Retrieval Enhanced Ensemble Model Framework For Rumor Detection On Micro-blogging Platforms

Rishab Sharma
Department of Computer Science
University of British Columbia
Kelowna, Canada
rishab.sharma@alumni.ubc.ca

Apurva Narayan
Department of Computer Science
University of British Columbia
Kelowna, Canada
apurva.narayan@ubc.ca

Fatemeh H. Fard
Department of Computer Science
University of British Columbia
Kelowna, Canada
fatemeh.fard@ubc.ca

***Abstract*—Automatic rumor detection is the task of finding rumors on social networks. Previous techniques leveraged the propagation structure of tweets to detect the rumors, which makes the propagation of tweets necessary to detect rumors. However, current text-based works provide sub-optimal results as compared to propagation-based techniques. This work presents a retrieval-based framework that leverages the similar tweets from the given train set and chooses the best model from an ensemble of models to predict the test tweet label. Our proposed framework is based on transformers-based pre-trained models (PTM's). Experiments on two public data sets used in previous works, show that our framework can detect the tweets with equivalent accuracy as propagation-based techniques. The primary advantage of this work is in early rumor detection. The proposed framework can detect rumors in few minutes compared to propagation-based works, which requires a significant amount of propagation of tweets that can take hours before they can be detected.**

***Index Terms*—rumor detection, information retrieval, pre-trained models, ensemble models**

I. Introduction

Online social media platforms like Twitter are actively used for personal, and mass communication all over the world [1]. The increasing popularity of social media platforms has made users more vulnerable to harmful un-monitored rumors on social media. Rumors can influence the political, economic, and social well-being of the users [2]. Therefore, it becomes essential to detect these rumors early to reduce or stop the dissemination of false information over a vast network of users. However, detecting the rumors is not a trivial task, considering the size of the micro-blogs platforms. It is challenging to detect the rumors manually quickly. Therefore, automatic techniques are required for the detection of rumors. Initial studies leveraged the textual data from the micro-blogs with Recurrent and Convolution Neural Networks [3]–[7]. Researchers have also started leveraging the propagation structure of tweets [8]–[10]. A propagation tree of a given tweet defines how the propagation of the tweet has evolved with different users. It includes all the tweets/retweets posted at different time frames. However, the propagation tree of a tweet may be sparse. Moreover, propagation-based techniques require the initial transmission of the tweet to generate its propagation structure, which means the propagation of a tweet (which can be a harmful tweet) is required to detect the nature of the tweet. [11]

Recently, pre-trained models (PTM's) like BERT [12], based on transformer architecture [13] have shown the state-of-the-art performance on various tasks in natural language processing [13]. The current literature has acknowledged the use of PTM's for rumor detection [14]. However, unlike [14], which learns and then leverages the stance of the tweet for rumor detection with BERT, our proposed framework only uses the content of the source tweet with similar example retrieval and ensemble learning. Therefore, the technique presented in this work is different and novel when compared to existing literature. Therefore, in this work, we propose a framework based on the pre-trained models (BERT and Roberta) to detect rumors on a popular social media platform Twitter. We leverage and combine the knowledge from similar retrieved posts and choose the best-performing model from the ensemble of models based on the retrieved train label. The motivation behind the retrieval of similar micro-blogs from the train set is that similar tweets with similar semantic features should have similar labels.

Our main contributions are outlined as follows:

- We propose a framework that leverages retrieval of similar tweets and ensemble models to detect rumors on Twitter.
- Experiments on two publicly available datasets for Twitter [15], shows an improvement (by 1337 hours) in early rumor detection by only using 0.4% of total tweets/retweets in a tweet cascade.
- Our proposed framework can detect tweets with equivalent [11], or better performance [10] to that of propagation-based models.
- We provide brief explanations for the predictions made by the framework.

