Automatic Detection of Bot-generated Tweets

Julien Tourille
Université Paris-Saclay, CEA, List
F-91120, Palaiseau, France
julien.tourille@cea.fr

Babacar Sow*
Université Clermont Auvergne
Ecole Nationale Supérieure des Mines de Saint Etienne
LIMOS
babacar.sow@emse.fr

Adrian Popescu
Université Paris-Saclay, CEA, List
F-91120, Palaiseau, France
adrian.pospecu@cea.fr

ABSTRACT

Deep neural networks have the capacity to generate textual content which is increasingly difficult to distinguish from that produced by humans. Such content can be used in disinformation campaigns and its detrimental effects are amplified if it spreads on social networks. Here, we study the automatic detection of bot-generated Twitter messages. This task is difficult due to combination between the strong performance of recent deep language models and the limited length of tweets. In this study, we propose a challenging definition of the problem by making no assumption regarding the bot account, its network or the method used to generate the text. We devise two approaches for bot detection based on pretrained language models and create a new dataset of generated tweets to improve the performance of our classifier on recent text generation algorithms. The obtained results show that the generalization capabilities of the proposed classifier heavily depends on the dataset used to trained the model. Interestingly, the two automatic dataset augmentation proposed here show promising results. Their introduction leads to consistent performance gains compared to the use of the original dataset alone.

CCS CONCEPTS

• **Applied computing → Document management and text processing**.

KEYWORDS

Information extraction, social networks, textual deepfake detection, data augmentation

ACM Reference Format:

Julien Tourille, Babacar Sow, and Adrian Popescu. 2022. Automatic Detection of Bot-generated Tweets. In *Proceedings of the 1st International Workshop on Multimedia AI against Disinformation (MAD '22), June 27, 2022, Newark, NJ, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3512732.3533584

1 INTRODUCTION

Social media platforms have become important sources of information for a very large number of people around the world. Among these platforms, Twitter allows to spread information quickly around its user community. Twitter posts take the form of short texts, limited to 280 characters. This format is ideal for text generation algorithms because short texts written by bots are more difficult to distinguish from human-generated ones compared to longer texts [9]. Consequently, Twitter is an interesting vehicle for bot-powered disinformation campaigns.

Recent advances in automatic text generation enable the generation of short coherent text that imitates the style of the human-elicited text on which the models have been trained [2, 10, 26, 33]. Potential misuse of these models includes the fast spreading of disinformation. In the context of an increasing political polarization, this could be detrimental to democracy and carry on actions which may cause public troubles. Methods which automatically discriminate short texts which are generated by humans from those generated by bots should be investigated to prevent such actions. However, this task is very challenging when very little background information is available for an account. Early detection is therefore crucial in order to stop disinformation campaigns before they spread significantly on the network [28].

Previous research focused on the identification of bots based on the analysis and the identification of anomalous behavior [3, 28] or on the mining of profile information [7]. Closer to our work is the approach proposed by Fagni et al. [6]. They collected original tweets from human accounts (accounts the content of which is written by a person) and their fake bot counterparts maintained by people on the Twitter platform (accounts the content of which is written by text generation algorithms). They analyzed the performance of several classification models according to the technology used for tweet generation. They show that RoBERTa [17], a pretrained language model based on the transformer architecture [31] obtains the best performance across all configurations. The authors retrieved 23 bot accounts and 17 human accounts through the platform API. Although individual tweets are different in both train and test datasets, accounts are not unique in either part. This prevents from evaluating the generalization capabilities of their approach. Moreover, the authors highlight that tweets generated by GPT-2 [26] are more difficult to detect than tweets generated by older methods (e.g. AWD-LSTM). These approaches assume a significant amount of background information or of automatically- and human-generated text is available for each account. They do not enable the early detection of bot accounts, while detection is most useful before accounts start spreading misinformation.

We generalize the approach introduced in Fagni et al. [6] by proposing a method which detects bot-generated accounts after the occurrence of a single tweet by relying only on the text itself. We do not take into account the factuality of what is being exposed

*Work done during an internship at CEA

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
MAD '22, June 27, 2022, Newark, NJ, USA.
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9242-6/22/06…$15.00
https://doi.org/10.1145/3512732.3533584

in the tweet nor the network or profile information linked to the author. Put simply, our method aims to detect bots after only one tweet and has the potential to counter disinformation campaign in an effective manner. Our main contributions are the following:

- We create a new dataset for deep fake tweet detection which contains 47 political and public personalities Twitter accounts. We generate their fake counterparts using GPT-2.
- We investigate the use of the newly generated dataset to improve the performance of a RoBERTa classifier on the Fagni et al. [6] dataset.
- We investigate the generalization capabilities of a classification algorithm across Twitter accounts by creating a new dataset based on TweepFake.
- We develop a new method for bot-generated Twitter account detection which use only one single tweet.

