RMIT at PAN-CLEF 2020: Profiling Fake News Spreaders on Twitter

Notebook for PAN at CLEF 2020

Xinhuan Duan, Elham Naghizade, Damiano Spina, and Xiuzhen Zhang

RMIT University, Melbourne, Australia
s3713321@student.rmit.edu.au, {e.naghizade, damiano.spina, xiuzhen.zhang}@rmit.edu.au

Abstract Automatic detection of fake news in social media has become a prominent research topic due to its widespread, adverse effect on not only the society and public health but also on economy and democracy. The computational approaches towards automatic detection of fake news span from analyzing the source credibility, user credibility, as well as social network structure and the news content. However, the studies on user credibility in this context have largely focused on the frequency and times of engaging in a fake news propagation rather than profiling users based on the content of their tweets. In this paper, we approach this challenge through extracting linguistic and sentiment features from users' tweet feed as well as retrieving the presence of emojis, hashtags and political bias in their tweets. These features are then used to classify users into spreaders or non-spreaders of fake news. Our proposed approach achieves 72% accuracy, being among the top-4 results obtained by systems for the task in the English language.

1 Introduction

This paper describes our participation to the Profiling Fake News Spreaders on Twitter task at PAN-CLEF 2020 [12]. Given a Twitter feed, the aim is to determine whether or not its author is a spreader of fake news. Although the task includes both English and Spanish languages, we only addressed the problem for the English language.

Fake news has been proven to be both harmful and misleading to people. In the 2016 US election, most of the users in Twitter have encountered at least one fake news each day [1]. To prevent people from being misled by fake news, one of the main problems to address consists of identifying fake news spreaders. We tackle this problem by proposing a two-step learning approach that (i) aims to model sentiment, political presence, and use of language of fake news spreaders at tweet level, and (ii) generates a profile-level representation to feed a binary classifier used to classify profiles into spreaders or non-spreaders of fake news. Our model achieves a 70% accuracy in a 10-fold cross validation using the training set provided by the organizers.

The rest of the paper is organized as follows. Section 2 describes our proposed approach. Section 3 reports the implementation details needed to reproduce our approach. Section 4 discusses the results. Finally, Section 5 concludes the work and discusses future directions.

2 Proposed Approach

Several recent studies have focused on the role of users in social networks on the spread of fake news. These studies have largely focused on the information provided by users in their profile [9] or the number of times that users engage in fake news propagation, e.g., through retweeting them [13].

However, in this task, the focus is not on detecting the veracity of a piece of news but rather if a user is a fake news spreader given the content of their tweet feed. We proposed profiling fake news spreader as a supervised learning task. Figure 1 shows the main components of our proposed model for this task. As can be seen, the model carries two major steps, denoted as tweet-level and profile-level representation.

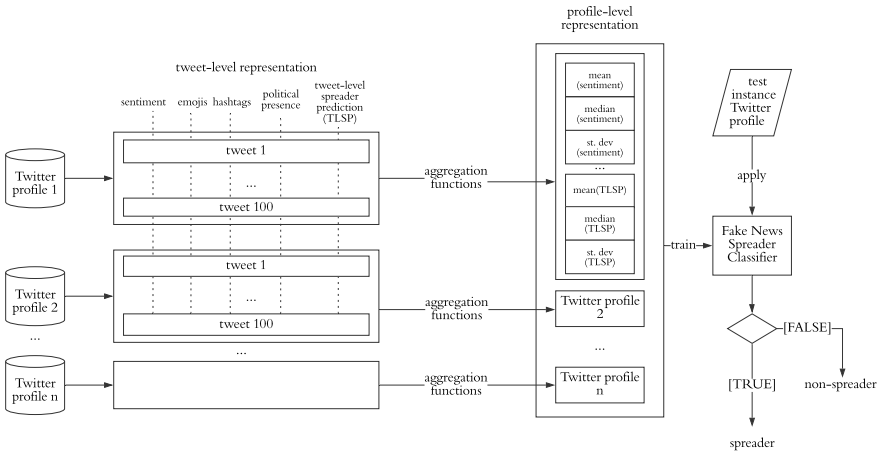

Figure 1. Architecture of the proposed approach.

