Detecting Conspiracy Theories from Tweets: Textual and Structural Approaches

Haoming Guo[1*], Adam Ash[1*], David Chung[1*], Gerald Friedland[1]
[1]University of California, Berkeley
mike0221@berkeley.edu,adamash@berkeley.edu
dachung@berkeley.edu,fractor@berkeley.edu

ABSTRACT

The sharing of biased and fake news on social media has skyrocketed in the past few years. These actions have caused real-world problems and harm. The Fake News Detection Task 2020 has two subtasks: NLP-based approach and graph-based approach (Analyzing the repost structure of social media posts). We present baseline models for these two different subtasks and their performance. For the NLP-based approach, Transformers yielded the best results with a Matthews Correlation Coefficient (MCC) score of 0.477. For the graph-based approach, the best results came from a Support Vector Machine (SVM) model with a MCC score of 0.366.

1 INTRODUCTION & RELATED WORK

This paper discusses social media natural language processing and graph-based processing on detecting conspiracy theories. We present our work two subtasks: one that classifies tweets based on their content and metadata (includes images), and another that classifies tweets solely based on their graph-based structure with very little metadata (relative time posted, friends, followers). The task overview paper[10] describes the dataset more in-depth as well as providing information on how the dataset was constructed.[11]

FNC-1, a similar benchmark on fake news and stance detection from texts has received much attention from researchers[4]. Using handcrafted features and a Multi-Layer Perceptron model has proved to perform well on the task[4]. Furthermore, Slovikovskaya et al. showed that fine-tuning transformers achieve state-of-the-art on the benchmark.[12]

Methods incorporating graph structure have proved to be fairly effective in detecting "fake news" consisting of an article shared on Twitter or other social media.[7] Moreover, deep learning methods on graphs of variable size and connectivity have been shown to be effective tools for classification.[3]

This paper presents several prediction models and features for an NLP approach as well as a graph-based approach. We present the performance of these predictors and describe our methodologies.

2 APPROACH

2.1 Bidirectional LSTM

We use a bidirectional LSTM (BiLSTM) as our baseline model for the NLP track. We tokenize and lemmatize each tweet into a list of 55 tokens (pad 0s if below 55), and for each token we get a 300-dimensional embedding from a pretrained word2vec model[6]. Then, we use the vectors to train a biLSTM for classifications. We choose the Adam optimizer, categorical cross entropy loss, 256 units for LSTM and two fully connected layers for our final prediction.

2.2 Transformers

We experiment with pretrained transformers to classify the tweets. BERT, which stands for Bidirectional Encoder Representations from Transformers, was introduced in 2018 and achieved state-of-the-art performance on most NLP tasks[2]. BERT uses a multi-layer bidirectional transformer encoder with a self-attention mechanism to learn a language representation of the input texts. Following BERT, two modifications, XLNet[13] and RoBERTa[5] were proposed to address some of the BERT's shortcomings and outperformed BERT on a variety of tasks.

We use a framework called flair[1] to obtain BERT, XLNet and RoBERTa embeddings separately as 3 sets of features. We use the base 768-dimensional versions of all transformers. Then, we train a fully connected neural network with 2 hidden layers on each set of features to classify the tweets. Best hyperparameters including hidden layer size, learning rate and number of training iterations are tuned separately for each of BERT, XLNet and RoBERTa.

2.3 Basic Graph Features

Some features are hand prepared for each retweet graph. Features are either categorized as being calculated solely based on graph structure, or calculated with the help of separate node information. Examples of features based solely off of graph structure include edge count, node count, number of connected components, and average clustering coefficient. Features based off of both node information and graph structure include average time to retweet, original tweeter's follower count, and percentage of original tweeter's followers who retweeted.

2.4 Computed Graph Features

About 60000 random subgraphs are sampled from graphs in the training set. To create each random sample, ten nodes and their corresponding edges are then randomly chosen from a graph in the training set. Each randomly sampled subgraph is given a 100-dimensional vector corresponding to the subgraph's flattened adjacency matrix and a label corresponding to the label of its source graph. A logistic regression classifier is run on all sampled subgraphs. For each graph in the test set, ten random subgraphs are similarly computed. The average of the model's predictions for

* indicates authors with equal contributions

MediaEval'20, December 14-15 2020, Online

Table 1: NLP Validation Results

Model	Three-class MCC	Two-class MCC
BiLSTM	0.292	0.378
BERT	0.355	0.391
XLNet	0.342	0.426
RoBERTa	0.449	0.471

Table 2: NLP Official Test Results

Model	Three-class MCC	Two-class MCC
XLNet	0.326	0.318
RoBERTa	0.459	0.477

these 10 random subgraphs is added to the test set features for their corresponding graph.

For each graph in the training set, the Graph2Vec package[8] is used to create 64-dimensional representations for the graph's largest (highest node count) subgraph. This representation is used as 64 features for each graph. Also, 64-dimensional representations are taken of each subgraph of each graph, and a weighted average is taken by the number of nodes in the subgraph is added to the features for each graph.

