Fake news spreader detection using neural tweet aggregation
Notebook for PAN at CLEF 2020

Oleg Bakhteev, Aleksandr Ogaltsov, and Petr Ostroukhov

Antiplagiat, Moscow, Russia;
Moscow Institute of Physics and Technology (MIPT), Moscow, Russia
Higher School of Economics, Moscow Institute of Physics and Technology
bakhteev@ap-team.ru, ogaltsov@ap-team.ru, ostroukhov@ap-team.ru

Abstract The paper describes the neural networks-based approach for Profiling Fake News Spreaders on Twitter task at PAN 2020. The problem is reduced to the binary classification with a set of tweets of the user as an object to classify and class labels corresponding to users that are likely to spread fake news and ordinary users. To deal with a set of tweets we employ two neural network architectures: either based on recurrent or convolutional neural networks. We try aggregate the whole information obtained from the tweets to decide whether the user can spread fake news or not. We also present an ensemble of models that consists of a neural network and a classification model that works on each tweet separately.

1 Introduction

Author profiling task is focused on investigating different aspects of the author style. This year the task considers the problem of fake news spreaders detection [10]: given a set of tweets written by an author, one should decide whether the author keen to spread fake news or not. The tweets are written in two languages: English and Spanish. The training dataset contains 300 sets of tweets for each language with 150 sets for users that are keen to spread fake news and 150 sets for ordinary users. The performance metric for the task is an accuracy.

The problem of detection fake news spreading in social media becomes more and more important nowadays. There are plenty of works devoted to classification distinct social media messages based on the fact they contain fake news or not. The methods of fake news detection significantly vary from usage of classical linguistic features [7] to the usage of contemporary deep learning-based models [13,4]. In [8] the authors propose an end-to-end deep learning-based approach to detect fakes using external resources, such as news datasets, which can improve the performance of the existing methods of fake news detection.

This work was supported by RFBR project No.18-07-01441.

Despite the significant amount of works devoted to the fake news detection, many of them cannot be implemented for this task straightforwardly: the analysed object for this task is not a unique text, but a collection of short texts. Therefore even if we have any fake news in the tweet collection, we don't know exactly, which tweet in collection really contains fake. Moreover, we don't have any guarantee that the set of tweets for the target fake news spreader really contains any fake news: we only know the fact that the author spreads them. Therefore the usage of external resources becomes less important for this task. In this way the fake news spreader detection is similar to previous author profiling task, such as gender profiling [12] or bot detection [11]. The key idea for many approaches for such tasks is to aggregate information that contained in all the texts written by the author. For such an aggregation we employ a neural network-based approach. Inspired by idea described in [14] we consider two architectures based on recurrent and convolutional neural networks. Both the architectures interpret the input object as a set without any knowledge about tweet order. We also analyse the performance of the ensemble of two models that work on different hierarchy levels of the dataset, both on the corpus of tweets level and on the distinct tweet level with classifier trained on the external corpus.

2 Methodology

The following section describes the proposed method of fake news spreader detection. Formally we consider the problem as a classification problem, where the classified object is a set of tweets. Given a labeled dataset $\mathfrak{D} = (x_i, y_i)$:

$$x_i = \{x_i^1, \dots, x_i^m\}, \quad x_i^j \in \mathbb{W}^+, \quad j \in \{1, \dots, m\}, \quad y_i \in \{0, 1\},$$

where $\mathbb{W}^+$ is a set of all possible strings written in the given language, $y_i = 1$ corresponds to users that are likely to spread fake news, $y_i = 0$ corresponds to ordinary user. The task is to find the binary classifier, that minimizes an empirical risk on the dataset $\mathfrak{D}$:

$$f = \arg\min_{f \in \mathfrak{F}} \sum_{x_i, y_i \in \mathfrak{D}} [f(x_i) \neq y_i],$$

where $\mathfrak{F}$ is a set of all considered classification models.

The following sections gives a brief overview of the used models.

