Detecting Sockpuppet Accounts in Wikipedia: A Quantitative Evaluation of Different Models

Skaistė Mielinytė
The University of Edinburgh

Björn Ross
The University of Edinburgh

Keywords: sockpuppets, fake identity, sockpuppet detection, machine learning, transformers

Introduction

The sockpuppetry problem, where the same person (called the sockmaster or sockpuppeteer) creates multiple accounts and uses them for malicious purposes, is a well-known fake identity problem in Wikipedia. Wikipedia editors detect and ban sockpuppets in a 2-step process: suspected sockpuppets are reported by other users and then manually reviewed by administrators, clerks and, if requested, checkusers. Arguably, the current procedure requires too much manual effort and is prone to human error. Researchers have tried to prevent this by suggesting automated sockpuppet detection solutions based on machine learning (ML). These can can be grouped into 3 categories based on the features used: textual features (Solorio et al., 2013), non-textual features (Yamak et al., 2016) and a combination of different type of features (Yu et al., 2021) (Sakib and Spezzano, 2022). However, the use of different evaluation datasets in these studies makes the reported results directly incomparable. Thus, the best model for automation is still not known. In addition, most previous studies were conducted before the emergence of transformer models in natural language processing, which have achieved much better classification results on many tasks. This leads to two research questions:

RQ1 How do sockpuppet detection models of different types compare when evaluated on the same dataset using standard metrics?

RQ2 Does switching to transformers improve classification results for the textual models?

Methods

Models implementation

We compare models representing three different groups: for textual models - (Solorio et al., 2013), for non-textual models - (Yamak et al., 2016) and for the combination of both - (Yu et al., 2021). Since published code is not available for any of these studies, the models were reimplemented from scratch. Models were recreated by following the research paper, so implementation details can be found in the respective papers. It is important to note, though, that we needed to make some changes to be able to directly compare models. Yamak's and Yu's models are binary classification models (classifying each account as a sockpuppet or legitimate), whereas Solorio's model was designed to match sockpuppet accounts with their sockmasters. Hence we changed Solorio's model to binary classification by using the same features for account classification purposes. As some of these models are rather dated in the approaches they use, and in particular the textual model performed poorly in initial experiments (see 1), we additionally tested transformer-based textual models. Transformers are arguably the most significant revolution in natural language processing from the last decade, hence we tested how well they work on sockpuppet detection. We tested the four most downloaded HuggingFace transformers whose pre-training data includes Wikipedia, namely RoBERTa, DistilBERT, BERT and XLNet, and used them to classify accounts into sockpuppets and legitimate ones.

Models comparison

We evaluated all models on the same dataset, using the same metrics. As no sockpuppetry dataset was publicly available, we created a new dataset from people's comments on Wikipedia's talk pages. The dataset was created by using the MediaWiki API service. We retrieved 20,361 sockpuppet accounts by using the API:Blocks endpoint. The API:Usercontribs and API:Compare endpoints were then used to collect all the individual contributions in talkpages (adding, editing and deleting comments) of a user. Since those users who did not contribute to talk pages were removed, number of sockpuppets dropped to 2,483. For each sockpuppet, a matching non-sockpuppet account, that contributed to the same talkpage and was active around the same time, was found with the help of the API:Revisions endpoint. Careful selection of control group accounts resulted in a balanced dataset (see 2, 3, 1). The final dataset consisted of 4,966 users (50% sockpuppets, 50% legitimate accounts) and their 146,886 contributions to talkpages made between 2001 and 2023. Collected dataset has a Research Ethics approval (application nr. 34872) and it was published together with the source code on this Github repository.

The dataset was divided into 3 different data splits (see 3). Each model was trained and tested on each split, we reported average metrics. This was done in order to check if the model's performance is independent of the data split. Each group was implemented on 5 different ML models (SVM, RF, NB, kNN and ADA) in order to find the most compatible ML model. For traditional ML models, parameter tuning was completed with CV (3-fold CV for time-consuming models and 5-fold CV for time efficient ones). For transformers, fine-tuning was completed using the Huggingface Trainer class.

