Analysis of Text Feature Extractors using Deep Learning on Fake News

Bilal Ahmed
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
bilal_ahmed@iba-suk.edu.pk

Gulsher Ali
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
gulsher@iba-suk.edu.pk

Arif Hussain
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
arif.hussain@iba-suk.edu.pk

Abdul Baseer Buriro
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
abdul.baseer@iba-suk.edu.pk

Junaid Ahmed
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
j.bhatti@iba-suk.edu.pk

Abstract-Social media and easy internet access have allowed the instant sharing of news, ideas, and information on a global scale. However, rapid spread and instant access to information/news can also enable rumors or fake news to spread very easily and rapidly. In order to monitor and minimize the spread of fake news in the digital community, fake news detection using Natural Language Processing (NLP) has attracted significant attention. In NLP, different text feature extractors and word embeddings are used to process the text data. The aim of this paper is to analyze the performance of a fake news detection model based on neural networks using 3 feature extractors: TD-IDF vectorizer, Glove embeddings, and BERT embeddings. For the evaluation, multiple metrics, namely accuracy, precision, F1, recall, AUC ROC, and AUC PR were computed for each feature extractor. All the transformation techniques were fed to the deep learning model. It was found that BERT embeddings for text transformation delivered the best performance. TD-IDF has been performed far better than Glove and competed the BERT as well at some stages.

Keywords-fake news; natural language processing; feature extractors; deep learning

I. Introduction

Due to the easy and excessive use of the internet and the incremented use of social media, the probability of fake news circulation is increased. This has impacted the trust on news from the media nowadays specially since the 2016 US elections. Authors in [1] conducted a survey in which they found that global trust on the news from the media of different countries ranged from 23% to 62%. The big challenge for the researchers is to encounter this problem and provide a feasible solution. The main aspects of this problem are the fact that the same sources may provide both fake and real news, the language used can be deceiving, and biasness of the dataset and machine learning models may occur. Also, due to advancement of Artificial Intelligence applications in natural language generation has brought considerable negative impact when used in generating fake news.

In [2], news sources were under consideration instead of single articles. The idea is based on the frequency of the news sources providing fake news. The more the frequency of fake news from a source, the more the chances are that that source will provide more fake news in the future. The motive of the authors was to build an algorithm that can identify the fake news in its source before it spread. In their paper they targeted multiple sources such as URLs, twitter accounts, Wikipedia pages and articles, and found articles to be more real than the other kind of sources. Automatic text generation using Artificial Intelligence is also a popular way to spread fake news at a fast pace [3].

Fake news is one of the most difficult and sensitive topics in the field of NLP. When dealing with fake news one must keep tricky things in mind like the source of the news, the language of the news, and its pattern. Different types of text transformation techniques, machine learning algorithms and state-of-the-art methods have been introduced and applied to address this problem. TD-IDF is one of the simplest text transformation techniques which transforms each word to a

Corresponding author: Bilal Ahmed

float number as its weight according to the frequency of words in documents. This simple technique is very useful when working on a simple task. But when encountering a tricky topic like fake news detection, TD-IDF can be less effective. To encounter the complexity of text-based data in machine learning, text transformation techniques, also called as embeddings, are developed which deal with the phenomenon of the relation of words with respect to their meanings. However, TD-IDF was used before the contextual embeddings came out and became common. Authors in [4] used TD-IDF vectorizer to extract features from news articles. Authors in [5] proposed a tool for fake news detection in which they used bag of words, bigram frequency, and TD-IDF vectorizer to extract features from news articles which were tested with probabilistic classification and linear classification. Authors in [6] analyzed different machine learning models including Naïve Bayes, Support Vector Machine (SVM), Logistic Regression and Recurrent Neural Networks (RNN) on a fake news dataset from Twitter. They concluded that the Naïve Bayes and SVM are the classifiers with the best performance. Ensemble or combination of machine learning models is a technique used by researchers to deal with complex machine learning tasks. Authors in [7] analyzed different ensembles of different machine learning models and finally came with the an ensemble of Decision Tree, Logistic Regression, Bagging Classifier used with hard-voting ensemble technique which gave accuracy of 88%. For other NLP tasks, ensemble techniques give nice results. Authors in [8] used the ensemble technique to improve the translation quality from English to Hindi and used 6 different machine translation engines.

