Using N-grams to detect Bots on Twitter
Notebook for PAN at CLEF 2019

Juan Pizarro[0000−0001−9598−1929]

Universitat Politècnica de València
jpizarrom@gmail.com

Abstract. This article describes the participation in the Bots and Gender Profiling shared task in PAN at CLEF 2019. We propose a Support Vector Machine Classifier with character and word n-grams features. Our model achieved the best average performance of 88.05% at the 7th International Competition on Author Profiling. In the task of determining whether the author of a set of tweets is a bot or a human, our model obtained an accuracy of 93.60% for English and 93.33% for Spanish. In the task of Gender Identification, obtained an accuracy of 83.56% for English and 81.72% for Spanish.

Keywords: Author Profiling · Gender Identification · Bot identification · Twitter · Spanish · English.

1 Introduction

Nowadays we communicate and interact through social media platforms on a daily basis, we use it as a source of information, a commercial or marketing channel, to buy products online even to speak with our bank representatives. Thus, social networks have a great influence on our lifestyle and affect the way decisions are made. It is known that social media bots (software controlled accounts) pose as humans to intervene in elections and decisions that affect many people [17]. They are also used to influence the perception of products through fake reviews.

The objective of this year's Author profiling tasks [15] in PAN [5] at CLEF 2019 is to determine if the author of a set of tweets in English or Spanish, is a human or a bot, and in case of being human, determining its gender.

The paper is organised as follows. Section 2 presents the related work, Section 3 describes the corpus, environment setup, preprocessing, features and models, and Section 4 shows the trained models with its accuracy on the dev set. Then in Section 5 we discuss about the obtained results in the test set. After that, Section 6 presents an overview of some experiments using deep learning. Finally, Section 7 draws some conclusions.

2 Related Work

In 2002, Koppel *et al.* [8] evaluate the use of function words and part of speech tagging to identify the author gender and document genre of a corpus consisting of 920 labelled documents. Pang *et al.* [12] explore the use of unigrams and bigrams, with Naive Bayes, maximum entropy classification, and support vector machines to classify the sentiment on movie-data. In 2004, Pang *et al.* [11] employ support vector machine and Naive Bayes to classify movie reviews as either positive or negative. In 2006, Schler *et al.* [16] were able to obtain an accuracy of 80.0% in gender identification in a corpus of 85.000 blogs using style and content words. Metaxas *et al.* [10] [9] found social bots supporting some candidates and attacking their opponents [19]. In 2014, Dickerson *et al.* [7] studied the problem of identifying bots on all of Twitter and identified 19 of the 25 top features they use are sentiment-related. They use grid search to find the best hyperparameters for each of the classifiers. In 2016, Bessi *et al.* [2] found social bots generating a large amount of content, possibly distorting online conversations, they noted that bots tweeting about Donald Trump generated the most positive tweets [19]. In 2018, Stella *et al.* [17] report a case of political manipulation on social media using sentiment analysis.

In the task of gender identification in PAN at CLEF 2017, Basile *et al.* [1] obtained 82.33% for English and 83.21% in Spanish using an SVM classifier trained with combinations of character and tf-idf n-grams. In the same task of gender identification in PAN at CLEF 2018, Daneshvar *et al.* [6] obtained 82.21% for English and 82.00% for Spanish using char and word n-grams as features, with a SVM classifier. Tellez *et al.* [18] obtained 81.21% for English and 80.05% for Spanish using a similar strategy.

3 Experimental Work

This section presents the methods and materials applied in the experiments. Subsection 3.1 describes the corpus, Subsection 3.2 shows the environment setup, Subsection 3.3 explain the preprocessing, Subsection 3.4 provides a description of the feature representations. Finally, the models are presented in Subsection 3.5 and in Subsection 3.6 all the hyperparameter are shown.

3.1 Corpus

The corpus consists of a set of files in the XML format, containing of 100 tweets, one file per author. It is balanced and annotated if the author is human or robot, and in case on human its gender, *male* or *female*. It is recommended to use the corpus partitions shown in Table 1 to avoid over-fitting (more than one file could be written by the same author). Also, it shows that there are more tweets for English than for Spanish.