II. Related Works

The literature for rumor detection models can be majorly divided into two categories (i) Hand-crafted features, (ii) Deep-Learning and Propagation structure-based models [16] to detect the rumors for the social media platforms. Extensive research has been conducted to investigate the veracity of the posts delivered on social media platforms. Researchers define

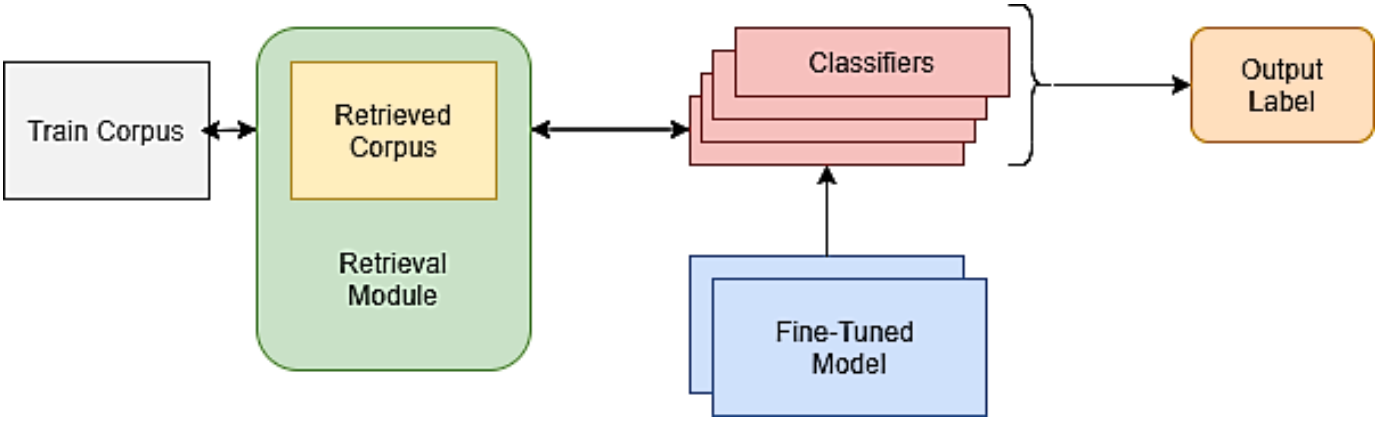

Fig. 1. High level structure of the Proposed Framework.

the problem of Rumor Detection as a multiclass classification problem. The scope of all the works discussed in this section is restricted to the two popular social media platforms (i) Twitter and (ii) Sina Weibo. Both Twitter and Weibo are microblogging platforms where users can communicate with other users using text (tweets), short videos, and images.

A. Hand-Crafted Features

Classification-based tasks like rumor detection require feature engineering techniques to extract/generate important features. This is because the quality of the dataset influences the quality of the trained models. Therefore in earlier models, features were hand-crafted from the tweets to learn good decision boundaries. [17] uses text features like word length, uncommon words, frequently occurring words, characters, and the average length of the word in a rumor, from the primary posts and comments of the same post to represent the post. In [18] a new approach is presented that uses time series of rumor's life-cycle to capture the temporal features. A sequential modeling technique is applied to the extracted features to gather the social context information. Syntactic elements of text like part of speech tags are used in [19]. Some of the works also incorporate polarity as a feature in their works [17]. [20] used visual elements like profile images of the users to classify the rumors further.

B. Deep-Learning and Propagation structure-based models

However, despite the success of the hand-crafted features, they are not effective for newer datasets. Designing hand-crafted features require expertise and time. Therefore, the recent research incorporates deep learning techniques which does not rely much on feature engineering.

Deep learning-based techniques can be further categorized based on the type of neural models used.

Recurrent Neural Network: The first recurrent neural network to detect rumors was used in [3]. In this work post, reposts and comments were considered separated entities and were batched together according to the time. Then TF-IDF was used to find top terms to represent each batch and ran RNN. An attentive RNN model was used in [3], [4]. It performed attention over the tokens to find the important tokens, which significantly affects the post classification. In [5], a recursive neural model uses a bottom-up and up-bottom tree structure to classify posts. [21] uses a Generative Adversarial Networks where the classifier and corresponding generator improve the discriminator by generating conflicting noise. [22] combines visual information and textual information in the RNN for rumor detection.

Convolutional Neural Network: In [6] every rumor event is split into several phases, which are then split into different groups. Doc2vec is used to represent each group passed through two-layer convolution neural networks, obtaining the final results of two-class classification. [7] extract high-level interactions among significant features using CNN, and later RNN is used on the data passed from CNN.

