Using N-grams and Statistical Features to Identify Hate Speech Spreaders on Twitter

Notebook for PAN at CLEF 2021

Eszter Katona[1], Jakab Buda[2] and Flora Bolonyai[3]

[1] *katona.eszter.rita@gmail.com*
[2] *jakab.buda@gmail.com*
[3] *f.bolonyai@gmail.com*

Abstract

In this notebook, we summarize our work process of preparing a software for the PAN 2021 Profiling Hate Speech Spreaders on Twitter task. Our final software was a stacking ensemble classifier of different machine learning models; a mixture of models using word n-grams as features and models based on statistical features extracted from the Twitter feeds. Our software uploaded to the TIRA platform achieved an accuracy of 70% in English and 79% in Spanish.

Keywords

Hate Speech Recognition, Text Classification

1. Introduction

The aim of the PAN 2021 Profiling Hate Speech Spreaders on Twitter task [18] was to investigate whether the author of a given Twitter feed is likely to spread hate speech. The training and test sets of the task consisted of English and Spanish Twitter feeds [19].

As online communication gains an increasingly important role, online hate speech also spreads. Although the question of how to tackle this problem is hard [6], and even the definition of hate speech is debated [8], its harm is evident [10]. Therefore, its detection is an essential task.

We used an ensemble of different machine learning models to provide a prediction for each user. All of our sub-models handle the Twitter feed of a user as a unit and determine a probability for each user how likely they are to be labelled as hate speech spreaders. For the final predictions, these sub-models are combined using a logistic regression.

In Section 2 we present some related works on classifying hate speech in social media and profiling hate speech spreaders. In Section 3 we describe our approach in detail together with the extracted features and models. In Section 4 we present our results. In Section 5 we discuss some potential future work and in Section 6 we conclude our notebook.

2. Related Works

There are many different approaches to identify hate speech. [21] uses Deep Convolutional Neural Network that could be called the state of the art as the DCNN model outperforms other classifiers. [1] compares 8 machine learning models. Based on their comparison SVM and Random Forest models show the highest, and KNN shows the lowest performance. Since we could not find an overall "best-practice", we decided to try some different models. We have found that SVMs [5, 9, 14, 20, 22], XGBoost [28], logistic regression [26] and random forest [3] models are also commonly used for author profiling and text classification purposes.

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-2936/paper-176.pdf

In the specific domain of classifying hate speech in social media [4] describes several models using both neural networks and statistical based predictive models such as SVMs. [2] concerned with identifying misogynistic language in social media found SVMs using token n-grams as predictive features to achieve the highest accuracy in the task out of a number of different machine learning models.

Using word n-grams as features for author profiling has been shown to be effective [5, 9, 14, 20, 22, 25], especially with TF-IDF weighting [27]. Statistical features, such as the number of punctuation marks [22, 26], medium-specific symbols (for example hashtags and at signs in tweets, links in digital texts) [11, 13, 20, 22, 24, 26], emoticons [11, 13, 20, 23, 26] or stylistic features [13] are also commonly used for text classification purposes. For text classification stacking methods are also commonly used [12].

In [8] the authors highlight "Othering Language" that turned out to be an important input for our research, as some of our models focus on the wording of the texts. Othering is an "us vs. them" that has an important role in dehumanization and prejudices against groups or people. [15] collected peer-reviewed publications for their systematic review on hate speech detection. They collected lexica of hate speech and m ention 8 published resources; however, they emphasize that most of the authors develop ad-hoc lexica.

The definition of hate speech is not an evident task. [8] provides a great overview of the different approaches. We did not have to deal with this question since the labelled dataset was provided for the task [19].

3. Our Approach

Our approach can be considered as an extension and improvement of the software we developed for the PAN 2020 Profiling Fake News Spreaders on Twitter task [17]. Compared to the software described in [7], we experimented with the addition of models that use topic probabilities based on topic models as predictive features, additional descriptive statistics among the features of the user-wise statistical models and a dictionary-based model relying on n-grams that are the most distinctive of the Tweets within each label.

3.1 The corpus and the environment setup

3.1.1 The corpus

The corpus for the PAN 2021 Profiling Hate Speech Spreaders on Twitter task [19] consists of one English and one Spanish corpus, each containing 200 XML files. Each of these files contains 200 tweets from an author. Because of the moderate size of the corpus, we wanted to avoid splitting the corpus into a training and a development set. Therefore, we used cross-validation techniques to prevent overfitting. Similarly to the PAN Author Profiling task in 2020, the dataset this year came pre-cleaned: all URLs, hashtags and user mentions in the tweets were changed to standardized tokens.

