.971

In our next experiments, we present our evaluation of the neural network architectures as presented in earlier with the top performing neural language model from the aforementioned experiments. The work presented in [20] achieved a 76.3% accuracy using a feed forward neural network architecture with BERT for document-level embeddings. We establish this as a baseline model for our experiments. Our experimental design evaluates the performance of documents converted to sequences from the neural language models as input to the neural network models that carry out the fake news classification. Previous work emphasized document-level embeddings whereas we focus on sequences of embeddings from transformer architectures.

The results presented in Table III demonstrate the accuracy for each of the models using the BERT neural language model.

 ISBN: 978-1-61208-986-7

TABLE III
COMPARISON OF BASELINE METHODS WITH BERT AND NEURAL NETWORK MODELS

Type	Method	Accuracy	Precision	Recall	F1
BASELINE	BERT+NN	0.763	0.798	0.721	0.757
MODELS	BERT+LSTM	0.981	0.981	0.981	0.981
	BERT+BiLSTM	**0.985**	**0.985**	**0.985**	**0.985**
	BERT+GRU	0.982	0.982	0.982	0.982
	BERT+BiGRU	0.983	0.983	0.983	0.984
	BERT+ConvNet	0.972	0.972	0.972	0.972

The BERT+BiLSTM model achieved the highest accuracy and was constructed by using bidirectional LSTM layers. It should also be noted that the BERT+BiGRU model is relatively comparable to the top performing model while using layers with fewer gates. The BERT+ConvNet model achieved a mean accuracy slightly lower than the LSTM and GRU models. All experimental models outperform the baseline models we defined earlier, which highlights the benefits of using sequences of inputs from neural language models for the fake news detection task.

TABLE IV
EVALUATION OF NEURAL LANGUAGE MODELS AND NETWORK ARCHITECTURES

Model	BERT	XLNet	DeBERTa	GPTJ	Average
LSTM	0.981	0.956	0.960	0.971	0.967
BiLSTM	**0.985**	0.955	0.974	0.970	0.971
GRU	0.982	0.962	0.973	0.973	**0.973**
BiGRU	0.983	0.955	0.962	0.971	0.968
ConvNet	0.972	0.947	0.958	0.959	0.959
Average	**0.981**	0.955	0.965	0.969	0.967

Our final task is to compare the mean performance of model architectures and neural language models. Table IV provides the mean accuracy for each neural network architecture and neural language model. As previously mentioned, the top performing neural language model for our experiments was BERT. However, the top performing model, on average, was obtained from the model that leverages layers using the GRU, which is marginally higher than BiLSTM.

The proposed BERT+BiLSTM model using sequences as input was able to achieve a 22.2% increase over the baseline BERT+NN model, which leverages document-level neural language model outputs as inputs to the model. Similarly, leveraging GRU layers with all neural language models achieves a 21% improvement while requiring less parameters than the BiLSTM layer. The mean performance for all proposed models using BERT and sequences as input achieved 98% accuracy, which is an improvement of 21.8%. Finally, leveraging sequences as inputs for all propose architectures and neural language models had a mean score of 96.7%, which was a 20.4% increase over the baseline.

The experiments presented here demonstrate the ability of recurrent neural networks when combined with state-of-the-art word embeddings to facilitate the classification of fake news. Our results also demonstrate that the additional training time and overhead required for some word embeddings do not necessarily yield better classification results for the fake news classification task as indicated in our experiments. Furthermore, more complex architectures for neural language models may improve semantic and syntactic understandings or relationships between words, but additional training data may be necessary to fully exploit these capabilities.

V. CONCLUSION AND FUTURE WORK

We presented a comparative analysis of various state-of-the-art methods for neural language models and neural network architectures for the fake news detection task. Our proposed BiLSTM+BERT model was able to achieve a 98.5% accuracy, which is an improvement over the baseline model. This demonstrates the effectiveness of bidirectional LSTM layers when combined with BERT for automating the classification of fake news articles. Given that the research continues to improve neural language models, future work will need to evaluate improvements in this space to determine how we can efficiently represent documents with the same model performance or improve upon the current model performance.

