Rumor Detection on Social Media

Farzaneh Jouyandeh
School of Computer Science
University of Windsor
Windsor, Canada
SID: 110034940

Prem Shanker Mohan
School of Computer Science
University of Windsor
Windsor, Canada
SID: 110036738

Zorawar Singh
School of Computer Science
University of Windsor
Windsor, Canada
SID: 110057433

***Abstract*—Rumors spreading widely on social media have had far-reaching consequences in both the online and offline worlds. The effects of rumor propagation are dreadful in time-critical events, such as during natural disasters. With so much unconfirmed data, it is unavoidable that there will be gossip in the environment. As a result, efficient rumor detection is an essential and trendy topic. The identification of rumors in online social media is represented as a classification problem in this paper, and the success of supervised machine learning algorithms is evaluated in real data. To address this problem, we proposed a text classification model using the BERT model and applied it to three datasets: LIAR, LIAR-PLUS, and ISOT dataset and compared the results. Our model performed best on the ISOT dataset and achieved 96% accuracy. For LIAR and LIAR-PLUS, our model achieved the accuracy of 68% and 70%.**

***Index Terms*—Rumor detection, Artificial Intelligence, Machine Learning, Social Media, text classification, BERT model**

I. Introduction

Rumors are a strong, pervasive, and persistent force that has an impact on individuals and groups. Since World War II, there has been a surge in interest in the psychology of rumors and their control, with early studies relying on factual but manual data collecting from books, newspapers, magazines, and interviews. Rumors have been defined in various ways, with the most common being "public communications imbued with private ideas about how the world works" and "means of creating sense to help us cope with our concerns and uncertainties". Rumors, as these definitions suggest, assist members of a society in learning about critical issues by providing individuals who engage with a collective problem-solving opportunity [21].

Misinformation campaigns are routinely carried out by misinformation producers for various commercial and political goals, given the growing expansion of social media. As a result, a substantial volume of false or unverified information has emerged and disseminated, affecting online social network members and having far-reaching consequences in the offline world. Thus, automatically detecting rumors is advantageous for taking early actions to lessen their detrimental impact.

Surprisingly, the importance of social networks is not confined to assisting in the organization of disruptive elements' operations. Many important government and news organizations are now using social media platforms to distribute information.

The social media era has made rumor spread even easier, as any piece of information can now be propagated by internet users without fear of being censored. Because persons getting information through this medium lack the capacity to verify the veracity of the information they receive, false rumors or unconfirmed information can spread more quickly and extensively. Some rumors have incorrectly hurt people's or organizations' reputations, and this negative role has gotten much attention in both study and society [22].

One of the most crucial study issues in the field of information credibility is rumor detection. It is frequently seen as a tall tale of event explanations passed down from person to person and relating to an object, event, or topic of public importance. Rumor dissemination is destructive to people's lives and society's stability, and it has become a major social network worry. Rumor detection is typically treated as a classification problem based on shallow message features such as content and blogger characteristics. However, in many circumstances, such superficial qualities are unable to discriminate between rumor and legitimate texts [20].

The remainder of this paper is organized as follows. Section II contains the problem statement and some definitions. Section III provides a literature review of rumor detection and machine learning techniques. The methodology followed to datasets, select, build and train the machine learning model is described in section IV followed by the experiment results, analysis and discussion in section IV. We then provide a Conclusion in section V, included summary, future research and open problems section.

II. problem Statement

A. Problem Description and Formulation

This section introduces some essential concepts about rumors detection and machine learning techniques.

Definition 1 - Rumor: A rumor is an information whose authenticity is questioned. Some rumors may prove to be genuine, while others may stay unconfirmed. False information is not always referred to as a rumor. Misinformation refers to unintentional mistakes made by people. On the other hand, there may be intentional rumors put to mislead people into believing them [1].