The experimental results indicate that it is difficult to generalize beyond training accounts, especially for bots that are using powerful deep language models, such as GPT-2. The automatic data augmentation approaches introduced in this paper have a positive effect on the obtained performance. However, the obtained accuracy is far from optimal and strong progress is needed in order to automate the bot detection task in a reliable manner. This is particularly true since the task is likely to become even more difficult if new and more complex language models are used to power Twitter bots.

2 RELATED WORK

Pretrained Language Models (PLMs) have become the foundation of most approaches developed in the NLP community [8, 24]. By leveraging large corpora during a pretraining phase, these models are able to encode general linguistic knowledge that is beneficial for downstream tasks [22]. The emergence of PLMs based on the transformer architecture [31] has further improved the modeling capabilities of such models [5, 10].

Text generation (also referred as Natural Language Generation - NLG), has benefited from these advances. This broad topic includes tasks such as machine translation [4], summarization [25], open text generation [26] or data-to-text generation [32]. In almost all cases, recent approaches include a transformer-based model [10, 14, 26, 27]. Following the increase in quality and complexity of the approaches developed for text generation, traditional information extraction tasks, such as temporal information [15] or event [21] extraction, are now investigated as text-to-text generation problems [18].

Manual and semi-automatic bot detection in Twitter were recently studied by Beatson et al. [1]. The authors find that the two methods have complementary results since they do not focus on the same features. This research is interesting insofar as it shows that the manual intervention of moderators is important. A survey of automatic Twitter bot account detection is available in Martino et al. [19]. The authors analyze research papers which focus on network analysis, natural language processing or a combination of both. Intuitively, they conclude that the combination of the two approaches optimizes bot detection performance. Such a combination is only possible if sufficient network-related data is available. It is not adapted for the early detection of bots needed in order to reduce the spread of disinformation on twitter.

An influential study which addresses fake tweet detection based on content and associated metadata was proposed by Kudugunta and Ferrara [12]. The authors combine GloVe embeddings [23] and an LSTM layer to represent tweet content. This method is interesting but needs metadata to perform reliable detection. Equally important, the detection accuracy of non-deep embedding is lower than that of their deep language models counterparts [29].

The method which inspired ours was introduced Fagni et al. [6]. The authors devise a dataset which includes pairs of verified accounts and one or several associated bots. The authors proposed to use pretrained language models to detect whether an individual tweet is uttered by a human or a bot. The method provides very promising detection accuracy (≈90%). Similar results for this task are reported by Tesfagergish et al. [30]. The authors focus on text augmentation and hyperparameter optimization. The same detection setting was recently explored by Saravani et al. [29], where a BiLSTM and NeXtVLAD [16] layers are combined to capture sequential dependencies and to perform pooling. The addition of these supplementary layers results in a performance gain of approximately 2% compared to Fagni et al. [6].

While very interesting, these approaches are biased toward the dataset they have been trained and tested on insofar as the accounts are similar within each corpus part. Instead, we tackle a more generic setting in which no prior knowledge on the account is needed. This setting is more difficult but also more realistic since it performs bot detection from individual tweets.

Recently, Kumarage et al. [13] analyzed data generation for automatic fake news detection. They focused on COVID-19 related news and showed that detection of news is possible even with limited training resources. They also highlighted the role of prior lexical analysis for the generation of good quality training data. They conclude that the use of diversified generators is beneficial. The result of their study is interesting but applies for long texts and need to be verified for short ones, whose detection is more challenging.

Fake online review detection is tackled by Kowalczyk et al. [11]. One of the main challenges of the task is that the reviews have an arbitrary length. The authors focus on the explainability of the process as understanding why a given piece of text is flagged as fake is important for moderation. Explainability is also important for tweet detection, but even more challenging due to their limited length.

3 PROPOSED METHOD

Bot detection methods need to be as fast as possible in order to reduce the effects of disinformation campaigns. As we discussed in Section 2, existing methods need an important amount of information related to the network activity and/or to the content issued by bots in order to detect them. Our main objective is to study whether automatic bot detection is possible from individual tweets issued by accounts which do not appear in the training set. We test this by splitting the dataset so as to have no common users between train and test subsets. Following Fagni et al. [6], we implement the bot tweet detection method using RoBERTa [17]. If successful, such an

	Human			Bot			Global
Model	P	R	F1	P	R	F1	Acc.
Fagni et al. [6]	0.901	0.890	0.895	0.891	0.902	0.897	0.896
RoBERTa	0.922 ±0.004	0.902 ±0.008	0.912 ±0.004	0.904 ±0.007	0.924 ±0.004	0.914 ±0.003	0.913 ±0.004
RoBERTa + init.	0.922 ±0.008	0.836 ±0.012	0.877 ±0.004	0.850 ±0.008	0.929 ±0.009	0.888 ±0.002	0.882 ±0.003
RoBERTa + feat.	0.922 ±0.013	0.906 ±0.020	0.913 ±0.005	0.908 ±0.016	0.923 ±0.015	0.915 ±0.002	0.914 ±0.003

