An Unsophisticated Neural Bots and Gender Profiling System

Notebook for PAN at CLEF 2019

Oren Halvani* and Philipp Marquardt

Fraunhofer Institute for Secure Information Technology SIT
Rheinstrasse 75, 64295 Darmstadt, Germany
{FirstName.LastName}@SIT.Fraunhofer.de

Abstract In recent years a sharp increase of bot-aided campaigns can be observed across social media networks. As a consequence, an own research discipline known as social bot detection has been established, to counteract these. In the context of the shared task "Bots and Gender Profiling" at the PAN workshop, we propose a simple neural network-based approach that determines for a given Twitter feed whether its author is a bot or a human, where in the latter case it distinguishes between male and female authors. On the official English test set, our approach achieves an accuracy of 92% and 83% for type and gender detection, respectively. For the Spanish test set, however, the results are lower (82% for type and 74% for gender detection).

1 Introduction

Bots and gender profiling can be seen as research tasks in the field of digital text forensics where, from the perspective of machine learning, both represent classification problems. In general, bots detection deals with the problem to judge if a piece of text (for instance, a Facebook post or a Twitter tweet) stems from a human or a bot, while gender profiling focuses on the question whether the text was written by a male or a female author. With the rise and growth of social networks, social bots became more and more present. As an attempt to counteract these, the organizers of the PAN workshop[1] invited researchers and practitioners to participate in the shared-task *bots and gender profiling*. In the context of this, we present a very simple approach based on a feed-forward neural network that was ranked 18th out of 55 participants.

2 Related Work

Over the years, many approaches have been proposed for both bot detection and gender profiling. In 2014, for example, Dickerson et al. [3] proposed their *SentiBot* system,

* Corresponding author.

[1] https://pan.webis.de/clef19

which uses sentiment to distinguish humans from bots on Twitter. More precisely, they considered four classes of features related to tweet syntax, tweet semantics, user behavior as well as network-centric user properties. *SentiBot* relies on an ensemble of six classifiers (Naive Bayes, SVMs, AdaBoost, Gradient Boosting, Random Forests and Extremely Randomized Trees) and achieved a score of 0.73 in terms of AUC on the *India Election Dataset*, which consists of 7.7 million tweets stemming from 550,000 Twitter accounts. One of the findings of Dickerson et al. was that sentiment related factors play a significant role in regard to the detection of bots and that considering the topics of interest to an application into account is highly important to identify bots associated with a specific application.

In 2017, Varol et al. [6] presented a similar framework for bot detection on Twitter. Based on a large number of tweets, their framework extracted 1,150 features, which they categorized into six different classes (user meta-data, friends/connected users, tweet content, sentiment, network patterns and activity time series. As an underlying model, the authors tried out a variety of classification algorithms (Random Forests, AdaBoost, Logistic Regression and Decision Tree classifiers), where the best performance was obtained using the Random Forest classifier. In contrast to the study of Dickerson et al. [3], here, Varol et al. state that both user meta-data and content features are the most promising classes to detect simple bots. To evaluate their approach, the authors used a dataset consisting of 14 millions twitter accounts of English-speaking active users. Their initial system yielded an AUC score of 0.95 on this dataset. Afterwards, the authors applied their approach on a more challenging dataset, where it also achieved a high score (0.94 AUC). In regard to their analysis, the authors made several interesting findings. They estimate, for example, that between 9% and 15% of the active Twitter accounts are bots. Also, they observed that simple bots tend to interact with bots that exhibit more human-like behaviors. Furthermore, the authors performed clustering analysis, where the resulting clusters point mainly to three subclasses of accounts (spammers, self promoters, and accounts that post content from connected applications).

3 Proposed Approach

In the following, we propose our bots and gender profiling method, which is essentially a simple feedforward-based neural network. However, before introducing the approach in more detail, we first mention the preprocessing steps that were performed on the respective documents.

3.1 Preprocessing

During the inspection of the provided corpora (more precisely, the inception of the underlying documents) we observed a large variety of noise such as citations, HTML encoded string such as `\&`, inconsistent apostrophe usage, etc. Initially, we attempted to clean the noise using a fine-grained preprocessing procedure based on truecasing [4], lexical normalization [7], accents / diacritics normalization[2], etc. However,

[2] `https://github.com/motss/normalize-diacritics`

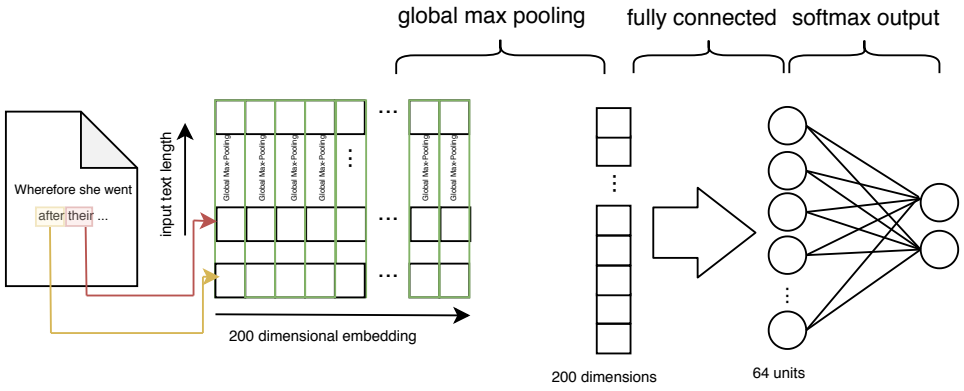

Figure 1. Architecture of our approach.