Graph Based: [23] has two different components. Initially, a Graph Convolution Network is used to encode a user's attributes and behaviors. The propagation tree encoder encodes the tree structure, and both GCN and tree encoders are then combined to identify rumors. Similar to [5] both the upward propagation and downward propagation of information for propagation tree is used; however, in this particular work [8] bidirectional GCN is used in place of recursive RNN. In [11] a simplified graph neural network is proposed to learn the relationship between rue rumors and false rumors. [9] uses a union graph to incorporate all tweets' propagation structures to deal sparseness of the graph structure.

BERT Based: [14] leverages the tweet's stance and the BERT model to recognize the rumor posts.

III. Methodology

A. Overview

Pre-trained models are "knowledge bases" trained on a large corpus of data, which can recall information to the available domain knowledge [24]. This is particularly one of the primary reasons for the pre-trained models' success in the natural language understanding tasks. Pre-trained models have shown their effectiveness in multiple domains like healthcare, natural language understanding, software engineering, etc. However,

the pre-trained model's boundaries for the rumor detection task have still not been tested exhaustively.

This paper proposes a framework that leverages an ensemble of classifiers and retrieval module based on BERT and Roberta for rumor detection. The overview of the framework is shown in Fig. 1. As shown in the figure, the proposed framework contains fine-tuned models BERT and Roberta. The output from the fine-tuned models is used with different classifiers. Ensemble of classifier interacts with the retrieval module, which is used to retrieve "similar tweet" and corresponding "label" from the training samples for each sample in the test set. The retrieved samples and labels are stored in the "Retrieved Corpus". The ensemble of classifiers are trained using the retrieved corpus and test corpus. Finally, the ensemble of models leverages the retrieved label to select the best performing model from the group of all the trained models and predict the output label. The details for each fine-tuning, ensemble classifiers, and retrieval module are explained in the following sections.

B. Fine-Tuning

Pre-trained language models learn the general natural language understanding of a language. Therefore, we need to fine-tune a PTM to a particular domain to understand a given domain better. Thus, in this work, we fine-tune the BERT and Roberta on the Twitter15 and Twitter16 datasets [15]. The fine-tuning helps the models adapt to a given dataset. The fine-tuned models are then used to extract the embedding for each tweet and then fed into the classifier. For the fine-tuning, We employ the widely used HuggingFace library [25] as it provides a single source for all the PTM's used in this work.

C. Classifiers

In our work, we experiment with different classifiers. We use three different classifiers, namely (i) Support Vector Machine, (ii) Logistic Regression, and (iii) a simple, fully connected deep-learning model with linear layers. The fine-tuned model is used to generate the input for each of the classifiers. To generate the input data, we use special tokens available within BERT and Roberta. Previous research has shown that the special tokens *[CLS] and [S]* tend to encode the complete information of the text [26]. We incorporate the information from all the 12 layers using the average vector representation of special tokens as shown in eq (1). The V represents the final vector representation and N is the number of layers in the encoder and cls_n is the vector representation of *[CLS]* token at layer n. For each tweet available with the dataset, we convert it to its vector representation V using the fine-tuned models. The vector representation V serves as input to train different classifiers.

$$V = \frac{\sum_{n=1}^{N} cls_n}{N} \tag{1}$$

D. Retrieval Module

The retrieval of similar documents and incorporating them during the testing phase helps the models to generate better results as seen in NLP and software engineering [27], and [28] respectively. The primary thesis behind retrieval-based techniques is based on similar documents that are complementary to each other [27]. We extend this thesis to "similar tweets should have similar rumor-based classes." Therefore, we use the retrieval-based technique in two different ways, as shown in Fig. 2.

In the first technique, we retrieve the most similar documents from the training set for each of the test examples. To find the most similar document corresponding to each test example, we use a cosine similarity score to find the vector distances between each test tweet and train tweet. We use cosine similarity because it is a standardized metric used to find vector distances and can easily find duplicates if both the vectors are same. As shown in eq (2), x represents the training sample, and Y represents the test sample. $\| x \|$ and $\| Y \|$ represents the modulus value of x and Y. Therefore, $sim(x, Y)$ represents the similarity score between sample x of the train set with sample Y from the test set. The $sim(x, Y)$ score ranges between 0 and 1. To represent sample x and Y to its vector representation, we use the vector representation V from the fine-tuned models as shown in eq (1).

$$sim(x, Y) = \frac{x.Y}{\| x \| \| Y \|} \tag{2}$$

Finally, the document with the highest similarity score is selected as shown in eq (3). The $RetDoc$ is the document with the highest similarity score, and n represents the number of samples in the train set.

