Profiling Hate Speech Spreaders on Twitter

Notebook for PAN at CLEF 2021

Àngel Andújar Carracedo [1] and Raquel Jiménez Mondéjar [1]

[1] Universitat Politècnica de València, Spain

Abstract

In this paper we summarize our participation in the CLEF conference 2021 regarding the Profiling Hate Speech Spreaders on Twitter task. We suggested a Support Vector Machine classifier that uses as features word n-grams. Our final software achieved an accuracy of 72% in English, 82% in Spanish and therefore, an average accuracy of 77%.

Keywords

Hate speech, n-gram, Support Vector Machine, Twitter.

1. Introduction

The evolution that social media have experienced, becoming an essential factor in the communication of today's society [1], has led to new data sources. That is why a lot of organizations use this data as a tool to analyze the feedback about some of their events, members, or products. However, organizations need to be able to discern between which opinions are written by users whose based solely on hating, and which are not to be able to do an objective analysis.

The goal of Profiling Hate Speech Spreaders on Twitter task is to identify possible hate speech spreaders as a first step towards preventing hate speech from being propagated among online users. Thus, in order to distinguish between authors, we use character and word n-grams as a feature with a Support Vector Machine (SVM) classifier and we prove different preprocessing strategies to provide a prediction for each user.

In Section 2 we expose various related works on this task. In section 3 we present our method and different models and preprocessing strategies that we have tested. In Section 4 we show our results and finally, in Section 5, we expose the conclusions we have reached.

2. Related Works

Most social networks have imposed rules on users that prohibit hate speech. However, controlling that these standards are met requires a large amount of manual work to review user reports. Due to this fact many of these platforms have increased the number of people in charge of controlling the generated content. Therefore, developing systems that are capable of detecting hateful users streamlines the review process by helping moderators to dismiss false reports. In order to develop automated hate speech detection systems, it should be noted that there are different approaches to this task.

On the one hand, there are the approaches based on combining some traditional machine learning model, such as Naive Bayes, SVM, Random Forest among others, with the extraction of features using character and word n-grams calculated from the Term Frequency - Inverse Document Frequency (TF-IDF) [2], [3], [4], [5]. On the other hand, there are the approaches based on deep learning and the use of different neural architectures to learn abstract feature representation from the input texts [6], [7], [8], [9].

3. Our Method

This section presents the dataset and the models utilized in the experimentation. For this we have used python and toolkits of emoji[2], keras [10], sklearn [11], tensorflow [12] and xgboost [13].

3.1. Corpus

The corpus of this task is composed of two sub corpuses, one of them with tweets in English and the other with tweets in Spanish. In addition, each sub corpus contains 200 XML files, which correspond to the authors, and each file contains 200 tweets from an author. It should be noted that tweets have been pre-cleaned, and hashtags, URLs and user mention in tweets have been converted to regulated tags.

3.2. Preprocessing

Firstly, we have grouped in a single chain all the tweets belong to the author, so there are 200 samples per corpus and then some preprocessing strategy applies to them.

Consequently, we have based on the preprocessing method of Pizarro [14], the winner of Author Profiling Task at PAN 2020 [15], which consist of determining if maintain letter case of the characters, replace repeated character sequences, replace digits by a tag, replace emojis by words representations and replace the regulated tags by other anonymized tags or eliminate it.

Table 1
Preprocessing options for tweets.

Name	Description
Preserve-case	Whether to maintain letter case or downcase for everything except for emoticons.
Reduce-len	Whether to replace repeated character sequences.
Replace-digits	Whether to replace numbers by xnumber.
Demojify	Whether to convert emojis into their word representations.
Replace-anon	whether to replace anonymized tags or eliminate it. #URL# by xurl #USER# by xusr #HASHTAG# by xhst

3.3. Classifiers

In our experimentation we have developed different types of machine learning models such as support vector machines (SVM), random forest (RF), XGBoost classifiers (XGB) and a neuronal model based on pre-trained BERT transformer.