A Deepwalk[9] algorithm is also used to generate a length 64 feature for each node in both the training and test sets. A logistic regression classifier is trained on the Deepwalk feature vectors for each node in each graph in the training set. For each graph in the test set, the average of the predictions for each node was used as a feature.

3 RESULTS AND ANALYSIS

3.1 Text-based Approaches

We split the data into an 80% training set and a 20% validation set. We evaluate our results using Matthews correlation coefficient (MCC), which is considered a balanced measure even for unbalanced data distributions. We present validation results and the official test set results in tables 1 and 2.

All transformers outperforms the baseline BiLSTM as in many other NLP tasks, but among the transformers RoBERTa significantly outperforms the other. We analyze the reasons behind it. First, BERT and XLNet are pretrained on 16GB of Book Corpus and English Wikipedia, while RoBERTa is pretrained on an additional 144GB of CommonCrawl News dataset, Web text corpus, and Stories from Common Crawl. We think that this additional data not only improves RoBERTa's generalizability, but it also makes the model more suitable for news subjects and informal language. Secondly, RoBERTa removes the Next Sentence Prediction (NSP) training objective. The NSP objective was hypothesized to improve performance on tasks that require reasoning on pairs of sentences, which is not a key element in our task. Liu et al. also showed in his paper the uselessness of the NSP objective in many settings. Therefore, the removal of NSP loss is another possible reason why RoBERTa performs the best.

Table 3: Structural Approach (Graph Only) Validation Results

Model	Three-class MCC	Two-class MCC
SVM	0.308	0.306
Random Forest	0.321	0.304
Neural Net	0.288	0.326

Table 4: Structural Approach Validation Results

Model	Three-class MCC	Two-class MCC
SVM	0.276	0.389
Random Forest	0.370	0.115
Neural Net	0.263	0.338

3.2 Structure-based Approaches

We evaluate our results using MCC as discussed in the previous subsection. Once again, the data is split into an 80% training set and a 20% validation set. Different models are tested for both the two-class and three-class problems. Models tested are SVM, neural nets, and random forest. The neural net has three layers of 64 Rectified Linear Units each, with a Sigmoid function output layer.

For the two-class problem, a SVM model with a radial basis function kernel outperforms the other models on validation sets. For the three-class problem, a random forest model with 40 estimators and a maximum depth of four nodes outperforms the others. Average validation results using features computed only from the graph structure are shown in Table 3, and average validation results using features computed from all available data are shown in Table 4.

Our final classifiers are run on a test set roughly one third the size of our training set. Our two-class SVM model receives an MCC of 0.370. Our three-class random forest model receives an average MCC of 0.318. It is not surprising that our two-class classifier performs better, as using two classes instead of three leads to a dataset with much less ambiguity.

4 DISCUSSION AND OUTLOOK

Above, we presented and experimented with several methods to detect conspiracy theories from social media content based on their text and graph structure. Overall, a transformer-based approach exhibited the best performance for text-based classification, while SVM/Random Forest trained on our crafted graph features proved to be the best on structure-based classification.

There are many ways to extend the methodologies described above. Below we list some possible ways of furthering our work.

(1) Our preliminary experimentation with the provided metadata yielded worse results than our transformers-based approaches. Further experiments could be done to determine whether training a classifier on the metadata would yield better results.

(2) In this paper, we focused on text-based approaches and structure-based approaches separately for the specific sub-tasks. Incorporating different modalities such as analyzing tweet texts, tweet structures, metadata, and images associated with the tweet could prove to be useful.

REFERENCES

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*. 1638–1649.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[3] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. 2015. Convolutional Networks on Graphs for Learning Molecular Fingerprints. (2015). arXiv:cs.LG/1509.09292

[4] Andreas Hanselowski, Avinesh P.V.S., Benjamin Schiller, Felix Caspelherr, Debanjan * Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*. http://tubiblio.ulb.tu-darmstadt.de/105434/

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[6] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[7] Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M. Bronstein. 2019. Fake News Detection on Social Media using Geometric Deep Learning. (2019). arXiv:cs.SI/1902.06673

[8] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. 2017. graph2vec: Learning Distributed Representations of Graphs. (2017). arXiv:cs.AI/1707.05005

[9] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. *CoRR* abs/1403.6652 (2014). arXiv:1403.6652 http://arxiv.org/abs/1403.6652

[10] 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*.

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

[12] Valeriya Slovikovskaya. 2019. Transfer Learning from Transformers to Fake News Challenge Stance Detection (FNC-1) Task. *CoRR* abs/1910.14353 (2019). arXiv:1910.14353 http://arxiv.org/abs/1910.14353

[13] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *CoRR* abs/1906.08237 (2019). arXiv:1906.08237 http://arxiv.org/abs/1906.08237