Table 1. Tweets by corpus partitions.

Lang	Train all	Train	Dev
es	3000	2080	920
en	4120	2880	1240

3.2 Environment Setup

The models were trained on a Jupyter notebook environment known as Colaboratory[1]. We opted to use mainly the next software tools to build our models: nltk[2], sklearn[3], hyperopt[4].

3.3 Preprocessing

The XML files are parsed using the Python 3 library xml.etree.ElementTree[5] to be able to work with its content. Then, for each author, their 100 tweets are concatenated forming a long string, and a custom tag is used to separate each of the tweets. After that, we applied a lowercase conversion and the strings are tokenized using nltk TweetTokenizer [3], each URL, user mention and hashtag are replaced by one fixed tag respectively, following what was done by Daneshvar *et al.* [6][6].

3.4 Features

Based on a quick experimentation, we choose to evaluate char and word n-grams with different n-gram orders. Also we opted to represent each document using term frequency–inverse document frequency (TF-IDF). Finally, to join both TF-IDF feature representations, the char and the word n-grams, we employ FeatureUnion[7], in order to use Pipelines[8] obtaining an end to end model.

3.5 Models

Taking into account our hardware resource limitation and the reason that we want to try several hyperparameters by each model, we opted to include only a few classical machine learning algorithms.

[1] https://colab.research.google.com
[2] https://www.nltk.org/
[3] https://scikit-learn.org/
[4] http://hyperopt.github.io/hyperopt/
[5] https://docs.python.org/3/library/xml.etree.elementtree.html
[6] https://github.com/pan-webis-de/daneshvar18/blob/5542895062f2404fd5b5a07493ff098132308457/pan18ap/train_model.py
[7] https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.FeatureUnion.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

Table 2. SVM hyperparameters.

Param	Values
C	hp.loguniform('C', np.log(1e-5), np.log(1e5))
tol	hp.loguniform('tol', np.log(1e-5), np.log(1e-2))
intercept_scaling	hp.loguniform('intercept_scaling', np.log(1e-1), np.log(1e1))

Table 3. MultinomialNB hyperparameters.

Param	Values
alpha	hp.loguniform('nb_alpha', -3, 5)

As a result of our research, we ended up selecting: LinearSVC[9], LogisticRegression[10], MultinomialNB[11].

3.6 Hyperparameter Tuning

The hyperparameter tuning was done by hand at first, obtaining poorly results, because of that a parameter search tool was used.

In this work we opted to explore the use of hyperopt[12], a Python library for serial and parallel optimisation over awkward search spaces, which may include real-valued, discrete, and conditional dimensions.

After some experiments and looking of how others do the hyperparameters search with hyperopt we define each parameter range as shown in Table 2, Table 3 and Table 4, for LinearSVC, MultinomialNB and LogisticRegression respectively.

Also, a hyperparameter search was done to find the best parameters for feature representation, what are shown in Table 5.

4 Trained Models

A model was trained for each task and language separately, 4 models in total. The best 5 configurations by language and task are shown in Table 6. The models were trained with the training set and evaluated with the dev set as shown in Table 1.

The best results for the task of determining whether the author is a bot or a human in English were obtained using SVM with char n-grams with range (1, 3) and word n-grams with range (2, 3). For the task of gender identification the best results were obtained also with SVM but with char ngram with range (1, 3) and word ngram with range (1, 3).

[9] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[10] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[11] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
[12] http://hyperopt.github.io/hyperopt/

Table 4. Logistic Regression hyperparameters.

param	values
C	hp.choice('lr_C', [0.25, 0.5, 1.0])

Table 5. Feature representation hyperparamenters.

N-gram type	Param	Values
word	ngram_range	(1, 2),(1, 3),(2, 3)
word	max_df	$0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0$
word	min_df	$0.0001, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.1, 1, 2, 5$
char	ngram_range	(1, 3),(1, 5),(2, 5),(3, 5),(1, 6),(2, 6)
char	max_df	$0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0$
char	min_df	$0.0001, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.1, 1, 2, 5$

In the Spanish task the same configuration allowed for obtaining the best result on both tasks, a SVM classifier with char n-grams with range (3, 5) and word n-grams with range (1, 3).

