GPT-generated Text Detection: Benchmark Dataset and Tensor-based Detection Method

Zubair Qazi
zqazi004@ucr.edu
University of California, Riverside
Riverside, CA, USA

William Shiao
wshia002@ucr.edu
University of California, Riverside
Riverside, CA, USA

Evangelos E. Papalexakis
epapalex@cs.ucr.edu
University of California, Riverside
Riverside, CA, USA

ABSTRACT

As natural language models like ChatGPT become increasingly prevalent in applications and services, the need for robust and accurate methods to detect their output is of paramount importance. In this paper, we present **G**PT **R**eddit **D**ataset (GRiD), a novel Generative Pretrained Transformer (GPT)-generated text detection dataset designed to assess the performance of detection models in identifying generated responses from ChatGPT. The dataset consists of a diverse collection of context-prompt pairs based on Reddit, with human-generated and ChatGPT-generated responses. We provide an analysis of the dataset's characteristics, including linguistic diversity, context complexity, and response quality. To showcase the dataset's utility, we benchmark several detection methods on it, demonstrating their efficacy in distinguishing between human and ChatGPT-generated responses. This dataset serves as a resource for evaluating and advancing detection techniques in the context of ChatGPT and contributes to the ongoing efforts to ensure responsible and trustworthy AI-driven communication on the internet. Finally, we propose GPTEN, a novel tensor-based GPT text detection method that is semi-supervised in nature since it only has access to human-generated text and performs on par with fully-supervised baselines.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Data mining**; **Content ranking**.

KEYWORDS

Benchmark Dataset, GPT-text Detection, Tensor Decomposition, Out-of-distribution detection, Semi-supervised

1 INTRODUCTION

Detection of Generative Pretrained Transformer (GPT)-generated content has gained significant relevance with the proliferation of large language models over the internet. These models, including

GPT-3, produce human-like text that can be seamlessly integrated into various applications and platforms [1]. However, the potential misuse of such generated content for misinformation, spam, or other malicious purposes has raised the importance of detecting GPT-generated text [2]. In diverse contexts like social media, customer service, and content generation, distinguishing between human-authored and AI-generated text has become crucial to maintaining trust, security, and the integrity of online discourse. Detecting GPT-generated text helps to mitigate the risk of spreading disinformation, ensures ethical AI use, and enhances content quality and reliability in applications harnessing AI language models.

There are quite a few approaches that exist for GPT detection [2]. We can sort a majority of the approaches into a few categories: traditional supervised machine learning, deep learning methods, transfer learning methods, and unsupervised methods.

Traditional supervised machine learning methods have been extensively employed for GPT detection [6]. These approaches leverage labeled datasets, where human-generated and machine-generated text samples are used to train classifiers. One of the key advantages of this approach is its interpretability, as it allows for the examination of features used by classifiers to make predictions. However, traditional supervised methods often require substantial manual annotation efforts to create labeled datasets, which can be time-consuming and resource-intensive. Additionally, they require large amounts of training data, with the risk of overfitting, and may struggle to adapt to evolving GPT models and the diverse ways in which they are employed, making them less effective in dynamic environments such as the modern web.

Deep learning methods, on the other hand, are prominent for their ability to automatically learn complex patterns from data [6]. These methods, such as neural networks, can effectively capture the nuanced characteristics of GPT-generated text. Deep learning models excel in handling unstructured data, but they tend to be data-hungry and may demand large training datasets for optimal performance. They are widely used due to their robustness and adaptability, especially when substantial labeled data is available.

Transfer learning methods have emerged as a highly practical solution for GPT detection. By leveraging pre-trained models and fine-tuning them on specific tasks, transfer learning allows for efficient use of available resources while inheriting the knowledge and capabilities of the pre-trained models, which can be particularly advantageous in scenarios with limited training data [7]. However, transfer learning methods may not always generalize well to diverse GPT variants and applications, which can restrict their usefulness.

Unsupervised methods represent a different paradigm, where GPT detection is achieved without the need for labeled data. These methods rely on various statistical and linguistic cues to identify

machine-generated content [4]. Unsupervised approaches are advantageous for their independence from labeled datasets but can be less accurate and robust compared to supervised or deep learning methods. They are less commonly used in practice due to their limitations, especially in the face of evolving GPT models and sophisticated adversarial techniques.

Traditional supervised machine learning and deep learning methods are commonly favored for their accuracy and adaptability, while transfer learning methods offer a pragmatic balance between data efficiency and effectiveness. Unsupervised methods, although less commonly used, offer a label-free alternative but may lag in terms of accuracy and robustness, especially in complex and evolving GPT environments. Our contributions in this paper are:

- **Dataset**: we present **G**PT **R**eddit **D**ataset (GRiD), a dataset designed and built for GPT detection. We make our dataset publicly available.
- **Novel Method**: we propose GPTEN: a novel *semi-supervised* tensor-based method with comparable results to existing *fully supervised* approaches for GPT detection
- **Experimental Evaluation**: we extensively evaluate how state-of-the-art existing approaches behave on our dataset.