Definition 2 - Rumor Detection: Rumor detection is a four-step procedure that starts with gathering information from the various social media sites under consideration. This information must be arranged consistently so that relevant features may be extracted. Consolidation, cleansing, transformation, and reduction are all part of the preprocessing process. The essential features (including content-based, pragmatic, and network-specific features) are extracted, and each dataset is identified as a rumor or not a rumor using a variety of machine learning approaches such as Naive Bayesian, Support Vector Machines, and others [1].

The rumor detection problem is defined as follow: A story x is defined as a set of n pieces of related messages

$$M = m_1,\ m_2,\ \dots,\ m_n$$

m_1 is the source message (post) that initiated the message chain, which could be a tree structure having multiple branches. For each message m_i, it has attributes representing its content, such as text and image. Each message is also associated with a user who posted it. The user also has a set of attributes, including name, description, avatar image, past posts. The rumor detection task is then defined as: Given a story x with its message set M and user set U, the rumor detection task aims to determine whether this story is *true, false* or *unverified* (or just *true* or *false* for datasets having just two labels). This definition formulates the rumor detection task as a veracity classification task. The definition is the same as the definition used in many studies [2]. This study aims to address the problem of how to detect whether a statement is a rumor or not using the text attributes. The problem is treated as a supervised learning problem on three datasets.

B. Motivation

With the rapid growth of the Internet, social media has evolved into a useful online platform for people to gather information, voice their opinions, and engage with one another. As more individuals participate in debates about current events and share their perspectives on social media, many rumors emerge. Rumors can spread rapidly and quickly on social media due to the high number of users and simple access to social media, inflicting significant harm to society and significant economic losses. As a result, given the potential for panic and danger generated by rumors, it is critical to developing a system for detecting rumors on social media quickly and as soon as feasible.

C. Justification

Due to the large volume and fast spreading of rumors in social media, detecting rumors quickly and efficiently becomes critical. This study is highly beneficial as it can be used as a baseline for comparison with future machine learning methods using for rumor detection on social media, and also it can be considered a starting point for future research in this area.

III. Literature Review

Many rumor detection models have been proposed in the literature. A brief overview of some of the current rumor detection research methodologies and their findings are provided in this section.

Takahashi and Igata suggested a methodology for detecting rumor candidates on Twitter. They proposed an algorithm that was accurate enough. They discovered a clue keyword in the rumor data and confirmed that it may be used to find additional rumors as well [3]. During the Boston event, Gupta et al. discovered over 6,000 fraudulent accounts on Twitter, which were eventually suspended by the social media platform. In the interaction network of these suspended profiles amongst themselves, they discovered closed community structure and star formation [4]. Shirai et al. created a model to simulate the propagation of rumors. They discovered several rumors' features on Twitter, however they did not disclose how to spot rumors [5]. Yang et al. conducted a study on rumor detection on Sina Weibo for the first time. Sina Weibo is China's most popular microblogging platform. To detect rumor, Yang et al. examined data such as the number of retweets, location, and content type of Tweets. Yang et al. investigated how to detect rumor from a mixed mixture of true and untrue data [6].

Aggrawal gave a thorough presentation on rumor detection. The cosine similarity method outperforms the adjective similarity approach in high accuracy, with 94 percent and 74 percent, respectively. Aggrawal created a lexical classifier that outperformed rule-based Naive Bayes [7]. Hamidian and Diab used common datasets to perform a supervised rumor categorization job. They improved the precision of the rumor retrieval task by 0.972 by using the Tweet Latent Vector (TLV) feature, which constructs a 100-d vector representing each tweet. They also created a belief score and investigated how rumor posters' beliefs changed between 2010 and 2016 [8]. By matching rumor tweets with verified rumor articles, Jin et al. suggested an interpretable and reliable technique for detecting rumor tweets. 8 million tweets from Twitter were evaluated with a 94.7 percent rumor prediction rate using five algorithms [9].