2.1 Recurrent network-based architecture

The proposed architecture is illustrated in Figure 1. It consists of 3 main components:

1. Recurrent network $\mathbf{f}_{\text{RNN}}$ that works with the word embeddings of the current tweet.
2. Weighting layer $\mathbf{f}_{\text{WL}}$, which determines the impact of each tweet in the final decision.
3. The feedforward network with softmax layer $\mathbf{f}_{\text{SM}}$ that makes a final decision on the class of the tweet collection author.

Once the recurrent network $\mathbf{f}_{\text{RNN}}$ processed the tweet x_i^j with hidden state $\mathbf{h}_i^j$ we employ the weighting layer in order to determine the weight of it in total sum:

$$w_i^j = \mathbf{f}_{\text{RNN}}(\mathbf{h}_i^j), w_i^j \in [0, 1].$$

After that we sum all the hidden states with their weights normalized by softmax:

$$\mathbf{h}_i = \sum_{j=1}^{m} \hat{w}_i^j \mathbf{h}_i^j, \quad \hat{w}^i = \mathbf{softmax}(\mathbf{w}^i).$$

The final stage of the classification is done by the feedforward network $\mathbf{f}_{\text{SM}}$ with a softmax layer:

$$f(x^i) = \operatorname*{arg\,max}_{j \in \{0,1\}} f_{\text{SM}}(\mathbf{h}^i)^j, \tag{1}$$

where $f_{\text{SM}}(\mathbf{h}^i)^j$ represents the component j of the softmax output $\mathbf{f}_{\text{SM}}(\mathbf{h}^i)$.

We use GRU for the $\mathbf{f}_{\text{RNN}}$ component and a one-layer neural network with sigmoid activation as a weighting layer $\mathbf{f}_{\text{WL}}$.

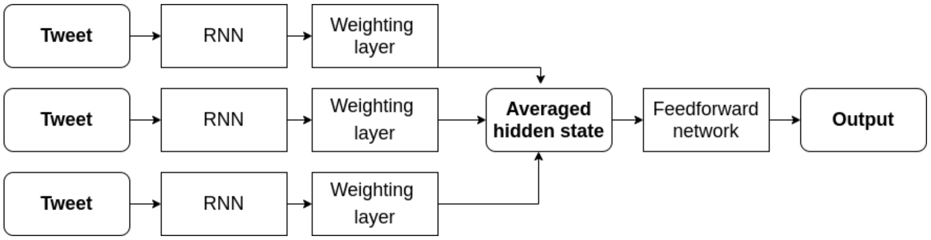

Figure 1. The scheme of the proposed Recurrent network-based architecture. All the RNNs and Weighting layers share their weights.

2.2 Convolutional network-based architecture

Another approach to aggregate collection of tweets by given author is convolutional network. CNNs are known for powerful hierarchical representation of matrix data. So, we need to convert tweet collection into matrix. We do it in a following way:

1. Take top-k most frequent words from all tweets.
2. Form a matrix using word embeddings by preserving order of words in top.
3. Apply convolutions to obtained matrix and get probability distribution via feedforward neural network.

More formally, we do "matrization" of each collection

$$x_i \longrightarrow \mathrm{M}_{\mathbf{i}}^{\mathbf{dxk}},$$

where d is an embedding size and k is number of most frequent words to take. Than we get vector of high-level features from filters:

$$\mathbf{u}_i = \mathbf{f}_{\text{CNN}}(\mathbf{M}_\mathbf{i}^{\mathbf{dxk}})$$

And finally, we obtain class label by feedforward neural network:

$$f(x^i) = \underset{j \in \{0,1\}}{\arg\max}\, f_{\text{SM}}(\mathbf{u}_i), \tag{2}$$

The intuition behind such conversion on the one hand is to make representation independent of tweets order in collection x_i and on the other hand to construct abstract representation of topics and sentiments of the collection. We apply common filters and pooling with fully-connected network with one hidden layer on the top to predict final class label. The scheme of this aggregation method is on Figure 2.

