News Authenticity Identification System Using BERT: An Empirical Study on the Performance of BERT in comparison to LSTM

Tani Hossain
Department of Computer Science
Georgia State University
Atlanta, GA, USA
thossain4@student.gsu.edu

Karishma Mahendra Wadhe
Department of Computer Science
Georgia State University
Atlanta, GA, USA
kwadhe1@student.gsu.edu

***Abstract*— BERT was first introduced by Google AI researchers in 2018. Within just a few months these algorithms replaced previous NLP algorithms in the Google Search Engine and achieved state-of-the-art performance on a number of natural language understanding tasks. Text classification is one of the most classic understanding tasks. In this project, the task of categorizing fake and real news has been taken using the BERT model, and the performance is also measured. Moreover, we will use build the same system with the same news dataset using a well-known NLP algorithm called LSTM and will compare the performance of the two algorithms based on their Accuracy.**

Keywords—Text Classification, Deep Learning, NLP, BERT, LSTM

I. Introduction

Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, particularly how to program computers to process and analyze large amounts of natural language data. It is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. 'Human-like language processing' reveals that NLP is considered a discipline within Artificial Intelligence (AI). And while the full lineage of NLP does depend on a number of other disciplines, since NLP strives for human-like performance, it is appropriate to consider it an AI discipline [1]. Natural language processing has its roots in the 1950s. Already in 1950, Alan Turing published an article titled "Computing Machinery and Intelligence" which proposed what is now called the Turing test as a criterion of intelligence, a task that involves the automated interpretation and generation of natural language, but at the time not articulated as a problem separate from artificial intelligence. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze text and then assign a set of pre-defined tags or categories based on its content [2].

Text classification problems have been widely studied and addressed in many real applications over the last few decades. Especially with recent breakthroughs in Natural Language Processing (NLP) and text mining, many researchers are now interested in developing applications that leverage text classification methods [3]. Text classification is becoming an increasingly important part of businesses as it allows to easily get insights from data and automate business processes. Some of the most common examples and use cases for automatic text classification include the following:

a) Sentiment Analysis: the process of understanding if a given text is talking positively or negatively about a given subject (e.g. for brand monitoring purposes).
b) Authentication Detection: identifying if a news article or social media post is authentic or not.
c) Topic Detection: the task of identifying the theme or topic of a piece of text (e.g. know if a product review is about Ease of Use, Customer Support, or Pricing when analyzing customer feedback).
d) Identification of illegal product advertisements: identifying advertisement posts for selling products that are illegal such as drugs.

The extensive spread of fake news has the potential for extremely negative impacts on individuals and society. Therefore, fake news detection on social media has recently become an emerging research that is attracting tremendous attention [4]. Fake news detection on social media presents unique characteristics and challenges that make existing detection algorithms from traditional news media ineffective or not applicable. First, fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content; therefore, we need to include auxiliary information, such as using social engagements on social media, to help make a determination. Second, exploiting this auxiliary information is challenging in and of itself as users' social engagements with fake news produce data that is big, incomplete, unstructured, and noisy. Because the issue of fake news detection on social media is both challenging and relevant, we used more powerful algorithms to work with this problem.

There are several algorithms for text classification such as The Naïve Bayes Classifier (NBC), k-nearest neighbor (KNN) [20]. Support Vector Machine (SVM) these are some classic text classification algorithms [3]. These algorithms are implemented, and their results are tested and proven by now. In our research, we will use two comparatively modern algorithms one is named BERT and another one is named LSTM.

BERT stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. It is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google. In 2019, Google announced that it had begun leveraging BERT in its search engine, and by late 2020 it was using BERT in almost every English-language query. A 2020 literature survey concluded that "in a little over a year, BERT has become a ubiquitous baseline in NLP experiments", counting over 150 research publications analyzing and improving the model. Therefore, it has become a state of art algorithm. The state of the art (sometimes cutting edge or leading edge) refers to the highest level of general development, as of a device, technique, or scientific field achieved at a particular time. However, in some contexts, it can also refer to a level of development reached at any particular time as a result of the common methodologies employed at the time [6].

