KU-CST at the Profiling Fake News spreaders Shared Task

Notebook for PAN at CLEF 2020

Manex Agirrezabal

Centre for Language Technology (CST)
Department of Nordic Studies and Linguistics
University of Copenhagen / Københavns Universitet
2300 Copenhagen (Denmark)
manex.aguirrezabal@hum.ku.dk

Abstract In this document we present our approach for profiling fake news spreaders. The model relies on semantic features, part-of-speech tag related features and other simple features. We have reached an accuracy of 0.697 and 0.810 for English and Spanish, respectively, on validation data. Test accuracies using these same models reach 0.690 and 0.725 for English and Spanish data. We believe that this is a simple and robust model that could potentially be used as a baseline for this task.

1 Introduction

In this paper, we present our method for the Shared Task on profiling fake news spreaders [12]. The method that we present here is a relatively simple model that could be seen as a simple baseline that relies on semantics, word classes and some other simple features. All the code is available at this repository.[1]

We expect that the topics (or meaning) that a fake news spreader covers will differ from the ones that other users cover. Besides, we expect that the used part-of-speech (POS) tags will be good predictors, as this is a common source for author profiling. We also included the average tweet length in characters and also the uppercase/lowercase letter ratio, with the expectation to capture fake news spreaders.

This document is structured as follows. First, we mention the resources that we have employed. Then, we give more information about how the representation of each user is built. We continue mentioning the classifiers that we have tested. Finally, we discuss the results and provide some insights for possible future directions.

[1] https://github.com/manexagirrezabal/PAN-PFNS2020

2 Resources

We have trained our models on the data published by the organizers of the Shared Task on Profiling Fake News Spreaders[2] [13]. This data set contains a tweet feed of 100 tweets of 300 different users. 150 out of 300 users are fake news spreaders.

In order to build the required user representations, we employed a number of resources that are presented below.

Currently, semantic representations are built using word embeddings, for which we employed a collection trained on Twitter [4]. The authors of these embeddings include representations for several languages[3], and English and Spanish are among them.

For the part-of-speech (POS) tagger, we decided to build our own POS tagger, as commonly used POS taggers may not work so well with the language from Twitter because of shortened words, specific slang, and so on. We built a Hidden Markov Model POS tagger [2,10] trained on Twitter data [3,14].[4,5]

3 Representation of each user

Following the expectations mentioned at the beginning of the article, we assume that using the average word embedding representation from all words that a user has written, we get an approximation of the semantic content that they published. Hence, we represent a user as an average embedding (200 dimensions). We also include the standard deviation of each dimension. We do not do any further lemmatization, stemming or preprocessing to the tweets.

We include a bag-of-pos, which encodes the frequency of each part-of-speech, but we normalize it by dividing with the most frequent part-of-speech frequency, and hence, all numbers are at the range $0 \ldots 1$. While in the English version this bag contains 53 different tags, the Spanish tagger can capture 18 different tags.

We then add some commonly used simple features, such as, the average length of tweets in characters, and also the ratio of uppercase letters. We calculate this last number by just counting the uppercase letters and dividing them by the sum of the uppercase and lowercase letters.

4 Classifiers

We decided to use two linear classifiers, especially Logistic Regression and Linear SVM, as they can be trained very fast, and they give a good insight of how a set of features work. We further decided to include a non-linear model, such as the Multilayer Perceptron and the Random Forest, because of its popularity in text classification tasks. The MLP classifier was trained with three hidden layers of size 50. All other classifiers were trained using the default parameters from the Scikit-learn package [9].

[2] https://zenodo.org/record/3692319

[3] https://www.spinningbytes.com/resources/wordembeddings/

[4] https://gate.ac.uk/wiki/twitter-postagger.html

[5] https://www.clarin.si/repository/xmlui/handle/11356/1078

5 Results on development and test data

In the table below we can see the results of the different classifiers. We validated the models using Stratified K-Fold Cross-Validation with $K = 5$.

Classifier	Accuracy	
	English	Spanish
Most frequent	0.500	0.500
Logistic Regression	0.677	0.720
Linear SVM	0.503	0.550
Multilayer Perceptron	0.677	0.703
Random Forest	**0.697**	**0.810**

Considering these results, we decided to use the Random Forest model as our final model for testing. We got a test accuracy of 0.690 for English data and 0.725 for Spanish.

Further experiments

We further experimented including grammatical errors. We included information about misspellings by using the Python package `pyspellchecker`[6] to detect whether there are misspelled words, and afterwards, we control which letter becomes which letter. Therefore, if the alphabet has 27 letters, we create a vector of 27^2 numbers and we save how often each variation happens. The goal of this representation was to capture systematic errors that a user may make, hoping that this would be representative.