Our dataset and implementation are publicly available at https://github.com/madlab-ucr/GriD.

2 GPT REDDIT DATASET DESCRIPTION

The **G**PT **R**eddit **D**ataset (GRiD) is a comprehensive collection of text data obtained from two distinct sources: Reddit and the OpenAI API. It is structured to encompass a total of 6513 samples, further categorized into two primary groups: 1368 samples represent text content generated by the GPT-3.5-turbo model, whereas 5145 samples denote text authored by human contributors. Each individual sample within the dataset is labeled to indicate its source of generation, differentiating between GPT-generated and human-generated text. In order to minimize the potential mixture of GPT-generated data with human-generated data, all data from human contributors is dated from October 31 2022 or earlier, which is the official release date of the ChatGPT web application.

The dataset is stored in a structured CSV (Comma-Separated Values) format. Each line in the CSV file consists of a data sample and its corresponding label. The GPT-generated data contained within this dataset is a result of interactions with the GPT-3.5-turbo model provided by the OpenAI API. To solicit responses from the model, a specific prompt was employed: "You are a frequent user of the subreddits <subreddit_names>. Answer anything relevant." The model's responses to these prompts are integral to the GPT-generated portion of the dataset. In order to promote further research in this direction, we make this dataset publicly accessible to the research community on GitHub[*].

2.1 Dataset Collection

The human-generated data is gathered from Reddit using the PRAW Python library and GPT-generated content is gathered from the OpenAI API. We sourced the Reddit data from three different subreddits: AskHistorians, AskScience, and ExplainLikeImFive. In order

to consider a post from each subreddit, they had to satisfy all of the following criteria:

(1) The post must be dated before November 2022.
(2) The post must have at least a score (upvotes) of 1000.
(3) The post does not contain adult content.
(4) The post is in English.
(5) The post title is formatted as a question.
(6) The post itself cannot be deleted.

We can justify each criteria separately. The following is a justification for each criteria:

(1) ChatGPT was officially released to the public in November 2022. In order to ensure minimal representation of GPT in the human-generated data, we only considered posts before that date.
(2) We only consider the top posts from each subreddit. Posts with at least 1000 upvotes were generally appropriate for the dataset.
(3) Since the dataset is for academic research purposes, we avoid any adult posts.
(4) The dataset is only intended to be comprised of English based content at this time.
(5) Each post title is fed into GPT as a prompt, so in order to reduce noise and increase consistency between the human-generated and GPT generated content.
(6) Some posts on Reddit are deleted by moderators of the subreddit, but their metadata still exists on the subreddit. For fairness, we filter out these posts from the dataset.

For the current dataset, we gather up to the top 500 posts from each subreddit which satisfy the above criteria. For each post, we gather up to 5 of the top comments based on score (upvotes), and then feed the post title into GPT and store the corresponding response. Each comment only needs to satisfy a simpler criteria to be considered: (1) The comment is in English; (2) The comment is not deleted.

2.2 Dataset Processing

Both the human-generated and GPT-generated data need to be processed to an acceptable state. Since the origin of the data differs, the processing techniques applied also differ.

2.2.1 Reddit Data Processing. In order to reduce unwanted bias towards human-generated content, we remove any features that exist in the Reddit data that does not exist in GPT data. Specifically, we remove links and any other non-text multi-modal information from the Reddit data. Links can exist in both markdown formatting (text)[link] as well as general URL formatting, so we must handle both uniquely. We extract the text from links in markdown formatting and remove the link and special characters. For generic URLs, we simply remove them since GPT3.5 does not generate links.

Special characters that are not representative of typical punctuation are also removed. An example is bold characters in markdown, which are encapsulated by * characters. We also remove newline characters from human-generated content, as they are not present in GPT-generated text. Other bias exists in human-generated content, such as personal anecdotes and nuanced contextual understanding, but said bias can be harnessed to discern human-generated content from GPT-generated content since GPT can attempt to replicate said biases through more advanced prompting techniques.

---

[*]https://github.com/madlab-ucr/GriD

We filter out Reddit data with profanity or other inappropriate content using the better-profanity library, since GPT does not typically use any profanity or generate inappropriate content unless specifically prompted to do so. The better-profanity library can filter most inappropriate content automatically, but any left over data has been manually filtered. We also filter out any human-generated content under 100 characters in length, as these comments are short and typically lacking in substance.