3.1.2 Environment setup

We developed our software using the Python language (version 3.7). To build our models we mainly used the following packages: scikit-learn[2], xgboost[3], spacy[4], emoji[5], lexical-diversity[6], pandas[7] and numpy[8]. Our codes are available on GitHub[9].

3.2 Our models

3.2.1 N-gram models

We experimented with a number of machine learning models based on word n-grams extracted from the text. Precisely, we investigated the performance of regularized logistic regressions (LR), random forests (RF), XGBoost classifiers (XGB) and linear support vector machines (SVM). For all four models, we ran an extensive grid search combined with five-fold cross-validation to find the optimal text pre-processing, vectorization technique and modeling parameters. We tested the same parameters for the English and Spanish data. We investigated two types of text cleaning methods for all models. The first method (M1) removed all non-alphanumeric characters (except #) from the text, while the second method (M2) removed most non alphanumeric characters (except #) but kept emoticons and emojis. Both methods transformed the text to lower case. Regarding the vectorization of the corpus, we experimented with a number of parameters. We tested different word n-gram ranges (unigrams, bigrams, unigrams and bigrams) and also looked at different scenarios regarding the minimum overall document frequency of the word n-grams (3, 4, 5, 6, 7, 8, 9, 10) included as features. Additionally, we included a wide range of hyperparameter values for each model in our grid search. For a more detailed description of the tested hyperparameters, see [7].

Table 1

The best performing text cleaning methods, vectorization parameters and model hyperparameters for the n-gram based machine learning models

Language	Model	Text cleaning	Vectorization		Model hyperparameters[10]
			N-grams	Min. global occurrence	
EN	LR	M2	bigrams	9	C=0.1
	RF	M2	unigrams	10	B=300 min_samples_leaf=7
	SVM	M2	unigrams	3	C=1
	XGB	M2	uni- and bigrams	8	eta= 0.3 max_depth=4 colsample_bytree=0.5 subsample=0.7 n_estimators=200
ES	LR	M1	unigrams	9	C=100
	RF	M1	bigrams	8	B=400 min_samples_leaf=5
	SVM	M1	unigrams	10	C=100
	XGB	M1		7	eta= 0.01

[2] https://scikit-learn.org/
[3] https://xgboost.readthedocs.io/
[4] https://spacy.io/
[5] https://pypi.org/project/emoji/
[6] https://pypi.org/project/lexical-diversity/
[7] https://pandas.pydata.org/
[8] https://numpy.org/
[9] https://github.com/pan-webis-de/katona21
[10] Parameter names in the relevant Python package/function. Detailed description in [7].

	uni- and bigrams	max_depth=5 colsample_bytree=0.5 subsample=0.7 n_estimators=200

3.2.2 User-wise statistical model

Apart from the n-gram based models, we constructed a model based on statistical variables describing all tweets of each author, thus giving one more prediction per author. The variables used in this model are as follows:

- the mean length of the 200 tweets of the authors both in words and in characters;
- the minimum length of the 200 tweets of the authors both in words and in characters;
- the maximum length of the 200 tweets of the authors both in words and in characters;
- the standard deviations of the length of the 200 tweets of the authors both in words and in characters;
- the range of the length of the 200 tweets of the authors both in words and in characters;
- the number of retweets in the dataset by each author;
- the number of URL links in the dataset by each author;
- the number of hashtags in the dataset by each author;
- the number of mentions in the dataset by each author;
- the number of emojis in the dataset by each author;
- the number of ellipses used at the end of the tweets in the 200 tweets of the authors;
- the number of words in all capitals (except for the masked URLs, RTs and user mentions) in the 200 tweets of the authors;
- a stylistic feature, the type-token ratio to measure the lexical diversity of the authors (in the dataset each author has 200 tweets thus the number of tokens per author does not differ as much that it would cause a great diversity in the TTRs).

This gives a total of 18 statistical variables. Since we used an XGBoost classifier, we did not normalize the variables and the linear correlation between the variables posed no problem.

To find the best hyperparameter set, we used a five-fold cross-validated grid search and finally refitted the best model on the whole data. The cross-validated accuracies achieved this way are 70% and 74% for the English and Spanish data respectively. Table 2 contains the best hyperparameters found.