Shu et al. presented a comprehensive review of characterizing fake news and detection methods. They discuss fake news in the context of traditional media and social media and explored the detection techniques from a data mining perspective exploring existing algorithms and feature extraction methods where they stated that accuracy, f-score, precision and recall are the most commonly used machine learning evaluation metrics [11]. Mandical et al. proposed a system that classifies fake news by utilizing machine learning algorithms such as Naive Bayes, Passive Aggressive Classifier and Deep Neural Networks [12] and claimed that it is feasible to detect fake news using machine learning algorithms whereas Gravanis et al. compared the performance of ensemble classifiers and simple classifiers on three fake news datasets finding that the AdaBoost classifier performed the best with a 95% accuracy [13].

In the context of social media, Aphiwongsophon et al. collected Twitter data and compare the performance of Naive Bayes, Neural Network and Support Vector Machine algorithms in detecting fake news. They find that Naive Bayes has an accuracy of 96.08% whereas both Neural Network and Support Vector Machine have an accuracy of 99.9% [14]. On the other hand, Bhutani et al. proposed a sentiment analysis approach to detect fake news and evaluate it using three data sets. They found that using tf-idf was the best approach using accuracy as an evaluation metric when compared to other methods [15].

BERT (Bidirectional Encoder Representations from Transformers) is a natural language processing (NLP) model that was created to pre-train deep bidirectional representations from unlabeled text and then fine-tune them using tagged text for various NLP applications [19]. Many similar types of research have been done from various perspectives based on BERT's innovative work. Sun et al. [24], for example, looked at different fine-tuning approaches of BERT on text classification tasks, such as long-text preprocessing, layer selection, layer-wise learning rate, catastrophic forgetting, and low-shot learning difficulties. However, they only addressed long-text datasets and ignored short-text datasets and concealed vector selection scenarios. In [25], Xu et al. developed a novel post-training strategy on BERT to improve the fine-tuning performance of BERT for review reading comprehension. They also used the proposed post-training to do certain other review-based tasks like aspect extraction and sentiment categorization.

These findings guided our approach to selecting suitable algorithm, sentiment analysis and techniques for our study.

IV. Methodology

A. Material and Data

In this paper, we applied our model on three different datasets: 1. LIAR, 2. LIAR-PLUS, and 3. ISOT.

- LIAR dataset is a new, publicly available dataset for fake news detection. We collected a decade-long, 12.8K manually labeled short statements in various contexts from POLITIFACT.COM, which provides a detailed analysis report and links to source documents for each case. This dataset can be used for fact-checking research as well. Notably, this dataset is an order of magnitude larger than previously largest public fake news datasets of similar type [10]. In this dataset, six fine-grained labels are considered for the truthfulness ratings: *pants-fire, false, barely-true, half-true, mostly-true*, and *true*. The LIAR dataset has a relatively well-balanced label distribution, as shown in Figure 1; except for 1,050 pants-fire cases, all other labels have between 2,063 and 2,638 incidents. The definition of labels in LIAR dataset:
  1) true – The statement is accurate and there's nothing significant missing.
  2) mostly-true – The statement is accurate but needs clarification or additional information.
  3) half-true – The statement is partially accurate but leaves out important details or takes things out of context.
  4) barely-true – The statement contains an element of truth but ignores critical facts that would give a different impression.
  5) false – The statement is not accurate.
  6) pants-fire – The statement is not accurate and makes a ridiculous claim. a.k.a. "Liar, Liar, Pants on Fire!"
- LIAR dataset has been extended to the LIAR-PLUS dataset by automatically extracting for each claim the justification that humans have provided in the fact-checking article associated with the claim. Most of the articles end with a summary that has a headline "our ruling" or "summing up". This summary usually has several justification sentences that are related to the statement. All sentences in these summary sections or the last five sentences were extracted in the fact-checking article when no summary exists. The sentence that has the verdict and related words has been filtered. These extracted sentences can support or contradict the statement, which is expected to enhance the accuracy of the classification approaches [32]. The label balance for LIAR and LIAR-PLUS datasets is shown in Fig.2.
- The third dataset is a public dataset containing two types of fake news and real news called ISOT. This dataset was collected from real-world sources; the truthful articles were obtained by crawling articles from Reuters.com (News website). As for the fake news articles, they were collected from different sources. The fake news articles were collected from unreliable websites that were flagged by Politifact (a fact-checking organization in the USA) and Wikipedia. The dataset contains different types of articles on different topics. However, the majority of articles focus on political and World news topics. It contains more than 12,600 real news extracted from articles from reuter.com, and it also contains more than 12,600 fake news extracted from articles from different fake news outlet resources. Each article contains the following information: article title, text, type, and the date the article was published on [31]. The label balance in this dataset is shown in Fig.3. Also, some details about both datasets are provided in Table.I.