Table 1: Experiment results on the TweepFake dataset. We run 5 experiments per configuration. We report mean Precision (P), Recall (R) and F1-score (F1) for each account type (Human vs. Bot) and the mean global accuracy (Acc.). We present also the standard deviation for each score. In the two first lines, we report the performance obtained with our baseline, RoBERTa, finetuned solely on the TweepFake dataset as well as the score obtained by Fagni et al. [6] with the same configuration. In the two last lines, we report the performance obtained with our proposed approaches.

Model	Global	Human	GPT-2	RNN	Other
Fagni et al. [6]	0.89	0.87	0.74	1.00	0.95
RoBERTa	0.913 ±0.004	0.902 ±0.008	0.826 ±0.015	0.995 ±0.003	0.942 ±0.012
RoBERTa + init.	0.882 ±0.003	0.836 ±0.012	0.856 ±0.012	0.987 ±0.002	0.938 ±0.014
RoBERTa + feat.	0.914 ±0.003	0.906 ±0.020	0.820 ±0.033	0.996 ±0.003	0.942 ±0.016

Table 2: Experiment results on the TweepFake dataset. We run 5 experiments per configuration. We report mean accuracy and standard deviation. In the two first lines, we report the performance obtained with our baseline, RoBERTa, finetuned solely on the TweepFake dataset as well as the score obtained by Fagni et al. [6] with the same configuration. In the two last lines, we report the performance obtained with our proposed approaches.

approach would enable early detection of bots since they would be flagged after uttering a single message. A second important objective is to test how well detection works in absence of information regarding the deep language model used to automatically generate tweets. The difficulty of detection is correlated with the performance of the backbone model used for text generation. We address this point by analyzing the accuracy issues which occur for the detection of GPT-2 based tweets, which are the most challenging following Fagni et al. [6]. To improve their detection, we augment the TweepFake dataset with a set of tweets generated with GPT-2. These supplementary tweets are leveraged within two approaches which differ in the way fine-tuned models are aggregated to obtain the final prediction.

Generalization capabilities

As me mentioned in the introduction, one limitation of Fagni et al. [6] work is that Twitter accounts are not unique in their dataset parts. This experimental setup hinders the possibility to assess the generalization capabilities of their approach. We address this issue by devising a new dataset split across the human(s)-bot(s) pair dimension. In order to limit the effect of a biased random split, we sample 10 random datasets. We use the mapping between human and bot accounts presented in the original paper.

Improve GPT-2 tweet detection

In their study, Fagni et al. [6] show that tweets generated with GPT-2 are more difficult to detect than those generated with other methods with a performance gap of almost 25 points in accuracy. This decrease is due the quality of the tweets generated by GPT-2. The model is larger and more complex than other types of models (e.g. AWD-LSTM [20]) and allows to obtain natural text that look like original tweets. We hypothesize that the performance gap can be reduced by increasing the number of GPT-2 tweets seen during training. Based on this assumption, we generate a new set of tweets with GPT-2 and devise two methods to leverage them during model training.

Tweet generation. We generate a new dataset of fake tweets using the GPT-2 model. We collect original tweets from 47 political and public personalities[1] and split the resulting collections into two parts, one for text generation, the other for tweet classification, following the 50/50 ratio. The part dedicated to classification is further divided into train, validation and test sets following the 80/10/10 ratio.

With the first half of the collected tweets, we finetune one GPT-2 model per account and generate fake tweets that will complement the corpus part dedicated to classification. For each account and each corpus part, there are the same amount of original tweets and fake tweets, resulting in a balanced dataset.

Feature extraction. We devise a first approach based on two RoBERTa models. The first model is finetuned on the generated dataset to recognize generated from original tweets. This first RoBERTa model is then used as a feature extraction module in our architecture. For a given input, tokens are passed through both models. The two outputs of the [CLS] token are then concatenated and fed to a feedforward neural network which outputs the final classification

[1]These accounts are not present in TweepFake

score. An overview of the proposed architecture is presented in Figure 1.