after using these in our preliminary analysis, we noticed a strong decrease in terms of accuracy. Therefore, we only performed "low-level" preprocessing steps including:

- Concatenation of all tweets in each XML-file into a one long document
- Lowercasing of the entire text
- Substitution of noisy elements with a dummy token as, for example, twitter handles (@ → §AT§), URLs (http... → §URL§), hashtags (# → §HASHTAG§), numbers ([0-9]+ → §NUMBER§), Emojis (... → §EMJOI§), punctuation marks ([.,?¿]+ → §PUNCTATION§), retweets (RT → §RT§).

3.2 Network Architecture

Our approach represents a simple feedforward neural network[3], which involves a single hidden layer. The architecture is illustrated in Figure 1). As can be seen, we first tokenize a given document and map each token into an embedding[4] vector. Next we apply global max pooling on the embedding dimensions over the sequence of tokens and concatenate the resulting pooled values to a compact representation vector, which is then fed into a simple fully connected hidden layer. The output layer performs the binary classification using the Softmax function. We used the same architecture for both classification scenarios human vs. bot and male vs. female. Furthermore, the architecture was used for both languages English and Spanish.

3.3 Hyperparameter Optimization

To optimize the hyperparameters of the network, we applied **Random Search** [1]. From the pool of all constructed configurations, we picked the one that led to the most stable

[3] We use the open-source neural-network framework Keras (`https://keras.io`)
[4] Note that we learn embeddings from scratch rather than using pretrained models.

results at the expense of accuracy. The hyperparameters of this configuration are listed in Table 1. Due to the varying lengths of the documents, we performed the following

Hyperparameter	Value
Vocabulary size	10,000
Input text length	2,500 characters
Embedding dimension	200
Dropout	0.5
Epochs	35
Batch size	64 (= number of units in the hidden layer)
Loss function	Categorical cross entropy
Optimizer	Adam (learning rate = 0.001)
Activation function	ReLu (hidden layer), Softmax (output layer)

Table 1. Hyperparameters of our approach.

strategy: Short documents with $< 2,500$ tokens were padded with zero values, while longer texts were truncated after the 2,500-th token.

In addition to dropout, we made use of **Early Stopping** [2] to counteract overfitting. Here, we observed that in many cases only few epochs (≤ 10) were needed, until the network reached a state, where the accuracy stopped to improve. Here, we also used the Keras callback function *ReduceLROnPlateau* to reduce the learning rate by 1e-1, where 1e-8 was the minimum value.

4 Evaluation

In order to reduce overfitting, we trained our approach on the provided training set (*truth-train.txt*) and evaluated the learned model on the development set (*truth-dev.txt*), as suggested[5] by the PAN organizers. On the validation set our approach achieved an accuracy of 97.69%. Afterwards, we applied the learned model on the official test set hosted on the TIRA[6] [5] platform. The results are listed in Table 2.

Language	Type (bot vs. human)	Gender (male vs. female)
English	91.59%	82.73%
Spanish	82.39%	73.78%

Table 2. Results for the official test set (*test-dataset2-2019-04-29*).

[5] `https://pan.webis.de/clef19/pan19-web/author-profiling.html`
[6] https://www.tira.io/

5 Conclusion and Future Work

We proposed a simple feedforward-based neural network that aimed to distinguish for a given Twitter feed whether its author is a bot or a human, where in the latter case the gender (male/female) is also classified. Although, the proposed method is quite simple, we observed in preliminary experiments that it was able to outperform more advanced approaches based on CNN and LSTM building blocks. In the near future, we plan to experiment with more sophisticated techniques such as Transformer-based networks that are able to capture fine-grained patterns in the embedding space.

References

1. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305 (Feb 2012), http://dl.acm.org/citation.cfm?id=2188385.2188395 3
2. Caruana, R., Lawrence, S., Giles, L.: Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In: Proceedings of the 13th International Conference on Neural Information Processing Systems. pp. 381–387. NIPS'00, MIT Press, Cambridge, MA, USA (2000), http://dl.acm.org/citation.cfm?id=3008751.3008807 4
3. Dickerson, J.P., Kagan, V., Subrahmanian, V.S.: Using Sentiment to Detect Bots on Twitter: Are Humans More Opinionated Than Bots? In: Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 620–627. ASONAM '14, IEEE Press, Piscataway, NJ, USA (2014), http://dl.acm.org/citation.cfm?id=3191835.3191957 1, 2
4. Lita, L.V., Ittycheriah, A., Roukos, S., Kambhatla, N.: tRuEcasIng. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. pp. 152–159. Association for Computational Linguistics, Sapporo, Japan (Jul 2003), https://www.aclweb.org/anthology/P03-1020 2
5. 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) 4
6. Varol, O., Ferrara, E., Davis, C.A., Menczer, F., Flammini, A.: Online Human-Bot Interactions: Detection, Estimation, and Characterization. In: Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017. pp. 280–289. AAAI Press (2017), https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15587 2
7. Xu, K., Xia, Y., Lee, C.: Tweet normalization with syllables. In: ACL (1). pp. 920–928. The Association for Computer Linguistics (2015) 2