B. Proposed Method

Data preprocessing can often have a significant impact on the generalization performance of a supervised ML algorithm. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively [16]. In the first step, to make the data clear, we removed the duplicate rows and dropped the rows even with single NaN and single missing values. After applying the preprocessing methods on both datasets.

In working with the text data type, as tokens are the building blocks of Natural Language, the most common way of processing the raw text happens at the token level. Tokenization is commonly understood as the first step of any natural language text preparation. The major goal of this early (pre-linguistic)

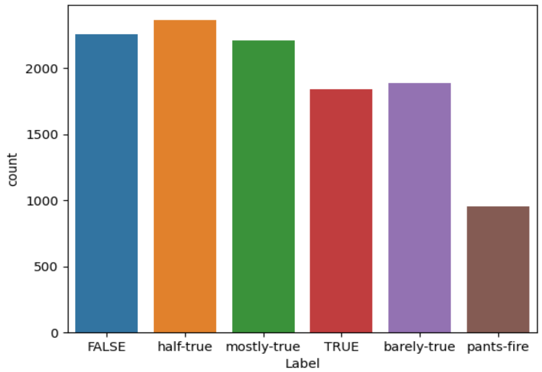


Fig. 1. Label Balance: LIAR and LIAR-PLUS dataset

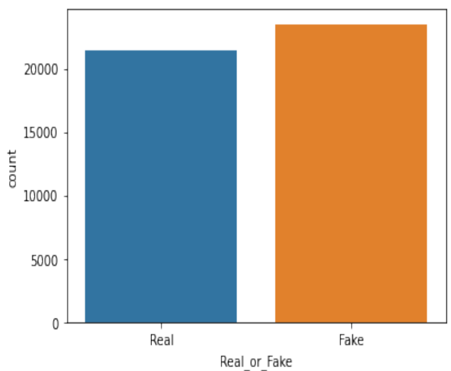


Fig. 2. Label Balance: ISOT dataset

task is to convert a stream of characters into a stream of processing units called tokens [18]. Tokenization is the process of splitting the text into these smaller pieces, and it often involves preprocessing the text to remove punctuation and transform all tokens into lowercase. Decisions made during tokenization have a significant effect on subsequent analysis [26]. As a result, we used the NLTK package in Python to tokenize news text, textblob package was used to perform sentiment analysis of the tweet text to classify it as positive, negative, and neutral. Also, in order to have a better performance, we merged the labels in the LIAR and LIAR-PLUS dataset to ”True” and ”False” to have just two labels, such as real-world data. After applying all of these techniques on the initial datasets, these texts and related columns were extracted in the CSV format files, with each column representing a feature of the texts and the final column which forms the predictor (Final Label for each row of the dataset).

TABLE I
STATISTICS OF LABELS IN THE USED DATASETS

Dataset	#false	#true	#half true	#barely true	#mostly true	#pants fire
LIAR	2,510	2,062	2,638	2,107	2,466	1,050
LIAR-PLUS	2,510	2,062	2,638	2,107	2,466	1,050
ISOT	23,503	21,418				

Given excessive amounts of raw information, the task of feature extraction, that is, transforming input data into features that can be useful for a learning algorithm, is as critical as ever for the successful application of machine learning. The major goal of feature extraction is to increase the accuracy of learned models by compactly extracting salient features (understandable to the learning algorithm) from the input data while also potentially removing noise and redundancy from the input [17]. In this regard, we measured the correlation of each feature with the final label and chose those features that had a higher correlation, and extracted them in a separate file.