BERT’s key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The paper’s results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. The researchers detail a novel technique named Masked LM (MLM) which allows bidirectional training in models in which it was previously impossible [5]. BERT is a computational model that converts words into numbers. This process is crucial because machine learning models take in numbers (not words) as inputs, so an algorithm that converts words into numbers allows you to train machine learning models on your originally textual data. BERT is pre-trained on an absurd amount of data. The original BERT model comes in two sizes: BERT-base (trained on BooksCorpus: ~800 million words) and BERT-large (trained on English Wikipedia: ~ 2,500 million words). Both of these models have huge training sets! As anyone in the machine learning field knows, the power of big data is pretty much unbeatable. When you’ve seen 2,500 million words, you’re going to be pretty good, even on new words. What this means is that because BERT was pre-trained so well that it can be applied on small datasets and still have good performance.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems [7]. We can say that, when we move from RNN to LSTM (Long Short-Term Memory), we are introducing more & more controlling knobs, which control the flow and mixing of Inputs as per trained Weights. And thus, bringing more flexibility in controlling the outputs. It was first introduced in 1997 but largely unappreciated until recently. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition,[2] speech recognition[3][4] and anomaly detection in network traffic or IDSs (intrusion detection systems). LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

Though BERT is pre-trained with a numerous number of data already which is not the case for LSTM, comparison of these two models has become popular nowadays. In quest of searching the answer for the question if we can use a large pre-trained model like BERT and get better results than simple models like LSTM data scientists are comparing their performance on several domains [9]. Surprisingly in some cases, authors also got inverse intuitive results where LSTM outperforms BERT [8]. As both BERT AND LSTM are getting popular almost at the same timeframe, but both have different origin times to assess the performance of BERT we will compare the accuracy result of the news detection system implemented using BERT with the system implemented using LSTM.

II. RELATED WORKS

Detecting fake news is an important task, which not only ensures users receive authentic information but also helps maintain a trustworthy news ecosystem. The majority of existing detection algorithms focus on finding clues from news contents, which are generally not effective because fake news is often intentionally written to mislead users by mimicking true news. Therefore, we need to explore auxiliary information to improve detection. The social context during news dissemination process on social media forms the inherent tri-relationship, the relationship among publishers, news pieces, and users, which has the potential to improve fake news detection. For example, partisan-biased publishers are more likely to publish fake news, and low-credible users are more likely to share fake news. In this paper, the authors studied the novel problem of exploiting social context for fake news detection and propose a tri-relationship embedding framework TriFN, which models publisher-news relations and user-news interactions simultaneously for fake news classification [10].A large body of recent works has focused on understanding and detecting fake news stories that are disseminated on social media. To accomplish this goal, these works explore several types of features extracted from news stories, including source and posts from social media. In addition to exploring the main features proposed in the literature for fake news detection, the authors presented a new set of features and measure the prediction performance of current approaches and features for automatic detection of fake news. Their results reveal interesting findings on the usefulness and importance of features for detecting false news. Finally, they discussed how fake news detection approaches can be used in the practice, highlighting challenges and opportunities [11]. The proliferation of misleading information in everyday access media outlets such as social media feeds, news blogs, and online newspapers have made it challenging to identify trustworthy news sources, thus increasing the need for computational tools able to provide insights into the reliability of online content. In this paper, the authors focused on the automatic identification of fake content in online news. Our contribution is twofold. First, they introduce two novel datasets for the task of fake news detection, covering seven different news domains. They describe the collection, annotation, and validation process in detail and present several exploratory analysis on the identification of linguistic differences in fake and legitimate news content. Second, They conducted a set of learning experiments to build accurate fake news detectors. In addition, they provided comparative analyses of the automatic and manual identification of fake news [12].