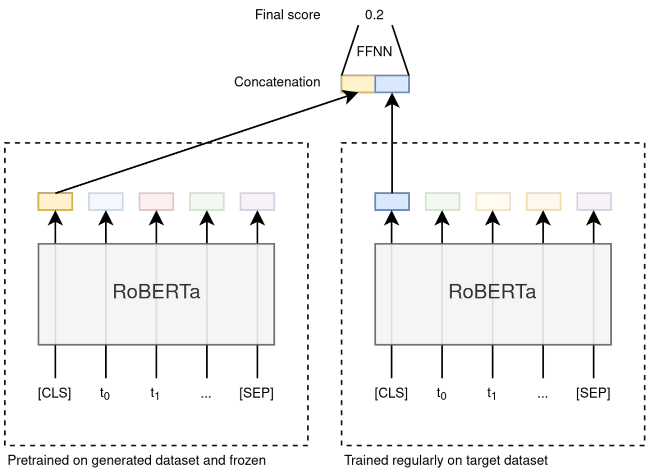


Figure 1: Architecture overview of our feature-based approach. A RoBERTa model for fake tweet classification is trained on our generated dataset. The model is then used during training on TweepFake as a feature extraction module. Weights remain frozen during training. A second RoBERTa model is used in parallel in a regular fashion. [CLS] token representations from both RoBERTa models are then concatenated and fed to a 2-layer feedforward neural network which produces the classification score.

Model initialization. In our second approach, we finetune a first RoBERTa model on our generated dataset and used its weights to initialize a second RoBERTa model which will be finetuned on TweepFake. We believe that the knowledge contained in the initial model could be useful during training and could improve the performance of our classifier.

4 EVALUATION

First, we present the datasets that will be used in our experiments. Then, we describe our experimental setup. In a third section, we present our experimental results. In the last section, we discuss the limitations of our work and possible research avenues.

4.1 Dataset

All our experiments and reported results are based on the TweepFake dataset, a dataset which contains 25,572 tweets (half humans, halt bots). Tweets are generated with several algorithms. Further information about the human accounts and their bot counterparts is available in the original paper [6].

The newly collected dataset contains 47 accounts and 725,227 original tweets. As me mentioned earlier, we keep half of them to finetune one GPT-2 model per account and the other half is split into train (290,081), dev (36,260) and test sets (36,282). We generate the same amount of fake tweets for each part of the corpus resulting in a balanced dataset. Information concerning the collected tweets is presented in Table 5 in Appendix A. We report the Twitter handle, the number of collected tweets and their partition across dataset parts.

4.2 Experimental setup

Our classifiers are based on the RoBERTa model (small version) to which we add a classification head. The head is composed of a 2-layer feedforward neural network whose hidden layer has the same size as the input layer. We use tanh as activation function an a dropout rate of 0.1 on both input and hidden layers. The classification model is trained to minimize a Binary Cross Entropy loss. We train for 5 epochs with a learning rate of $1e^{-5}$ for the transformer model and $1e^{-3}$ for the classification head and a linear warmup of a 1/2 epoch (10%). We use AdamW as optimizer and a mini-batch size of 16. We implement our models using the library *transformers*[2]. We keep the best performing model on the development set (tested at the end of each epoch) and apply it on the test set at the end of training. To assess the robustness of our model to the random seed, we run 5 experiments per configuration and report mean scores along with their standard deviations.

Concerning generation, we finetune the large version of GPT-2 (774M parameters) for one epoch with a learning rate of $1e^{-5}$ and a mini-batch size of 16. We use a temperature of 0.7 during tweets generation. Our implementation is based on the library *aitextgen*[3].

4.3 Results

We implement the same baseline as the one presented in Fagni et al. [6]. We train a RoBERTa classifier in the conditions mentioned above and report its performance on the test part of TweepFake. We report the score for comparison with other methods introduced in the paper in Table 3, 1 and 2. We obtain a global accuracy of 0.913 with f1-scores of 0.912 and 0.914 for human and bots respectively. We outperform the scores obtains by Fagni et al. [6] with the same model. The performance bump is in part due to a better detection of tweets generated with GPT-2 with a score increasing from 0.74 to 0.826. However, this score remains lower than the ones obtained for RNN and other methods with average scores of 0.995 and 0.942 respectively.

Generalization capabilities. Performance obtained for the different corpus splits is presented in Table 3 along with our baseline. Depending on the random split, the global accuracy is varying from 0.615 to 0.939 with several standard deviation going beyond four points, suggesting that in some cases, the model encounters difficulties to generalize across accounts. These results show that generalization is a key issue for detecting generated tweets at an early stage. Models and algorithms that are proposed in the literature must ensure that they are able to generalize beyond the accounts used in their datasets.

