Artificial text detection via magnitude functions

Hassan Iftikhar Pavel Gurevich

Abstract

Rapid advancement of generative language models has blurred the distinction between human- and machine-generated text, posing significant challenges for fakes detection, verification, and academic integrity enforcement. Current detection methods often rely on lexical or statistical features, which struggle to generalize as the models evolve. To address this, we propose a novel geometric framework that utilizes magnitude functions, a tool from topological data analysis (TDA), to quantify structural differences in text representations derived from language model embeddings.

By mapping texts into high-dimensional embedding spaces, we analyze their geometric "shape" through the magnitude function, which captures the multiscale topological properties of point-cloud distributions. We hypothesize that human-authored texts exhibit distinct organizational patterns (e.g., coherence, semantic consistency) compared to AI-generated outputs, which may manifest as quantifiable disparities in magnitude function profiles.

This work bridges machine learning and topological analysis to advance AI-generated text detection, offering a scalable, model-agnostic solution. Our findings underscore the potential of geometric features as interpretable markers for synthetic text, with implications for trustworthiness in online social, scientific and educational content.

1. Introduction

Modern generative Large Language Models, such as GPT-4, Mistral, and DeepSeek, performed a revolution in the text creation area. Now, generated texts are extremely similar to human-made ones. LLMs are used for text translation, polishing, grammar check, style correction and many other tasks. However, such models may also be used for some bad purpose, like online bullying, cheating, fraud activities, or spam. That leads to the need of some effective and robust detection technique, which could allow to distinguish LLM-generated and human-written texts with high accuracy.

Three major text detection approaches could be highlighted:

1. Perplexity-based algorithms (Kushnareva et al., 2024)
2. Statistics/stylistic based techniques (Juzek & Ward, 2025)
3. Supervised models trained on text embeddings (Hu et al., 2023)

Unfortunately, all of these methods have drawbacks that will be discussed in Section 2.

We propose the paradigm shift: instead of analyzing text through lexical or probabilistic lenses, we interrogate the geometric organization of language in embedding spaces. Here we introduce magnitude functions (Leinster & Meckes, 2017), a tool that quantifies multi-scale geometric complexity in point cloud data. By treating a text's embedding sequence as a topological space, we compute its magnitude function, a curve encoding the "shape" of the text's semantic trajectory at varying scales. We posit that magnitude profiles capture intrinsic differences in how humans and LMs organize information.

This work bridges machine learning and computational topology, offering insights into the geometric fingerprints of machine intelligence. Beyond detection, our findings illuminate fundamental differences in how humans and LLMs structure language — a step toward more interpretable AI accountability frameworks.

2. Related work

Many researchers have tried to figure out how to spot AI-generated text. The most significant methods are listed below:

Linguistic Feature-Based Methods: (Ippolito et al., 2020) discovered how often common words (like "the" or "and") appear, how long sentences are, or how punctuation is used. They found AI texts often overuse certain words, which can be a giveaway. This method is extremely fast and easy to get, but it does not work well with newer AI models that seem more human. It is great for catching obvious fakes but not so good for tricky ones.

Supervised Learning: (Hu et al., 2023) proposed RADAR, a framework to improve AI-generated text detection by adversarially training a paraphraser and a detector iteratively, enhancing robustness against paraphrasing attacks. They evaluated RADAR across 8 diverse LLMs and 4 datasets, demonstrating superior detection accuracy compared to existing methods, especially when paraphrasing is used.

Watermarking: (Kirchenbauer et al., 2023) developed a watermarking framework for proprietary LLMs to embed imperceptible, algorithmically detectable signals into generated text, enabling misuse detection without compromising output quality. Their method selectively promotes ”green” tokens during text generation and uses an open-source detection algorithm with statistical tests to identify watermarked content, even without access to the model's API or parameters.

Statistical and Entropy-Based Methods: (Gehrmann et al., 2019) made a tool called GLTR that looks at how predictable words are in a text. AI tends to pick ”safe” words that are very likely, making its writing less surprising than human text. This is a smart idea and works for some AI models, but advanced ones that mix things up can fool it, just like they fool word-based methods.

To see how our approach stacks up, here's a comparison with two of these methods.