Language model pre-training has proven to be useful in learning universal language representations. As a state-of-the-art language model pre-training model, BERT (Bidirectional Encoder Representations from Transformers) has achieved amazing results in many language understanding tasks. In this paper, the authors conducted exhaustive experiments to investigate different fine-tuning methods of BERT on text classification task and provide a general solution for BERT fine-tuning. Finally, they proposed solution obtains new state-of-the-art results on eight widely-studied text classification datasets [13]. Machine learning algorithms are often vulnerable to adversarial examples that have imperceptible alterations from the original counterparts but can fool the state-of-the-art models. It is helpful to evaluate or even improve the robustness of these models by exposing the maliciously crafted adversarial examples. In this paper, The authors present TextFooler, a simple but strong baseline to generate adversarial text. By applying it to two fundamental natural language tasks, text classification and textual entailment, we successfully attacked three target models, including the powerful pre-trained BERT, and the widely used convolutional and recurrent neural networks. We demonstrate three advantages of this framework: (1) effective—it outperforms previous attacks by success rate and perturbation rate, (2) utility-preserving—it preserves semantic content, grammaticality, and correct types classified by humans, and (3) efficient—it generates adversarial text with computational complexity linear to the text length [14]. Modern text classification models are susceptible to adversarial examples, perturbed versions of the original text indiscernible by humans which get misclassified by the model. Recent works in NLP use rule-based synonym replacement strategies to generate adversarial examples. These strategies can lead to out-of-context and unnaturally complex token replacements, which are easily identifiable by humans. The authors presented BAE, a black box attack for generating adversarial examples using contextual perturbations from a BERT masked language model. BAE replaces and inserts tokens in the original text by masking a portion of the text and leveraging the BERT-MLM to generate alternatives for the masked tokens. Through automatic and human evaluations, they showed that BAE performs a stronger attack, in addition to generating adversarial examples with improved grammaticality and semantic coherence as compared to prior work [15].

Neural network models have been demonstrated to be capable of achieving remarkable performance in sentence and document modeling. Convolutional neural network (CNN) and recurrent neural network (RNN) are two mainstream architectures for such modeling tasks, which adopt totally different ways of understanding natural languages. In this work, The experimental results show that the C-LSTM outperforms both CNN and LSTM and can achieve excellent performance on these tasks [16]. This paper utilizes 2D convolution to sample more meaningful information of the matrix. Experiments are conducted on six text classification tasks, including sentiment analysis, question classification, subjectivity classification and newsgroup classification. Compared with the state-of-the-art models, the proposed models achieve excellent performance on 4 out of 6 tasks. Specifically, one of the proposed models achieves highest accuracy on Stanford Sentiment Treebank binary classification and fine-grained classification tasks [17]. The traditional text classification methods are based on machine learning. It requires a large amount of artificially labeled training data as well as human participation. However, it is common that ignoring the contextual information and the word order information in such a way, and often exist some problems such as data sparseness and latitudinal explosion. With the development of deep learning, many researchers have also been using deep learning in text classification. This paper investigates the application issue of NLP in text classification by using the Bi-LSTM-CNN method. For the purpose of improving the accuracy of text classification, a kind of comprehensive expression is employed to accurately express semantics. The experiment shows that the model in this paper has great advantages in the classification of news texts [18].

