Enriching Content Analysis of Tweets using Community Discovery Graph Analysis

Andrew Magill, Lia Nogueira De Moura, Maria Tomasso, Mirna Elizondo, Jelena Tešić
Computer Science, Texas State University, San Marcos TX U.S.A.
{a_m730,l_n63,met48,mirnaelizondo,jtesic}@txstate.edu

ABSTRACT

This paper describes the proposed solutions of the Texas State University Data Lab team to the MediaEval 2020 FakeNews benchmark. We have responded to the text-based and structural-based tasks with lexical, graph, and community labeling approaches. Our lexical analysis approach using logistic regression produces the best results at 0.43 MCC, we describe a promising community labeling model, and discuss our attempts to find predictive structural features in retweet graphs of conspiracy promoting tweets.

1 INTRODUCTION

Our analysis assumes that people in the same social network community who agree on fake news also write in a similar style, discuss similar topics, produce similar content, and share similar values. We relate content of the tweets using lexical analysis, employ community discovery by building a network of re-tweets, and employ network analysis on structural data provided.

2 RELATED WORK

Analysis of tweet content spans from the use of *Bag-Of-Words* features in classification model to capture most likely terms associated with fake news [1] to using lexical analysis to characterise writing style in fake news articles [6]. Community-based modeling of social networks that leverages the spread of information in social media through re-tweets and comments has been shown to improve NLP-based modeling [6], and entire-graph clustering has shown promise in community identification on large scale [5]. Structural modeling is in its infancy for social networks, based on the promising direction on applying deep neural network classification of URLs as fake or trusted based on the propagation patterns [2].

3 APPROACH

3.1 Text-Based Misinformation Detection

Twitter restricts tweet content to 280 characters, a constraint that tends to influence a writing style that differs from that found in most corpora. To achieve brevity, users employ a lexicon that includes abbreviations, colloquialisms, *hashtags*, and *emoticons*. Tweets may also contain frequent misspellings. The context of a tweet is also richer as it resides in a rich network of retweets and replies. To this end, we employ lexical-based analysis and community analysis for tweet content and context.

Lexical Analysis Pipeline implements transformation of twitter content, feature extraction, and modeling, to make predictions for the NLP-based task [4]. We analysed the efficacy of common preprocessing techniques and tokenization patterns to extract the most useful features for prediction. Effective preprocessing techniques included: transforming text to lowercase, removing terms very common to all classes (*stopwords*), removing punctuation, preserving URLs, and normalizing several specific terms ('u.k.' to 'uk'). Tokenization patterns that preserved and uniquely encoded emoticons and punctuation did not improve predictive performance. Normalizing our text using stemming and lemmatization, methods that map similar but distinct terms to the same encoding, produced mixed results. We decided that it may be more beneficial to preserve distinctive characteristics of our text, and we opted to leave these processes out of our final pipeline.

Feature extraction in text can be accomplished by encoding terms as vectors representing either the occurrence of terms in text (*Bag-Of-Words*), or the impact of terms to a document in a corpus (TF-IDF). We employed a TF-IDF vectorizer to reduce the impact of repeated terms within individual tweets, but did not see improvements over the *Bag-Of-Words* model in our pipeline. We trained a set of **classifiers** (Naive Bayes, SVC, Random Forest, and Logistic Regression) on our extracted feature vector, and analysed the performance of both the fine-grained four class and coarse-grained two class classification sub-tasks. To account for imbalance in data, we have experimented with data augmentation, generating fake tweets using the most predictive or most common terms for each class. This approach led to overfitting of most classifiers, and we have dropped it early on. We have extended the feature set in the tweets using Optical Character Recognition of images embedded in tweets. We have also adjusted class weights to account for imbalanced data, when possible. Logistic regression showed superior performance in all our test runs, and in our submission we have included runs labeled LR (Logistic Regression) and LR-OCR (Logistic Regression with Optical Character Recognition text augmentation) in Table 1.