5 Results

In order to evaluate our model on the test set, a TIRA [13] account was given, to install the software dependencies and to deploy our model in testing mode.

The task organisers offer us to evaluate our models in an early birds dataset, allowed us to verify the configuration of the environment and give us an early approximation of our model behaviour. Table 7 shows the results in the early birds dataset (dataset1) and Table 8 shows the results the final datatest (dataset2).

It is possible to see that in the gender identification task the results for the dataset2 are a couple of points better than for the dataset1, but in the task of determining the author, the results are similar for both languages and both datasets.

Finally, the results obtained with our proposed model, are better than the baselines defined by the task organisers. The baseline models as shown in Table 8 are majority, random and LDSE [14].

Table 6. Best model parameters by language and task.

Lang	Task	Classifier	Loss	Feats	word.ngram_range	char.ngram_range
en	human_or_bot	LinearSVC	-0.945968	word_char	(1, 2)	(2, 6)
en	human_or_bot	LinearSVC	-0.945968	word_char	(1, 2)	(2, 6)
en	human_or_bot	LinearSVC	-0.945968	word_char	(1, 2)	(2, 6)
en	human_or_bot	LinearSVC	-0.945968	word_char	(2, 3)	(1,3)
en	human_or_bot	LinearSVC	-0.945161	word_char	(2, 3)	(1, 5)
en	gender	LinearSVC	-0.804839	word_char	(1,3)	(1,3)
en	gender	LinearSVC	-0.803226	word_char	(1, 2)	(1, 3)
en	gender	LinearSVC	-0.801613	word_char	(1, 2)	(1, 3)
en	gender	LinearSVC	-0.801613	word_char	(1, 2)	(1, 3)
en	gender	LinearSVC	-0.801613	word_char	(1, 2)	(1, 3)
es	human_or_bot	LinearSVC	-0.922826	word_char	(1, 3)	(3,5)
es	human_or_bot		-0.918478	word_char	(1, 2)	(1, 5)
es	human_or_bot	LinearSVC	-0.918478	word_char	(1, 3)	(2, 6)
es	human_or_bot	LinearSVC	-0.918478	word_char	(1, 3)	(2, 6)
es	human_or_bot	LinearSVC	-0.918478	word_char	(1, 3)	(2, 6)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)

Table 7. Results in the early birds dataset.

pan19-author-profiling-test-dataset1-2019-03-20	Bot or Human		Gender	
	en	es	en	es
Our model	0.9394	0.9278	0.7879	0.7611

Table 8. Results in the test set and baselines.

pan19-author-profiling-test-dataset2-2019-04-29	Bot or Human		Gender	
	en	es	en	es
Our model	0.9360	0.9330	0.8356	0.8172
MAJORITY	0.5000	0.5000	0.5000	0.5000
RANDOM	0.4905	0.4861	0.3716	0.3700
LDSE	0.9054	0.8372	0.7800	0.6900

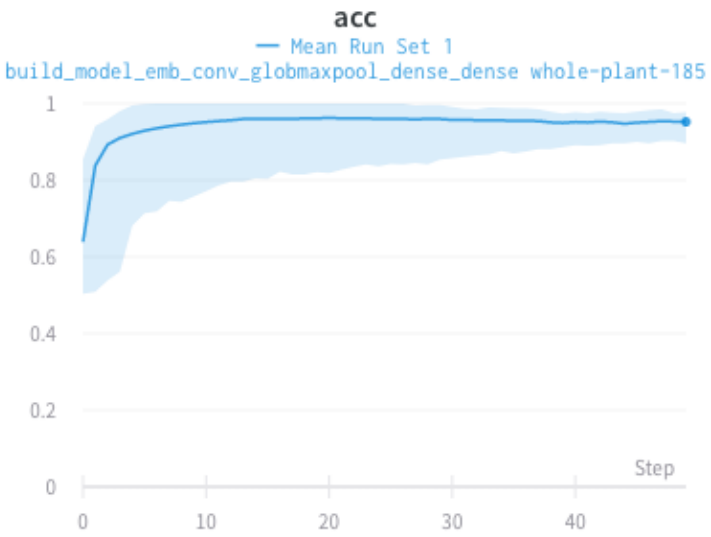

Fig. 1. Training accuracy.