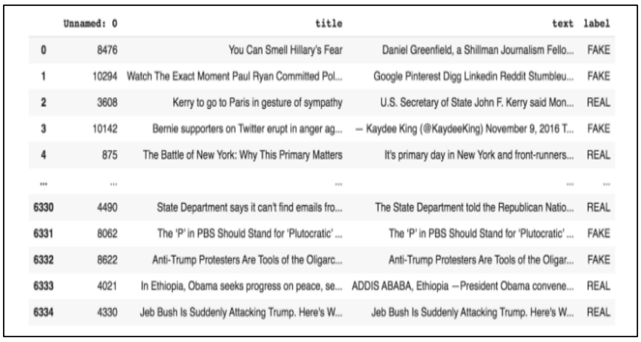
The BERT model has arisen as a popular state-of-the-art machine learning model in the recent years that is able to cope with multiple NLP tasks such as supervised text classification without human supervision. Its flexibility to cope with any type of corpus delivering great results has make this approach very popular not only in academia but also in the industry. Although, there are lots of different approaches that have been used throughout the years with success. In this work, the authors first presented BERT and include a little review on classical NLP approaches. Then, they empirically tested with a suite of experiments dealing different scenarios the behaviour of BERT against the traditional TF-IDF vocabulary fed to machine learning algorithms. Their purpose of this work wass to add empirical evidence to support or refuse the use of BERT as a default on NLP tasks. Experiments show the superiority of BERT and its independence of features of the NLP problem such as the language of the text adding empirical evidence to use BERT as a default technique to be used in NLP problems [19]. Traditional sentiment construction in finance relies heavily on the dictionary-based approach, with a few exceptions using simple machine learning techniques such as Naive Bayes classifier. While the current literature has not yet invoked the rapid advancement in the natural language processing, they constructed in this research a textual-based sentiment index using a novel model BERT recently developed by Google, especially for three actively trading individual stocks in Hong Kong market. On the one hand, they demonstrated a significant enhancement of applying BERT in sentiment analysis when compared with existing models. By combining with the other two existing methods commonly used on building the sentiment index in the financial literature, i.e., option-implied and market-implied approaches, they proposed a more general and comprehensive framework for financial sentiment analysis, and further provide convincing outcomes for the predictability of individual stock return for the above three stocks using LSTM (with a feature of a nonlinear mapping), in contrast to the dominating econometric methods in sentiment influence analysis that are all of a nature of linear regression [8]. In another paper, the experimental results show that bidirectional LSTM models can achieve significantly higher results than a BERT model for a small dataset and these simple models get trained in much less time than tuning the pre-trained counterparts. We conclude that the performance of a model is dependent on the task and the data, and therefore before making a model choice, these factors should be taken into consideration instead of directly choosing the most popular model [9].

III. PROBLEM FORMULATION

Implementing a fake and real news categorization system using a powerful pre-trained state-of-art Natural Language Processing Algorithm called BERT and depicting the higher accuracy with respect to a simpler model (LSTM).

A. Input Dataset

In this paper, the REAL and FAKE News Dataset from Kaggle was used [20]. The dataset contains 7795 rows and 4 columns which are an arbitrary index, title, text, and the corresponding label.



	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

Fig. 1. Snippet of raw dataset

B. Output

We will fine-tune a BERT model using a training portion of the dataset and show its performance by testing with the other portion of the dataset test. We will train also an LSTM model using the same training portion of this dataset. The rest of the dataset we will use as testing data set. We will test both the model and observe the confusion matrix and finally will compare the accuracy of both the models.

IV. EXPERIMENTAL DESIGN

The experiment in this paper has been divided into three parts. First of all, we pre-processed the raw data. Then we used the pre-processed data to implement the system using BERT. After that, we have used the same pre-processed dataset to implement the LSTM system.

A. Data Preprocessing

Data Preprocessing is an important step in data mining. It is the process of transforming raw data into an understandable format. The quality of the data should be checked before applying Algorithms. It removes outliers and scales the features to an equivalent range. This step is also used to split data in the Training and test set.

First, we converted REAL to 0 and FAKE to 1. Then concatenated title and text to form a new column titletext as we are going to use both the title and text to figure out if the news is real or fake. Then we Dropped rows with empty text and trimmed each sample to the first n words (Here, n = 200). Finally we have split the full dataset according to train_test_ratio(.10) and the train dataset according to train_valid_ratio(.80) and got the resulting data frames into train.csv, valid.csv, and test.csv files.

B. Designing the system with BERT

BERT consists of L identical transformer encoder layers. Each of these layers contain two types of sublayer. The First layer is a multi-head self-attention mechanism. This layer encodes specific word and also look at other words in sequence to derive contextual meaning. The second layer is a fully connected feedforward network (FFN). This layer consists of two linear transformations W1, W2.

$(W_1 \varepsilon R^{d_{model} \times d_{ff}}, b_1 \varepsilon R^{d_{ff}}), (W_2 \varepsilon R^{d_{model} \times d_{ff}}, b_2 \varepsilon R^{d_{ff}})$ such that

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

FFN uses GELU activation. The Gaussian Error Linear Unit, or GELU, is an activation function. The GELU activation function is x Φ (x), where the standard Gaussian cumulative distribution function. The GELU nonlinearity weights inputs by their percentile.