REFERENCES

[1] M. Barthel, A. Mitchel, and J. Holcomb, Many americans believe fake news is sowing confusion, *Pew Research Center*, 2016.
[2] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, 12(ARTICLE): pp. 2493–2537, 2011.
[3] C. Domonoske, Students have 'dismaying' inability to tell fake news from real, study finds, *National Public Radio*, 23, 2016.
[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*, 2013.
[5] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

 ISBN: 978-1-61208-986-7

[6] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 797–806, 2017.
[7] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "defend: Explainable fake news detection," In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 395–405, 2019.
[8] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, 19(1): pp. 22–36, 2017.
[9] H. Rashkin, E. Choi, J. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: analyzing language in fake news and political fact-checking," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2931–2937, 2017.
[10] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," *Proceedings of the 20th international conference on World wide web*, pp. 675–684, 2011.
[11] K. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, Teaching machines to read and comprehend, 2015.
[12] Q. Le and T. Mikolov, Distributed representations of sentences and documents, 2014.
[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint* 2018.
[14] X. Zhou and R. Zafarani, A survey of fake news: fundamental theories, detection methods, and opportunities, *ACM Comput. Surv* 2020.
[15] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, arXiv:1906.08237, 2019.
[16] P. He, Z. Liu, J. Gao, and W. Chen, DeBERTa: Decoding-enhanced BERT with Disentangled Attention, arXiv:2006.03654, 2020.
[17] T. Brown et al, Language Models are Few-Shot Learners, arXiv:2005.14165, 2020,
[18] B. Wang and A. Komatsuzaki, GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model, https://github.com/kingoflolz/mesh-transformer-jax, 2021.
[19] M. Peters et al, Deep contextualized word representations, *arXiv* 2014.
[20] A. Mackey, S. Gauch, and K. Labille, "Detecting Fake News Through Emotion Analysis," *eKNOW* 2021.
[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* 2013.
[22] J. Tucker, et al, "Social media, political polarization, and political disinformation: A review of the scientific literature," *Political polarization, and political disinformation: a review of the scientific literature (March 19, 2018)* 2018.
[23] E. Shearer, More than eight-in-ten Americans get news from Digital Devices, *Pew Research Center*, 2021.
[24] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473* 2014.
[25] Z. Jin, J. Cao, Y. Jiang, and Y. Zhang, "News credibility evaluation on microblog with a hierarchical propagation model," *2014 IEEE international Conference on Data Mining*, pp. 230–239,2014.
[26] H. Shen, F. Ma, X. Zhang, L. Zong, X. Liu, and W. Liang, "Discovering social spammers from multiple views," *Neurocomputing*, pp. 49–57,2017.
[27] C. Matthews. How does one fake tweet cause a stock market crash?, *Time*, 2013.
[28] H. Parkinson. Click and elect: How fake news helped Donald Trump win a real election, *The Guardian*, 2016.
[29] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of sentences and documents," *Advances in neural information processing systems*, 2013.

 ISBN: 978-1-61208-986-7

TABLE V
FAKE NEWS CLASSIFICATION METHODS WITH EACH OF THE PROPOSED MODELS

Type	Method	Accuracy	Precision	Recall	F1
BASELINE	WE+LSTM	0.978	0.979	0.977	0.978
	WE+BiLSTM	0.979	0.979	0.979	0.979
	WE+GRU	0.977	0.977	0.977	0.977
	WE+BiGRU	0.976	0.976	0.976	0.977
	WE+ConvNet	0.966	0.968	0.965	0.967
LSTM	BERT+LSTM	0.981	0.981	0.981	0.981
	XLNet+LSTM	0.956	0.956	0.956	0.956
	DeBERTa+LSTM	0.960	0.960	0.959	0.960
	GPTJ+LSTM	0.971	0.971	0.971	0.971
BILSTM	BERT+BiLSTM	0.985	0.985	0.985	0.985
	XLNet+BiLSTM	0.955	0.955	0.955	0.955
	DeBERTa+BiLSTM	0.974	0.974	0.973	0.974
	GPTJ+BiLSTM	0.970	0.970	0.969	0.969
GRU	BERT+GRU	0.982	0.982	0.982	0.982
	XLNet+GRU	0.962	0.963	0.962	0.963
	DeBERTa+GRU	0.973	0.973	0.972	0.972
	GPTJ+GRU	0.973	0.973	0.973	0.973
BIGRU	BERT+BiGRU	0.983	0.983	0.983	0.984
	XLNet+BiGRU	0.955	0.956	0.955	0.956
	DeBERTa+BiGRU	0.962	0.963	0.962	0.962
	GPTJ+BiGRU	0.971	0.971	0.971	0.971
CONVNET	BERT+ConvNet	0.972	0.972	0.972	0.972
	XLNet+ConvNet	0.947	0.948	0.946	0.947
	DeBERTa+ConvNet	0.958	0.960	0.957	0.959
	GPTJ+ConvNet	0.959	0.959	0.959	0.959
AVERAGE		0.969	0.970	0.969	0.970

 ISBN: 978-1-61208-986-7