Generative AI Detection Using Simple Feature Selection and SVM

Notebook for PAN at CLEF 2025

Joseph Larson[1]

[1]*Indiana University, Bloomington, IN., USA*

Abstract

Generative AI detection has been of interest for at least the past decade, but especially since the emergency of transformer powered LLMs. This paper treats the task as a binary classification problem, where $y \in [0, 1]$. I chose to use a traditional Support Vector Classifier (SVC) with sets of features chosen from examination of the training data set to determine what features human authors, as opposed to AI, are more likely to employ. I found the top 40 unigram and bigrams, along with the top 15 punctuation features, to be the most informative. When combined and input into my SVC, I achieved a mean (the mean of all scores used for the task) score of 94.89.

Keywords

PAN 2025, AI Detection, Feature Selection, SVC

1. Introduction

As Large Language Model (LLM) decoder models become readily more available, the demand for systems which can distinguish texts written by them from human authors has skyrocketed. Although generated text has its uses, concerns have arisen regarding its improper use e.g. phishing schemes [1], phony product reviews [2] and fake news [3, 4, 5]. In addition, research has shown that human evaluators perform only slightly better than chance at identifying machine-authored text [1, 6].

I approach this task [7] that is at the PAN workshop [8] as a binary authorship attribution task. Feature selection is a common approach to this task, since different authors employ different linguistic and stylometric features. Character and n-gram counts have been common features employed in authorship attribution models even as recently in the past decade [9, 10, 11]. Other common features employed are POS-tags [12, 13], Topic Modeling (e.g. LDA, DADT and AT) [14, 15, 16] and LIWC [5]. For this task, I experiment primarily with n-gram features and stylometric features like punctuation. My reasoning for this will become clear in the next section of this paper. I submitted the final work through Tira, like the other task participants [17].

2. Related Work

In terms of impressionistic feature differences between human and A.I. authors, it has been stated that overall LLMs are more focused (that is to say, they never leave the subject matter at hand), more objective and highly formal. In contrast, their human counterpoints employ more subjective language, less formal and demonstrate an increased propensity to stray from the topic. Linguistically, humans employ less nouns and conjunctions and LLMs employ less punctuation and adverbs. Dependency relations for humans are also much shorter. Lastly, humans show more 'creativity in terms of word choices', therefore human texts on average show a higher type to token ratio. [6].

To this last point, it has been demonstrated that LLMs sample from a limited amount of tokens to generate natural looking text, e.g. through mass sampling [18] or k-max sampling [19]. For this reason, [20] found relative success using n-gram tf-idf features (unigrams and bigrams) to distinguish between human and GPT2 redacted web pages. They created a dataset using three sampling methods: k-sampling

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

joelarso@iu.edu (J. Larson)

(sampling the highest probability tokens until a threshold of specified tokens is reached), *p*-sampling (sampling from the smallest possible set of words until a cumulative probability is reached) and so-called 'pure' sampling (a.k.a temperature sampling, where lower 'temperatures' are associated with higher probabilities for tokens). Since *k* samples overproduce common words, they were the easiest to detect. [21] use three statistical features: probability of each word, absolute rank of each word and the entropy of the word's distribution. It was found that GPT2 oversamples certain words, allowing the model (in this case, BERT) to easily detect LLM generated text.
In terms of models, [3] has claimed that the best detector of LLM generated text are LLMs themselves. The aforementioned Gehrmann et al. study found that finetuning GPT2 did not yield better results. Inspired by this, my goal was to develop an LLM text detector that relied on purely statistical models rather than transformers; this type of method would be computationally inexpensive and easily employable on a person's local machine.

3. Task Overview

The present task involves a binary classification task, whereby documents in the dataset are classified as either being human authored or machine authored. I experiment with systems that return both binary labels and probabilities, where scores approaching 0 indicate probable human authorship and scores approaching 1 indicate probable machine authorship. Scores of 0.5 indicate that the system is unsure.

3.1. Dataset

The training set used had a total of 23,707 documents while the validation set had a total of 3,589. The human class was lower for both splits, with it representing roughly two fifths of training set and roughly a third of the validation set. Various models were used to create the machine documents. Table 1 contains a more detailed summary.

Table 1
Information on dataset used in Task.

