Spot the Bot: Distinguishing Human-Written and Bot-Generated Texts Using Clustering and Information Theory Techniques

Vasilii Gromov[1][0000−0001−5891−6597], Quynh Nhu Dang[1][0000−0003−0450−7063]

National Research University Higher School of Economics, Moscow, Russia
stroller@rambler.ru, dqnhu00@gmail.com

Abstract. With the development of generative models like GPT-3, it is increasingly more challenging to differentiate generated texts from human-written ones. There is a large number of studies that have demonstrated good results in bot identification. However, the majority of such works depend on supervised learning methods that require labelled data and/or prior knowledge about the bot-model architecture. In this work, we propose a bot identification algorithm that is based on unsupervised learning techniques and does not depend on a large amount of labelled data. By combining findings in semantic analysis by clustering (crisp and fuzzy) and information techniques, we construct a robust model that detects a generated text for different types of bot. We find that the generated texts tend to be more chaotic while literary works are more complex. We also demonstrate that the clustering of human texts results in fuzzier clusters in comparison to the more compact and well-separated clusters of bot-generated texts.

Keywords: Semantic analysis · Clustering · Information theory

1 Introduction

With the development of NLP methods it has become increasingly more difficult to distinguish computer-generated texts from human literature. Many advances have been made in bot detection in various fields. However, state-of-the-art solutions are obtained using supervised methods and depend heavily on labelled data. Not as many works concentrate on self-supervised or unsupervised learning and those that do usually deal with particular bots. Our main objective is to conduct a careful study of semantic paths of both literature and bot-generated texts to find a black-box method for spotting bots. The goal is to find a procedure that distinguishes human-written texts from bot-generated texts without prior knowledge about the bot.

Our study provides a general view on how human-written texts and bot-generated texts differ on a semantic level and studies the compactness, separability and noisiness of clusters, as well as the types of text series (deterministic/chaotic/stochastic). Our hypothesis is that these characteristics should differ

for human-written and bot-generated texts, and the findings can be used to create an algorithm for bot identification. The advantage of this algorithm lies in its universality and its ability to work with bots of different types — from simple Recurrent Neural Network models to more advanced GPT bots. Our study has shown that different methods highlight various properties of the semantic space. The analysis of the characteristics of semantic paths has shown that human-written texts are more complex, while the bot-generated texts tend to be simpler and more chaotic. The clustering of data has resulted in more compact and well-separated clusters for bot-generated texts and fuzzier clusters for human-written texts. The rest of this paper is organized as follows. In the next section we review recent advances in the bot detection field. Section 3 outlines the methods we have used for the analysis of semantic space. Section 4 provides the description of conducted experiments and presents the results. In Section 5 we give our conclusions.

2   Literature Review

Recent years have seen a surge of interest in the bot detection task. Most studies employ feature-based supervised learning algorithms and centre around constructing features which are then used to build a classification model. There are a variety of methods to build such features. [9] use simple lexical and syntactic features like letter frequency or average word length. [8] derive sentiment qualities of English and Dutch tweets by calculating their polarity. [3] model a Twitter user through a set of stylistic features and distinguish bots from human accounts by analysing the consistency of their post style. [4] combine text feature engineering and graph analytics. Similarly, [6] propose SentiBot, an architecture that combines graph-based and sentiment and semantic analysis techniques. In our study, we focus on unsupervised machine learning algorithms, rather than supervised learning methods, and engineer features by clustering texts, examining the resulting semantic space and extracting various characteristics.

Other approaches are based on information theory. [5] characterise the differences between bot and human activity on Twitter by calculating the entropy of account activity statistics. They have found that humans have higher entropy than bots, which highlights their more complex timing behaviour. In our work we apply similar ideas to semantic trajectories of text data instead of meta-data. In [7] the authors study a natural language as an integral whole and ascertain that it is a self-organised critical system, whereas a separate literature text is 'an avalanche' in a semantic space. The latter fact further reinforces the argument for considering a trajectory in a semantic space as a unified object.

3   Methodology

3.1   Data

For the human written corpus, the literary books were obtained via open sources. See Table 1 for corpora details and python libraries used for each language. To

Table 1. Literature corpora details.

language	corpus size	unique bigrams	library
English	11008	8m	spacy
Russian	12692	3m	natasha
Vietnamese	1071	6m	pyvi

obtain bot-generated texts two models were utilised — a simple Long Short Term Memory recurrent neural network (LSTM), and a GPT-3. We use different models in order to design a working identification algorithm on both simple and complex bots. We train the LSTM model on subsets from the literary corpora and select pretrained GPT models from the huggingface database. To generate texts, for every 500th word from a literary piece we generate a text abstract of 500 words (the conventional size of a book page), therefore, the texts are generated of similar lengths as literary texts.

