Multilingual Detection of Fake News Spreaders via Sparse Matrix Factorization

Notebook for PAN at CLEF 2020

Boško Koloski[1,2], Senja Pollak[1], and Blaž Škrlj[1]

[1]Jožef Stefan Institute, Ljubljana
[2]Faculty of Information Science - University of Ljubljana , Slovenia
blaz.skrlj@ijs.si

Abstract Fake news is an emerging problem in online news and social media. Efficient detection of fake news spreaders and spurious accounts across multiple languages is becoming an interesting research problem, and is the key focus of this paper. Our proposed solution to PAN 2020 fake news spreaders challenge models the accounts responsible for spreading the fake news by accounting for different types of textual features, decomposed via sparse matrix factorization, to obtain easy-to-learn-from, compact representations, including the information from multiple languages. The key contribution of this work is the exploration of how powerful and scalable matrix factorization-based classification can be in a multilingual setting, where the learner is presented with the data from multiple languages simultaneously. Finally, we explore the joint latent space, where patterns from individual languages are maintained. The proposed approach scored second on the 2020 PAN shared task for identification of fake news spreaders.

1 Introduction

The notion of fake news refers to distortions of news with the intention to affect the political landscape and to create confusion and divisions in society. Even if the phenomenon of fake news is not new, the scale and impact of fake news has never been so important than today, which can be attributed to the digital transformation of the news industry, and especially to the rise of social media as a news distribution channel. [6]

One of the crucial problems is the recognition of *fake news spreaders*. For example, Twitter bots (fake accounts) are capable of generating fake information and propagating it through their follower networks, which can impact real-life entities such as stock markets and possibly even elections [4]. Automatic detection of such spreaders is thus becoming one of the key approaches to minimize the manual annotation costs employed by the social media owners. This work fits under the framework of the PAN author profiling tasks [21,19], and describes our approach submitted to the PAN 2020 shared task on Profiling Fake News Spreaders on Twitter [22].

This paper is structured as follows. In Section 2 presents related work, Section 3, we discuss the problem addressed in this work. Next, in Section 4, we discuss the proposed method, followed by empirical evaluation and discussion.

2 Related work

A critical mass of fake news can have serious, real-life consequences, and can for example impact election process [3]. Distinguishing between real and fake news content has been addressed by linguistic approaches focusing on text properties, such as the writing style and content [18] and by network approaches, where using network properties and behavior are ways to complement content-based approaches that rely on deceptive language and leakage cues to predict deception. [1] A very relevant subtopic of fake news research is detection of fake news spreaders. Commonly, fake news spreaders are implemented as bots [23], and are able to carry out the spreading process in completely *automated manner*. It is still researched, whether active prevention of fake news spreading is a viable tactic, and to what extent it can be implemented in real-life online systems [15]. Further, previous PAN submissions on the topic of bot prediction indicate (e.g., [11]), that the best models perform well when different types of textual features, entailing semantic, as well as morphological information, are used.

Twitter fake news spreaders can be captured in their own social bubbles, which was shown to be an efficient defense tactic [10]. Here, simple tweet frequency distributions were already indicative of spurious behavior. Classification via features, such as the account age and similar was also shown to work well [7]. In a recent survey [24], the authors emphasize that fact-checking is an important step in maintaining online social media *quality*. By employing automated systems, capable of prioritizing potentially interesting users, less time is spent on manual curation, which can be an expensive and time-consuming process.

Traditional classifiers with extensive feature engineering seem to be pervasive in the literature about distinguishing between bots and humans but there was also some attempts to tackle the task with neural networks. In the recent work, [5] proposed a behavior enhanced deep model (BeDM) that regards user content as temporal text data instead of plain text and fuses content information and behavior information using a deep learning method. They report an F1-score of 87.32% on a Twitter-related dataset. Finally, low-dimensional representations have recently been shown to perform well for social media-based profiling [20].

3 Problem description

Provided a timeline of chosen tweets of ground truth labeled data consisting of fake news spreaders and non-spreaders, the goal is to decide if a new author is a spreader of fake news or not. Formally, we are given a decision problem which states:
Given an author A who tweets in language $L \in \{English \vee Spanish\}$ and from the collection of tweets C, given a subset of tweets C_A (of an author A),

$$C_A = t_1, t_2, \ldots, t_n \quad \text{where} \quad t_i \quad \text{represents a tweet content,}$$

find a decision function that maps $f : C_A \mapsto$ author reliability, hence

$$f(C(A)) = \begin{cases} 0 & \text{a non fake-news spreader;} \\ 1 & \text{a fake-news spreader;} \end{cases}$$

This decision problem is specialization of the problem of *author profiling*. It requires *learning* a representation from C_A, suitable for approximating f. The provided data consists of tweets by 300 English and 300 Spanish authors respectively, respectively.

