FakeNews Detection Using Pre-trained Language Models and Graph Convolutional Networks

Nguyen Manh Duc Tuan
Toyo Unversity, Japan
ductuan024@gmail.com

Pham Quang Nhat Minh
Aimesoft JSC., Vietnam
minhpham@aimesoft.com

ABSTRACT

We introduce methods for detecting FakeNews related to coronavirus and 5G conspiracy based on textual data and graph data. For the Text-Based Fake News Detection subtask, we proposed a neural network that combines textual features encoded by a pre-trained BERT model and metadata of tweets encoded by a multi-layer perceptron model. In the Structure-Based Fake News Detection subtask, we applied Graph Convolutional Networks (GCN) and proposed some features at each node of GCN. Experimental results show that textual data contains more useful information for detecting FakeNews than graph data, and using meta-data of tweets improved the result of the text-based model.

1 INTRODUCTION

In this paper, we present our methods for two subtasks of the FakeNews Detection Task at MediaEval 2020 [9, 10]. We formalize the FakeNews detection task as a classification problem. In text-based subtask, we applied BERT model [2] which is the state-of-the-art model in many NLP tasks. BERT model has been shown to be effective in many NLP tasks including text classification. We used Covid-Twitter-BERT [8] (CT-BERT), which was trained on a corpus of 160M tweets about the coronavirus. The data used to train CT-BERT has the same domain as the domain of data provided for the FakeNews detection task, and we expect that we can obtain better results with CT-BERT compared with the general BERT models trained on open-domain data. We combined metadata-based features with textual features obtained by CT-BERT and fine-tuned CT-BERT on our task-specific data. Experimental results show that combining metadata with textual features is better than using textual features only. In the structure-based subtask, we adopted Graph Convolutional Networks (GCN) [12] to capture the relations of nodes in retweet graphs.

2 RELATED WORK

One of the approaches to fake news detection is using the content of the news. Content-based features are extracted from textual aspects and visual aspects. Textual information can be extracted by layers of CNN [4]. From textual information, we can observe features that are specific to fake news, such as writing style or emotions [3, 13, 16]. Furthermore, both textual and visual information can be used together to detect fake news [5, 16, 17].

MediaEval'20, December 14-15 2020, Online

We can use social network information to detect fake news by analyzing user-based features and network-based features. User-based features are extracted from users' profiles [7, 11]. Network-based features can be extracted from propagation posts or tweets on the graph [18].

3 APPROACH

In this section, we describe our methods for two subtasks: text-based misinformation detection and structure-based misinformation detection.

3.1 Text-Based Misinformation Detection

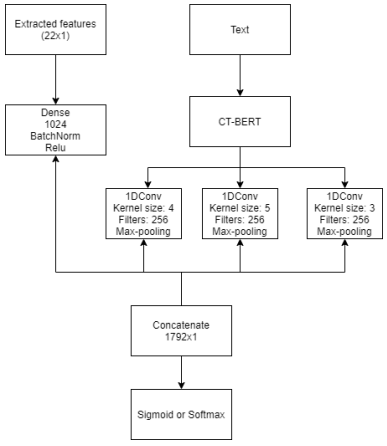

Figure 1: Text-based Fake News Detection Model.

Since tweet data is very noisy, we performed pre-processing steps as follows before putting data into CT-BERT model.

- We deleted mentions and emojis with tweet-preprocessor, a pre-processing library for tweet data.
- We changed the words into lowercase forms.
- There are some emojis written in text format such as ":)", ":(", etc. We changed those emojis into sentiment words "happy" or "sad".
- We deleted punctuation characters that are not useful such as ";", ":", "-", "=".
- We did tokenization, word normalization, word segmentation with ekphrasis [1], a text analysis tool for social medias.

FakeNews detection data is unbalanced, in which the number of tweets labeled as a conspiracy is much smaller than the number of tweets labeled as non-conspiracy. Therefore, we balanced the dataset with Easy Data Augmentation (EDA) method [14].

Pre-processed and augmented data was then put into neural networks. In our work, we conducted experiments with two models as follows.

In the first model, we simply passed a tweet text into CT-BERT and used the hidden vector at [CLS] token as the representation of the tweet. The hidden state at [CLS] is then put into a sigmoid layer for 2-class classification or into a softmax layer for 3-class classification.

In the second model, we combined text-based features with metadata based features in a neural network shown in Figure 1. First, we get the embedding vector of a tweet text using CT-BERT. After that, we used 1D-CNN [6] with different filter sizes. By doing that, we can use more information from various sources for prediction. We passed metadata-based features into a fully-connected layer with batch normalization. Finally, we concatenated metadata features with all outputs from 1D-CNN and passed them into a sigmoid layer for 2-class classification or a softmax layer for 3-class classification. In addition to provided metadata, we extracted other features including the number of retweets, favorites, characters, words, question marks, hashtags, mentions, and URLs in the tweet, the posted time of the tweet, and a binary feature to indicate whether or not it is a sensitive tweet. From users' profiles, we extracted the number of friends, followers, groups, favorites, and statuses that users have posted. We also used the created time and whether or not the users' profiles have been edited, and whether they are verified accounts or not. In total, we extracted 22 features including metadata features.

In experiments, we used the implementation of BERT in the library **Transformers** of HuggingFace [15].

3.2 Structure-Based Misinformation Detection

We applied Graph Convolutional Network [12] (GCN) for structure-based subtask. The model uses traditional GCN on first-order proximity matrix and second-order proximity matrix. The first order proximity is created by adding edges in the original adjacency matrix in order to a directed graph into an undirected graph. The second-order proximity matrix is also an undirected graph and is created by taking into account shared neighbors of each two nodes.