$$RetDoc = argmax[sim(x_1, Y), ...sim(x_n, Y)] \tag{3}$$

To form the retrieved corpus, each retrieved document and its corresponding test document are added together as shown in eq (4). This retrieved corpus is then stored in the memory and used to predict test documents' labels. We also experimented with the weighted addition between the retrieved tweet and test tweet to form the retrieved corpus. However, we got better results with the non-weighted addition as shown in eq (4).

$$RetCorpus = RetDoc + Y \tag{4}$$

E. Ensemble Module

We train Support Vector Machine, Logistic Regression, and fully connected model using the train set from the fine-tuned model and retrieved corpus from the retrieval technique. We found that there was no single model that outperforms others. To overcome this problem, in this approach, we select the best-performing model from the ensemble of models and use the chosen model to make the final prediction for the test sample. The best performing model is selected based on the value of best accuracy from the ensemble of models. Now, to select the best model from the ensemble of models, we leverage the label of the most similar tweet from the training set for a given test sample, as shown in Fig. 2. For each most similar document selected from the training dataset, we extract

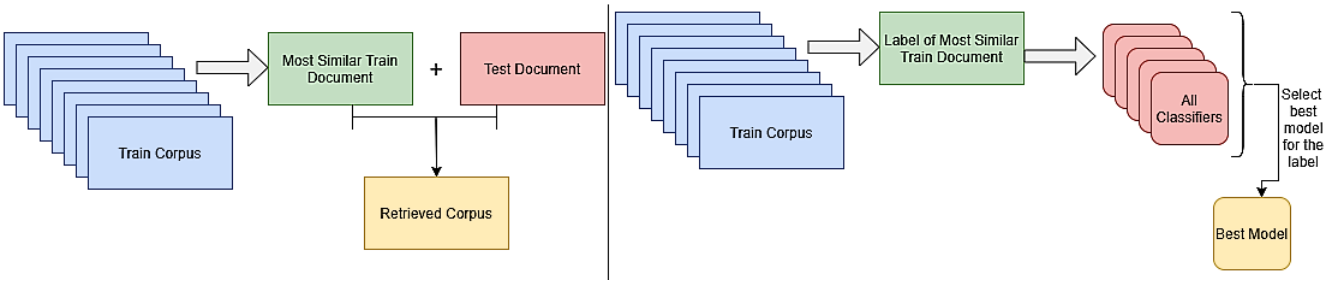

Fig. 2. Retrieval module of the Proposed approach

its label associated with the train set. We use the extracted label to choose the best-performing model from a collection of trained models and use it as a final model for predicting the corresponding test sample. Finally, the best model takes the input and generates the output/prediction for the test example.

IV. Experiments

A. Dataset

We test our proposed framework on two publicly available datasets, namely Twitter15 and Twitter16 [15]. Each dataset contains source tweets and their propagation threads in a tree structure. A propagation thread includes all reposts/retweets made for the tweets. Each sample within the dataset is labeled to one of the four categories, namely (i) non-rumors(NR), (ii) false rumors(FR), (iii) true rumors(FR), and (iv) unverified rumors(UR). The dataset contains 1490 and 818 samples in Twitter15 and Twitter16 datasets, respectively. For the baselines, we directly use the results reported in [5] and [11] and round them to the closest two decimal places.

B. Experimental settings

In our experiments, we use batch sizes 32 and 16 respectively during the fine-tuning of BERT and Roberta's model for Twitter15 dataset and batch size of 32 for the Twitter16 dataset. We used the same train, valid, and test splits as provided in the original dataset to make a direct comparison with existing baselines. We used the HuggingFace library [25] for the fine-tuning, and for the SVM and Logistic regression, we used the sklearn library [29]. All the experiments were conducted on TESLA P100, 16 GB GPU and best performing model is selected based on best accuracy.