Split	Class	Size	Genres	Models
Train	Human	9,101	Essays, Fiction and News	Llama, O3, GPT, Deepseek, Gemini, Falcon, Bison, Qwen, Ministral
	Machine	14,606		
Val	Human	1,058		
	Machine	2,531		

3.2. Model Selection

As stated in section 2, my goal for this project was specifically to not use a transformer model. I wanted to use a model that could be easily employed on more traditional, less computationally expensive model. SVC's and SVM's have both proved to be successful in the domain of authorship attribution, so I chose to use an SVC as my model. In terms of hyperparameters, I experimented with different kernels and class weights. I found RBF to be the most dynamic kernel and 'balancing' (see equation 1: w_c is the class weight, N_d is the number of documents, N_c is the number of class and $B(y)$ is the bin count of the classes).

$$w_c = \frac{N_d}{N_c} B(y) \tag{1}$$

3.3. Feature Selection

My initial approach to feature selection was examining the dataset for anomalies. I wanted to first analyze the claim that human texts haver higher type to token ratios. Figure 1 shows a density plot for

the training dataset of type to token ratios. This graph shows that although the human texts show a normally distributed curve and the machine generated text is much more irregular distributed, there is a lot of overlap, meaning this feature probably would not be helpful for classification.
Next, I examined the top unigrams, bigrams and trigrams for both classes, to see if certain n-grams were more common in one class. I came to the conclusion that the top forty unigrams and bigrams were the most informative in distinguishing between the two classes. After this, since the literature had stated AI uses less punctuation, I considered of using this as a feature for the SVC. After careful examination of the dataset, the first 15 punctuation patterns proved to be the most informative. A summary of the features used for my model can be found in Table 2.

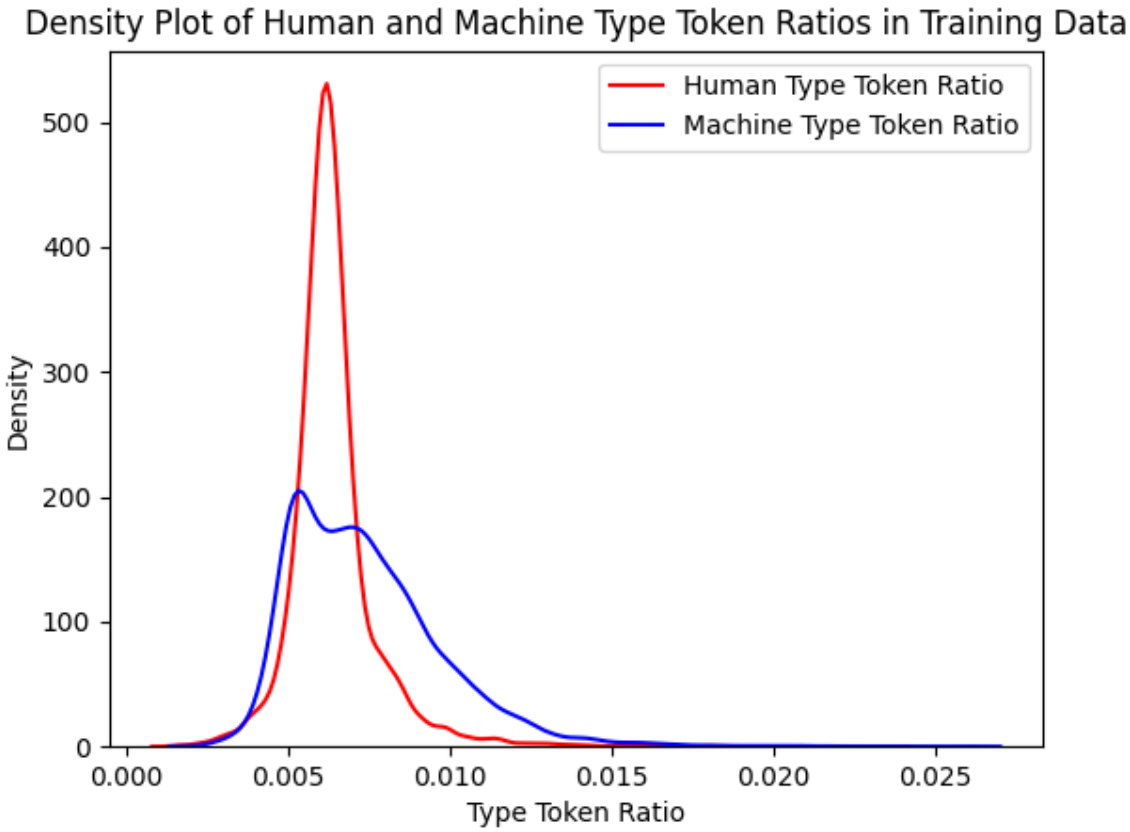

Figure 1: Density plot showing the type to token ratios for both Human and AI authored texts.