The newest breakthrough in natural language understanding tasks is Bidirectional Encoder Representations from Transformers (BERT), which outperforms numerous predecessors such as the Generative Pretrained Transformer (GPT) and Embeddings from Language Models (ELMo) [27]. The BERT family of models uses the Transformer encoder architecture to process each token of input text in the full context of all tokens before and after. As we had to work on text data in this study, the BERT model was used for text classification. This pre-trained Deep Learning Language Model is helpful because it represents how a language works, simplifying downstream operations. As a result, we trained a classifier by adding a final layer to our pre-trained language model and fine-tuning its final layer to this specific purpose, requiring less annotated data and producing better results.

BERT-base model contains an encoder with 12 Transformer blocks, 12 self-attention heads, and a hidden size of 768. BERT takes an input of a sequence of no more than 512 tokens and outputs the representation of the sequence. The sequence has one or two segments that the first token of the sequence is always [CLS], which contains special classification embedding and another special token [SEP], is used to separate segments. For text classification tasks, BERT takes the final hidden state h of the first token [CLS] to represent the whole sequence. A simple softmax classifier is added to the top of BERT to predict the probability of label c:

$$p(c|h) = softmax(Wh), \tag{1}$$

where W is the task-specific parameter matrix. We fine-tune all the parameters from BERT as well as W jointly by maximizing the log-probability of the correct label [30].

C. Conditions and Assumptions

These are the assumptions that we considered for the implementation of this study: 1. The datasets used in this project have true information and are not biased. 2. The LIAR and LIAR-PLUS labels have been merged to perform similarly in the real world. 3. Some columns have been filtered, and these are the list of columns that we used in each dataset: The statement and subject columns in the LIAR dataset, the statement and justification columns in the LIAR-PLUS dataset, and the title column in the ISOT dataset are considered.

D. Simulation Analysis

In this study, we first split each dataset into train and validation sets using train_test_split. Then, we converted all inputs and labels into torch tensors, the required datatype for our model. The DataLoader needs to know our batch size for training, and for fine-tuning BERT on a specific task, a batch size of 16 or 32 is recommended, so we used 32 in this experiment. After the completion of each training epoch, we measured our performance on the validation set.

V. Computational Experiments

A. Experiments

Methodologies and libraries used in this project:

1) Data preparation using Pandas and scikit-learn: loading datasets from CSV files, doing some basic introspection, text preprocessing with nltk package, and text tokenization using transformers.
2) Splitting data into train, test and validation splits for ML.
3) Applying BERT model on data from both datasets.
4) Training and evaluating the model using TensorFlow , Keras, Pandas and scikit-learn.

B. Evaluation metrics

Accuracy: Accuracy is the ratio of correct predictions to total predictions. An accuracy metric is used to measure the algorithm's performance in an interpretable way and is usually determined after the model parameters and is calculated in the form of a percentage. It is the measure of how accurate the model's prediction is compared to the true data.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

Loss: A loss function is used to optimize a machine learning algorithm. The loss is calculated on training and validation, and its interpretation is based on how well the model is doing in these two sets. It is the sum of errors made for each example in training or validation sets. Loss value implies how poorly or well a model behaves after each iteration of optimization. As for the loss function, Cross entropy is used for evaluation in this project.

As the final part of our evaluation, the model was checked against the test set. The data required for this was created, and the model's 'evaluate' method in TensorFlow was used to compute the metric values for the trained model. This method returns the loss value and metrics values for the model in test mode.