C. Metrics

All the result reported in Table I and Table II represents the accuracy of different models, we use the same metrics as reported in [11] and [5]

V. Performance for Rumor Detection

Table I and Table II shows the results obtained by different baselines and model proposed in this work. We use two graph based models GCNII [10] and SAGNN [11]. DTR [30], DTC [17], RFC [31] and SVM-TS [18] models are based on handrafted features. MM-BERT-RET and MM-ROBERTa-RET has been proposed in this work. BU-RvNN [5] and TD-RvNN [5] uses recursive neural networks with bottom up and top down traversal of propagation tree. As shown in Table I and Table II, the proposed model achieves equivalent accuracy to the graph-based models. For both the datasets, we see that either the models based on Graph Convolution Network, Aggregated network, or the approach proposed in this paper performs better than the other baselines. Manual feature engineering-based techniques perform worst, except for the "Non-Rumor" category. Compared to the vanilla BERT and Roberta model, we see a 4% increase in the retrieval framework's accuracy. Therefore the results confirm our thesis that similar tweets should have similar labels. We also see the Roberta-based model achieves the best accuracy within our models as shown in Table I and II. We attribute the better performance of Roberta to the dedicated and focused attention on a particular set of words, contrary to the distributed attention for BERT as shown in Fig. 3. The attention behavior in BERT makes the model attend even to inefficient tokens, which may not be useful in a particular scenario. Hence retrieval of similar tweets can help the model detect the rumors better. The increase in "unverified rumor" performance is most evident using the ensemble models and retrieval technique. We see an increase of almost 5.3% for "unverified" rumor, which insinuates that this category had the highest most similar tweets. Also, as compared to BERT, we see that Roberta had better performance. In the case of the ensemble model, we see the model based on Roberta performed better than the BERT-based model. We attribute Roberta's better performance over BERT to its pre-training object, which has shown that model generalizes well without the next sentence prediction objective [32]. Adding an LR and SVM layer over the BERT and Roberta seems to increase Roberta's performance marginally. This is interesting because the machine learning model used with PTM's performs better than neural network-based models used in combination with PTM's.

Between the graph neural network models, namely SAGNN and GCNII, SAGNN seems to have better performance, which the authors attribute is the advantage of removing sparseness from the graphs.

For the Twitter16 dataset, we don't see a single model performing better than all the baselines. Overall, contrary to Twitter15, we see that the GCNII model performs better than all the baselines; still, the proposed model emerges as

Method	Acc	NR	FR	TR	UR
Decision Tree-based Ranking (DTR) [30]	0.41	0.50	0.31	0.36	0.47
Decision-Tree Classifier (DTC) [17]	0.45	0.73	0.35	0.32	0.41
Random Forest Classier (RFC) [31]	0.56	0.81	0.42	0.40	0.54
SVM Time Series (SVM-TS) [18]	0.54	0.80	0.47	0.40	0.48
SVM Bag of Words (SVM-BOW) [5]	0.55	0.56	0.52	0.58	0.51
SVM Hybrid Kernel (SVM-HK) [33]	0.49	0.65	0.44	0.34	0.34
SVM Tree Kernel (SVM-TK) [18]	0.67	0.62	0.67	0.77	0.64
GRU-RNN [3]	0.64	0.68	0.63	0.69	0.57
BU-RvNN [5]	0.71	0.69	0.73	0.76	0.65
TD-RvNN [5]	0.72	0.68	0.76	0.82	0.65
BERT [12]	0.79	0.77	0.78	0.84	0.76
ROBERTa [32]	0.79	0.77	0.80	0.85	0.76
BERT-LR	0.79	0.76	0.78	0.85	0.76
BERT-SVM	0.79	0.76	0.78	0.84	0.77
ROBERTa-LR	0.80	0.75	0.80	0.86	0.78
ROBERTa-SVM	0.80	0.76	0.80	0.86	0.78
GCNII [10]	0.79	0.77	0.79	**0.87**	0.73
SAGNN [11]	**0.82**	**0.82**	**0.83**	0.86	0.77
MM-BERT-RET(**Ours**)	0.80	0.76	0.82	0.86	0.77
MM-ROBERTa-RET(**Ours**)	**0.82**	0.78	0.82	**0.87**	**0.80**

TABLE I
RESULTS FOR TWITTER15 DATASET.