6 Trained Deep Learning Models

After the submission of our run, we carried out a couple of further experiments that we could not evaluated on the test dataset and that we present in this section.

We conduct several experiments of different deep learning architectures with Keras [4] and the data partitions shown in Table 1 (train and dev). The same preprocessing process shown in Subsection 3.3 was done. In addition to that, each emoji was replaced with a word using emoji[13].

The model shown in Figure 3 obtained an accuracy of 94.524 ± 0.00167 evaluated in 10 runs. We also opted to evaluate 100 experiments with different hyperparameters for the same architecture. Figure 1 shows the accuracy on the training set and Figure 2 shows the accuracy on the dev set. It is possible to see that the results were good, independently of the hyperparameters used.

[13] https://github.com/carpedm20/emoji/

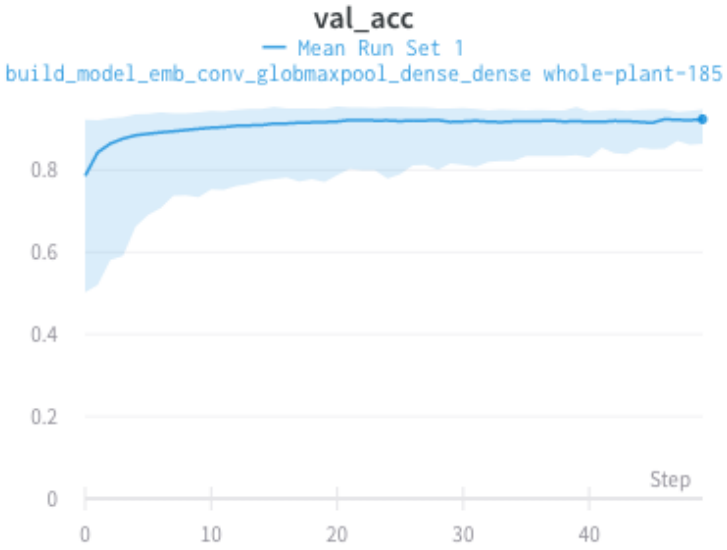

Fig. 2. Validation accuracy.

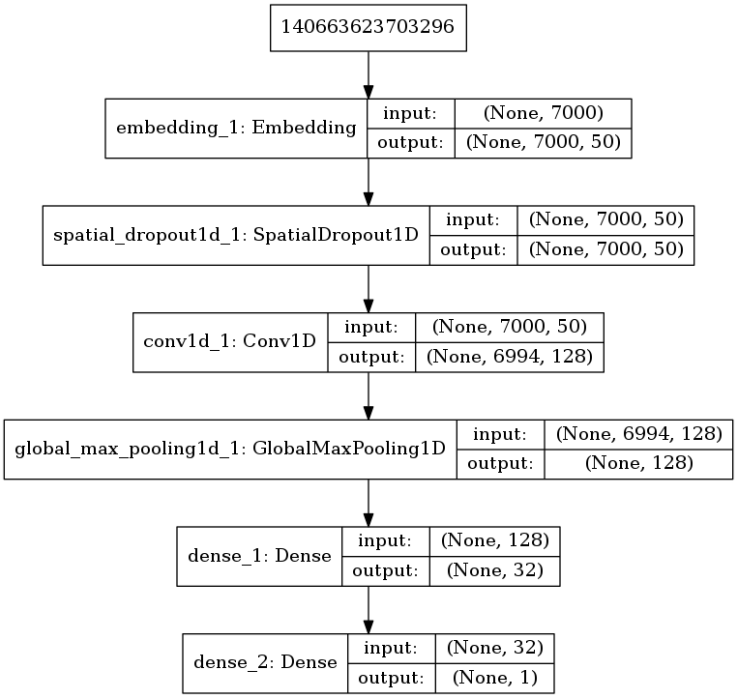

Fig. 3. Deep learning model.