3.4. Evaluation

To evaluate the performance of the model, the metrics used by the task were:

- **AUC** (a.k.a. AUC Roc score): The area under the curve score.
- **C@1** (a.k.a. Classification at 1): The percentage of instances where the top score was the correct one.
- $\mathbf{F}_{0.5}$: The harmonic mean of precision and recall, with $\beta = 0.5$.
- **F1**: The harmonic mean of precision and recall, with $\beta = 1$.
- **Brier**: The complement of the Brier Score Loss, which is the mean squared difference between the predicted class and actual class.
- **Mean**: The arithmetic mean of all previous scores.
  All metrics naturally produce a score between 0 and 1 and are multiplied by 100.

Table 2
Feature rankings for N-gram and Punctuation Features used in the final SVM

N-gram Features				Punctuation Features	
Rank	**Feature**	**Rank**	**Feature**	**Rank**	**Feature**
1	*the*	21	*not*	1	,
2	*of*	22	*but*	2	.
3	*and*	23	*in the*	3	’
4	*to*	24	*by*	4	"
5	*in*	25	*at*	5	;
6	*that*	26	*this*	6	."
7	*his*	27	*their*	7	,"
8	*with*	28	*be*	8	?
9	*was*	29	*from*	9	!
10	*he*	30	*an*	10	?"
11	*as*	31	*my*	11	:
12	*her*	32	*have*	12	!"
13	*it*	33	*him*	13	.’
14	*for*	34	*to the*	14	,’
15	*of the*	35	*they*	15	...
16	*had*	36	*we*		
17	*is*	37	*said*		
18	*on*	38	*are*		
19	*you*	39	*were*		
20	*she*	40	*one*		

4. Results

To obtain my results, I experimented with different numbers of features. While experimenting with just unigrams and bigrams, I ran models for up to 1,000 count features. My conclusion was that only the top 40 impacted the document class. For the punctuation features, I found there to be 153 unique punctuation patterns within the dataset. I ran experiments with varying numbers of these patterns and found the top 15 to be the most distinctive. My hypothesis was then that combining the features would yield an improvement in the model; I was proven correct. The best model I was able to train was with the top 15 punctuation features and the top 40 unigram and bigram features. Figure 2 shows a PCA of the SVM’s feature space. Table 3 shows my final results on the validation set and table 4 shows my results on the final test set compared to the other baselines.

Table 3
Performance comparison of different feature sets on validation set provided by organizers

Feature Set	**AUC**	**C@1**	F_{05}	**F1**	**Brier**	**Mean**
Punctuation only	81.77	81.77	86.47	89.36	85.48	84.97
Words only	91.16	91.16	93.53	94.02	92.23	92.42
Words and Punctuation	93.33	93.99	95.49	96.09	94.90	94.89

5. Discussion and Conclusion

With this short paper, I have shown that feature selection is still an effective method in AI detection. LLM models clearly still produce text that over sample certain words, as found to the case by [21, 20]. In addition, punctuation patterns prove to be a distinguishing factor: humans use wider ranges of punctuation patterns and use them with higher frequency. I have been able to demonstrate all of this without finetune a State-of-the-Art transformer model, with experiments I have run on my local machine