Method	Acc	NR	FR	TR	UR
Decision Tree-based Ranking (DTR) [30]	0.41	0.39	0.27	0.67	0.34
Decision-Tree Classifier (DTC) [17]	0.46	0.64	0.39	0.42	0.40
Random Forest Classier (RFC) [31]	0.58	**0.75**	0.41	0.55	0.56
SVM Time Series (SVM-TS) [18]	0.57	**0.75**	0.42	0.57	0.53
SVM Bag of Words (SVM-BOW) [5]	0.58	0.55	0.56	0.65	0.58
SVM Hybrid Kernel (SVM-HK) [33]	0.51	0.65	0.43	0.47	0.45
SVM Tree Kernel (SVM-TK) [15]	0.66	0.64	0.62	0.78	0.65
GRU-RNN [3]	0.63	0.62	0.71	0.58	0.53
BU-RvNN [5]	0.72	0.72	0.71	0.78	0.66
TD-RvNN [5]	0.74	0.66	0.74	0.83	0.71
BERT [12]	0.77	0.72	0.75	0.86	0.75
ROBERTa [32]	0.76	0.65	0.77	0.85	0.76
BERT-LR	0.77	0.71	0.76	0.87	0.75
BERT-SVM	0.78	0.71	0.76	0.88	0.75
ROBERTa-LR	0.77	0.68	0.77	0.86	0.77
ROBERTa-SVM	0.78	0.70	0.80	0.86	0.76
GCNII [10]	**0.81**	0.73	**0.83**	**0.91**	0.75
SAGNN [11]	0.80	0.71	0.80	**0.91**	0.77
MM-BERT-RET(**Ours**)	0.79	0.74	0.80	0.88	0.76
MM-ROBERTa-RET(**Ours**)	0.80	0.74	0.79	0.88	**0.79**

TABLE II
RESULTS FOR TWITTER16 DATASET.

the second-best baseline in each category. However, we still see that the proposed model outperforms all the different models in unverified rumors. Overall we see a decrease in the performance from the Twitter15 dataset, which we think is due to the reduced number of samples.

VI. EARLY RUMOR DETECTION

Detecting rumors at an early stage is crucial to stop rumor dissemination. Propagation-based techniques SAGNN [11], and DCNII [10] use the complete propagation structure to detect the rumors with the best accuracy. This means if a source tweet is posted at time x and the last reply/retweet was made to the source tweet after y hours, then the propagation-based techniques require all the information posted between x and y hours for rumor detection. Twitter15 and Twitter16 datasets have an average time of 1,337 and 848 hours, respectively. Hence, the propagation-based techniques require a very high average time and the number of posts (223 and 251) to detect rumors with the best possible accuracy. However, the framework proposed in this paper only uses the source tweet, and no propagation of the tweet is required. Therefore, the possible time taken by the proposed framework for rumor detection accounts for inference time of the models and finding the most similar tweet from the train set, which is few minutes as compared to 1337 and 848 hours. Therefore, the results on Twitter15 and Twitter16 show a significant improvement in early rumor detection for both datasets.

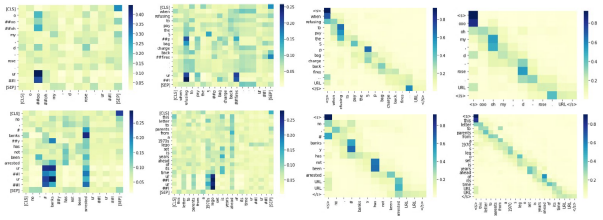

Fig. 3. Attention Distribution for BERT (left) and Roberta (right)

VII. THREATS TO VALIDITY

Internal Validity: We ran our experiments multiple times and have checked for the reproducibility of our work. Therefore we think that there is the least threat to the internal validity of our work.

External Validity: The results reported for the baselines have been used directly from their respective papers. Therefore we assume their correctness as they all are published works. Moreover, we only chose the baselines that maintained the original train, valid, and test split of the dataset and did not change the dataset ordering or reduced the dataset's size for some reason. Therefore, there may be a slight chance of an external threat to the validity.

Construct Validity: The proposed approach in this paper leverages the text of the tweets for rumor detection. However, rumors can also be spread using images, gifs, where propagation-based techniques can be useful compared to the proposed approach. Also, our work highly depends on the similarity of the dataset, so the proposed work may not work effectively on datasets with dissimilar tweets. To the best of our knowledge, we could not find a work that studies the similarities and dissimilarities within the splits of the dataset. Therefore, we believe designing and analyzing the non-similar sampled dataset requires an independent and separate study.