Improving GPT-2 tweet detection. To assess the quality of our generated dataset, we sample 10 random datasets, split across the human(s)-bot(s) pairs, and learn one[4] classification model (RoBERTa) per split. Results are presented in Table 4. The performance obtained on these datasets is higher than the one obtained on GPT-2

[2]https://github.com/huggingface/transformers
[3]https://github.com/minimaxir/aitextgen
[4]We do not run 5 experiments per split due to low computing budget

Dataset	Human			Bot			Global
#	P	R	F1	P	R	F1	Acc.
TweepFake	0.922 ±0.004	0.902 ±0.008	0.912 ±0.004	0.904 ±0.007	0.924 ±0.004	0.914 ±0.003	0.913 ±0.004
1	0.705 ±0.018	0.854 ±0.035	0.771 ±0.010	0.816 ±0.027	0.641 ±0.043	0.716 ±0.021	0.747 ±0.011
2	0.667 ±0.020	0.788 ±0.054	0.721 ±0.016	0.743 ±0.028	0.603 ±0.059	0.663 ±0.026	0.695 ±0.010
3	0.945 ±0.018	0.809 ±0.028	0.871 ±0.010	0.834 ±0.017	0.952 ±0.018	0.889 ±0.004	0.880 ±0.007
4	0.932 ±0.009	0.822 ±0.010	0.873 ±0.003	0.841 ±0.006	0.940 ±0.009	0.888 ±0.003	0.881 ±0.003
5	0.582 ±0.033	0.837 ±0.036	0.685 ±0.017	0.705 ±0.034	0.392 ±0.096	0.498 ±0.082	0.615 ±0.036
6	0.950 ±0.018	0.928 ±0.024	0.939 ±0.009	0.930 ±0.020	0.951 ±0.019	0.940 ±0.008	0.939 ±0.008
7	0.640 ±0.021	0.764 ±0.041	0.696 ±0.008	0.708 ±0.016	0.568 ±0.059	0.628 ±0.030	0.666 ±0.011
8	0.598 ±0.018	0.897 ±0.044	0.716 ±0.011	0.799 ±0.041	0.393 ±0.068	0.522 ±0.053	0.645 ±0.019
9	0.961 ±0.013	0.850 ±0.019	0.902 ±0.010	0.866 ±0.014	0.965 ±0.013	0.913 ±0.008	0.908 ±0.009
10	0.947 ±0.018	0.848 ±0.024	0.894 ±0.007	0.863 ±0.017	0.951 ±0.018	0.905 ±0.004	0.900 ±0.000

Table 3: Results for cross-account experiments with RoBERTa on TweepFake. We split randomly the TweepFake dataset into train, dev and test sets along the human(s)-bot(s) pair axis following the 80/10/10 ratio. We repeat this step 10 times and we run 5 experiments with each dataset. We report mean Precision (P), Recall (R) and F1-score (F1) for each account type (Human vs. Bot) and the mean global accuracy (Acc.). We present also the standard deviation for each score. For comparison, we report a simple RoBERTa baseline on TweepFake original split.

tweets from TweepFake, suggesting that our generated tweets are easier to discriminate. We select the best performing model on bot accounts (number 3 in our case) and keep model weights for the next experiments.

Split	Human			Bot			Global
#	P	R	F1	P	R	F1	Acc.
1	0.927	0.946	0.936	0.945	0.925	0.935	0.936
2	0.935	0.946	0.941	0.946	0.935	0.940	0.940
3	0.941	0.951	0.946	0.951	0.941	0.946	0.946
4	0.953	0.899	0.925	0.904	0.955	0.929	0.927
5	0.916	0.904	0.910	0.905	0.917	0.911	0.911
6	0.945	0.912	0.929	0.915	0.947	0.931	0.930
7	0.922	0.932	0.927	0.931	0.921	0.926	0.926
8	0.970	0.914	0.941	0.919	0.972	0.944	0.943
9	0.927	0.945	0.936	0.944	0.926	0.935	0.935
10	0.928	0.907	0.917	0.909	0.930	0.919	0.918

Table 4: Results for experiments on the generated dataset. We split randomly the TweepFake dataset into train, dev and test sets along the human(s)-bot(s) pair axis following the 80/10/10 ratio. We repeat this step 10 times and we run 1 experiment with each dataset. We report mean Precision (P), Recall (R) and F1-score (F1) for each account type (Human vs. Bot) and the mean global accuracy (Acc.).

Performance for our two proposed approaches is presented in Tables 1 and 2. Our initialized model performs worse than the baseline in terms of accuracy. The model seems to have difficulty to leverage the knowledge contained in the pretrained weights. Our feature-extraction based model is on par with the baseline (0.914 for global accuracy). However when we look at performance according to the type of algorithm used to generate fake tweets, we see that the initialized model improve the recognition of GPT-2 tweets by 3 points, which is not the case for the other approach. This suggests that our initialized model do in fact capitalize on the knowledge contained in the pre-trained weights for this type of tweets but failed to generalize to other methods. Scores for *RNN* and *Other* decrease by 0.8 and 0.4 points respectively.