3.2   Embeddings

Word embeddings are obtained using the SVD of a document-term matrix [1] and the word2vec models [12]. The decision to use these two techniques was based on their semantic properties – both SVD and word2vec embeddings capture the structural relationships between words. In order to study word order correlations, we split the texts into n-grams and obtain final embeddings by concatenating word embeddings for each word in an n-gram. The collection of n-gram embeddings for each text is further referred to as a semantic path.

3.3   Clustering

To analyse the semantic space, we use Wishart (density-based) [16] and K-Means [11] clustering techniques[1]. We additionally explore fuzzy implementations of these algorithms to allow for the noisiness and imprecise nature of real-life data. We consider two algorithms: fuzzy clustering C-Means, [2], which is similar to K-Means, and Wishart clustering on fuzzy numbers.

To fuzzify the data, we use the notion of fuzzy numbers with trapezoidal membership functions[13]. For each j-th component of an m-dimensional object x we define the value for the fuzzy membership function as $\mu_j(x_j) = \frac{n_j}{\max_j n_j}$, where n_j is the normalised frequency of j-th component in the text. With fixed parameter values of $l_j, r_j, \Delta c = m_{2j} - m_{1j}$ we construct the fuzzy number. The ordered set of fuzzy numbers for each component of x is the fuzzification of x.

To fuzzify n-grams, we join fuzzifications of the words from n-grams accordingly to the fuzzy logic, i.e. take the minimum of fuzzy number membership functions. Finally, to use Wishart clustering algorithm (which only requires pairwise distances) on fuzzy data, we calculate the fuzzy distance as defined in [13].

---

[1] Each algorithm has its advantages — K-Means separates spherical clusters well, whereas Wishart algorithm does not make any assumptions about cluster shapes.

3.4   Entropy-complexity plane

The second method proposed in [14] distinguishes chaotic semantic paths from deterministic and stochastic ones. In order to test our hypothesis that the bot-generated texts are less complex and more chaotic, we calculate complexity and entropy measures of the word permutations. The position of the point in relation to the lower and upper theoretical boundaries points to the type of the series in question. Namely, simple deterministic processes occupy the bottom left corner of the plane, stochastic processes, the bottom right corner, whereas chaotic (complex deterministic) processes occupy areas adjacent to the vertex of the upper curve [14]. We also propose a modified variation for multidimensional use: for m-dimensional time series $(x^t)_{t=1}^L$, $x_t \in \mathbb{R}^m$ for each of m components of an n-gram we obtain permutation π_d as in one-dimensional case. For multidimensional case we define the final permutation as $\Pi = (\pi_1, \pi_2, \ldots, \pi_m)$.

4   Results

4.1   Clustering

Prior to text feature extraction using clustering results, we run experiments with the total collections of n-grams found in text corpora. For each type of corpora (human/bot, different languages) 3 million unique n-grams are selected. In order to differentiate bot-generated texts and human-written clusterisations, we study the compactness, separability and noisiness of their clusters. Both the

Table 2. Wilcoxon test p-values for RMSSTD distribution.

	Russian		English		Vietnamese	
	LSTM	GPT	LSTM	GPT	LSTM	GPT
K-Means	5.63e-3	8.61e-88	7.47e-4	4.93e-2	2.12e-3	1.50e-2
Wishart	5.92e-3	8.15e-28	4.51e-3	2.29e-2	1.32e-5	9.33e-3

Wishart and K-Means algorithms result in more compact and less separated clusters for bots measured by the RMSSTD and RS metrics [17]. The three languages share a resemblance — the clusters for literature corpora are less compact compared to those of bots. The nonparametric Wilcoxon test [15] shows statistically significant differences between RMSSTD distributions of literature and bots corpora: p-values are less than 0.05 (see Table 2).