C. implementation details

BERT relies on a Transformer (the attention mechanism that learns contextual relationships between words in a text). A basic Transformer consists of an encoder to read the text input and a decoder to produce a prediction for the task. Since BERT's goal is to generate a language representation model, it only needs the encoder part. The input to the encoder for BERT is a sequence of tokens, which are first converted into vectors and then processed in the neural network. But before processing can start, BERT needs the input to be massaged and decorated with some extra metadata:

1) Token embeddings: A [CLS] token is added to the input word tokens at the beginning of the first sentence, and a [SEP] token is inserted at the end of each sentence.
2) Segment embeddings: A marker indicating Sentence A or Sentence B is added to each token. This allows the encoder to distinguish between sentences.
3) Positional embeddings: A positional embedding is added to each token to indicate its position in the sentence.

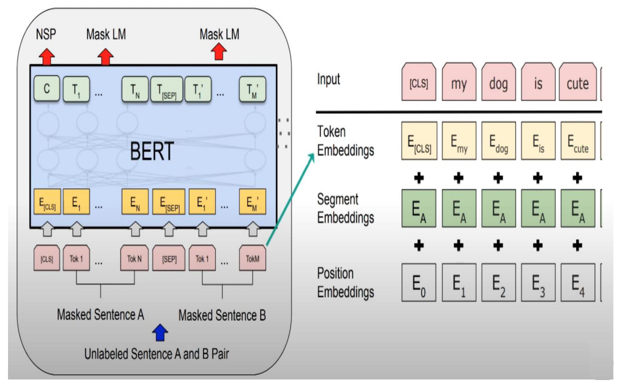


Fig. 3. BERT Architecture

D. Results and Analysis

After setting up our model, our training data, our validation data, and our training callbacks, we trained the model. Once we have trained the model, we showed the behavior loss function and accuracy metrics using the 'plot_history' function on both the training and validation sets, as shown in Figure 4. We also plot the training history for BERT on LIAR, LIAR-PLUS, and ISOT. For the final part of our evaluation, we checked the model against the test set, and in the next step, we predicted with the model.

To evaluate the BERT model for rumor detection and text classification tasks, loss and accuracy measures are used on three different datasets introduced in the previous subsection. Based on the evaluation in Figure 5, BERT has significantly better performance on the ISOT dataset, Since the accuracy of BERT on the ISOT dataset is 0.96, and for the LIAR and LIAR-PLUS dataset, the accuracy is 0.68 and 0.70.

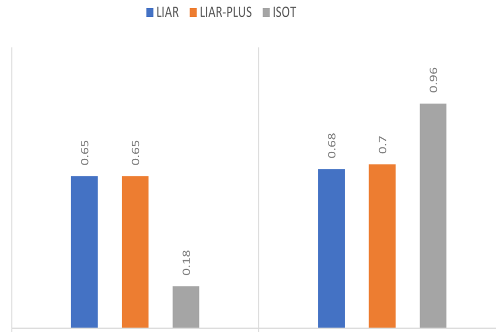
Several possible reasons can be mentioned for the models' poor performance on LIAR and LIAR-PLUS and its high performance on ISOT: 1. The language used in LIAR and LIAR-PLUS datasets are ambiguous compared to ISOT dataset 2. The lack of dates translates to a lack of historical information on LIAR and LIAR-PLUS. 3. There is a prevalence of missing speaker jobs and affiliations on LIAR and LIAR-PLUS, which means that these features may not have been beneficial for determining which piece of news was fake. 4. Some of the articles in the LIAR dataset are from the wrong set of data and are tagged with a truth value. As a result, those data points do not help train the model because they are mislabeled. 5.

Fig. 4. Visualization BERT-based text classification on LIAR, LIAR-PLUS and ISOT

ISOT dataset has a more significant number of rows comparing with two other datasets. Hence, we can conclude that as the dataset size increase, BERT performance grows.