For NLP problems, neural networks provide a great range of algorithms to process and learn on sequence-based or textual data. Dense, RNN, LSTM, 1D-Convolutional (Conv1d) and GRU layers are being used in processing of textual or sequence-based data. Authors in [9] used convolutional neural networks to build a model that can classify the Arabic text. Authors in [10] used convolutional layers and bidirectional RNNs on large movie review and sentiment treebank datasets. Embeddings do not consist a feasible solution for every text-based problem, as there are two kinds of texts, static and context-based and so there should be two kinds of embeddings. Thus, the contextual embedding has been developed which works well on the text data having context with respect to every word. This paper discusses and analyzes different word transformation algorithms on fake news datasets using deep learning for each word transformation technique. It covers some well-known word transformation techniques such as BERT [11], Glove [12] and TD-IDF [13]. The main objective of this paper is to analyze whether contextual embeddings outperform the static word transformation techniques or not.

II. THE DATASETS

Two balanced datasets openly available on Kaggle were used. Both datasets contain both true and false news about western politics and international issues. The first dataset [14] contains 6335 articles about different topics mostly on politics and has 3164 fake and 3171 true articles and the second [15] contains 20,800 articles of the same nature with 10,387 fake articles and 10,413 valid articles (Figure 1). In both datasets, article text was only analyzed and the other columns were dropped to avoid probability of false pattern learning and to reduce computational cost while training the model. For both datasets, train/test ratio was 80/20 and from the train data, 10% was kept for validation to monitor the model behavior during training.

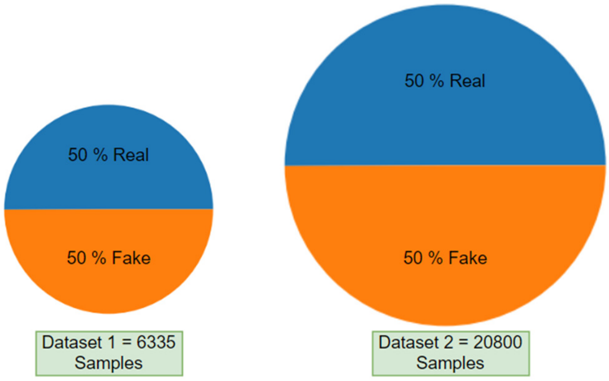

Fig. 1. Pictorial presentation of the used datasets

III. METHODOLOGY AND FEATURE EXTRACTORS

The objective of this study is to analyze the performance of 3 feature extractors, BERT embedddings, Glove embeddings and TD-IDF vectorizer using ANNs on two fake news datasets. The ANN model contains two dense hidden layers and an output layer with 8, 16 and 1 neuron(s) respectively. The feature extractors were chosen because they cover all broader classes of text feature extractors. The old fashioned TD-IDF vectorizer computes the word count (frequency) of a word, Glove is a static embedding context-independent method which works on the principle of computing the similarity between words according to their semantics, and BERT is a contextual embedding model which does not just compute the similarity between words but takes care of the context in which a specific word is used, since the word's meaning may be varied according to the context. ANNs were used as classifiers due to their ability to handle efficiently large datasets in comparison with other machine learning models. Other ANN types (LSTM, RNN, etc.) were not used, since, in this paper, the analysis of feature extractor has not been conducted on the basis of different ANN models. Also, a simple ANN requires less computational time. The BERT model was itself quite heavy. For extracting features of our dataset from Glove and TD-IDF, a personal I7 laptop with 8GB RAM was used. Feature extraction and model training took hardly 2-3 minutes for TD-IDF and 4-5 minutes for Glove. However, for BERT the required computational power surpassed the available resources, as feature extraction requires very high computational power. For BERT, the cloud-based Kaggle platform was utilized, which offers free GPU and TPU usage for higher computational tasks. For this task 15 GB GPU and 16 GB RAM were used to extract the features and pass to the ANN classification model which took about 28 minutes in total for each dataset.

This study was carried out using Python language and its available tools. The datasets were loaded from Kaggle, then they were split into train and test groups with 80/20 ratio. The feature extractors were applied on each dataset and the output was fed to the ANN model. We used Sklearn library to implement the TD-IDF vectorizer and Keras for the implementation of the ANN model having TensorFlow at the backend. We used built-in functions in Sklearn for the evaluation of the models trained on each feature extractor for both datasets. Accuracy, precision, recall, F1, AUC ROC and AUC PR score were computed for both datasets using all three feature extractors to compare their performances. The flowchart of the process from dataset loading to the final evaluation is given in Figure 2.