Table 1. Comparison of Detection Methods

Method	Strengths	Weaknesses
Magnitude Function	- Checks the shape of word points, a new idea. - Looks at multiple scales for small and big differences. - Can mix with other methods for better results.	- Needs a lot of computer power for math. - Hard to explain what the numbers mean. - Relies on a good word-to-point model.
Linguistic Feature-Based	- Quick and easy to use. - Clear why it flags a text. - Catches simple AI errors well.	- Struggles with human-like AI. - Misses deeper text structure.
Supervised Classification	- Very accurate when trained well. - Spots complex text patterns. - Can learn new AI tricks with retraining.	- Needs tons of data and computer power. - May fail on new AI models. - Ignores word shapes.

3. Project plan

The main steps of our further work are listed below:

1. **Data Collection**: Collect the same number of real texts (like articles or essays by people) and AI-generated texts.
2. **Embedding Texts**: Use a BERT-like model to turn tokens to vectors. This makes a “point cloud” for each text, representing them in some text vector space.
3. **Magnitude Function**: Make a matrix Z_{tA} where each entry is $e^{-t \cdot d(x_i, x_j)}$ (distance between points, scaled by t). Solve $Z_{tA} w = 1$ to get a vector w, and the magnitude $|tA|$ is the sum of w_i. Do this for different t values to see the shape at different zoom levels.
4. **Feature Extraction**: Pick numbers from the magnitude function, like $|tA|$ at $t = 1, 5, 10$, to use as features for the classifier.
5. **Classification**: Train a simple model to guess if a text is human or AI based on these features.

3.1. Promises

Here we select the major points why we believe that our approach will work:

- **Geometry-Based**: This method looks at how words are spread out in a math space, not just what words are used. Real texts might have words that “cluster” together, like ideas that connect, while AI texts could be more random.
- **Multi-Scale Analysis**: By checking the shape at different t values, we can see both tiny details and the big picture. Maybe AI texts look weird when you zoom in or out.
- **Can Complement Other Methods**: We could combine my shape features with word counts or other tricks to make a stronger classifier.

3.2. Challenges

- **Slow Computation**: The math for the magnitude function, especially flipping big matrices, takes a long time. We will try to figure out how we can speed this thing.
- **Hard to Explain**: It's hard to say why a text is AI just by looking at shape numbers.
- **Depends on Embeddings**: If the model turning words into points isn't great, our plan won't work well. We need to pick a good one and experiment with different embedding techniques.

3.3. Goals

Here's what we are planning next:

- Build a dataset with real and AI-generated texts.
- Obtain text embeddings using the BERT-like model.
- Implement the magnitude function calculation.
- Make a simple classifier to guess real vs. generated texts.
- Compare our method with other approaches.
- Prepare Final Report and store the code of all our experiments into the Github repo.

4. Theoretical Background

Consider the finite metric space A. Write Z_A for the $A \times A$ matrix as:

$$Z_A(a,b) = e^{-d(a,b)},$$

where $a, b \in A$, and d is some distance function.

If Z_A is invertible, the magnitude of A is:

$$|A| = \sum_{a,b \in A} Z_A^{-1}(a,b) \in \mathbb{R}$$

Magnitude may be understood as the *effective number of points*.

Magnitude assigns to each metric space not just a *number*, but a *function*.

For $t > 0$, write tA for A scaled by a factor of t.

The magnitude function of a metric space A is the partially defined function:

$$(0, \infty) \to \mathbb{R}$$

$$t \mapsto |tA|$$

A magnitude function has only finitely many singularities. It is increasing for $t \gg 0$, and $\lim_{t\to\infty} |tA| = \text{cardinality}(A)$.

The magnitude function also contains information about the geometrical features of the space and encodes its dimension. Deeper math could be found in the paper (Leinster & Meckes, 2017).

5. Results and discussion

5.1. Data

For our task we choose a dataset with human-written and AI-generated texts from Kaggle. There, all detailed information about dataset is presented.

To transform tokens to embeddings BERT model was used. For the first experiments, a subset of 1000 texts (500 for human-written; 500 for AI-generated) was taken and transformed to matrices with embedding size equal to 768 and number of tokens per text equals 128.

5.2. Magnitude function

Magnitude function calculation was implemented, Manhattan metric was taken as a distance.

Magnitude function was calculated for text-embedding matrices for the 50 values of t uniformly distributed from 0.005 to 0.04. As shown in Figure 1, there is some difference between the magnitudes of human and AI texts, which makes the magnitude a promising informative feature.