2.1 Tweet-Level Representation

The feature extracted in the tweet-level representation phase are listed in Table 1, and the justifications for the features are listed as below:

– Sentiment: Recent studies suggest that the sentiment of tweets can help detect the credibility of a piece of news or its spreader [5,6]. As a result, we include sentiment polarity as one of our extracted features.

Table 1. Features used to in the tweet-level representation.

Feature	Description
Sentiment	Sentiment intensity of the tweet, represented as a numeric value between -1 (negative) and 1 (positive).
Hashtags	Number of hashstags in the tweet.
Emojis	Number of emojis in the tweet.
Political Presence	Number of political entities (e.g., "Trump") mentioned in the tweet.
Tweet-Level Spreader Prediction (TLSP)	Content-based fake-news spreader prediction at tweet-level.

- Emojis and hashtags: To help strengthen the sentiment signal, we include the number of emojis and hashtags in the tweets as additional features.
- Political presence: Recent studies suggest that fake news spreaders and non spreaders have different political presence [14], hence we proposed to study the relation between political preferences and the behavior of spreading fake news.
- Content: It has been proved that spreading fake news on social media is a very rare behavior [7]. Only features based on user profiles may not be sufficient. We train a language model (a fine-tuned BERT model) using the tweet content to extract relevant language features that can distinguish the desired two classes of users. Figure 2 illustrates this process.

The mapping of training data is based on the following rule: all tweets belonging to a profile labeled in the training set as a fake news spreader will be labeled as *spreader*; analogously, tweets that belong to a non-spreader profile in the training set will be labeled as *non-spreader*.

2.2 Profile-Level Representation

After extracting the features at the tweet-level representation phase, we build the profile-level representation (as shown in Figure 1). Given the tweets for a Twitter profile, the values for each feature are aggregated using three functions: mean, media, and standard deviation. The result of this process is a profile-level vector that includes these three aggregated scores for each of the feature in the tweet-level representation, which can be used to train a profile-level fake news spreader classifier.

3 Implementation

The implementation of our submission to the evaluation campaign is publicly available[1] and the details are described below.

[1] http://github.com/rmit-ir/pan2020-rmit

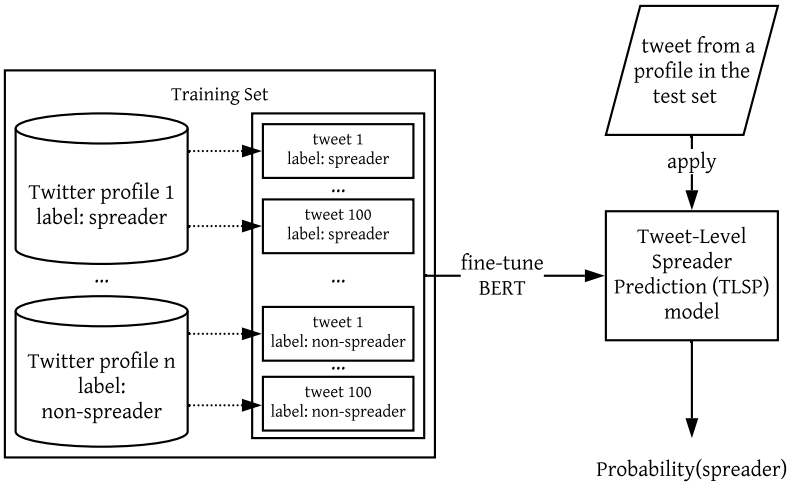

Figure 2. Architecture of the Tweet-Level Spreader Prediction (TLSP) model.

3.1 Data Preprocessing

The training data provided by the organizers contain tweets in two languages: English and Spanish [12]. Each language consists of 300 Twitter profiles, and each profile has 100 tweets. Due to time constraints, we only addressed the task for the English language. The training dataset is balanced, i.e., half of the Twitter profiles are fake-news spreaders, while the other half are labeled as non-spreaders. Only the content of tweets is provided. In our experiments, the tweet content is preprocessed using the tokenizer provided by the library used to implement the TLSP model, described in Section 3.2.

3.2 Tweet-Level Spreader Prediction

The TLSP has been instantiated as a fine-tuned BERT binary classifier. BERT stands for Bidirectional Encoder Representations Transformers, and is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers [4]. The pre-trained BERT model are used to get the word embedding of the text and the embeddings are then fed into a Gated Recurrent Units (GRU) [3] to produce a prediction of the probability of the tweet belonging to a fake-news spreader.