$$GELU(x) = 0.5x(1 + tanh(\sqrt{2/\prod}(x + 0.044715x^3)))$$

Each encoder layer has a residual connection and layer normalization such that the output of each sublayer is:

$$LayerNorm(x + Sublayer(x))$$

There are two major versions of BERT. First version is BERT-base: L=12, $dmodel$ = 768, h=12, $df\ f$ = 3072 (110M total parameters). Second version is BERT-base: L=24, $dmodel$ = 1024, h=16, $df\ f$ = 4096 (340M total parameters). Where L is number of layers, $dmodel$ is the dimensionality of input and output of each layer, h is the number of attentions heads in a self-attention sublayer and dff is the number of hidden units in feed-forward sublayer. In our case, we have used the first version because it is smaller and it takes less time to fine-tune the model using this version.

Here, we used five epochs to fine-tune the model using our dataset. A suitable learning rate of 2e-5 and Adam optimizer was used. We added save and load functionalities for model checkpoints and training metrics, respectively. The save function for model checkpoint does not save the optimizer. We did not save the optimizer because the optimizer normally takes very large storage space and we assumed no training from a previous checkpoint is needed. The training metric stores the training loss, validation loss, and global steps so that visualizations regarding the training process can be made later. We used BinaryCrossEntropy as the loss function since fake news detection is a two-class problem.

C. *System Design with LSTM*

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. It has a similar control flow as a recurrent neural network. It processes data passing on information as it propagates forward. The differences are the operations within the LSTM's cells. These operations are used to allow the LSTM to keep or forget information which makes it more memory efficient.

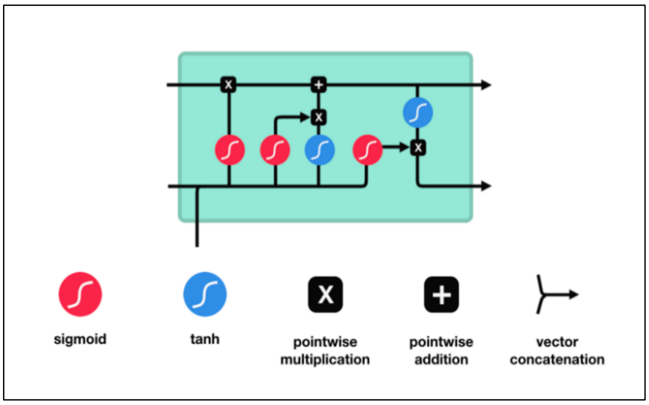


Fig. 2. LSTM cell and its Operations

We constructed the LSTM class that inherits from the nn.Module. Inside the LSTM, we constructed an Embedding layer, followed by a bi-LSTM layer, and ending with a fully connected linear layer. In the forward function, we pass the text IDs through the embedding layer to get the embeddings, passed it through the LSTM accommodating variable-length sequences, learn from both directions, pass it through the fully connected linear layer, and finally sigmoid to get the probability of the sequences belonging to FAKE (being 1). We trained the LSTM with 10 epochs and save the checkpoint and metrics whenever a hyperparameter setting achieves the best (lowest) validation loss. It took less than two minutes to train.

V. EXPERIMENTAL RESULTS

In this part, we will observe and analysis the results obtained from the testing of both of the models. At the end of this section, we will also see the comparison of training efficiency and accuracy of both systems.

A. *Evaluation of BERT System*

We evaluated our model parameters against the validation set. We save the model each time the validation loss decreases so that we end up with the model with the lowest validation loss, which can be considered as the best model.