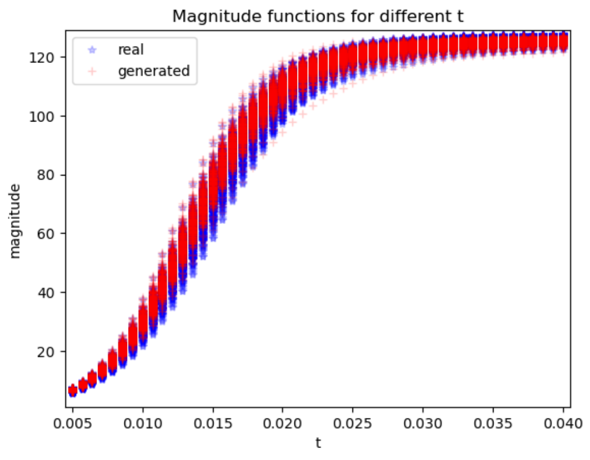

Figure 1. Magnitude function

However, for the single point of t, the difference is not significant to clearly separate two classes. As presented in Figure 2, the magnitudes of real and generated texts have shifted distributions, but they are close to each other.

Interestingly, that for the higher values of t, the distributions "swap", so the real texts become have higher magnitudes (Figure 3).

All this means, that the magnitude function values for different properly selected t may be valuable feature vector for real and generated texts distinguishing.

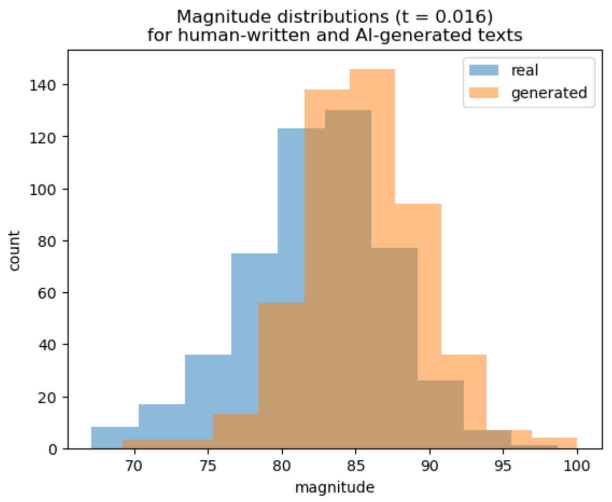

Figure 2. Magnitude distributions for $t = 0.016$

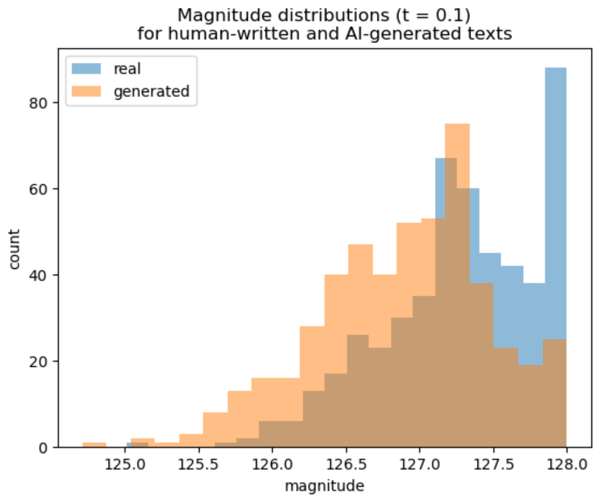

Figure 3. Magnitude distributions for $t = 0.1$

References

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

Hu, X., Chen, P.-Y., and Ho, T.-Y. Radar: robust ai-text detection via adversarial learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Ippolito, D., Duckworth, D., Callison-Burch, C., and Eck, D. Automatic detection of generated text is easiest when humans are fooled. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1808–1822, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.164. URL https://aclanthology.org/2020.acl-main.164/.

Juzek, T. S. and Ward, Z. B. Why does ChatGPT "delve" so much? exploring the sources of lexical overrepresentation in large language models. In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S. (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 6397–6411, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.426/.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Kushnareva, L., Gaintseva, T., Magai, G., Barannikov, S., Abulkhanov, D., Kuznetsov, K., Tulchinskii, E., Piontkovskaya, I., and Nikolenko, S. Ai-generated text boundary detection with roft, 2024. URL https://arxiv.org/abs/2311.08349.

Leinster, T. and Meckes, M. W. *The magnitude of a metric space: from category theory to geometric measure theory*, pp. 156–193. De Gruyter Open, December 2017. ISBN 9783110550832. doi: 10.1515/9783110550832-005. URL http://dx.doi.org/10.1515/9783110550832-005.