We passed three created graphs into two layers of GCN, with the filter size of 64. After that, we concatenated three output graphs horizontally and then used global max pooling to get the embedded vector of the entire graph. Finally, we passed it into a fully-connected layer of 512 nodes with dropout then added a sigmoid layer for 2-class classification or a softmax layer for 3-class classification.

In GCN, from the input graph, for each node, by using networkx library[1] we created nine features: page-rank, in/out-degree, hub, and authority, betweenness centrality, closeness, number of triangles, eigenvector centrality. For the first run, we use only the nine extracted features as node features. For the second run, we include provided metadata features into node features.

4 RESULTS AND ANALYSIS

4.1 Text-Based Misinformation Detection

We submitted two runs for each of two-class classifier and three-class classifier.

[1]https://networkx.org

Table 1: Evaluation Results for Text-based Subtask

Run	2-class	3-class
Run-1: Tweet only	0.361	0.412
Run-2: Tweet + other features	0.396	0.419

Table 2: Evaluation Results for Structure-based Subtask

Run	2-class	3-class
Graph+extracted features	0.151	0.088
Graph+metadata+extracted features	-0.081	0.151

- **Run-1**: We use the first model presented in Section 3.1 to generate results.
- **Run-2**: We use the second model presented in Section 3.1.

Table 1 shows results of our submitted runs. For the first run with tweets only, we obtained 0.361 of Matthews correlation coefficient (MCC) and 0.412 of MCC for 2-class and 3-class classification, respectively. In the second run, using tweets and other features, we obtained 0.396 of MCC and 0.419 of MCC for 2-class and 3-class classification, respectively.

4.2 Structure-Based Misinformation Detection

We submitted two runs in the structure-based subtask.

- **Run-1**: We used 9 extracted features as node features in graphs.
- **Run-2**: We included metadata-based features along with 9 extracted features as node features.

Table 2 shows the results for two runs. For the first run, we got 0.151 of MCC for 2-class classification and 0.088 of MCC for 3-class classification. For the second run, we got -0.081 of MCC for 2-class classification and 0.151 of MCC for 3-class classification. We can see that metadata-based features did not show their benefits in our GCN model.

In 2-class classification, the second run is better than the first run when we evaluated on the development set. We obtained 0.30 and 0.31 of MCC, respectively. The reason for the performance gap might be that the way we standardized features or split training data is not good.

5 CONCLUSIONS AND FUTURE WORK

We have presented our proposed methods for the two subtasks at MediaEval 2020 FakeNews Detection Task. In the text-based subtask, we have shown that using metadata-based features and other proposed features outperformed the model with only text features. The MCC scores of our proposed models are still low, especially in the structure-based subtask. In future work, we plan to use external resources to compare different information sources and calculate the probability that a piece of information is false. We believe that it is a natural way to detect misinformation.

REFERENCES

[1] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 747–754.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[3] Souvick Ghosh and Chirag Shah. 2018. Towards automatic fake news classification. *Proceedings of the Association for Information Science and Technology* 55, 1 (2018), 805–807.

[4] Rohit Kumar Kaliyar, Anurag Goswami, Pratik Narang, and Soumendu Sinha. 2020. FNDNet–A deep convolutional neural network for fake news detection. *Cognitive Systems Research* 61 (2020), 32–44.

[5] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. MVAE: Multimodal Variational Autoencoder for Fake News Detection. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2915–2921. https://doi.org/10.1145/3308558.3313552

[6] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. (2014). arXiv:cs.CL/1408.5882

[7] S. Krishnan and M. Chen. 2018. Identifying Tweets with Fake News. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. 460–464. https://doi.org/10.1109/IRI.2018.00073

[8] Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. COVID-Twitter-BERT: A Natural Language Processing Model to Analyse COVID-19 Content on Twitter. (2020). arXiv:cs.CL/2005.07503

[9] Konstantin Pogorelov, Daniel Thilo Schroeder, Luk Burchard, Johannes Moe, Stefan Brenner, Petra Filkukova, and Johannes Langguth. 2020. FakeNews: Corona Virus and 5G Conspiracy Task at MediaEval 2020. In *MediaEval 2020 Workshop*.

[10] Daniel Thilo Schroeder, Konstantin Pogorelov, and J. Langguth. 2019. FACT: a Framework for Analysis and Capture of Twitter Graphs. *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (2019), 134–141.

[11] Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. 2019. The Role of User Profile for Fake News Detection. (2019). arXiv:cs.SI/1904.13355

[12] Zekun Tong, Yuxuan Liang, Changsheng Sun, David S. Rosenblum, and Andrew Lim. 2020. Directed Graph Convolutional Network. (2020). arXiv:cs.LG/2004.13970

[13] Yaqing Wang, Fenglong Ma, Z. Jin, Ye Yuan, G. Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018).

[14] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6383–6389. https://www.aclweb.org/anthology/D19-1670

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and others. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* (2019), arXiv–1910.

[16] Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. 2018. TI-CNN: Convolutional Neural Networks for Fake News Detection. (2018). arXiv:cs.CL/1806.00749

[17] Xinyi Zhou, Jindi Wu, and Reza Zafarani. 2020. SAFE: Similarity-Aware Multi-Modal Fake News Detection. (2020). arXiv:cs.CL/2003.04981

[18] Xinyi Zhou and Reza Zafarani. 2019. Network-based Fake News Detection: A Pattern-driven Approach. (2019). arXiv:cs.SI/1906.04210