The Wishart clustering algorithm can also be used to find noisy data. We have found that out of all types of texts, those generated by LSTM model are the noisiest, while human written and GPT-generated texts are similar in the noise percentage (see Figure 1). We propose a following interpretation for this observation — the LSTM texts are semantically simpler and the diversity of the texts are mainly achieved by the noise generation.
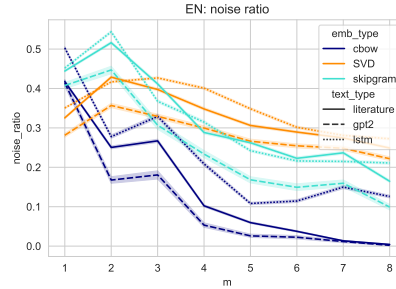
Fig. 1. Noise ratio in English data (found with Wishart algorithm on fuzzified data).

Table 3. Classification performance (accuracy) with intercluster distance measures.

	Literature vs.	**LSTM+GPT**		**LSTM**		**GPT**	
Language	**Algorithm**	Train	Test	Train	Test	Train	Test
English	K-Means	0.947	0.975	1.0	1.0	0.903	0.881
	Wishart	**0.953**	**0.975**	1.0	1.0	0.904	0.881
	C-Means	0.943	0.970	0.999	1.0	0.897	0.921
	Wishart+Fuzzy	0.945	0.947	1.0	1.0	**0.907**	**0.94**
Russian	K-Means	0.912	0.934	0.999	1.0	0.871	0.916
	Wishart	**0.937**	**0.954**	0.999	1.0	**0.913**	**0.944**
	C-Means	0.882	0.894	0.999	1.0	0.838	0.857
	Wishart+Fuzzy	0.882	0.913	0.991	1.0	0.904	0.911
Vietnamese	K-Means	0.862	0.903	1.0	1.0	0.887	0.881
	Wishart	0.902	0.896	1.0	1.0	0.893	0.900
	C-Means	0.887	0.893	1.0	1.0	0.871	0.871
	Wishart+Fuzzy	**0.929**	**0.942**	1.0	1.0	**0.893**	**0.926**

Based on these findings, we move on to clustering n-grams for each text in order to extract features. As previous experiments have shown that bots have more compact and less separated clusters, we use inter-cluster distances (average, maximum and minimum) as features. Simple SVC models (separate models for each set of parameters and text types) with L2 regularisation are trained and cross-validated on data subsets (1000 texts for each corpus). Table 3 shows the best results for each language. We found that the texts are better distinguished with features extracted from the Wishart algorithm. It is possible that K-Means, as well as its fuzzy variance C-Means, perform worse due to the abstract form of the noisy clusters. It is worth noting that fuzzification improves classification performance on English and Vietnamese texts.

4.2 Entropy-Complexity Plane

For certain parameter sets the entropy-complexity measures can fall into noise or deterministic areas, in which it is difficult to identify different types of texts. To account for this nuisance, we first analyse the values of m and n for which the
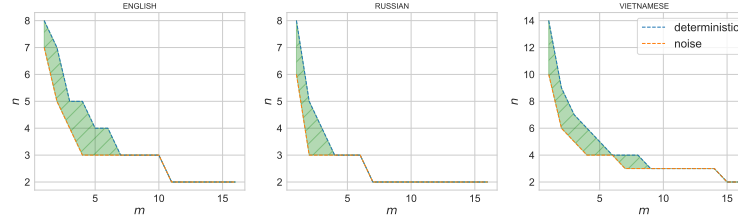
Fig. 2. Chaotic area parameter values for English, Russian and Vietnamese data with Skip-gram embeddings.

literary texts fall into chaotic area on the entropy-complexity plane (i.e. close to the upper theoretical boundary). Such parameter sets are marked with the green area in Figure 2. Sets below the area border result in texts appearing in noise area, above — deterministic area. Values differ significantly for each language: longer sequences fall into the chaotic area with values of n varying from 10 to 14 for $m = 1$ for Vietnamese, whereas for English and Russian the sequences are shorter — n varies from 7 to 8 and from 6 to 8 accordingly.
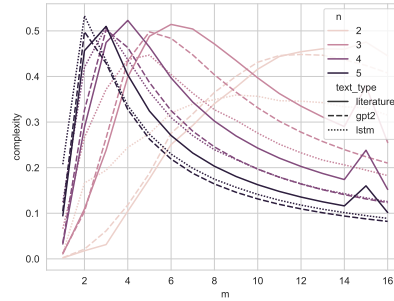

Fig. 3. Mean complexity measure on English data.

On average, the literary texts are more complex, although it is worth noting that for bigrams the more complex texts are LSTM-generated ones (Figure 3). We believe this happens due to the vast variety of bigrams themselves: more logically coherent texts written by humans or generated by GPT models do not include as many bigrams as the simpler LSTM-generated texts.