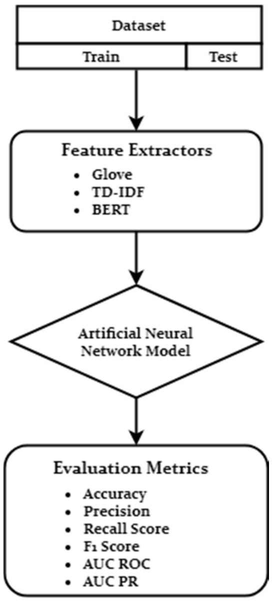

Fig. 2. Flowchart of ANN model analysis for 3 feature extractors.

A. TD-IDF Vectorizer

TD-IDF vectorizer is one of the simplest techniques to transform text into numerical values which can be fed to a machine learning model for processing. It statistically computes and finds the relevancy of a word from a document in other documents. It is computed by multiplication of two metrics: how often a word appears in a document and how rare it appears in other documents. TD-IDF vectorizer has nothing to do with the similarity between words because it is not an embedding. It is very commonly used as feature extractor for various NLP tasks [16, 17].

B. Glove Embeddings

Glove embedding is an unsupervised model for word representation in the form of vectors. These embeddings are achieved by mapping words to a meaningful space in which the distance between words is related to their semantic meaning. Cosine similarity and Euclidean distance are used in Glove embeddings to compute the distance between words. Glove comes with the advantage that it does not just depend on local context (surrounding) information of words but on the global co-occurrence of words in a given corpus by creating a co-occurrence matrix of words in a given corpus unlike Word2Vec which relies on local contextual (surrounding) word information. Glove embeddings has been used in many text problems [18, 19]. Embedding comes in some versions with respect to the size of tokens used. We have used a pre-trained Glove model with 6 billion tokens, each of 300-dimensional vector size.

C. BERT Embeddings

Contextual embeddings [20, 21] differ from static embeddings like Word2Vec [22] and Glove. These embeddings do not just compute the similarity between words which similarity in their semantics, but also they compare the context as well in which the words are used. They are more efficient on contextual problems like sentiment analysis, sentence classification, text summarization, etc.. An interesting thing about BERT is that it does not just compute words or token embeddings, but also sentence embeddings to differentiate the sentences and positional embedding of the word in a given sequence. These combined embeddings can clearly help context each word in a given corpus. Another interesting thing in the development of the BERT model is the use of the concept of masked language modeling. This means that they hid 15% of words and used their positional embeddings to address or infer them to make the learning more effective. As a result, the BERT model outperformed all state-of-the-art existing language models even before its convergence. Authors in [23] used a BERT model with Bayesian Network to classify text data of people's livelihood governance. The BERT model comes in various versions with respect to the number of layers, heads, hidden units, cased, uncased and for languages other than English. We used the BERT Base Uncased model with 12 layers with 768 hidden units, 12 attention heads which has 110 million learning parameters.

IV. CLASSIFICATION MODEL AND EXPERIMENTS

Features from the split data into train and test sets were fed to a simple classifier 2-layer feed forward ANN model which evaluated each feature extractor (TD-IDF, Glove and BERT). The same ANN model was used for all 3 feature extractors' outputs to balance and rationalize the experimental results.

A. The Artificial Neural Network

A simple ANN was selected to classify the news in two classes (fake and real) after getting the features from all 3 feature extractors. The ANN contains two hidden dense layers having 8 and 16 neurons respectively with a final output layer with a single neuron as this is a binary classification problem. Relu activation function was used in hidden layers and sigmoid in the output layer with Adam optimizer to update the leaning weights while training.

B. Experimental Results

In this section, the results obtained from the 3 feature extractors are analyzed and compared. Accuracy is commonly used as an evaluation metric to analyze performance [24]. Figures 3, 4 illustrate the obtained results including accuracy, precision, and AUC ROC for both datasets. Outperforming the TD-IDF and Glove, BERT achieved 96% accuracy on the first dataset and 99% accuracy on the second dataset. TD-IDF achieved 93% on the first dataset and 96% on the second.

In the first dataset, BERT outperformed Glove and TD-IDF by 12% and 3% in accuracy. The precisions of BERT and TD-IDF do not differ much but achieved better by 13% than Glove. The AUC ROC scores of TD-IDF and BERT are similar and better than Glove. In the second dataset, Glove performed better. This time BERT outperformed TD-IDF by 3% as was in first dataset and Glove by 7% in accuracy. In precision, BERT outperformed Glove and TD-IDF by 7% and 3% respectively, and on the AUC ROC score, both BERT and TD-IDF achieved 3% higher than Glove. The reason because Glove performed a bit better in the second dataset (but not better than BERT and TD-IDF), is that the second dataset is 3 times the size of the first dataset as we implemented an ANN which requires large datasets to perform better. All feature extractors' performances are better on the second dataset. The detailed comparisons of all discussed evaluation metrics are given in Tables II and II.