Community Analysis Pipeline applies community finding work [5] to assign discovered communities in large social networks. We extend the provided dataset with an auxiliary dataset that contains tweets related to the hashtags #Coronovavirus, #Covid19, and #Covid-19, collected from March to September 2020, with over 3.2 million users and 8 million tweets [5]. First, we create three different networks from the raw data: *User Connections* from provided data: vertex is a user, and each edge is the connection between two users by either a retweet, quote, reply, or mention; *Hashtag Connections* from provided data: vertex in the network is a hashtag, and edge exists between two hashtags if they were used together in the same tweet; and *User Connections 8M*: a network created from provided data and the auxiliary dataset of over 8M tweets, where vertices and edges of the network created the same way as the *User Connections* network. Next, we extract the degree of connectivity for each of the provided conspiracy labels (5G, non, and other) driven by observation that if vertices are well connected their content is

MediaEval'20, December 14-15 2020, Online

similar. We employ the *Louvain Community* discovery method to discover communities in all three networks, and apply to specific tweets information from each network analyzed [5]. We labeled each community with one of the three conspiracy categories (5G, non, other), based on majority of the labels for that community associated with the tweet label. If we found a community where 5G labels are larger than non or other, we will use 5G label to assign the label to unlabeled tweets in that community. These assignments were done based on the combination of communities found in all three networks. Tweets that did not belong to any community, or belonged to a community with tweets strictly originating from the test dataset, were assigned based on their degree of connectivity, and the remaining were assigned as *Unknown*. Many unknowns were found because a large number of tweets did not have any connections with other users in the given datasets (no retweets, replies, quotes, mentions, or hashtags). The community discovery approach can be useful for datasets where the users are well connected to each other, as shown in runs labeled as CL in Table 1. **Fusion Run** for both methods is labeled LR-CL in Table 1: it implements simple fusion algorithm: for all tweets where confidence for LR is under certain threshold, CL label is used. Details are described in Sec. 4.

3.2 Structure-Based Misinformation Detection

Standard summary statistics for graphs are used here as feature vectors. We extract 19-dimensional feature vector from each provided graph adjacency matrix using 'igraph' R package, and the dimensions represent: number of nodes, number of edges, diameter of the graph, mean distance, edge density, reciprocity, global transitivity, local transitivity, number of triangles, mean in-degree, maximum in-degree, minimum in-degree, mean out-degree, maximum out-degree, minimum in-degree, mean total degree, maximum total degree, and minimum total degree. Feature vectors are normalized and fed into a series of python scikit-learn classifiers: (1) Decision tree with no maximum depth and the Gini impurity or entropy criterion; (2) Linear discriminant analysis (LDA) with SVD, LSQR, and LSQR plus shrinkage, and (3) Naive Bayes. An 80/20 train/test split of the development data was used to train the models. Each classifier was trained for coarse classification and 4 way fine classification.

4 RESULTS AND ANALYSIS

Text-Based Misinformation Detection: for each of the coarse-grained and fine-grained classifications the team has submitted one run of predictions from our community labeling model, two runs from our lexical analysis pipeline, and one run combining the two approaches, for a total of eight sets of predictions. Table 1 summarizes our returned test set results, and our own evaluations on the development set using the provided ground truth labels. **Lexical Analysis Pipeline** using logistic regression produces the highest MCC for both classification subtasks. Note that OCR text augmentation does not improve the MCC on test set, even as it shown improvements for multi-class subtask for development set.

Community Analysis Pipeline only run does not achieve high MCC on test set, but it does contribute to comparable MCC on test set (0.363), and higher precision, and comparable recall and accuracy on development set. We ran out of time to implement meaningful fusion of lexical and community runs. Internal analysis showed that number of tweets that are isolated from the network degrades the performance of community based approach.