For each author 100 tweets are provided making a total of 300000 English and 300000 Spanish tweets. The balance of classes is consistent for both languages, both having 150 negative and 150 positive samples, as shown in Table 1.

Table 1. Dataset distributions

Language	spreaders	non-spreaders
English	150	150
Spanish	150	150

4 Method description

The following section includes description of the proposed method with the corresponding intermediate steps.

4.1 Pre-processing

First, the tweets from each author are concatenated, and only the printable characters are kept, which means no non-printable characters are preserved. Data pre-processing for both English and Spanish includes the following steps:

1. From the original data punctuation is removed
2. URL and hashtags are removed from the result of step (1)
3. stop-words are removed from the output of step (2).

4.2 Automatic feature construction

For each author's collection of tweets we initially define a collection of candidate n features from the pre-processed data which are iteratively selected and weighted, similarly to Martinc et. al. [12]. Features generated in the construction are based on choosing following feature types:

- character based: each of the texts is tagged with character n-grams of size 2 and 3 characters and generates a predetermined maximum allowed number of features ranging from $\frac{n}{2}$ up to 15000 features.
- word based: each of the texts is tagged with word n-grams of size 1 and 2 words and generates a preconditioned maximum allowed number of features ranging from $\frac{n}{2}$ up to 15000 features.

At this we have prepared word and character features from each author's collection of tweets, ready to be used in the feature selection step.

4.3 Dimensionality reduction via matrix factorization

Next, we perform sparse singular value decomposition (SVD)[1] [8] that can be summarized via the following expression:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T.$$

The final representation (embedding) $\boldsymbol{E}$ is obtained by multiplying back only a portion of the diagonal matrix ($\boldsymbol{\Sigma}$) and $\boldsymbol{U}$, giving a low-dimensional, compact representation of the initial high dimensional matrix. Note that $\boldsymbol{E} \in \mathbb{R}^{|D| \times d}$, where d is the number of diagonal entries considered. The obtained $\boldsymbol{E}$ is suitable for a given down-stream learning task, such as classification (considered in this work). Note that performing SVD in the text mining domain is also commonly associated with the notion of *latent semantic analysis*.

4.4 Classifier selection

Classification model we aimed for in this task was to be robust yet highly flexible, one that will score well on the prepared data without using many features or extensive processing power. Following this goal we conducted a series of experiments, trying different representations with corresponding linear models as presented in Section 5. The classifiers used were the following (from scikit-learn [17]): Random Forest, Logistic Regression and the Support Vector Machines [9].

5 Conducted experiments

Considering the size of the dataset and the distribution of the data within the dataset, we preformed a series of experiments. All of them aimed to test the pipeline described in the Section 4. The experiments conducted can be divided into two main categories, based on the language considered by a given model:

1. Multilingual - Both languages' data is fused together and is subject to the same feature construction and representation creation steps.
2. Monolingual - For each language in the dataset, *English* and *Spanish*, we create a separate pipeline, that is also executed exclusively on the data from a given language.

For both approaches we performed extensive grid search over parameter space to find best hyper-parameter configuration with the help of Scikit's Learn GridSearchCV function. By doing 10-fold cross validation, the grid consisted of reducing the dimensions parametrized by k in the following interval:

$$k \in [128, 256, 512, 640, 768, 1024]$$

and the number of generated n features from the interval

$$n \in [2500, 5000, 10000, 20000, 30000].$$

[1] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html

Monolingual variant was based on splitting the data from each language separately into training 90% and 10% validation set, obtaining 270 training examples $C_{training}$ and 30 validation examples $C_{validation}$. Such splits were obtained for each language. Only training data was used for feature construction and dimensionality reduction.
Multilingual variant merged the data from both languages and after that the same approach as previously was applied. Merging the data from both languages potentially reduces the computational load required to train two separate models. Data was split into training 90% and 10% validation set, obtaining 540 training examples $C_{training}$ and 60 validation examples $C_{validation}$. In each iteration we generated n features in $\mathbb{R}^{540xn}$, reduced them to dimension k obtaining a matrix from the space $\mathbb{R}^{540xk}$.

$$g(C_{training}, n \text{ features}) : \mathbb{R}^{Nxn} \xrightarrow{\text{SVD}} \mathbb{R}^{Nxk} \quad \text{where } g \text{ denotes the 4.3 process.}$$

Once constructed, the feature space was subject to learning. We experimented with both logistic regression and linear SVMs and in initially some experiments were conducted with RandomForest model, of which hyperparameters we optimized in 5-fold cross validation considering the size of the dataset. Finally, we tested the performance on the $C_{validation}$ set.