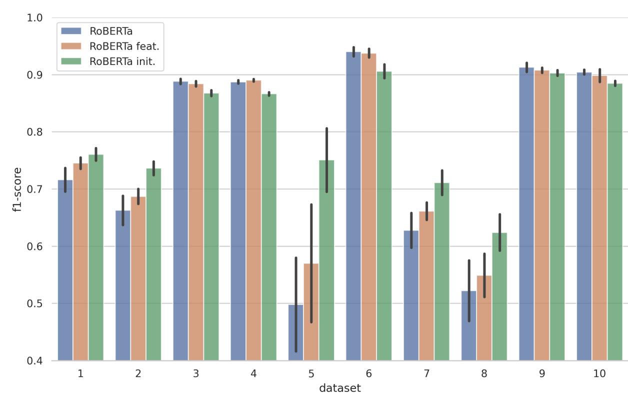


Figure 2: Results for cross-account experiments with RoBERTa and our two proposed approaches on the TweepFake splits. We run 5 experiments with each dataset. We report mean F1-score (F1) for bots with its standard deviation.

We analyze the performance per account and per type of accounts. We report accuracies in Figure 3. As aggregated scores suggested, we find that bot accounts which use GPT-2 are more difficult to detect than others. For these accounts, our proposed approaches seem to improve the performance of the classifier. Interestingly, *deep_potus*, a bot account generated with GPT-2, is very easy to discriminate. This suggests that either this account is not filtered by its maintainer or that the model is not correctly optimized. For the other two categories of generation algorithms, we observe that the performance is very high for a large majority of accounts, with several of them reaching an accuracy of 1.0. Across all types of accounts, our approaches based on a initialization of RoBERTa do sometimes decrease drastically the performance (e.g. *Musk_from_Mars* and *UtilityLimb*).

We apply our two proposed approaches (RoBERTa + init and RoBERTa + feat) to the new TweepFake splits created in this paper. Results are presented in Figure 2. Our approaches fail to improve the performance for splits whose baseline score is close or beyond 0.9 of f1-score. This result is in line with our findings on the original TweepFake split. However, both approaches are able to increase the performance for low performing splits (i.e. splits 1, 2, 5, 7 and 8), suggesting that the classifier is able to exploit the knowledge contained in the pretrained weights. For these configurations, RoBERTa init. gains are higher than those obtains by RoBERTa feat.

4.4 Analysis and Perspectives

Our study highlights the need for building models that generalize better beyond the accounts and the technologies used in the training datasets. Our experiments on random splits of TweepFake show that performance can vary from 0.615 to 0.939 according to the split. By generating fake tweets with GPT-2 and incorporating them within two simple approaches, we are able to boost the classification performance for splits with low scores.

Although our two proposed approaches seem to improve the performance for several experimental configurations, it is difficult to draw any conclusion for cases where the classifier is already performing well (beyond 0.9 accuracy). The almost neutral contribution of our proposed approaches could be explained by the low quality of the generated tweets used for training the first RoBERTa model.

Our experiments open new research avenues in the domain. First, the quality of generated tweets needs to be improved. Finding a way of selecting hard examples could allow us to generate tweets that will help the classifier to take decisions. Second, results obtained for GPT-2 tweets suggest that the use of text generation algorithm will become more and more difficult to detect. With an increasing number of parameters, new models such as T5 [27] or GPT-3 [2] will likely be harder to detect. Finally, further research in the domain needs to take into consideration the diversity of algorithms used for generating tweets. Our experiments show that focusing on a specific method (GPT-2 in our case) does improve the performance for that particular type of tweets but decrease it for others.

5 CONCLUSION

We presented a study on the detection of fake tweets without any knowledge on the account it has been posted. We highlight the difficulty of detecting text that has been generated with recent algorithms such as GPT-2. Our approaches based on the creation of unfiltered GPT-2 tweets show promising results and needs further investigation. Text is not the only feature that can be used to detect bot-generated accounts. Tweet and account metadata as well as embedded media can be used together with text to better detect twitter bots. On the semantic level, fact checking systems could also be used.

6 ACKNOWLEDGMENTS

This work was supported by the European Commission under European Horizon 2020 Programme, grant number 951911 - AI4Media. This work was supported by the Fondation MAIF. It was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council.

REFERENCES

[1] Oliver Beatson, Rachel Gibson, Marta Cantijoch Cunill, and Mark Elliot. 2021. Automation on Twitter: Measuring the Effectiveness of Approaches to Bot Detection. *Social Science Computer Review* (2021).

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1877–1901.

[3] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2016. DeBot: Twitter Bot Detection via Warped Correlation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 817–822.

[4] Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, 4171–4186.

[6] Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *PLOS ONE* 16, 5 (2021).

[7] Maryam Heidari, James H Jones, and Ozlem Uzuner. 2020. Deep Contextualized Word Embedding for Text-based Online User Profiling to Detect Social Bots on Twitter. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 480–487.