For the selected parameter sets we build classification model with entropy and complexity measures as features. Again, for the model we use a simple SVC with L2 regularisation. We originally tried classifying texts with the addition of m and n as numeric features, but such a model only achieved 0.57 accuracy on test set. The models for separate parameter sets perform much better, see Table 4

Table 4. Classification performance (accuracy) based on entropy-complexity measures.

Literature vs.	LSTM+GPT		LSTM		GPT	
Language	Train	Test	Train	Test	Train	Test
English	0.937	0.965	0.999	1.0	0.997	1.0
Russian	0.879	0.890	0.991	0.992	0.889	0.893
Vietnamese	0.981	0.989	1.0	1.0	0.991	0.995

for the best models. LSTM texts are well separated on the entropy-complexity plane, a simple SVC achieves 100% accuracy. GPT texts are also distinguished well — for English and Vietnamese the accuracy is 99%, for Russian — 90%. The binary classification model for both bots achieves highest accuracy on Skip-gram data, $m = 1, n = 3$ in English; for Russian — Skip-gram, $m = 1, n = 8$; for Vietnamese — SVD, $m = 3, n = 3$.

5   Conclusions and further directions

In order to differentiate generated texts from human literature, we have employed different techniques, such as crisp and fuzzy clustering and entropy-complexity plane construction. We have found that these methods, supplemented by a careful parameter selection, can be used to obtain features with significant differences for different text types. We are therefore able to build robust identification algorithms without prior knowledge of bot-model architecture. The final classification models achieve up to 99% accuracy for English and Vietnamese data and 94% for Russian. These methods do not require a lot of labelled data and thus can be easily downstreamed to other tasks, such as fraud detection. As a possible future direction for this work, we also propose an analysis of the methods of this research in application to other languages of varying language families.

Acknowledgements

References

1. Bellegarda, J.R.: Latent semantic mapping: Principles & applications. Synthesis Lectures on Speech and Audio Processing **3**(1), 1–101 (2007)
2. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. Computers & geosciences **10**(2-3), 191–203 (1984)
3. Cardaioli, M., Conti, M., Di Sorbo, A., Fabrizio, E., Laudanna, S., Visaggio, C.A.: It'sa matter of style: Detecting social bots through writing style consistency. In: 2021 International Conference on Computer Communications and Networks (IC-CCN). pp. 1–9. IEEE (2021)

4. Chakraborty, M., Das, S., Mamidi, R.: Detection of fake users in twitter using network representation and nlp. In: 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS). pp. 754–758. IEEE (2022)
5. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Transactions on dependable and secure computing **9**(6), 811–824 (2012)
6. Dickerson, J.P., Kagan, V., Subrahmanian, V.: 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. IEEE (2014)
7. Gromov, V.A., Migrina, A.M.: A language as a self-organized critical system. Complexity **2017** (2017)
8. Heidari, M., James Jr, H., Uzuner, O.: An empirical study of machine learning algorithms for social media bot detection. In: 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). pp. 1–5. IEEE (2021)
9. Kang, A.R., Kim, H.K., Woo, J.: Chatting pattern based game bot detection: do they talk like us? KSII Transactions on Internet and Information Systems (TIIS) **6**(11), 2866–2879 (2012)
10. Kostenetskiy, P., Chulkevich, R., Kozyrev, V.: Hpc resources of the higher school of economics. In: Journal of Physics: Conference Series. vol. 1740, p. 012050. IOP Publishing (2021)
11. MacQueen, J.: Classification and analysis of multivariate observations. In: 5th Berkeley Symp. Math. Statist. Probability. pp. 281–297 (1967)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Novák, V., Perfilieva, I., Mockor, J.: Mathematical principles of fuzzy logic, vol. 517. Springer Science & Business Media (2012)
14. Rosso, O.A., Larrondo, H., Martin, M.T., Plastino, A., Fuentes, M.A.: Distinguishing noise from chaos. Physical review letters **99**(15), 154102 (2007)
15. Wilcoxon, F.: Individual comparisons by ranking methods. In: Breakthroughs in statistics, pp. 196–202. Springer (1992)
16. Wishart, D.: Numerical classification method for deriving natural classes. Nature **221**(5175), 97–98 (1969)
17. Xiong, H., Li, Z.: Clustering validation measures. In: Data Clustering, pp. 571–606. Chapman and Hall/CRC (2018)