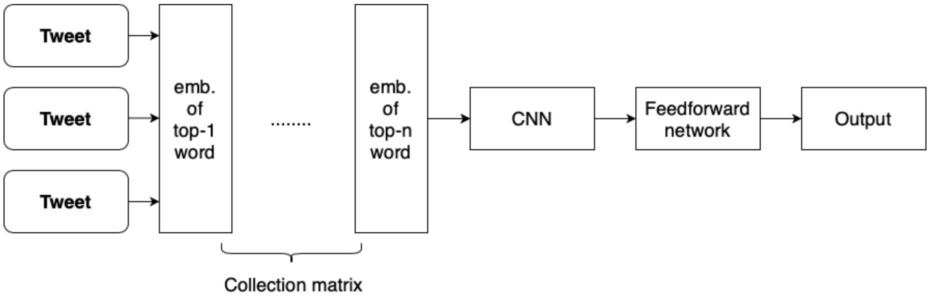

Figure 2. The scheme of the proposed CNN-based architecture.

2.3 Model ensembling

In order to analyse the tweet collection on both the hierarchical levels, either on the whole tweet set level or on the distinct tweet level, we employ an ensemble of two models: neural network for aggregation of total information about the author and a per-tweet classification model. For the per-tweet model we train a classification model f_{LR} on the dataset from [15]. This is a binary classification dataset, where objects are tweets about events from 2016, and labels are the indicators of whether particular tweet is rumour or not. The percentage of rumour tweets in this collection is about 37%.

As a classification model we use a l_2-regularized logistic regression model with TF-IDF representation as tweet features.

As a final decision rule we use the following formula:

$$f(x_i) = \begin{cases} 0, \text{ if } \min_{\{x_i^j \in x_i\}} f_{LR}(x_i^j) \leq \alpha_0, \\ 1, \text{ if } \min\{x_i^j \in x_i\} f_{LR}(x_i^j) \geq \alpha_1, \\ f_{NN}(x_i) \text{ otherwise.,} \end{cases} \tag{3}$$

where f_{LR} is a probability of the rumour prediction for the logistic regression model, $\mathbf{f}_{NN}$ is either CNN or RNN described in (1), (2), α_0, α_1 are the hyperparameters tuned on the validation set. The described scheme gives us an opportunity to use some tweet-level information straightforwardly without neural networks for the two cases:

1. if all the tweets of the user seem to be very usual and not suspicious;
2. if any of the tweets is likely to be fake or rumour with high probability and we don't need to use neural network for the tweet collection.

3 Experiment Details

3.1 Preliminary dataset analysis

For the preliminary dataset analysis we vectorized all the tweets from English part of dataset using Universal Sentence Encoder [2]. We clusterized them using DBSCAN [3] from scikit-learn package [6]. The visualization of T-SNE projection [5] of the clusterization results is represented in Figure 3. Despite the fact the cluster structure is not rather clear, we can see that the clusters of messages with high amount of users that are likely to spread fake news are concentrated on the right part of the projection. Although we do not have any ground truth for the distinct tweets, we believe that clusters with high amount of messages from the users that are likely to spread fake news contain our point of interest. A brief analysis of these messages showed that the most part of them is devoted to the three topics:

1. news about politics;
2. news about pop-starts and actors;
3. news about sport.

A significant part of such messages also mention different celebrities famous in one of these areas. We tried to use this information in the following experiments.

3.2 Experiment results

In order to validate our models we conducted a computational experiment. As a preprocessing step we lowercased tweets and removed stop-words and punctuation. We did not use any special preprocessing.

For the word embeddings we used fastText [1] trained on Common Crawl and Wikipedia with dimension set to 100. For the hyperparameters tuning we used 5-fold validation both for the neural networks (1),(2) and the tweet classification model f_{LR}. We tuned the following hyperparameters:

1. number of layers and hidden dimension for the RNN model;
2. number of top-k most frequent words, number of filters and padding size for the CNN model;
3. learning rate, l_2 and dropout rate;
4. α_0 and α_1 for the model ensemble (3).