We used the BERT implementation in the `Transformers` library[2] and the GRU implementation in the `torch.nn` library[3]. The parameters used to fine-tune the model are listed in Table 2.

[2] https://huggingface.co/transformers/model_doc/bert.html
[3] https://pytorch.org/docs/master/generated/torch.nn.GRU.html

Table 2. Parameters used to fine-tune BERT using Gated Recurrent Units (GRU) [3] with the `torch.nn.GRU` library. For the pre-trained BERT model in the `Transformers` library, we instantiated a configuration with the defaults, which is a similar configuration to that of the BERT `bert-base-uncased` architecture.

Parameter	Value	Description
Training/Validation split	90%/10%	Percentage of data used for training and validation, respectively.
Number of epochs	5	Number of passes of the entire training dataset.
Batch Size	128	Number of training samples in each iteration.
`hidden_size`	256	Number of features in the hidden state.
`num_layers`)	2	Number of recurrent layers.
`bidirectional`	True	Use bidirectional GRU.
`batch_first`	True	The input and output tensors are provided as (batch, seq, feature).
`dropout`	0.25	The dropout layer on the outputs of each GRU layer.

The TLSP model is trained with the all the training data available. The model is then applied to the tweets in the validation set. The predicted probability of being a *fake-news spreader* tweet is then used as the TLSP feature in the tweet-level representation.

3.3 Tweet-Level Feature Representation

Besides the TLSP score, we computed the rest of the tweet-level features as follows:

- Sentiment: We used the VADER sentiment analysis system to compute the sentiment intensity of each tweet [8], using the implementation provided by the authors.[4] Given a tweet, the sentiment polarity is represented as a numeric score between -1 (negative) and 1 (positive).
- Emojis and hashtags: For each tweet, the frequency of emojis and hashtags is computed. To extract emojis, We used the `spacymoji` library.[5]
- Political Presence: We aimed to model a political profile of the Twitter users. However –due to time limitations– the current version only indicates the presence of the term `trump`.

3.4 Profile-Level Classification

As it is shown in the Figure 1, the final model building firstly transforms the tweet-level representation to the user profile-level representation. After aggregating the features that are retrieved at the tweet-level to retrieve the profile-level features, this vector is fed to an SVM classifier[6] to create the final model.

[4] https://github.com/cjhutto/vaderSentiment

[5] https://spacy.io/universe/project/spacymoji

[6] We used the scikit-learn implementation of C-Support Vector Classification (`sklearn.SVM.SVC`) with default paramenters.

4 Results

4.1 Preliminary Results Using the Training Data

In order to define the run used to make the official submission, we tried different combinations of features for the tweet-level representation, as well as different machine learning algorithms as the fake news spreader classifier. Table 3 shows the effectiveness in terms of Accuracy, Precision, and Recall for different runs, using 10-fold cross-validation over the training data.[7] Using all features and SVM as the profile-level classifier obtained the best results, and this is the run we submitted via the TIRA Integrated Research Architecture [10].

Table 3. Results using 10-fold cross-validation over the training data. Submitted run in boldface.

Tweet-Level Representation	Fake-News Spreader Classifier	Accuracy	Precision	Recall
TLSP + Sentiment	Logistic Regression	0.6567	0.6707	0.6469
	Random Forest	0.6300	0.6392	0.6518
TLSP + Sentiment	SVM	0.6667	0.6837	0.6514
TLSP + Sentiment + Emojis		0.6633	0.6799	0.6431
TLSP + Sentiment + Emojis + Hashtags		0.6700	0.6775	0.6608
All Features	**SVM**	**0.7000**	**0.7140**	**0.6917**

4.2 Official Results

Table 4 reports the best performing systems according to the accuracy results for the English language released by the organizers.

The 0.72 accuracy obtained by our system (duan20) is among the top-4 accuracy scores obtained by the submissions for the English language, and 4% less accurate than the best system –which achieved a 0.75 accuracy, followed by the baseline SYMANTO (LDSE) [11].[8]

Our official accuracy result is higher than the scores we obtained in our cross-validation settings using the training data. This suggests that our approach would benefit from using larger datasets for training.