Evaluation Set		**Test**	**Development**			
Run	Model	MCC	MCC	Prec	Recall	Acc
001	LR	**0.397**	0.457	0.562	0.545	0.760
002	LR-OCR	0.363	0.465	0.599	0.565	0.767
003	CL	0.081	0.170	0.388	0.229	0.281
004	LR-CL	*0.363*	0.442	0.462	0.430	0.725
011	LR	**0.436**	0.516	0.780	0.737	0.862
012	LR-OCR	0.428	0.499	0.769	0.732	0.854
013	CL	0.091	0.219	0.604	0.615	0.748
014	LR-CL	0.091	0.244	0.613	0.631	0.743

Table 1: Multi-class (runs 001 - 004) and coarse (runs 011 - 014) text-based information detection scores for test run return by benchmark (MCC), and development set released ground truth (MCC, Precision, Recall, Acc). Model abbreviations: LR for logistic regression; LR-OCR for logistic regression w OCR; CL for community labeling; LR-CL for fusion run.

Evaluation Set		**Test**	**Development**			
Run	Model	MCC	MCC	Prec	Recall	Acc
101	NB	0.0116	0.176	0.469	0.423	0.691
102	LDA-SVD	**0.0124**	0.060	0.387	0.356	0.707
103	LDA-LSQRS	-0.0145	0.057	0.398	0.351	0.711
104	DT-gini	-0.0288	0.048	0.356	0.356	0.572
105	LDA-LSQR	0.0124	0.049	0.375	0.354	0.702
111	NB	-0.0014	0.158	0.581	0.576	0.859
112	LDA-LSQR	**0.0327**	0.070	0.579	0.516	0.896
113	LDA-SVD	**0.0327**	0.070	0.579	0.516	0.896
114	DT-gini	0.0068	0.015	0.507	0.508	0.809
115	DT-entropy	-0.0481	0.017	0.508	0.509	0.811

Table 2: Multi-class (runs 101 - 105) and coarse (runs 111 - 115) structure-based detection scores for test run return by benchmark (MCC), and development set released ground truth (MCC, Precision, Recall, Acc). Model abbreviations: NB for Naïve Bayes; LDA for linear discrimination analysis; SVD for singular value decomposition; LSQR for least square; LSQRS for least square shrinkage; DT-gini for decision tree with Gini index; DT-entropy for decision tree with entropy.

Structure-Based Misinformation Detection runs are reported in Table 2, where the highest overall MCC on the test set was 0.0327 for LDA based coarse classifiers. There is not enough information to mine in the provided structure data to capture meaningful relations, as some of the runs produce MCC lower than random. Note that proposed community-based network produces higher MCC scores (Table 1) alone as it takes into account hashtags, retweets, and community discovery over larger corpora.

5 DISCUSSION AND OUTLOOK

Lexical based analysis produced the highest MCC score on the test set. Our community discovery method showed some promise in the fusion approach with increased precision. Community-based and structure-based methods will likely contribute more if we consider conspiracy vs non-conspiracy classification, as recent work has shown different dispersion patterns regardless of the conspiracy topic [3]. Next steps are refined fusion approach and use of community-based scores as features for structure-based approach.

REFERENCES

[1] Indra et al. 2016. Using logistic regression method to classify tweets into the selected topics. In *Intl. Conf. on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, NY, 385–390.

[2] Monti et al. 2019. Fake News Detection on Social Media using Geometric Deep Learning. (2019). arXiv:cs.SI/1902.06673

[3] Vosoughi et al. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

[4] Andrew Magill and Maria Tomasso. 2020. Fake News Twitter Data Analysis. https://github.com/DataLab12/fakenews. (2020).

[5] Lia Nogueira. 2020. *Social network analysis at scale: Graph-based analysis of Twitter trends and communities*. Master's thesis. Texas State University. https://digital.library.txstate.edu/handle/10877/12933

[6] Zhou and Zafarani. 2019. Fake News Detection: An Interdisciplinary Research. In *WWW Proceedings*. ACM, NY, 1292.