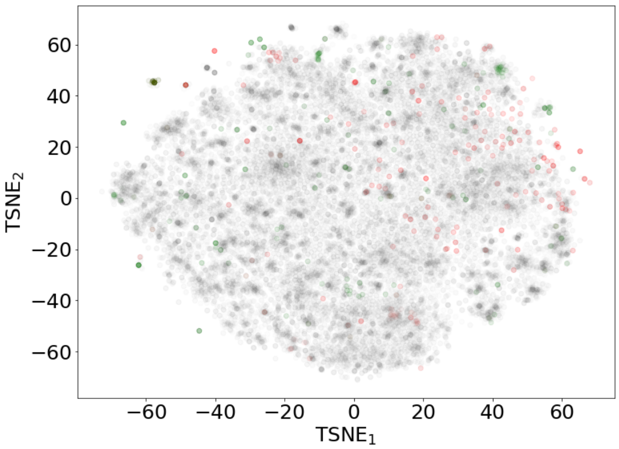

Figure 3. T-SNE projection of the vectorized tweets. The color of the point corresponds to the percentage of messages from users that are likely to spread fake news in cluster: from green (only ordinary users) to red (all the tweet authors in clusters are the users that are keen to spread fake news). Grey coloured points correspond to the clusters with only one point per cluster.

Based on the preliminary analysis from subsection 3.1 we also considered the proposed models with an addition indicator: whether the token corresponds to the celebrity or not. The celebrity list was mined from the web-resources devoted to the political, sports, music and cinema news.

The model evaluation was done using TIRA environment [9]. The results of our model performance for the cross-validation is shown in Table 1. As we can see, the RNN-based architecture gives a slightly better performance than CNN-based architecture. The usage of ensemble also allowed us to slightly increase the resulting accuracy. An additional feature for the celebrity in tokens did not give us any improvement and lowered the predictional performance.

Language	**CNN**	**RNN**	**CNN with celebrity indicator**	**RNN with celebrity indicator**	**CNN with ensemble**	**RNN with ensemble**
English	0.74 ± 0.05	0.76 ± 0.05	0.7 ± 0.05	0.7 ± 0.05	0.75 ± 0.05	$\mathbf{0.77 \pm 0.05}$
Spanish	0.77 ± 0.06	$\mathbf{0.78 \pm 0.05}$	0.74 ± 0.06	0.75 ± 0.04	—	—

Table 1. The results for the proposed models.

4 Conclusion

The paper describes the neural networks-based approach for Fake news spreaders detecion task. We proposed two neural network architectures based on RNN and CNN. The resulting performance gave us an accuracy about 77% for the English dataset and 78% for the Spanish dataset. We believe that the proposed approach can be considered as one of the baselines for the further problem development. The future work include a detailed analysis of the dataset and more advanced usage of external resources, such as list of celebrities mentioned in tweets or external datasets of fake news in tweet messages.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
2. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder for english. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 169–174 (2018)
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. vol. 96, pp. 226–231 (1996)
4. Jiang, Y., Petrak, J., Song, X., Bontcheva, K., Maynard, D.: Team bertha von suttner at semeval-2019 task 4: Hyperpartisan news detection using elmo sentence representation convolutional network. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 840–844 (2019)
5. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
7. Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R.: Automatic detection of fake news. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 3391–3401 (2018)
8. Popat, K., Mukherjee, S., Yates, A., Weikum, G.: Declare: Debunking fake news and false claims using evidence-aware deep learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 22–32 (2018)
9. 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)
10. 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 Workshop Proceedings (Sep 2020), CEUR-WS.org
11. Rangel, F., Rosso, P.: Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter. In: Proceedings of the CEUR Workshop, Lugano, Switzerland. pp. 1–36 (2019)

12. Rangel, F., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. Working notes papers of the CLEF pp. 1613–0073 (2017)
13. Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., Gao, J.: Eann: Event adversarial neural networks for multi-modal fake news detection. In: Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining. pp. 849–857 (2018)
14. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: Advances in neural information processing systems. pp. 3391–3401 (2017)
15. Zubiaga, A., Hoi, G.W.S., Liakata, M., Procter, R.: Pheme dataset of rumours and non-rumours (2016). https://doi.org/10.6084/M9.FIGSHARE.4010619, https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619