In both datasets, BERT performed better than the other feature extractors. TD-IDF also performed well, but Glove embeddings did not because the problem at hand is a contextual one and Glove computes static embeddings. Surprisingly, TD-IDF performed far better than Glove and reached BERT's performance at some stage in the first dataset. There are 3 explanations for this. The first reason for this surprise performance of TD-IDF is that the word length of the document for Glove and BERT was fixed. TD-IDF uses all words in given documents and normally we do not fix the size of TD-IDF word length in its implementation, but for embeddings a size must be defined. The second reason is that Glove and BERT might have over fit the data as their vocabulary size is too large. The third reason may be that signal from embeddings is noisy as they have complex architecture in their implementation which can cause the model to learn false information from the given data during training.

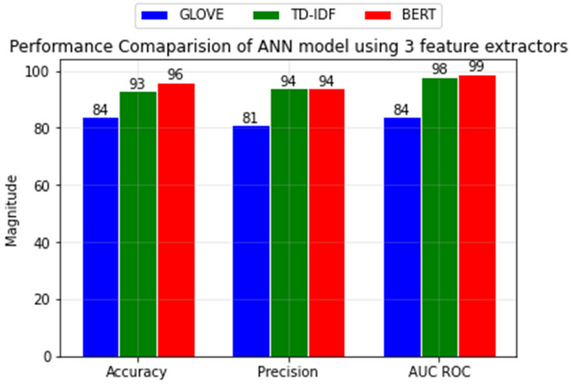

Fig. 3. Analysis of feature extractors' performance on the first dataset.

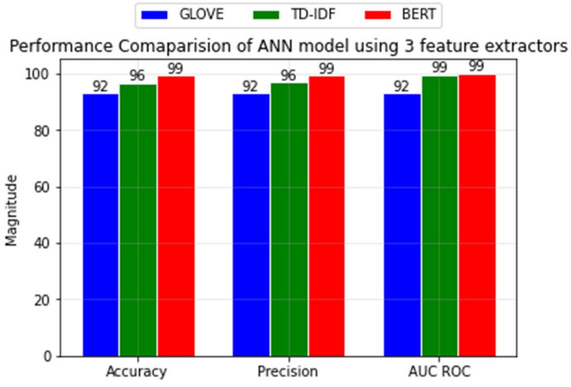

Fig. 4. Analysis of feature extractors' performance on the second dataset.

TABLE I. PERFORMANCE EVALUATION OF FEATURE EXTRACTORS ON DATASET 1

Feature extractor	Accuracy	Precision	Recall	F1 Score	AUC ROC score	AUC PR score
BERT	96.37	94.71	98.27	96.46	99.08	98.71
TD-IDF	93.68	94.85	92.47	93.65	98.70	98.77
Glove	84.21	81.80	87.31	84.47	84.26	87.67

TABLE II. PERFORMANCE EVALUATION OF FEATURE EXTRACTORS ON DATASET 2

Feature extractor	Accuracy	Precision	Recall	F1 Score	AUC ROC score	AUC PR score
BERT	99.23	99.14	99.33	99.24	99.97	99.98
TD-IDF	96.61	96.71	96.57	96.64	99.46	99.48
Glove	92.95	92.99	92.94	92.97	92.95	94.73

V. CONCLUSION

The easy access of social media to everyone has obvious advantages but also it has some disadvantages, such as the rapid quick spread of fake news. It is a very tedious job to check every news item manually, so, in order to overcome this problem, researchers are developing algorithms to detect fake news automatically. Fake news identification is a contextual problem in which the meaning of the same words may be different depending on the context. Various feature extractors have been built to efficiently solve this problem. In this paper, we analyzed two publicly available fake news datasets using three different feature extractors: TD-IDF vectorizer, Glove static embeddings, and BERT contextual embeddings on the fake news datasets and the outputs were fed to an ANN model for classification. It was found experimentally that the BERT model outperformed the TD-IDF and Glove in both datasets. TD-IDF outperformed Glove for both datasets and competed well with BERT.

REFERENCES

[1] T. Lima-Quintanilha, M. Torres-da-Silva, and T. Lapa, "Fake news and its impact on trust in the news. Using the Portuguese case to establish lines of differentiation," *Communication & Society*, vol. 32, no. 3, pp. 17–32, Apr. 2019, https://doi.org/10.15581/003.32.3.17-32.