2.2.2 GPT Data Processing. The GPT data requires some minimal processing before it can be utilized. The output of GPT is limited to 100 tokens to match the typical length of Reddit comments so as to avoid any bias in length. Since the token limit can result in incomplete sentences, any incomplete sentences in the GPT responses are removed. In order to ensure the preservation of the underlying patterns, while also avoiding the introduction of human bias into the GPT data, so we don't perform any further processing.

3 PROPOSED METHOD
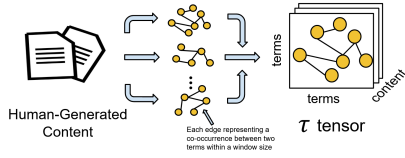
Figure 1: Tensor construction method. We generate a unique graph for each document, where each edge represents the co-occurrence between two unique terms within a set window size, and then stack the graphs to create the tensor.

We propose GPTEN, a novel method for anomaly detection that leverages tensor decomposition to identify underlying patterns in the data. Specifically, we are leveraging the fact that comparing the reconstruction of a tensor built from its decomposed components to the original may show that the original tensor contains anomalies, if a significant difference between the two exists. This type of tensor representation has been successfully applied to two very different language tasks: fake news detection [3] and humor recognition [9]. So its reasonable that this method can also be applied directly to GPT detection, and we have applied it to the task using this dataset. The proposed method is structured as a pipeline and consists of a few major components, which this section will describe in detail. This approach is considered semi-supervised based on the first step.

The first step of the pipeline is to construct a three-dimensional tensor of the corresponding input data. In most cases, the tensor should represent the in-distribution data. If this is the case, then any positive data points will be excluded from the tensor, hence the semi-supervised approach.

We build the tensor as follows. For each document in the data, we build a co-occurrence matrix for each term and its neighbors within a window size (typically 5 to 10). Each co-occurrence matrix is an $M \times M$ matrix, where M represents the number of unique terms in the entire collection of documents. So, given a collection C with N documents, we will construct an $N \times M \times M$ tensor where each slice of the tensor is an $M \times M$ co-occurrence of document n_i; there are N such co-occurrence matrices (Fig. 1).

An important distinction is that we are only using human-generated content to build the tensor. Since we require labeled non-GPT data in order to build the tensor, we consider this method semi-supervised. Simultaneously, we only use terms from the human-generated content to build the the co-occurrence matrices in order to avoid any potential contamination from the test set.

The second step of the pipeline is to decompose the tensor. For our proposed method, we employ the Canonical Polyadic Decomposition (CPD)[3] to decompose the tensor into factor matrices.

The third step of the pipeline is to project and reconstruct each slice of the tensor using the decomposed factor matrices. Specifically, we want to construct a vector r of length N which contains the reconstruction error of each slice in the tensor (Fig. 2). Since the input tensor is three dimensional, CPD decomposition calculates three corresponding factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of dimensions $M \times r$, $M \times r$, $N \times r$ respectively, where r represents the rank of the decomposition. We will then project each slice of the tensor, denoted as $\mathbf{S}_i$, through the factor matrices $\mathbf{A}$ and $\mathbf{B}$, to get the projection $\mathbf{P}_i$ as $\mathbf{P}_i = (\mathbf{A}^\dagger \cdot \mathbf{S}_i) \cdot \mathbf{B}$. We can then calculate the reconstruction $\mathbf{S}'_i$ for each slice $\mathbf{S}_i$ as $\mathbf{S}'_i = \mathbf{A} \cdot (\mathbf{P}_i) \cdot \mathbf{B}^\dagger$ The reconstruction $\mathbf{S}'_i$ will be of dimension $M \times M$, which is the same as $\mathbf{S}_i$. We can then take the Frobenius norm to calculate the reconstruction error e_i of each slice $\mathbf{S}_i$: $e_i = ||\mathbf{S}'_i - \mathbf{S}_i||_F$.

The reconstruction errors follow a distribution which can be modelled by both supervised and unsupervised models, however, given that our method only has access to negative labels (i.e., human-generated text), we employ unsupervised anomaly detection to model the reconstruction error distribution and identify positive (i.e., GPT-generated text) as O.O.D points.

## 4 EXPERIMENTAL EVALUATION

4.1 Baseline methods

We use three distinct models—Random Forest, Support Vector Machine (SVM), and BERT on the dataset to assess their efficacy in GPT-generated text detection. The selection of these models is intended to explore a spectrum of approaches [2]. For both SVM and Random Forest, we applied a simple TF-IDF vectorizer to transform the text data into numerical data[5], which is then fed into both models. All results are obtained via 10-fold cross-validation.

BERT-based models stand as state-of-the-art models in natural language processing and represents the cutting edge of deep learning-based text understanding [2]. Its ability to contextualize words within a sentence and grasp intricate semantic relationships makes it an ideal candidate for the task of distinguishing between human-generated and GPT-generated text. For this reason, we have chosen a pre-trained baseline BERT model for experimentation.