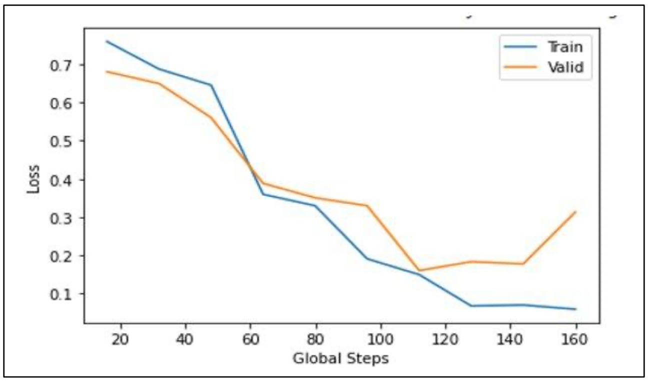
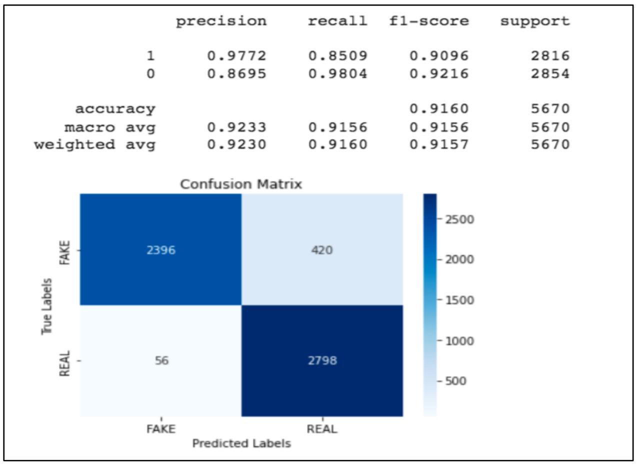


Fig. 3. Training loss and Valid loss for BERT

Fig. 3. shows that the training loss decreases with each step which means the model is getting to know the data better. but we can see that after a certain step valid loss starts to increase which means the system starts to overfit, So, we can say the point where valid loss is 0.2 is our best model.



```
              precision    recall  f1-score   support

           1     0.9772    0.8509    0.9096      2816
           0     0.8695    0.9804    0.9216      2854

    accuracy                         0.9160      5670
   macro avg     0.9233    0.9156    0.9156      5670
weighted avg     0.9230    0.9160    0.9157      5670
```

Fig. 4. Classification Report for BERT

Fig. 4. shows the Confusion table and the accuracy measurement of the News authentication detection system built using BERT. From the result, we can see that the system has an overage of 92.3% accuracy rate.

B. *Evaluation of LSTM System*

We also evaluated our model parameters against the validation set. Figure 5 shows that though the training loss decreases with each step, the valid loss starts to increase and then changes arbitrarily after a certain point but do not decrease after that. This means that even we used 10 epochs to train the model after a certain number of epochs the system becomes saturated.

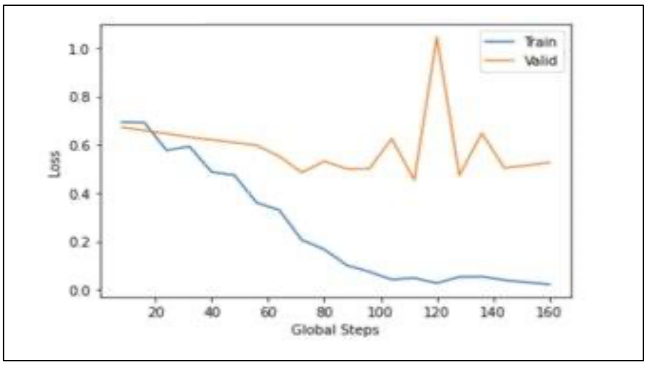
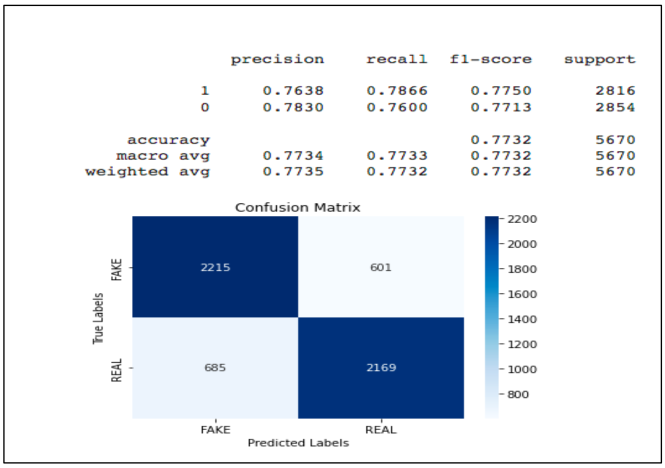