Table 2
The best model hyperparameters for the XGBoost model using statistical features

Parameter name	Parameter values	
	EN	ES
Column sample by node	0.8	0.8
Column sample by tree	0.9	0.8
gamma	1	4
Learning rate	0.1	0.1
Max. depth	2	2
Min. child weight	2	4
Number of estimators	150	200
alpha	0.7	0.7
Subsample	1	0.6

For our early-bird submission on TIRA [16], we used a stacking ensemble as described in 3.2.4 of the n-gram based models and the XGBoost model using descriptive statistics. In order to achieve higher

accuracy with our final software, we experimented with the addition of new models. We tested models using n-grams from dictionaries containing the most typical tokens for each label and models that rely on features extracted from running topic modeling on the training data.

3.2.3 Dictionary-based n-gram models

We also built an XGBoost model for each language based on a dictionary of n-gram features retrieved from fitting separate vectorizers on each target group subcorpora. To construct this dictionary we first selected the most frequent n-grams of each subcorpora (the number of the most frequent features selected was tuned during the hyperparameter search from 200, 400, 800, and 2000 features). Then we discarded those that are also amongst the most frequent n-grams in the other group (the number of the most frequent n-grams considered here is provided as a ratio of the feature selection number and was also tuned during the hyperparameter search, the ratios examined were 1, 1.2 and 1.5). Finally, we combined the two separate dictionaries and according to a boolean hyperparameter variable, we decided whether to also include the most frequent n-grams in the whole corpus or not. For this model we investigated three more text cleaning methods in addition to the previously mentioned M1 method, all three based on M1: M3 is changing standardized tokens in such a way that the vectorizer can differentiate them from the words of the tweets; M4 is a lemmatized version of M3, and M5 is a version of M4 where all stopwords are removed[11]. To find the optimal hyperparameters we first defined a large space (373240 grid points) and randomly tested 10% of the grid points, then narrowed down the hyperparameter space and made a more exhaustive grid search.

To find the best hyperparameter set, we used a five-fold cross-validated grid search and finally refitted the best model on the whole data. The cross-validated mean accuracies achieved this way are 65.7% and 80.5% for the English and Spanish data respectively. Table 3 contains the best hyperparameters found.

Table 3
The best performing text cleaning methods, dictionary forming and vectorization parameters and model hyperparameters for the dictionary-based n-gram models

Parameter type and name		Parameter value	
		EN	ES
text cleaning		M5	M1
dictionary forming and vectorization	number of features	200	800
	n-grams	uni-, bi- and trigrams	uni-, bi- and trigrams
	min. global occurrence ratio	0.005	0.005
	max. global occurrence ratio	0.95	0.99
	whether to add most frequent features from full corpus	False	True
model hyperparameters	subsample	0.7	0.8
	column sample by tree	0.9	0.7
	column sample by node	1	1
	max depth	6	3
	number of estimators	50	50
	alpha	0.1	0.1

3.2.4 Stacking ensemble

[11] We used spacey built-in language models to lemmatize and remove stopwords.

After identifying the best hyperparameters for the six mentioned models with cross-validation, we had to find a reliable ensemble method. To avoid overfitting this ensemble model to the training set, we did not train it using the predictions of the five final trained models. Instead, we wanted to create a dataset that represents the predictions that are produced by our models on previously unseen data. To do this, we refitted the six sub-models with the cross-validated hyperparameters five times on different chunks of the original training data (each consisting of tweets from 160 users). The predictions given by these five models to the 40 remaining users were appended to the training data of the ensemble model, thus this training set consisted of predictions given to all 200 users in the training data, but these predictions were given by five different models in case of each model type. The sample created this way can be interpreted as an approximation of a sample from the distribution of the predictions of the final five models on the test set. We created a test set with the same method but with a different split of the training data.

We then used these constructed training and test sets to find the best ensemble from the following three methods: majority voting, linear regression of predicted probabilities (this includes the simple mean), and a logistic regression model. The best and most reliable results were given by the logistic model; therefore, we used this model as our final ensemble method. Table 4 summarizes the logistic regression coefficients for the probabilistic predictions of each model for both languages.

Table 4
Logistic regression coefficients for the predicted probabilities by each sub-model

Model	Coefficient values	
	EN	ES
LR	0	4.26
SVM	2.74	0
RF	0	0
XGB	0	0
Statistical XGB	0	2.59
Dictionary-based n-gram XGB	0.73	-0.52

The validity of this method is backed by the fact that our results on the training sets (an accuracy of 69% and 81% for the English and Spanish set respectively) were approximately the same as the final results. Compared to the results last year (when only the coefficients of the RF model were 0 for both languages), it seems that the different n-gram models are more similar this year. It is also worth pointing out that in the case of the English ensemble model the prediction of the model based on descriptive statistics is not taken into account either as opposed to the Spanish stacking model, where it is an important feature. This could both mean that the information that can be captured based on descriptive statistics is also mostly captured by the n-gram models or that in the case of English language different statistical variables should be considered.