4.2 Results

Our results indicate that BERT outperformed both SVM and Random Forest in terms of ROC-AUC values (Table 1). This observation aligns with the expectations set by the choice of models, with BERT leveraging its advanced contextual understanding to discern the nuances of GPT-generated text.

SVM, representing a traditional ML approach, demonstrated respectable performance, while Random Forest, as an ensemble method, showcased its ability to capture certain patterns but fell
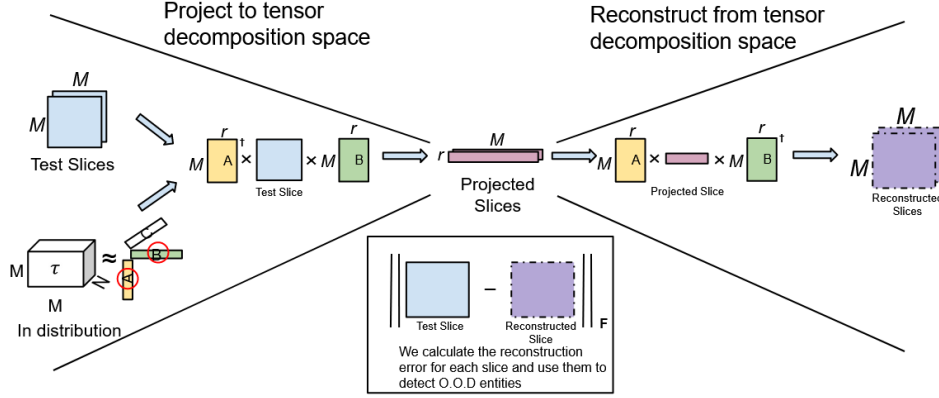
Figure 2: Each slice from the test set is projected via the decomposition factors A and B, and then reconstructed. The reconstruction error for each slice is the Frobenius norm between the slice and its associated reconstruction.

short of the performance achieved by BERT. Despite being outperformed, SVM and Random Forest remain viable options for GPT-generated text detection due to their interpretable structure and efficiency in handling high-dimensional feature spaces. These traditional machine learning approaches offer transparency in model predictions and can serve as practical alternatives, particularly when computational resources are constrained or interpretability is an important consideration.

Table 1: Performance metrics on the GPT Detection Dataset. Note that GpTen, while semi-supervised, performs comparably to fully-supervised baselines.

	F1	AUC
BERT	0.934	0.984
SVM	0.813	0.845
Random Forest	0.787	0.825
GpTen (proposed)	0.667	0.708

For GpTen we compare the results of applying an unsupervised model for anomaly detection on the calculated reconstruction errors (Fig. 2). We apply models from the PyOD library, a comprehensive Python library for outlier detection [8]. PyOD supports both supervised and unsupervised models, and in both cases provides a simple abstraction to gather predictions and metrics from the model. For unsupervised models, the anomaly scores given to each data point by the model are converted to a prediction by applying a threshold. Thus, we can compare metrics such as F1-score and ROC AUC score against other supervised models. The best performing unsupervised model from our experiments was the KDE model, which assesses the likelihood of each data point by estimating its probability density function based on a non-parametric approach, identifying anomalies as instances with lower likelihoods.

5 CONCLUSIONS

In this paper we introduced **GPT Reddit D**ataset (GRiD), a novel benchmark dataset for the detection of GPT-generated text and

demonstrated the performance of fully-supervised methods on it. Furthermore, we proposed GpTen, a tensor-based method which only has access to human-generated data and is able to perform on par with fully-supervised baselines.

6 ACKNOWLEDGEMENT

REFERENCES

[1] Tom B. Brown et al. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165
[2] Evan Crothers, Nathalie Japkowicz, and Herna Viktor. 2023. Machine Generated Text: A Comprehensive Survey of Threat Models and Detection Methods. arXiv:2210.07321 [cs.CL]
[3] Gisel Bastidas Guacho, Sara Abdali, Neil Shah, and Evangelos E. Papalexakis. 2018. Semi-supervised Content-based Detection of Misinformation via Tensor Embeddings. arXiv:1804.09088 [cs.LG]
[4] Zied Haj-Yahia, Adrien Sieg, and Léa A Deleris. 2019. Towards unsupervised text classification leveraging experts and word embeddings. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*. 371–379.
[5] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21.
[6] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2020. A Survey on Text Classification: From Shallow to Deep Learning. *CoRR* abs/2008.00364 (2020). arXiv:2008.00364
[7] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *NAACL: Tutorials*. 15–18.
[8] Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research* 20, 96 (2019), 1–7. http://jmlr.org/papers/v20/19-011.html
[9] Zhenjie Zhao, Andrew Cattle, Evangelos Papalexakis, and Xiaojuan Ma. 2019. Embedding lexical features via tensor decomposition for small sample humor recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.