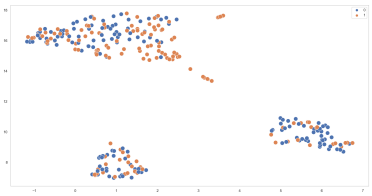


Figure 1. English Distribution

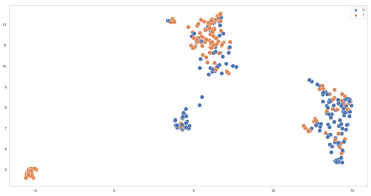


Figure 2. Spanish Distribution

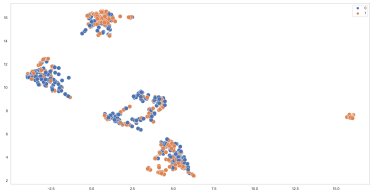


Figure 3. Merged Distribution

Figure 4. Visualization of the latent spaces used to train the final models. The orange color corresponds to spread and the blue to non-spreader. The plots indicate the number of clusters is maintained in the latent space.

We visualise the distribution of the dataset reduced to 2 dimensions using UMAP [13] dimensionality reduction in Figure 4. Figures 1 and 2 represent the visualization

with the best monolingual model described in Chapter 6, Figure 3 represents the joint latent space generated by the multilingual model described in the same chapter.

6 Results

We constructed two baselines one that was based on TF-IDF on Logistic Regression (LR) with L_1 regularization and the second was doc2vec modeled with RandomForest (RF) as classifier. The array of experiments conducted yielded the results presented in Table 2, and the outcomes of our final submission in Table 3.

As discussed in Section 5 all training was conducted by using C_{training} data and the validation was done on $C_{\text{validation}}$ set. The next presented Table 2 shows the model results as measured on TIRA training evaluation on the whole $C_{\text{validation}} \cup C_{\text{training}}$ data.

name	type	#features	#dimensions	model	EN ACC	ES ACC
tfidf_large	multi	5000	768	LR	0.9633	**0.9867**
tfidf_tweet_tokenizer	multi	5000	768	LR	0.9633	0.9533
tfidf_small	mono	5000	512	SVM,SVM	**0.9700**	0.4900
tfidf_cv	mono	10000	768	SVM,SVM	0.9100	0.9367
tfidf_no_hash	multi	10000	768	LR	0.9300	0.9067
doc2vec_baseline	mono	100	#	RF,SVM	0.6428	0.6971
tfidf_tpot_baseline	mono	30000	#	LR,SVM	0.7500	0.7400
tfidf_baseline	mono	10000	#	LR,LR	0.5567	0.7033

Table 2. Final training data on TIRA.

The final un-official evaluation as reported on TIRA's page is presented in Table 3.

name	type	#features	#dimensions	model	EN ACC	ES ACC
tfidf_large	multi	5000	768	LR	**0.7150**	**0.7950**
tfidf_cv	mono	10000	768	SVM,SVM	0.7000	**0.7950**

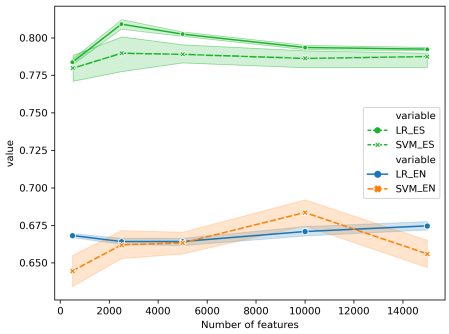
Table 3. Un-official evaluation on test data on TIRA

The Model column in Table 2 refers to the classifiers used, such that if two classifiers are present the model is monolingual - the first classifier is for English and the second one for Spanish and in case the model is multilingual only one classifier is used. The type column discriminates between the number of languages the model is trained on. Name column consists of vectorizer used and is followed by dimension size or type of tokinizer used or, dimensions column denotes the number of dimenstions SVD reduces to.

As it can be seen the highest evaluation score on our training data was obtained by the multilingual model *tfidf_large*, with the following hyper-parameters: $k = 768$ dimensions, $n = 5000$ features, Logistic Regression classifier with λ_2 = 0.002 and fit_intercept= $False$.

Monolingual model that preformed best is *tfidf_cv* which for English is paramatrized as SVM model with the following hyper-parameters: $\alpha = 0.001$, λ_1 = 0.8 while penalizing elastic-net, loss-function = hinge and power_t = 0.5 and for Spanish of SVM model with hyper-parameters: $\alpha = 0.0005$, λ_1 = 0.25 while penalizing elastic-net, loss-function = hinge and power_t = 0.9 .

The more detailed insight into the performance of the best performing models over the inference of the number of word and char n-grams and the accuracy on the 5fCV of the models is also given in Figures 5 and 6. The figures show the performance of the best mono and multilingual models – the confidence intervals indicate the variability obtained when repeating the experiments



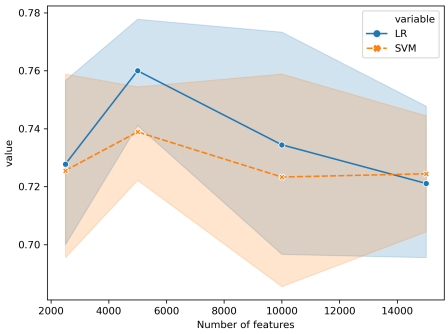


Figure 5. Best monolingual model on eval data. **Figure 6.** Best multilingual model on eval data.