3.2.5 Other experiments

As hate speech tends to evolve around well-defined themes (it is often directed at women, religious groups or ethnicities), we decided to experiment with topic models. We wanted to see how much fitting a topic model as dimension reduction could help the discrimination of the categories. We used LDA (latent dirichlet allocation) MALLET model from the Gensim package. We created 20 topics based on two different methods: we examined the coherence scores of different runs and we also used a Hierarchical Dirichlet Process (HDP) from the Gensim[12] package. Both methods showed that 20 topics could work best for our data. To see how much topic models are able to contribute to the discrimination of the groups, we first fitted the model on the whole set of training data. In this scenario, we saw that machine learning models using the topic probabilities as predictive features perform comparably to our other models. However, keeping in mind that in the test phase the model would receive unknown texts,

[12] https://radimrehurek.com/gensim/

we decided to conduct some further experiments and investigated how the machine learning models using the topic probabilities as features perform in a 5-fold cross-validation, i.e. on unseen data. In this scenario the performance of our models proved to be significantly worse compared to the other models, so we decided not to include our LDA model in our final software.

4. Results

Overall, we tested two versions of our software. For the early bird testing, we used our approach from 2020 fitted to the 2021 data. In our final submission we added one more feature to the descriptive statistical model and the dictionary-based model. As Table 5 shows, this resulted in some improvement in the case of the English corpus but the accuracy on the Spanish test set slightly decreased.

Table 5
Accuracies achieved by the two versions of our software during the cross-validation process and on the test set

Language	Early bird software		Final software	
	CV (training set)	Test set	CV (training set)	Test set
ES	80%	80%	81%	79%
EN	69%	68%	69%	70%

5. Future Work

An interesting phenomenon that we already faced during the PAN20 Author Profiling task [14] and which remained typical for our submissions in the PAN21 competition is that our models in general perform better in Spanish than in English. This is true about all of our individual models regardless of the features they used, and about the final ensemble model as well. Investigating and understanding this issue offers a challenging opportunity for further research.

6. Conclusion

In this notebook, we summarized our work process of preparing a software for the PAN 2021 Profiling Hate Speech Spreaders on Twitter task [18]. We originally started from our software developed for the PAN 2020 Profiling Fake News Spreaders task consisting of a stacking ensemble of different machine learning models based on n-grams and descriptive statistical features of the text. In the hopes of higher accuracies, we included some new descriptive statistical features and a new model relying on n-grams from a dictionary containing the most frequent n-grams in the tweets belonging to each group. In order to achieve the highest accuracies, we conducted an extensive grid-search combined with cross-validation to find the best hyperparameters for each of the models. Using the best hyperparameters, the models were refitted on the full training dataset. To get a final prediction for each user, we trained a logistic regression that used the probabilistic predictions of the sub-models as features. Using the ensemble model, we were able to achieve nearly the same accuracy on the test set as during the cross-validation process. Overall, our final software was able to identify hate speech spreaders with a 70% accuracy among users that tweet in English, and with an 79% accuracy among users that tweet in Spanish.