The results of this experiment can be seen below, using the same classifiers as before, and also validated under the same conditions (Stratified K-Fold Cross-Validation with $K = 5$).

Classifier	Accuracy	
	English	Spanish
Logistic Regression	0.577	0.720
Linear SVM	0.570	0.677
Multilayer Perceptron	0.600	0.693
Random Forest	**0.720**	**0.773**

Unfortunately, these last test was done out of competition time, and therefore, we could not test this models performance on test data.

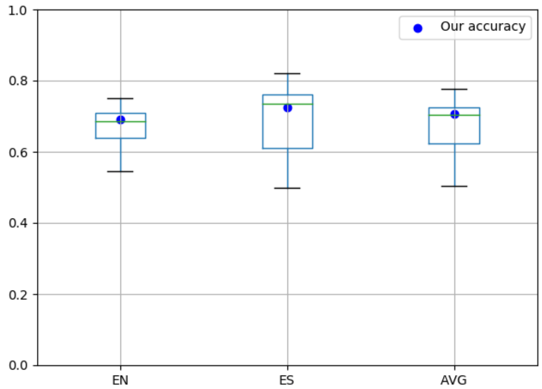
6 Discussion and Future work

In this paper, we presented a model that could potentially be used for capturing fake news spreaders. Considering average accuracy, our model ranked 31st out of 66 participants in the competition. The ranking also includes five different baselines: A model

[6] `https://pypi.org/project/pyspellchecker/`

that was used for language variety identification [11], an SVM trained on character n-grams, a Neural Network trained on word n-grams, an emotion based model (*Emotionally Infused Network*) [5], an LSTM-based implementation and a random classifier. Our model performs better than the last three baselines, but it is still worse than [11] and the character-based SVM.

In the box plot below, we illustrate how our model performs compared to the other participants. Note that outliers such as authors that have not participated in specific language configurations, have been discarded.



The presented model is relatively simple and efficient, but we believe that results can still be improved. We mention here some possible future directions.

As the character-based SVM model performs better than our model, we believe that adding character-aware representations can boost our performance. This could be done either using character n-grams or using a character-based Recurrent Neural Network to build representations.

Apart from that, we have not done any preprocessing in this work. Considering the language use on Twitter [1,6], we believe that having a normalization step could improve our results. We could also perform lemmatization or stemming. By doing this, the number of retrieved embeddings would be expected to be much higher.

In the current work, we trained a very simple Hidden Markov Model for POS tagging. This model may fall short because of the high number of misspellings in social media language. This effect could be reduced by training a POS tagger using a character-level tagger, such as a BiLSTM+CRF model [7,8].

Acknowledgements

The author would like to acknowledge all participants from the course Language Processing 2 (spring semester on 2019/2020) at the *M.S. program in IT & Cognition* at the University of Copenhagen, as the majority of the ideas presented here flourished in the discussions during class.

References

1. Alegria, I., Aranberri, N., Comas, P.R., Fresno, V., Gamallo, P., Padró, L., San Vicente, I., Turmo, J., Zubiaga, A.: Tweetnorm: a benchmark for lexical normalization of spanish tweets. Language resources and evaluation 49(4), 883–905 (2015)
2. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
3. Derczynski, L., Ritter, A., Clark, S., Bontcheva, K.: Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics (2013)
4. Deriu, J., Lucchi, A., De Luca, V., Severyn, A., Müller, S., Cieliebak, M., Hofmann, T., Jaggi, M.: Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In: Proceedings of the 26th international conference on world wide web. pp. 1045–1052 (2017)
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. Gupta, I., Joshi, N.: Tweet normalization: A knowledge based approach. In: 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS). pp. 157–162. IEEE (2017)
7. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 260–270 (2016)
8. Ling, W., Dyer, C., Black, A.W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L., Luís, T.: Finding function in form: Compositional character models for open vocabulary word representation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1520–1530 (2015)
9. 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)
10. Rabiner, L., Juang, B.: An introduction to hidden markov models. ieee assp magazine 3(1), 4–16 (1986)
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 Workshop Proceedings (Sep 2020), `CEUR-WS.org`
13. Rangel, F., Roso, P., Ghanem, B., Giachanou, A.: Profiling fake news spreaders on twitter (Feb 2020), `https://doi.org/10.5281/zenodo.3692319`
14. Rei, L., Mladenic, D., Krek, S.: A multilingual social media linguistic corpus. In: Proceedings of the 4th Conference on CMC and Social Media Corpora for the Humanities, Ljubljana, Slovenia (2016)