[8] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 328–339.

[9] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[10] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv:1909.05858 [cs]* (2019). arXiv:1909.05858

[11] Peter Kowalczyk, Marco Röder, Alexander Dürr, and Frédéric Thiesse. 2022. Detecting and Understanding Textual Deepfakes in Online Reviews. In *Hawaii International Conference on System Sciences*.

[12] Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *Information Sciences* 467 (2018), 312–322.

[13] Tharindu Kumarage, Amrita Bhattacharjee, Kai Shu, and Huan Liu. 2021. Data Generation for Neural Disinformation Detection. (2021).

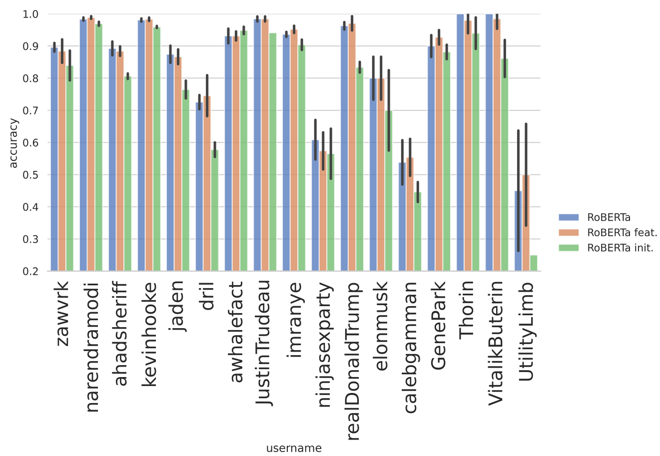
[14] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 7871–7880.

[15] Chen Lin, Timothy Miller, Dmitriy Dligach, Farig Sadeque, Steven Bethard, and Guergana Savova. 2020. A BERT-based One-Pass Multi-Task Model for Clinical Temporal Relation Extraction. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*. Association for Computational Linguistics, 70–75.

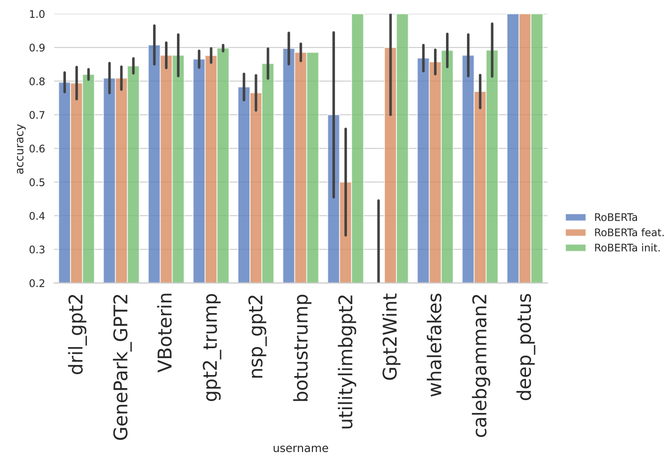
[16] Rongcheng Lin, Jing Xiao, and Jianping Fan. 2018. Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

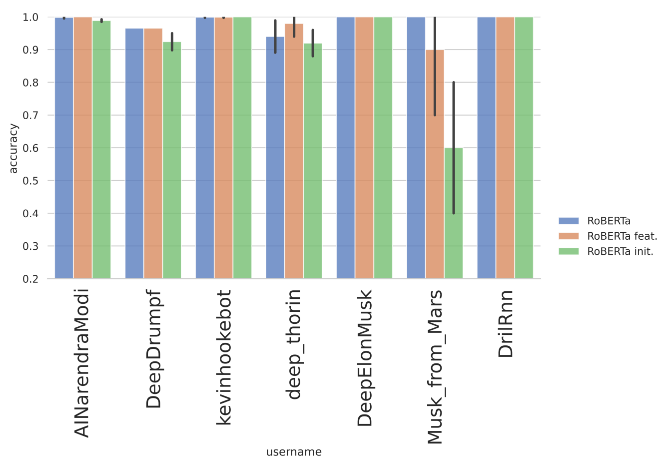
[18] Aman Madaan and Yiming Yang. 2021. Neural Language Modeling for Contextualized Temporal Graph Generation. In *Proceedings of the 2021 Conference of the*