7. References

[1] Abro, S., Shaikh, Z. S., Khan, S., Mujtaba, G., Khand, Z. H. (2020). Automatic Hate Speech Detection using Machine Learning: A Comparative Study. In: International Journal of Advanced Computer Science and Applications 11(8) pp. 484-491. (2020)
[2] Anzovino M., Fersini E., Rosso P.: Automatic Identification and Classification of Misogynistic Language on Twitter. In: Proc. 23rd Int. Conf. on Applications of Natural Language to Information Systems, NLDB-2018, Springer-Verlag, LNCS(10859), pp. 57-64 (2018)
[3] Aravantinou, C., Simaki, V., Mporas, I., Megalooikonomou, V.: Gender Classification of Web Authors Using Feature Selection and Language Models. In: Speech and Computer Lecture Notes in Computer Science, pp. 226–33. (2015)
[4] Basile V., Bosco C., Fersini E., Nozza D., Patti V., Rangel F., Rosso P., Sanguinetti M.: SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. Proc. SemEval 2019 (2019)
[5] Boulis, C., Ostendorf, M.: A quantitative analysis of lexical differences between genders in telephone conversations. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05. Morristown, NJ, USA: Association for Computational Linguistics, pp. 435-442 (2005)
[6] Brown, A.: What is so special about online (as compared to offline) hate speech? In: Ethnicities. 18(3) pp. 297–326. (2018)
[7] Buda, J., Bolonyai, F.: An Ensemble Model Using N-grams and Statistical Features to Identify Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro, N., N´ev´eol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEURWS.org (Sep 2020)
[8] Fortuna P., Nunes S.: A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR) 51.4 (2018)
[9] Garera, N., Yarowsky, D.: Modeling Latent Biographic Attributes in Conversational Genres. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp 710-718 (2009)
[10] Gelber, K., McNamara, L.: Evidencing the harms of hate speech. In: Soc Identities. 22 pp. 324–341. (2016)
[11] Gonzalez-Gallardo, C. E., Torres-Moreno, J. M., Rendon, A. M., Sierra, G.: Efficient social network multilingual classification using character, POS n-grams and Dynamic Normalization. In: IC3K 2016 - Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. SciTePress, pp. 307-314. (2016)
[12] Hagen, M., Potthast, M., Büchner, M., and Stein, B. (2015). Webis: An ensemble for twitter sentiment detection. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 582– 589.; Stiven Zimmerman, Udo Kruschwitz, Cris Fox (2018). Improving hate speech detection with deep learning ensembles. In Proc. of the Eleventh Int. Conf. on Language Resources and Evaluation (LREC 2018)
[13] Marquardt, J., Farnadi, G., Vasudevan, G., Moens, M., Davalos, S., Teredesai, A., De Cock, M.: Age and gender identification in social media. In: CLEF 2014 working notes, pp. 1129-1136. (2014)
[14] Peersman, C., Daelemans, W., Van Vaerenbergh, L.: Predicting Age and Gender in Online Social Networks. In: Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents. New York, NY, USA: Association for Computing Machinery, pp. 37-44. (2011)
[15] Poletto, F., Basile V., Sanguinetti M., Bosco C., Patti V.: Resources and benchmark corpora for hate speech detection: a systematic review. Language Resources & Evaluation. https://doi.org/10.1007/s10579-020-09502-8 (2020)
[16] 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 - Lessons Learned from 20 Years of CLEF. Springer. (2019)
[17] 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: L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings.CEUR-WS.org (2020)

[18] Rangel F., De la Peña Sarracén G. L., Chulvi B., Fersini E., Rosso P.: Profiling Hate Speech Spreaders on Twitter Task at PAN 2021. In: Faggioli G., Ferro N., Joly A., Maistro M., Piroi F. (eds.) CLEF 2021 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings.CEUR-WS.org (2021)
[19] Rangel F., Chulvi B., De la Peña G., Fersini E., Rosso P.: Profiling Hate Speech Spreaders on Twitter [Data set]. Zenodo. http://doi.org/10.5281/zenodo.3692319 (2021)
[20] Rao, D., Yarowsky, D., Shreevats, A., Gupta, M.: Classifying latent user attributes in Twitter. In: SMUC '10: Proceedings of the 2nd international workshop on Search and mining user-generated contents. Pp. 37-44. (2010)
[21] Roy, P. K., Tripathy, A. K., Das, T. K., Gao, X. Z.: A Framework for Hate Speech Detection Using Deep Convolutional Neural Network. In: IEEE Access 8, pp. 204951-204962. (2020)
[22] Santosh, K., Bansal, R., Shekhar, M., Varma, V.: Author Profiling: Predicting Age and Gender from Blogs Notebook for PAN at CLEF 2013. in: Working Notes for CLEF 2013 Conference. (2013)
[23] Sboev, A., Litvinova, T., Voronina, I., Gudovskikh, D., Rybka, R.: Deep Learning Network Models to Categorize Texts According to Author's Gender and to Identify Text Sentiment. In: Proceedings - 2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016. Institute of Electrical and Electronics Engineers Inc., pp. 1101-1106. (2017)
[24] Schler, J., Koppel, M., Argamon, S., Pennebaker, J.: Effects of Age and Gender on Blogging. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs. American Association for Artificial Intelligence (AAAI), pp. 199- 205. (2006)
[25] Stout, L., Musters, R., Pool, C.: Author Profiling based on Text and Images Notebook for PAN at CLEF 2018. In: Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum. (2018)
[26] Volkova, S., Bachrach, Y.: On Predicting Sociodemographic Traits and Emotions from Communications in Social Networks and Their Implications to Online Self Disclosure. In: Cyberpsychology, Behavior, and Social Networking (Mary Ann Liebert Inc.) 2015/12, pp. 726-736. (2015)
[27] Yildiz, T.: A comparative study of author gender identification. In: Turkish Journal of Electrical Engineering and Computer Science 27, pp. 1052-1064. (2019)
[28] Zhang, X., Yu, Q.: Hotel reviews sentiment analysis based on word vector clustering. In: 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, pp. 260-264. (2017)