E. Discussions

A text classification model and one of its approaches, rumor detection, is presented in this study. For this purpose, we have demonstrated the performance of our model (BERT-based approach) for rumor detection. Furthermore, for a better understating of the result and better comparison, some visualizations were used. Based on our results and also our visualization, it can be conducted that BERT has some limitations in text classification on some datasets. We also noticed that BERT is weaker when it comes to datasets with a higher number of classes, and its performance gets better when we shrank the number of classes and merged them, as we tried it on LIAR and LIAR-PLUS dataset. Also, it performs better on larger datasets. Hence, in future studies, these problems should be addressed.



	Loss	Accuracy
LIAR	0.65	0.68
LIAR-PLUS	0.65	0.7
ISOT	0.18	0.96

Fig. 5. Result of rumor detection using BERT

VI. Conclusion

A. Summary

In this study, we performed a BERT model for text classification and rumor detection tasks. First, to better understand and explain the mechanisms of BERT, we explored the simplest possible formulation of this model. Then, we applied the model on three different datasets: LIAR, LIAR-PLUS, and ISOT, and made a comparison between our model performance on these datasets and LIAR got 68%, LIAR-PLUS got 70% and ISOT got 96% accuracy rate. Based on our results, we can conclude that this study is highly beneficial to the community because:

1) It shows excellent results for BERT's text classification task and performance for rumor detection.
2) BERT model has a higher performance and better efficiency on datasets with fewer labels and larger datasets.
3) It can be used as a baseline for comparison with future text classification models.
4) It can be considered as a starting point for future research in rumor detection.

B. Future Research

Despite many algorithms and ways to permute such algorithms, text classification poses many challenges for learning systems [28]. For future research, we can work on using and improving BERT for rumor detection on larger datasets or targeted datasets. We can also perform other embedding methods and compare their result with BERT. Besides, we can do some experiments on the parameters used in this study. In addition, after comparing and recognizing the index criteria in choosing the basic approach, we can propose a new idea for improving BERT methods such as BERT-CNN By adding CNN to the task-specific layers of the BERT model [29]. Also, BERT Large model can be used, which has more parameters. And finally, our model can be extended by combining two different pre-trained models, such as a BERT encoder and a decoder, to explore more possibilities.

C. Open Problems

Although BERT and other text classifiers have achieved great success in different fields, including rumor detection, it is remarkable that text classification models are not good enough to offer satisfying solutions for any dataset. In this section, we will state some open problems for further researches: 1. Designing a text classification model with high performance on any dataset, such as datasets with an abstract vocabulary. 2. Using different variations of BERT Models to compare and contrast the performance. 3. Include user features such as the user activity 4. Scaling up BERT. 5. Non-Structural Scenarios.

Acknowledgement

The project code and final results are accessible on GitHub: https://github.com/Zorawar920/Rumor-Detection

References

[1] Kumar A., Sangwan S.R. (2019) Rumor Detection Using Machine Learning Techniques on Social Media. In: Bhattacharyya S., Hassanien A., Gupta D., Khanna A., Pan I. (eds) International Conference on Innovative Computing and Communications. Lecture Notes in Networks and Systems, vol 56. Springer, Singapore. https://doi.org/10.1007/978-981-13-2354-6-23

[2] Li, Q., Zhang, Q., Si, L., Liu, Y. (2019). Rumor detection on social media: datasets, methods and opportunities. arXiv preprint arXiv:1911.07199.

[3] T. Takahashi, N. Igata, "Rumor detection on twitter", In The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems, pp. 452-457, IEEE, 2012.

[4] Gupta, A., Lamba, H., Kumaraguru, P. (2013, September). 1.00 per rt bostonmarathon prayforboston: Analyzing fake content on twitter. In 2013 APWG eCrime researchers summit (pp. 1-12). IEEE.

[5] T. Shirai, T. Sakaki, F. Toriumi, K. Shinoda, K. Kazama, I. Noda, M. Numao, S. Kurihara, "Estimation of false rumor diffusion model and estimation of prevention model of false rumor diffusion on twitter (in japanese)," in The 26th Anual Conference of the Japanese Society for Artificial Intelligence, 2012.