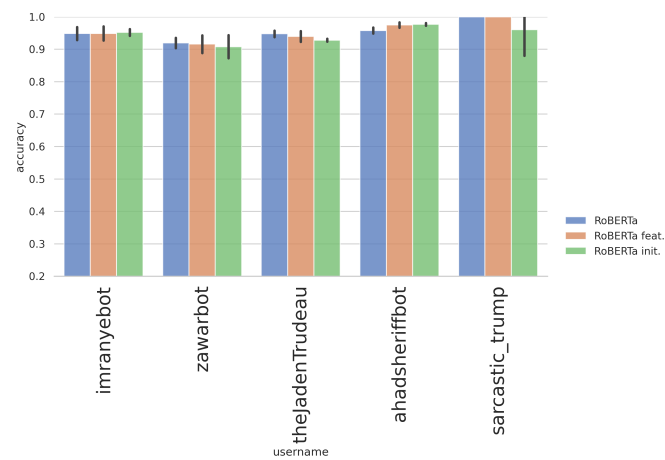
(a) *human* category



(b) *GPT-2* category



(c) *RNN* category



(d) *others* category

Figure 3: Accuracy on TweepFake test set reported by Twitter username. We regroup Twitter handle belonging to the same category: *human, GPT-2, RNN* or *others.*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 864–881.

[19] Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. A Survey on Computational Propaganda Detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 4826–4832.

[20] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and Optimizing LSTM Language Models. In *International Conference on Learning Representations*.

[21] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint Event Extraction via Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 300–309.

[22] Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*. Association for Computational Linguistics, 58–65.

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[24] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2227–2237.

[25] Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On Extractive and Abstractive Neural Document Summarization with Transformer Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 9308–9319.

[26] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. 21, 140 (2020), 1–67.

[28] Jorge Rodríguez-Ruiz, Javier Israel Mata-Sánchez, Raúl Monroy, Octavio Loyola-González, and Armando López-Cuevas. 2020. A one-class classification approach for bot detection on Twitter. 91 (2020).

[29] Sina Mahdipour Saravani, Indrajit Ray, and Indrakshi Ray. 2021. Automated Identification of Social Media Bots Using Deepfake Text Detection. In *International Conference on Information Systems Security*. Springer, 111–123.

[30] Senait G. Tesfagergish, Robertas Damaševičius, and Jurgita Kapočiūtė-Dzikienė. 2021. Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings and Deep Learning. In *Computational Science and Its Applications – ICCSA 2021*. Springer International Publishing, 523–538.
[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.
[32] Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2253–2263.
[33] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

A COLLECTED DATASET

Twitter Handle	lm	train	val	test
AndrewScheer	5,851	4,680	585	586
BarackObama	7,158	5,727	716	716
BernieSanders	8,312	6,650	831	832
ChuckGrassley	5,146	4,117	515	515
CoryBooker	17,476	13,981	1,748	1,748
DrJillStein	5,252	4,201	525	526
GavinNewsom	4,633	3,707	463	464
georgegalloway	19,385	15,508	1,938	1,939
HelenClarkNZ	7,416	5,932	742	742
HillaryClinton	4,831	3,865	483	484
JohnKasich	4,917	3,933	492	492
justinamash	4,129	3,303	413	413
KamalaHarris	7,509	6,008	751	751
keithellison	7,522	6,017	752	753
KremlinRussia_E	4,554	3,643	455	456
LindseyGrahamSC	4,069	3,255	407	407
marcorubio	6,153	4,923	615	616
marwilliamson	8,379	6,704	838	838
MassGovernor	4,913	3,930	491	492
MayorBowser	6,835	5,468	683	684
MikeBloomberg	5,595	4,476	559	560
NadineDorries	5,478	4,382	548	548
nenshi	4,815	3,852	482	482
newtgingrich	6,138	4,910	614	614
NickGriffinBU	8,008	6,406	801	801
NYCMayor	8,355	6,684	835	836
PeterTatchell	20,769	16,615	2,077	2,077
RandPaul	5,013	4,010	501	502
RonPaul	4,982	3,986	498	499
RonWyden	4,563	3,651	456	457
RoyalFamily	11,402	9,121	1,140	1,141
SarahPalinUSA	8,500	6,800	850	850
ScottWalker	8,999	7,199	900	900
seanhannity	8,179	6,544	818	818
SenatorLeahy	5,468	4,375	547	547
SenatorMenendez	5,766	4,613	577	577
SenatorSinema	5,820	4,656	582	583
SenBlumenthal	5,889	4,712	589	589
SenGilliBrand	8,004	6,404	800	801
SenJohnMcCain	6,007	4,805	601	601
SenRickScott	7,931	6,344	793	794
SenSchumer	7,936	6,348	794	794
SpeakerPelosi	4,584	3,667	458	459
SpeakerRyan	4,573	3,658	457	458
tedcruz	9,506	7,605	951	951
timkaine	4,827	3,861	483	483
WalshFreedom	31,057	24,845	3,106	3,106
Total	362,604	290,081	36,260	36,282

Table 5: Statistics on the 47 collected accounts. We divided the collections into two parts, one for text generation (lm) and one for tweet classification following the 50/50 ratio. We further divided the latter into train (train), validation (val) and test (test) sets following the 80/10/10 ratio.