VIII. CONCLUSION

We propose an ensemble models framework to detect rumors which leverages a retrieval mechanism to detect similar tweets and labels in the train set. Results on two public Twitter datasets show that our method can achieve equivalent or better performance (in some cases) than the existing baselines. Our framework can detect rumors in Twitter15 and Twitter16 datasets in few minutes, contrary to propagation-based techniques that take almost 1337 or 868 hours with almost equivalent accuracy. This behavior can be useful for early rumor detection. Moreover, we also explain possible

reasons for such advancement using the attention heat maps. Our work sets up a baseline based on linguistic features, which the researchers can combine with propagation-based techniques.

REFERENCES

[1] Y. Luo, J. Ma, and C. Yeo, "Bcmm: A novel post-based augmentation representation for early rumour detection on social media," *Pattern Recognition*, vol. 113, p. 107818, 01 2021.

[2] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018. [Online]. Available: https://science.sciencemag.org/content/359/6380/1146

[3] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and e. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, p. 3818–3824.

[4] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang, "Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection," *CoRR*, vol. abs/1704.05973, 2017. [Online]. Available: http://arxiv.org/abs/1704.05973

[5] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on Twitter with tree-structured recursive neural networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1980–1989. [Online]. Available: https://www.aclweb.org/anthology/P18-1184

[6] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3901–3907. [Online]. Available: https://doi.org/10.24963/ijcai.2017/545

[7] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news," *CoRR*, vol. abs/1703.06959, 2017. [Online]. Available: http://arxiv.org/abs/1703.06959

[8] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," 2020.

[9] K. Tu, C. Chen, C. Hou, J. Yuan, J. Li, and X. Yuan, "Rumor2vec: A rumor detection framework with joint text and propagation structure representation learning," *Information Sciences*, vol. 560, pp. 137–151, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520312469

[10] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," 2020.

[11] L. Zhang, J. Li, B. Zhou, and Y. Jia, "Rumor detection based on sagnn: Simplified aggregation graph neural networks," *Machine Learning and Knowledge Extraction*, vol. 3, no. 1, pp. 84–94, 2021. [Online]. Available: https://www.mdpi.com/2504-4990/3/1/5

[12] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[14] L. Tian, X. Zhang, Y. Wang, and H. Liu, "Early detection of rumours on twitter via stance transfer learning," in *Advances in Information Retrieval*, J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, and F. Martins, Eds. Cham: Springer International Publishing, 2020, pp. 575–588.

[15] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 708–717. [Online]. Available: https://www.aclweb.org/anthology/P17-1066

[16] J. Cao, J. Guo, X. Li, Z. Jin, H. Guo, and J. Li, "Automatic rumor detection on microblogs: A survey," *CoRR*, vol. abs/1807.03505, 2018. [Online]. Available: http://arxiv.org/abs/1807.03505

[17] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 675–684. [Online]. Available: https://doi.org/10.1145/1963405.1963500

[18] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1751–1754. [Online]. Available: https://doi.org/10.1145/2806416.2806607

[19] A. Hassan, V. Qazvinian, and D. Radev, "What's with the attitude? identifying sentences with attitude in online discussions," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, Oct. 2010, pp. 1245–1255. [Online]. Available: https://www.aclweb.org/anthology/D10-1121

[20] M. Gupta, P. Zhao, and J. Han, *Evaluating Event Credibility on Twitter*, pp. 153–164. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611972825.14

[21] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 3049–3055. [Online]. Available: https://doi.org/10.1145/3308558.3313741

[22] C. Boididou, S. Middleton, Z. Jin, S. Papadopoulos, D. T. Dang Nguyen, G. Boato, and I. Kompatsiaris, "Verifying information with multimedia content on twitter: A comparative study of automated approaches," *Multimedia Tools and Applications*, 09 2017.

[23] Q. Huang, C. Zhou, J. Wu, M. Wang, and B. Wang, "Deep structure learning for rumor detection on twitter," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[24] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, "Language models as knowledge bases?" 2019.

[25] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," 2020.

[26] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," 2019.

[27] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021.

[28] S. Liu, Y. Chen, X. Xie, J. K. Siow, and Y. Liu, "Retrieval-augmented generation for code summarization via hybrid {gnn}," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=zv-typ1gPxA

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[30] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," 05 2015, pp. 1395–1405.

[31] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *2013 IEEE 13th International Conference on Data Mining*, 2013, pp. 1103–1108.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[33] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *2015 IEEE 31st International Conference on Data Engineering*, 2015, pp. 651–662.