[6] F. Yang, Y. Liu, X. Yu, M. Yang, "Automatic detection of rumor on Sina Weibo", In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, p. 13, ACM, 2012.

[7] N. Aggrawal, "Detection of Offensive Tweets: A Comparative Study", Computer Reviews Journal, 1(1), pp. 75-89, 2018.

[8] S. Hamidian, M. Diab, "Rumor identification and belief investigation on twitter", In Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 3-8, 2016.

[9] Z. Jin, J. Cao, H. Guo, Y. Zhang, Y. Wang, J. Luo, "Detection and analysis of 2016 us presidential election related rumors on twitter", In International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation, pp. 14-24, Springer, Cham, 2017.

[10] Wang, W. Y. (2017). " liar, liar pants on fire": A new benchmark dataset for fake news detection. arXiv preprint arXiv:1705.00648.

[11] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.

[12] R. R. Mandical, N. Mamatha, N. Shivakumar, R. Monica, and A. N. Krishna, "Identification of Fake News Using Machine Learning," 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2020.

[13] G. Gravanis et al. "Behind the cues: A benchmarking study for fake news detection," 2019.

[14] S. Aphiwongsophon and P. Chongstitvatana, "Detecting Fake News with Machine Learning Method," 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2018.

[15] B. Bhutani, N. Rastogi, P. Sehgal and A. Purwar, "Fake News Detection Using Sentiment Analysis," 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 2019, pp. 1-5

[16] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised learning," International Journal of Computer Science, vol. 1, 2006.

[17] D. Storcheus, A. Rostamizadeh and S. Kumar, "A survey of modern questions and challenges in feature extraction", Proc. NIPS Workshop, pp. 1-18, Dec. 2015.

[18] M. Hassler and G. Fliedl, "Text preparation through extended tokenization," Data Mining VII: Data, Text and Web Mining and their Business Applications, 2006.

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

[20] Zhang, Q., Zhang, S., Dong, J., Xiong, J., Cheng, X. (2015). Automatic detection of rumor on social network. In Natural Language Processing and Chinese Computing (pp. 113-122). Springer, Cham.

[21] Kwon, S., Cha, M., Jung, K. (2017). Rumor detection over varying time windows. PloS one, 12(1), e0168344.

[22] DiFonzo N, Bordia P. Rumor psychology: Social and organizational approaches. American Psychological Association; 2007

[23] Kaliyar, R. K., Goswami, A., Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Multimedia Tools and Applications, 80(8), 11765-11788.

[24] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" May 2019, arXiv:1905.05583. [Online]. Available: https://arxiv.org/abs/1905.05583

[25] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," Apr. 2019, arXiv:1904.02232. [Online]. Available: https://arxiv.org/abs/1904.02232

[26] Mullen, L. A., Benoit, K., Keyes, O., Selivanov, D., Arnold, J. (2018). Fast, consistent tokenization of natural language text. Journal of Open Source Software, 3(23), 655.

[27] Chang, W. C., Yu, H. F., Zhong, K., Yang, Y., Dhillon, I. (2019). Taming Pretrained Transformers for Extreme Multi-label Text Classification. arXiv preprint arXiv:1905.02331.

[28] Lewis, D. D. (1996, January). Challenges in machine learning for text classification. In Proceedings of the ninth annual conference on Computational learning theory (pp. 1-ff).

[29] Zheng, S., Yang, M. (2019, October). A New Method of Improving BERT for Text Classification. In International Conference on Intelligent Science and Big Data Engineering (pp. 442-452). Springer, Cham.

[30] Sun, C., Qiu, X., Xu, Y., 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.

[31] Ahmed, H., Traore, I., Saad, S. (2017, October). Detection of online fake news using n-gram analysis and machine learning techniques. In International conference on intelligent, secure, and dependable systems in distributed and cloud environments (pp. 127-138). Springer, Cham.

[32] Alhindi, T., Petridis, S., Muresan, S. (2018, November). Where is your evidence: Improving fact-checking by justification modeling. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER) (pp. 85-90).