Results

Models were compared on false positive rate (FPR), true positive rate (TPR), precision and F-score. TPR (i.e. recall) measures if the classifier catches the majority of sockpuppets, however, FPR is as important because sockpuppetry is a serious accusation which might result in an account block. As the cost of false positives is high, precision is also of interest – we want a model to be certain when making a positive prediction. The F-score summarises the model's overall performance.

The recreated models based on textual features (see 1) performed poorly on this problem - the best one only detects 38.5% of sockpuppets (TPR of kNN), so most sockpuppets remain undetected. Models based on non-textual features (see 2) look more promising. The most promising model is ADA because it has the lowest FPR of all non-textual models, the highest precision and F-score. Among the recreated published models, combining textual and non-textual features resulted in the best performance. From the combined features group (see 5), RF have the best metrics and when compared with the ADA non-textual model, RF improved TPR by 8.2%, FPR by 4.8%, precision by 6.5% and F-score by 6.5%.

The picture changed dramatically when we fine-tuned transformer models for the detection task. The RoBERTa model (see 4) achieved the best results overall of 84.4% TPR, 9.8% FPR, 89.3% precision and 86.7% F-score.

Discussion and Conclusions

We directly compared three groups of approaches to detect sockpuppets on Wikipedia based on their talk page contributions. Recreating ML models and evaluating them on the same dataset (RQ1) revealed that models combining textual and non-textual features achieved the best results on the sockpuppet detection problem as they had the highest recall, precision, F-score and somewhat tolerable FPR. Interestingly enough, ML models relying on textual features alone turned out to be incompatible with the sockpuppetry problem because of very poor recall. This might have happened because, in order to allow for direct comparison, we changed the original model into a binary classification model, that is, we used the same features as the authors but for a different classification task. Even though those features were suitable for the original problem of identifying the sockmaster (Solorio et al., 2013), they might not work well when classifying users into sockpuppets and legitimate accounts.

The best performing results overall, however, were achieved when we used modern transformer models and fine-tuned them on our dataset (RQ2). Compared with all previously published models, the RoBERTa model achieved outstanding results. We believe that the model's performance may now be good enough to explore its integration with Wikipedia in order to catch sockpuppet accounts. Automating sockpuppet detection should improve decision objectivity, reduce error and allow to use Wikipedia's administrators time more efficiently. A fully automated solution, if possible at all, is likely still some time away: the RoBERTa model still has an FPR of 9.8%. Most accounts are legitimate, so 9.8% of accounts would translate into a high number of incorrectly blocked accounts. Inaccurate blocking can cause a user to lose trust in Wikipedia or even discontinue to use it. Future research could look into this and try to further decrease the number of false positives. Perhaps taking an account's edits to the articles themselves into account may help further increase model performance. Nevertheless, the presented model could arguably be used to flag potential sockpuppets for human review, or it could be used as an additional resource for those accounts which have already been reported by other users as suspected sockpuppets.

References

[Sakib and Spezzano2022] Mostofa Najmus Sakib and Francesca Spezzano. 2022. Automated detection of sockpuppet accounts in wikipedia. In *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 155–158. IEEE.

[Solorio et al.2013] Thamar Solorio, Ragib Hasan, and Mainul Mizan. 2013. A case study of sockpuppet detection in wikipedia. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 59–68.

[Yamak et al.2016] Zaher Yamak, Julien Saunier, and Laurent Vercouter. 2016. Detection of multiple identity manipulation in collaborative projects. WWW '16 Companion, page 955–960, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

[Yu et al.2021] Hang Yu, Feng Hu, Li Liu, Ziyang Li, Xiangpeng Li, and Zhimin Lin. 2021. Sockpuppet detection in social network based on adaptive multi-source features. In Victor Chang, Muthu Ramachandran, and Víctor Méndez Muñoz, editors, *Modern Industrial IoT, Big Data and Supply Chain*, pages 187–194, Singapore. Springer Singapore.

Textual features

	SVM	RF	NB	KNN	ADA
TPR AVG	0.329	0.239	0.378	0.385	0.268
TPR SD	0.029	0.043	0.046	0.086	0.059
FPR AVG	0.235	0.068	0.294	0.208	0.132
FPR SD	0.034	0.010	0.044	0.020	0.033
Precision AVG	0.559	0.665	0.548	0.604	0.607
Precision SD	0.010	0.010	0.006	0.031	0.003
F-score AVG	0.524	0.528	0.528	0.569	0.523
F-score SD	0.008	0.031	0.012	0.044	0.030

Table 1: Results of models based on textual features.