5 Conclusions and Future Work

We have described our participation in the Profiling Fake News Spreaders on Twitter task at PAN-CLEF 2020 [12]: given a Twitter feed, the aim is to determine whether or

[7] In order to avoid overfitting, the same training-test split is used to build both tweet-level (i.e., TLSP) and profile-level classifiers.

[8] All the results are available at https://pan.webis.de/clef20/pan20-web/author-profiling.html#results.

Table 4. Best performing systems according to the accuracy results for the English language released by the organizers for the Profiling Fake News Spreaders on Twitter task at PAN-CLEF 2020. Our system is in boldface and baselines are indicated with a – position.

Position	System	Accuracy
1	bolonyai20	0.7500
–	SYMANTO (LDSE)	0.7450
2	pizarro20	0.7350
3	babaei20	0.7250
3	datatontos20	0.7250
3	higueraporras20	0.7250
3	hoertenhuemer20	0.7250
3	ikae20	0.7250
3	tarela20	0.7250
3	vogel20	0.7250
4	andmangenix20	0.7200
4	**duan20**	**0.7200**
4	johansson20	0.7200
5	koloski20	0.7150
5	morenosandoval20	0.7150
5	pinnaparaju20	0.7150
6	carrodve20	0.7100
6	castellanospellecer20	0.7100
6	estevecasademunt20	0.7100
6	labadietamayo20	0.7100
6	shrestha20	0.7100
7	cosin20	0.7050
7	staykovski20	0.7050
8	kaushikamardas20	0.7000
8	saeed20	0.7000
–	NN + w nGrams	0.6900
–	SVM + c nGrams	0.6800
–	EIN	0.6400
–	LSTM	0.5600
–	RANDOM	0.5100

not its author is a spreader of fake news. Our approach consists of two steps: (i) we first model sentiment, political presence, and language features of fake news spreaders at tweet level; (ii) we then generate a profile-level representation to feed to a binary classifier to distinguish user profiles into spreaders or non-spreaders of fake news. Our model achieves a 70% accuracy in a 10-fold cross validation using the training set provided by the organizers. We have a number of improvements planned for our future work:

Modeling emotions in addition to sentiment. Instead of only analyzing the sentiment intensity of tweets, we plan to incorporate emotions (e.g. fear, disgust, joy, sadness, and anger) and follow up recent work that studies the relation between emotions and fake news spreading behaviors [6,5].

Modeling political presence. We believe extracting entities from tweets and characterizing those entities to create a political profile for tweets (e.g., by performing entity linking with a knowledge base) may lead to a better identification of fake news spreaders.

Understanding the impact of TLSP. Although the fine-tuned BERT model obtained promising results, we would like to explore other embedding transforms such as GPT-3 [2] to better understand the impact of the TLSP component w.r.t. to the overall performance of our approach.

Incorporate multilingual inputs. We plan to instantiate our proposed approach to other languages such as Spanish.

References

1. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. Journal of Economic Perspectives **31**(2), 211–36 (2017)
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
3. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14). pp. 1724–1734 (2014)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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 (NAACL-HLT'19). pp. 4171–4186 (2019)
5. Ghanem, B., Rosso, P., Rangel, F.: An Emotional Analysis of False Information in Social Media and News Articles. ACM Transactions on Internet Technology (TOIT) **20**(2), 1–18 (2020)
6. Giachanou, A., Rosso, P., Crestani, F.: Leveraging emotional signals for credibility detection. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19). pp. 877–880 (2019)
7. Guess, A., Nagler, J., Tucker, J.: Less than you think: Prevalence and predictors of fake news dissemination on Facebook. Science Advances **5**(1), eaau4586 (2019)
8. Hutto, C.J., Gilbert, E.: VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: Eighth international AAAI Conference on Weblogs and Social Media (ICWSM-14). pp. 216–225 (2014)
9. Liu, Y., Wu, Y.F.B.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Thirty-Second AAAI Conference on Artificial Intelligence. pp. 354–361 (2018)
10. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World. Springer (Sep 2019)
11. Rangel, F., Franco-Salvador, M., Rosso, P.: A Low Dimensionality Representation for Language Variety Identification. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 156–169. Springer (2016)
12. Rangel, F., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020)
13. Ruchansky, N., Seo, S., Liu, Y.: Csi: A hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM'17). pp. 797–806 (2017)
14. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter **19**(1), 22–36 (2017)