7 Availability

The code and the pilot experiments are freely available at https://gitlab.com/skblaz/pan2020.

8 Discussion and Conclusions

The series of experiments conducted as a part of this work indicates, n-grams for the task of Author Profiling are still sufficient and method compared to more complex methods as transformers and word2vec [14] alike, which can easily overfit when considering only hundreds of instances. As part of the initial experiments, we also attempted to include semantic features [25], however, the results were not significantly better (nor worse), but only added to the computational time, hence such features were omitted from the final solution. We tried to change the feature space by trying different NLTK

[2] tokenizers - TweetTokenizer and the TPOT [16] automatic model generation and selection, however results obtained were similar to the ones obtained by manual construction. The joint vector space, obtained by merging the data from both languages maintains the patterns, observed when projecting individual language data sets, indicating merging of the data is a suitable tactic that does not result in complete loss of information.

Further on we can focus on exploring the possibility for detecting fake news profiles across different languages by first considering Latent Semantic Analysis across different language settings, further segmenting the semantic space prior to learning.

9 Acknowledgements

The work of the last author was funded by the Slovenian Research Agency through a young researcher grant. The work of other authors was supported by the Slovenian Research Agency (ARRS) core research programme *Knowledge Technologies* (P2-0103), an ARRS funded research project *Semantic Data Mining for Linked Open Data* (financed under the ERC Complementary Scheme, N2-0078) and European Unionś Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media).

References

1. Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology Computer Science (2016)
2. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media (2009)
3. Bovet, A., Makse, H.A.: Influence of fake news in twitter during the 2016 us presidential election. Nature communications **10**(1), 1–14 (2019)
4. Brigida, M., Pratt, W.R.: Fake news. The North American Journal of Economics and Finance **42**, 564–573 (2017)
5. Cai, C., Li, L., Zengi, D.: Behavior enhanced deep bot detection in social media. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 128–130. IEEE (2017)
6. 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)
7. Gilani, Z., Kochmar, E., Crowcroft, J.: Classification of twitter accounts into automated agents and human users. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. pp. 489–496. ACM (2017)
8. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions (2009)
9. Hearst, M.A.: Support vector machines. IEEE Intelligent Systems **13**(4), 18–28 (Jul 1998). https://doi.org/10.1109/5254.708428, https://doi.org/10.1109/5254.708428
10. Lee, K., Caverlee, J., Webb, S.: Uncovering social spammers: social honeypots+ machine learning. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 435–442. ACM (2010)

11. Martinc, M., Blaž Škrlj Pollak, S.: Fake or not: Distinguishing between bots, males and. CLEF 2019 Evaluation Labs and Workshop – Working Notes Papers (2019)
12. Martinc, M., Skrlj, B., Pollak, S.: Multilingual gender classification with multi-view deep learning: Notebook for PAN at CLEF 2018. In: Cappellato, L., Ferro, N., Nie, J., Soulier, L. (eds.) Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. CEUR Workshop Proceedings, vol. 2125. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2125/paper_156.pdf
13. McInnes, L., Healy, J., Saul, N., Großberger, L.: Umap: Uniform manifold approximation and projection. Journal of Open Source Software **3**(29), 861 (2018). https://doi.org/10.21105/joss.00861, https://doi.org/10.21105/joss.00861
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013), http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
15. Mustafaraj, E., Metaxas, P.T.: The fake news spreading plague: was it preventable? In: Proceedings of the 2017 ACM on web science conference. pp. 235–239 (2017)
16. Olson, R.S., Urbanowicz, R.J., Andrews, P.C., Lavender, N.A., Kidd, L.C., Moore, J.H.: Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I, chap. Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pp. 123–137. Springer International Publishing (2016)
17. 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)
18. 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. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://www.aclweb.org/anthology/C18-1287
19. 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)
20. 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)
21. 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)
22. 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
23. Shao, C., Ciampaglia, G.L., Varol, O., Flammini, A., Menczer, F.: The spread of fake news by social bots. arXiv preprint arXiv:1707.07592 **96**, 104 (2017)
24. Zhou, X., Zafarani, R.: Fake news: A survey of research, detection methods, and opportunities. arXiv preprint arXiv:1812.00315 (2018)

25. Škrlj, B., Martinc, M., Kralj, J., Lavrač, N., Pollak, S.: tax2vec: Constructing interpretable features from taxonomies for short text classification. Computer Speech & Language p. 101104 (2020)