[2] Emoji https://github.com/carpedm20/emoji/

Relative to SVM, RF and XGB models, it should be noted that these models are being trained using features of character and word n-grams calculated from the TF-IDF of each author. In addition, we have run a grid search to find the best preprocessing and vectorization strategy and combination of hyperparameters for the models.

Table 2
Feature hyperparameters.

Parameter	Value
N-gram type	[word, char, char_wb]
Ngram_range	[(1,1) (1,2) (1,3) (1,4) (1,5) (1,6) (2,2) (2,3) (2,4)]
Max_df	[0.7, 0.85, 0.9, 1, 2, 4, 6]
Min_df	[0.3, 0.5, 0.8, 1, 2, 4, 6]

For the finetuning of the SVM model, we have applied different types of kernel and various values of hyperparameter C.

Table 3
Hyperparameters for SVM model.

Parameter	Value
Kernel	[poly, rbf, linear]
C	[0.1, 1, 10, 100]

Regarding the random forest (RF) model, we have experimented with the quantity of trees in the forest, the criteria for measuring the quality of partitions and the minimum number of samples required to partition an internal node.

Table 4
Hyperparameters for random forest model.

Parameter	Value
Number of trees	[10, 100, 150, 200]
Partition criterion	[gini, entropy]
Minimum number of samples	[1, 2, 4, 6]

Relative to XGBoost classifier, we have tested with the number of estimators, the learning rate, and the maximum depth of a tree.

Table 5
Hyperparameters for XGBoost model.

Parameter	Value
Number of estimators	[100, 200, 300]
Learning rate	[0.01, 0.1, 1]
Maximum depth of a tree	[1, 2, 4, 5, 6]

On the other hand, regarding the pre-trained BERT [16] model, we have used its own data preprocessing and encoder to generate the embeddings of the tweets. Furthermore, we have added an additional dense layer between the encoder output and the output layer of the classifier. Therefore, we have experimented with the number of neurons of the middle layer and the dropout to apply to the output of the encoder layer.

Table 6
Hyperparameters for BERT model.

Parameter	Value
Number of neurons	[32, 64, 128]
Dropout	[0, 0.1, 0.3]

4. Results

In the training phase, we have used a 10-fold cross validation strategy to finetune the parameters of models and to select the best of them. Therefore, in the table 7 is shown estimated accuracies of each model.

Table 7
Results in the training phase with 10-fold CV.

Model	Language		Average
	es	en	
SVM	81.50%	69.00%	75.25%
RF	75.50%	65.50%	70.50%
XGBoost	78.00%	67.50%	72.75%
BERT	70.00%	57.50%	63.75%

Relative to Spanish data, the best model we have obtained is the SVM with a linear kernel, a C value of 100 and for computing features we used word n-grams with a range of (1,6). In addition, the preprocessing strategy used is to downcase the chain of tweets, to replace digits by the tag xnumber, to demojize emojis, to substitute the tag #url# to xurl and to eliminate the tags of #user# and #hashtag#.

Regarding to English data, the best model we have obtained is the SVM with a linear kernel, a C value of 100 and for computing features we used char n-grams with a range of (1,4). Furthermore, the preprocessing strategy used is to downcase the chain of tweets and remove punctuation marks.

Table 8
Results in the Profiling Hate Speech Spreaders on Twitter task.

Dataset	Model	Language		Average
		es	en	
Train	10-fold CV	81.50%	69.00%	75.25%
Test	Our model	82.00%	72.00%	77.00%

Table 8 shows the performance of classifiers on the final unseen test set. We observe that our models have obtained an accuracy of 82.00% in Spanish tweets and 72.00% in English. We observe that accuracies obtained in the training phase with 10-fold cross validation are like those of the test.

5. Conclusions

In this paper, we summarized the submitted models through the TIRA platform [17] for the Profiling Hate Speech Spreaders on Twitter task [18] at PAN 2021 [19]. These consist of SVM as classifier, and TF-IDF of word n-grams feature for Spanish tweets, and char n-grams for English authors. Regarding the presented results in the notebook, we draw the following conclusions.

Firstly, it is worth noting the great influence that cleaning and tokenizing data has on the operation of classic classification models, we observed that for each language we have to tune specifically the preprocessing strategy used.