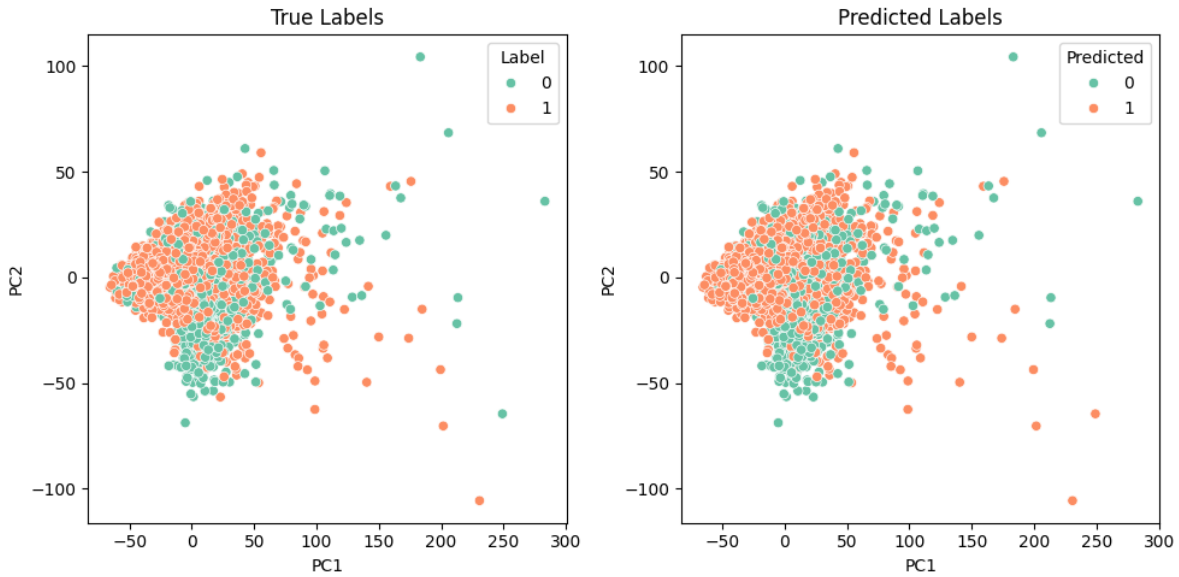

Figure 2: PCA graph showing the feature space for the true labels (left) and the predicted labels (right). 0 corresponds to human authored texts and 1 corresponds to machine generated texts.

Table 4
My final results on the test dataset compared to the three baselines provided by the task organizers

Software	ROC-AUC	Brier	C@1	F1	F0.5u	Mean	FPR	FNR
Baseline TF-IDF SVM	0.838	0.871	0.836	0.827	0.862	0.856	0.201	0.153
Baseline Binoculars Llama3.1	0.760	0.835	0.793	0.802	0.831	0.818	0.314	0.206
PPMd CBC	0.636	0.795	0.735	0.763	0.771	0.758	0.784	0.129
Larson	0.734	0.799	0.799	0.829	0.850	0.814	0.330	0.178

(when doing feature selection, I did at certain points have to use my university's super computer[1], however one feature selection was finished training for my models required less than a minute). While transformer models may still yield higher performance than the model I present in this paper, my work serves as a reminder that feature selection is still a powerful method within the domain of AI text detection.

My work, is of course, not without its limitations. I could have done more hyperparameter fine-tuning to possibly improve my model even more. I also could have carried a more profound analysis of what features were actually relevant in distinguishing the two classes. The top baseline for this task reports TF-IDF features as having achieved a mean of 0.98; while I experimented with TF-IDF features, my findings were that they did not outperform raw count features. Future work should focus upon creating more robust datasets that not only contain a diversity of different models, but also a diversity of sampling models, as demonstrated by [20] to be an important factor in AI detection.

6. Declaration on Generative AI

During the preparation of this work, the author used not a single AI tool for any purpose.

[1]I acknowledge the Indiana University Pervasive Technology Institute for providing supercomputing and storage resources that have contributed to the research results reported within this paper [22]

References

[1] M. Weiss, Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions., Technology Science 2019121801 (2019).

[2] D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, I. Echizen, Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection, 2019. arXiv:1907.09177.

[3] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against Neural Fake News, Curran Associates Inc., Red Hook, NY, USA, 2019.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[5] A. Uchendu, T. Le, K. Shu, D. Lee, Authorship attribution for neural text generation, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 8384–8395. URL: https://aclanthology.org/2020.emnlp-main.673. doi:10.18653/v1/2020.emnlp-main.673.

[6] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is chatgpt to human experts? comparison corpus, evaluation, and detection, 2023. arXiv:2301.07597.