Non-textual features

	SVM	RF	NB	KNN	ADA
TPR AVG	0.618	0.693	0.912	0.709	0.694
TPR SD	0.205	0.013	0.019	0.072	0.016
FPR AVG	0.324	0.300	0.781	0.388	0.278
FPR SD	0.229	0.007	0.028	0.082	0.000
Precision AVG	0.681	0.697	0.627	0.666	0.708
Precision SD	0.005	0.003	0.013	0.006	0.008
F-score AVG	0.626	0.696	0.506	0.657	0.708
F-score SD	0.025	0.003	0.018	0.012	0.008

Table 2: Results of models based on non-textual features.

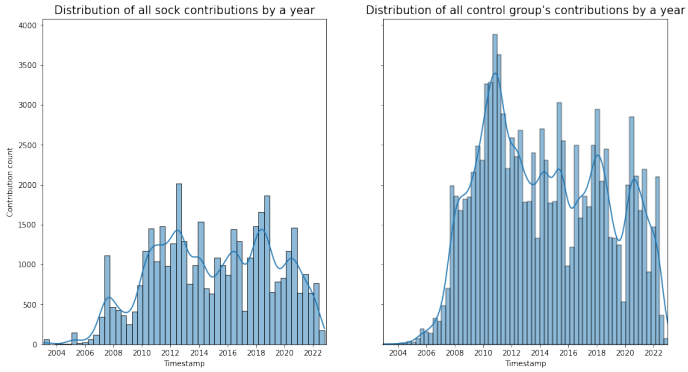

Figure 1: Distribution of all contributions by their timestamp.

	Training set (70%)		Testing set (30%)	
	#socks	#control	#socks	#control
SPLIT 1	1756	1720	727	763
SPLIT 2	1721	1755	762	728
SPLIT 3	1736	1740	747	743

Table 3: Description of different data splits.

Transformers

	RoBERTa	DistilBERT	BERT	XLNet
TPR AVG	0.844	0.766	0.744	0.632
TPR SD	0.010	0.017	0.005	0.335
FPR AVG	0.098	0.278	0.274	0.385
FPR SD	0.010	0.006	0.021	0.268
Precision AVG	0.893	0.727	0.725	0.646
Precision SD	0.012	0.012	0.009	0.060
F-score AVG	0.867	0.746	0.734	0.566
F-score SD	0.005	0.011	0.004	0.200

Table 4: Results of transformers (belong to textual group).

Combined features

	SVM	RF	NB	KNN	ADA
TPR AVG	0.709	0.776	0.910	0.654	0.749
TPR SD	0.009	0.008	0.011	0.009	0.010
FPR AVG	0.328	0.230	0.730	0.395	0.265
FPR SD	0.032	0.009	0.010	0.021	0.016
Precision AVG	0.691	0.773	0.653	0.629	0.742
Precision SD	0.012	0.003	0.007	0.009	0.008
F-score AVG	0.690	0.773	0.543	0.629	0.742
F-score SD	0.013	0.003	0.006	0.009	0.008

Table 5: Results of models based on combined features.

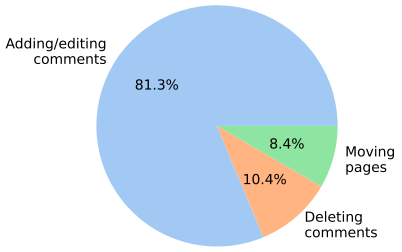

Figure 2: Sockpuppets' contributions.

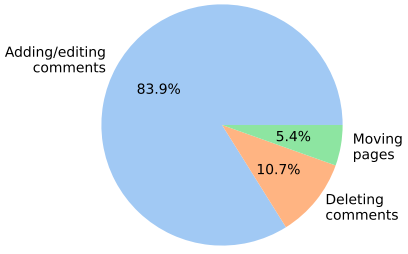

Figure 3: Non-sockpuppets' contributions.