Relative to obtained results in the training phase with 10-fold cross validation strategy, we contemplate that SVM model gives the best accuracy in both languages. In addition, we see neural model BERT provides the worst performance probably due to the small quantity of data.

Finally, comparing the results obtained in the training phase with the estimation made in the training phase, we observed that with Spanish tweets we have made a good estimate of the accuracy whereas with English we find a small difference.

6. References

[1] Kemp, S.: Digital 2020: 3.8 billion people use social media - We Are Social. We Are Social [online]. 2020. Avalaible in: https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media.

[2] MacAvaney, S. & Yao, H. & Yang, E. & Russell, K. & Goharian, N. & Frieder, O. (2019). Hate speech detection: Challenges and solutions. PloS one. 14. e0221152. 10.1371/journal.pone.0221152.

[3] Burnap, P. and Williams, M. L.: Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy and Internet, 7(2):223–242, 2015. doi:10.1002/poi3.85.

[4] Greevy, E. and Smeaton, A.: Classifying racist texts using a support vector machine. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’04, pages 468–469, New York, NY, USA, 2004. ACM. doi:10.1145/1008992.1009074.

[5] Davidson, T., Warmsley, D., Macy, M. and Weber, I.: Automated hate speech detection and the problem of offensive language. In Proceedings of the 11th Conference on Web and Social Media, Menlo Park, California, United States, 2017. Association for the Advancement of Artificial Intelligence.

[6] Gröndahl,T., Pajola, L., Juuti, M., Conti, M. and Asokan: 2018. All You Need is "Love": Evading Hate Speech Detection. In Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security (AISec '18). Association for Computing Machinery, New York, NY, USA, 2–12. DOI:https://doi.org/10.1145/3270101.3270103.

[7] Pitsilis, G.K., Ramampiaro, H. & Langseth, H.: Effective hate-speech detection in Twitter data using recurrent neural networks. Appl Intell 48, 4730–4742 (2018). https://doi.org/10.1007/s10489-018-1242-y.

[8] Raghad, A. & Hend, A.: (2020). A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. Applied Sciences. 10. 8614. 10.3390/app10238614.

[9] Ziqi, Z. & Lei, L.:(2018). Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. Semantic Web. Accepted. 10.3233/SW-180338.

[10] Chollet, François and others.: Keras. 2015. Available in: https://keras.io.

[11] Pedregosa, F., Gaël, V., Alexander,G., Vincent, M., Bertrand, T., Oliver,G., Mathieu, B., Peter, P., Ron, W., Vincent, D., Jake, V., Alexandre, P., David, C., Matthieu, B., Matthieu P., Édouard, D.: Scikit- learn: Machine learning in Python. Journal of Machine Learning Research. 2011, 12. 2825–2830.

[12] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. & others (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI (p./pp. 265—283).

[13] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785.

[14] Pizarro, J.: Using N-grams to detect 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).

[15] 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., N´ev´eol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020).

[16] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:181004805 [cs]. 2018.

[17] 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).

[18] Rangel, F., De La Peña Sarracén, G. L., BERTa Chulvi, Fersini, E. and Rosso,P.: Profiling Hate Speech Spreaders on Twitter Task at PAN 2021. In: Faggioli, G., Ferro, N., Joly, A., Maistro, M. and Piroi, F. (eds.) CLEF 2021 Labs and Workshops, Notebook Papers. CEUR-WS.org (2021).
[19] Bevendorff, J., BERTa Chulvi and De La Peña Sarracén, G. L., Kestemont, M., Manjavacas, E., Markov, I., Mayerl, M., Potthast, M., Rangel, F., Rosso,P., Stamatatos, E., Stein, B., Wiegmann, M., Wolska, M. and Zangerle, E.: Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection. In: Selcuk Candan, K., Ionescu, B., Goeuriot, L., Larsen, B., Müller, H., Joly, A., Maistro, M., Piroi, F., Faggioli, G. and Ferro, N. (eds). 12th International Conference of the CLEF Association (CLEF 2021).