Fig. 5. Training loss and Valid loss for LSTM

Fig. 6. shows the Confusion table and the accuracy measurement of the News authentication detection system built using LSTM. Here, we can see that the system has an overage of 77.5% accuracy.



```
              precision    recall  f1-score   support

           1     0.7638    0.7866    0.7750      2816
           0     0.7830    0.7600    0.7713      2854

    accuracy                         0.7732      5670
   macro avg     0.7734    0.7733    0.7732      5670
weighted avg     0.7735    0.7732    0.7732      5670
```

Fig. 6. Classification Report for LSTM

C. *Comparison of LSTM and BERT*

We have implemented the same system using both LSTM and BERT. While training the the LSTM model was faster it does not have a lesser amount of validation loss in comparison to fine tuning the BERT system. If we observe the evaluations closely, we can see that in our case of news classification the BERT algorithm has significantly higher performance. SO, we can come to this conclusion that for this particular dataset and systems that we have built in this paper BERT performs significantly better that LSTM.

VI. CONCLUSION AND DISCUSSION

News authentication itself is a very difficult task and it is also very difficult even for human to identify fake news all at once. In this paper, we have tried to classy news using supervised learning where the news articles already had the tag of fake and real and that is why we have got a significant amount of accuracy in case of both the algorithms. Both the algorithms are very powerful and also distinct in their implementation and applications. Our goal was to introduce a comparatively new algorithm to detect news and compare its performance with an existing simple yet powerful algorithm.

REFERENCES

[1] Liddy, Elizabeth D. "Natural language processing." (2001)

[2] Ikonomakis, M., Kotsiantis, S. and Tampakas, V., 2005. Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8), pp.966-974.

[3] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D., 2019. Text classification algorithms: A survey. *Information*, 10(4), p.150.

[4] Shu, K., Sliva, A., Wang, S., Tang, J. and Liu, H., 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1), pp.22-36.

[5] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[6] Jermann, Patrick, Amy Soller, and Martin Muehlenbrock. "From mirroring to guiding: A review of the state of art technology for supporting collaborative learning." In *European Perspectives on Computer-Supported Collaborative Learning*, no. CONF, pp. 324-331. 2001.

[7] Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2016. LSTM: A search space odyssey. I*EEE transactions on neural networks and learning systems*, 28(10), pp.2222-2232.

[8] Ezen-Can, A., 2020. A Comparison of LSTM and BERT for Small Corpus. *arXiv preprint arXiv:2009.05451*

[9] Hiew, J.Z.G., Huang, X., Mou, H., Li, D., Wu, Q. and Xu, Y., 2019. BERT-based financial sentiment index and LSTM-based stock return predictability. *arXiv preprint arXiv:1906.09024.*

[10] Shu, K., Wang, S. and Liu, H., 2019, January. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining* (pp. 312-320).

[11] Reis, J.C., Correia, A., Murai, F., Veloso, A. and Benevenuto, F., 2019. Supervised learning for fake news detection. IEEE Intelligent Systems, 34(2), pp.76-81.

[12] Pérez-Rosas, V., Kleinberg, B., Lefevre, A. and Mihalcea, R., 2017. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104.*

[13] Sun, C., Qiu, X., Xu, Y. and Huang, X., 2019, October. How to fine-tune bert for text classification?. In *China National Conference on Chinese Computational Linguistics* (pp. 194-206). Springer, Cham.

[14] Jin, D., Jin, Z., Zhou, J.T. and Szolovits, P., 2020, April. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 05, pp. 8018-8025).

[15] Garg, S. and Ramakrishnan, G., 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970.*

[16] Zhou, C., Sun, C., Liu, Z. and Lau, F., 2015. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630.*.

[17] Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H. and Xu, B., 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639.*.

[18] Li, C., Zhan, G. and Li, Z., 2018, October. News text classification based on improved Bi-LSTM-CNN. In *2018 9th International Conference on Information Technology in Medicine and Education* (ITME) (pp. 890-893). IEEE..

[19] González-Carvajal, S. and Garrido-Merchán, E.C., 2020. Comparing BERT against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.

[20] https://www.kaggle.com/nopdev/real-and-fake-news-dataset.