[7] J. Bevendorff, D. Dementieva, M. Fröbe, J. Karlgren, M. Mayerl, P. Nakov, A. Panchenko, M. Potthast, A. Shelmanov, E. Stamatatos, B. Stein, Y. Wang, M. Wiegmann, E. Zangerle, Overview of PAN 2025: Generative AI Authorship Verification, Multi-Author Writing Style Analysis, Multilingual Text Detoxification, and Generative Plagiarism Detection, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2025.

[8] J. Bevendorff, Y. Wang, J. Karlgren, M. Wiegmann, M. Fröbe, A. Tsivgun, J. Su, Z. Xie, M. Abassy, J. Mansurov, R. Xing, M. N. Ta, K. A. Elozeiri, T. Gu, R. V. Tomar, J. Geng, E. Artemova, A. Shelmanov, N. Habash, E. Stamatatos, I. Gurevych, P. Nakov, M. Potthast, B. Stein, Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2025, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[9] Y. Sari, A. Vlachos, M. Stevenson, Continuous n-gram representations for authorship attribution, in: M. Lapata, P. Blunsom, A. Koller (Eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 267–273. URL: https://aclanthology.org/E17-2043/.

[10] A. Sharma, A. Nandan, R. Ralhan, An investigation of supervised learning methods for authorship attribution in short hinglish texts using char word n-grams, 2018. URL: https://arxiv.org/abs/1812.10281. arXiv:1812.10281.

[11] A. Zečević, N-gram based text classification according to authorship, in: I. Temnikova, I. Nikolova, N. Konstantinova (Eds.), Proceedings of the Second Student Research Workshop associated with RANLP 2011, Association for Computational Linguistics, Hissar, Bulgaria, 2011, pp. 145–149. URL: https://aclanthology.org/R11-2023/.

[12] K. Sundararajan, D. Woodard, What represents "style" in authorship attribution?, in: E. M. Bender, L. Derczynski, P. Isabelle (Eds.), Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico,

USA, 2018, pp. 2814–2822. URL: https://aclanthology.org/C18-1238/.

[13] E. Ferracane, S. Wang, R. Mooney, Leveraging discourse information effectively for authorship attribution, in: G. Kondrak, T. Watanabe (Eds.), Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Asian Federation of Natural Language Processing, Taipei, Taiwan, 2017, pp. 584–593. URL: https://aclanthology.org/I17-1059/.

[14] Y. Seroussi, I. Zukerman, F. Bohnert, Authorship attribution with Latent Dirichlet Allocation, in: S. Goldwater, C. Manning (Eds.), Proceedings of the Fifteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 181–189. URL: https://aclanthology.org/W11-0321/.

[15] Y. Seroussi, F. Bohnert, I. Zukerman, Authorship attribution with author-aware topic models, in: H. Li, C.-Y. Lin, M. Osborne, G. G. Lee, J. C. Park (Eds.), Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 264–269. URL: https://aclanthology.org/P12-2052/.

[16] Y. Seroussi, I. Zukerman, F. Bohnert, Authorship attribution with topic models, Computational Linguistics 40 (2014) 269–310. URL: https://aclanthology.org/J14-2003/. doi:10.1162/COLI_a_00173.

[17] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241.

[18] J. Gu, K. Cho, V. O. Li, Trainable greedy decoding for neural machine translation, in: M. Palmer, R. Hwa, S. Riedel (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1968–1978. URL: https://aclanthology.org/D17-1210/. doi:10.18653/v1/D17-1210.

[19] A. Fan, M. Lewis, Y. Dauphin, Hierarchical neural story generation, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 889–898. URL: https://aclanthology.org/P18-1082. doi:10.18653/v1/P18-1082.

[20] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, J. Wang, Release strategies and the social impacts of language models, 2019. arXiv:1908.09203.

[21] S. Gehrmann, H. Strobelt, A. Rush, GLTR: Statistical detection and visualization of generated text, in: M. R. Costa-jussà, E. Alfonseca (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Florence, Italy, 2019, pp. 111–116. URL: https://aclanthology.org/P19-3019/. doi:10.18653/v1/P19-3019.

[22] C. A. Stewart, V. Welch, B. Plale, G. Fox, M. Pierce, T. Sterling, Indiana University Pervasive Technology Institute, Technical Report, Indiana University, 2017. URL: https://doi.org/10.5967/K8G44NGB. doi:10.5967/K8G44NGB.