7 Conclusions

Similarly as in previous years of the Author Profiling shared task in PAN, the SVM classifier with n-grams and TF-IDF features obtained very good results. The use of hyperparameter tuning tools showed to be one of the crucial parts of the model building process to obtain good results. As future work, it could very useful to explore the use of more features such as the use on lexicons, transform the emojis to custom tags, and also to try other feature representations such as word embeddings with neural networks. In Section 6 we showed the promising results of the preliminary experiments that we carried out.

References

1. Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., Nissim, M.: N-gram: New groningen author-profiling model. arXiv preprint arXiv:1707.03764 (2017)
2. Bessi, A., Ferrara, E.: Social bots distort the 2016 us presidential election online discussion (2016)
3. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
4. Chollet, F., et al.: Keras. https://keras.io (2015)
5. Daelemans, W., Kestemont, M., Manjavancas, E., Potthast, M., Rangel, F., Rosso, P., Specht, G., Stamatatos, E., Stein, B., Tschuggnall, M., Wiegmann, M., Zangerle, E.: Overview of PAN 2019: Author Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection. In: Crestani, F., Braschler, M., Savoy, J., Rauber, A., Müller, H., Losada, D., Heinatz, G., Cappellato, L., Ferro, N. (eds.) Proceedings of the Tenth International Conference of the CLEF Association (CLEF 2019). Springer (Sep 2019)
6. Daneshvar, S., Inkpen, D.: Gender Identification in Twitter using N-grams and LSA: Notebook for PAN at CLEF 2018. In: CEUR Workshop Proceedings. vol. 2125 (2018), http://ceur-ws.org/Vol-2125/paper_213.pdf
7. Dickerson, J.P., Kagan, V., Subrahmanian, V.S.: Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014). pp. 620–627 (Aug 2014). https://doi.org/10.1109/ASONAM.2014.6921650
8. Koppel, M., Argamon, S., Shimoni, A.R.: Automatically categorizing written texts by author gender. Literary and linguistic computing **17**(4), 401–412 (2002)
9. Metaxas, P.T., Mustafaraj, E.: Social media and the elections. Science **338**(6106), 472–473 (2012)
10. Mustafaraj, E., Metaxas, P.T.: From obscurity to prominence in minutes: Political speech and real-time search (2010)
11. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics. ACL '04, Association for Computational Linguistics, Stroudsburg, PA, USA (2004). https://doi.org/10.3115/1218955.1218990, https://doi.org/10.3115/1218955.1218990

12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. pp. 79–86. EMNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002). https://doi.org/10.3115/1118693.1118704, https://doi.org/10.3115/1118693.1118704
13. 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)
14. Rangel, F., Franco-Salvador, M., Rosso, P.: A low dimensionality representation for language variety identification. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing. pp. 156–169. Springer International Publishing, Cham (2018)
15. Rangel, F., Rosso, P.: Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019)
16. Schler, J., Koppel, M., Argamon, S., Pennebaker, J.W.: Effects of age and gender on blogging. In: AAAI spring symposium: Computational approaches to analyzing weblogs. vol. 6, pp. 199–205 (2006)
17. Stella, M., Ferrara, E., De Domenico, M.: Bots increase exposure to negative and inflammatory content in online social systems. Proceedings of the National Academy of Sciences **115**(49), 12435–12440 (2018). https://doi.org/10.1073/pnas.1803470115, https://www.pnas.org/content/115/49/12435
18. Tellez, E.S., Miranda-Jiménez, S., Moctezuma, D., Graff, M., Salgado, V., Ortiz-Bejar, J.: Gender identification through multi-modal tweet analysis using microtc and bag of visual words. In: Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018) (2018)
19. Yang, K.C., Varol, O., Davis, C.A., Ferrara, E., Flammini, A., Menczer, F.: Arming the public with artificial intelligence to counter social bots. Human Behavior and Emerging Technologies **1**(1), 48–61 (2019). https://doi.org/10.1002/hbe2.115, https://onlinelibrary.wiley.com/doi/abs/10.1002/hbe2.115