[2] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, and P. Nakov, "Predicting Factuality of Reporting and Bias of News Media Sources," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct. 2018, pp. 3528–3539, https://doi.org/10.18653/v1/D18-1389.

[3] R. Zellers *et al.*, "Defending Against Neural Fake News," *arXiv:1905.12616 [cs]*, Dec. 2020, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/1905.12616.

[4] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," in *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, Vancouver, Canada, Oct. 2017, pp. 127–138, https://doi.org/10.1007/978-3-319-69155-8_9.

[5] B. A. Asaad and M. Erascu, "A Tool for Fake News Detection," in *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Timisoara, Romania, Sep. 2018, pp. 379–386, https://doi.org/10.1109/SYNASC.2018.00064.

[6] Abdullah-All-Tanvir, E. M. Mahir, S. Akhter, and M. R. Huq, "Detecting Fake News using Machine Learning and Deep Learning Algorithms," in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, Sarawak, Malaysia, Jun. 2019, pp. 1–5, https://doi.org/10.1109/ICSCC.2019.8843612.

[7] S. Sangamnerkar, R. Srinivasan, M. R. Christhuraj, and R. Sukumaran, "An Ensemble Technique to Detect Fabricated News Article Using Machine Learning and Natural Language Processing Techniques," in *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, Jun. 2020, https://doi.org/10.1109/INCET49848.2020.9154053.

[8] D. Chopra, N. Joshi, and I. Mathur, "Improving Translation Quality By Using Ensemble Approach," *Engineering, Technology & Applied Science Research*, vol. 8, no. 6, pp. 3512–3514, Dec. 2018, https://doi.org/10.48084/etasr.2269.

[9] M. Biniz, S. Boukil, F. Adnani, L. Cherrat, and A. Moutaouakkil, "Arabic Text Classification Using Deep Learning Technics," *International Journal of Grid and Distributed Computing*, vol. 11, no. 9, pp. 103–114, Sep. 2018, https://doi.org/10.14257/ijgdc.2018.11.9.09.

[10] A. Hassan and A. Mahmood, "Efficient Deep Learning Model for Text Classification Based on Recurrent and Convolutional Layers," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, Mexico, Dec. 2017, pp. 1108–1113, https://doi.org/10.1109/ICMLA.2017.00009.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/1810.04805.

[12] 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)*, Doha, Qatar, Oct. 2014, pp. 1532–1543, https://doi.org/10.3115/v1/D14-1162.

[13] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011.

[14] "Fake News: Balanced dataset for fake news analysis," *Kaggle*. https://kaggle.com/hassanamin/textdb3 (accessed Mar. 19, 2021).

[15] "Fake news: Fake News Classifier Using Bidirectional LSTM," *Kaggle*. https://kaggle.com/saratchendra/fake-news (accessed Mar. 19, 2021).

[16] H. Christian, M. P. Agus, and D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 7, no. 4, pp. 285–294, Dec. 2016, https://doi.org/10.21512/comtech.v7i4.3746.

[17] B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based Framework for Text Categorization," *Procedia Engineering*, vol. 69, pp. 1356–1364, Jan. 2014, https://doi.org/10.1016/j.proeng.2014.03.129.

[18] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory With GloVe Features," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 5, no. 2, pp. 85–100, Dec. 2019, https://doi.org/10.26555/jiteki.v5i2.15021.

[19] U. Khan, K. Khan, F. Hassan, A. Siddiqui, and M. Afaq, "Towards Achieving Machine Comprehension Using Deep Learning on Non-GPU Machines," *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, pp. 4423–4427, Aug. 2019, https://doi.org/10.48084/etasr.2734.

[20] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, Oct. 2020, pp. 38–45, https://doi.org/10.18653/v1/2020.emnlp-demos.6.

[21] M. Zaheer *et al.*, "Big Bird: Transformers for Longer Sequences," *arXiv:2007.14062 [cs, stat]*, Jan. 2021, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/2007.14062.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Sep. 2013, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/1301.3781.

[23] S. Liu, H. Tao, and S. Feng, "Text Classification Research Based on Bert Model and Bayesian Network," in *2019 Chinese Automation Congress (CAC)*, Hangzhou, China, Nov. 2019, pp. 5842–5846, https://doi.org/10.1109/CAC48633.2019.8996183.

[24] A. Hussain, G. Ali, F. Akhtar, Z. H. Khand, and A. Ali, "Design and Analysis of News Category Predictor," *Engineering, Technology & Applied Science Research*, vol. 10, no. 5, pp. 6380–6385, Oct. 2020, https://doi.org/10.48084/etasr.3825.