Is this AI-generated? A comparative analysis of zero-shot and few-shot text classification with small GPT-based language models

Abstract

The rapid proliferation of AI-generated content has created an urgent need for low-resource detection methods that serve as more cost-effective alternatives to supervised methods like fine-tuning. This study evaluates a small GPT-based language model, *Gemini-1.5-fast*, for distinguishing human- versus AI-generated text under zero-shot and few-shot learning paradigms. Experiments on a balanced dataset, comprising topic-aligned human and AI-generated samples, indicate that few-shot learning substantially outperforms zero-shot across all evaluated metrics. These findings establish a baseline for AI text detection relying only on prompt optimization techniques and smaller GPT-based models, offering insights into cheaper detection solutions.

1. Introduction: background, motivation, scope and contributions of the study

Generative Pretrained Transformers (GPT) have become highly effective tools in natural language processing, driven by their multi-layered decoder architectures, attention mechanisms, large-scale training, and extensive context windows (*Vaswani et al., 2017; Radford & Narasimhan, 2018*). This advancement has also unleashed a flood of AI-generated content across the web, underscoring the need for effective methods to distinguish machine-generated text from human-written material. Traditional text classification approaches often rely on manually annotated datasets and models fine-tuning, which can be resource-heavy. As a result, there is growing interest in more cost-effective techniques that leverage prompt optimization techniques for GPT-based models (Su & Wu, 2024). Thus, this study explores detection methods for binary text classification, distinguishing AI-generated text from human-written material, by utilizing a smaller GPT-based model (*gemini-1.5-fast* with 8B parameters) and leveraging zero-shot and few-shot learning paradigms, where zero-shot involves no labeled examples and few-shot incorporates a small set of labeled examples in the prompt (*Brown et al., 2020; Anglin & Ventura, 2024; Edwards & Camacho-Collados, 2024)*. We hypothesize that while few-shot learning will outperform zero-shot learning, performance may still be inconsistent for reliable classification when relying solely on prompt optimization. Finally, our objective is to establish baseline performance metrics that provide a foundation for further research on cost-effective detection solutions. Key contributions are:

- Creation of human- and AI-generated text datasets, guided by GPT-based topic labeling to ensure aligned topics, style, and tone.
- Evaluation of dataset similarity by computing the overlap of the most frequent lemmas.
- Systematic evaluation of the performance of a smaller and cheaper GPT-based model (gemini-1.5-fast) in binary text classification under zero-shot and few-shot scenarios.
- Introduction of baseline performance metrics to guide the development of cost-effective detection solutions.

2. Methodology: datasets, experimental setup, evaluation metrics

2.1 Datasets

The research begins with the creation of three main datasets stored in CSV format: one containing human-generated blog posts (226), another with corresponding AI-generated texts (99), and a third comprising a balanced mix of the first two, used in the classification tasks (198). First, blog posts by Paul Graham were collected through web scraping, resulting in a raw dataset containing columns for the resource link and the blog post text. Then, the human contents were classified and labeled with respect to their main topic using a small AI wrapper powered by gemini-1.5-fast. Artificial blog posts were generated using gemini-1.5-fast with few-shot learning, configured with a temperature of 1 to balance creativity and adherence to examples. Each generation round used three randomly selected human examples within a specific topic category to guide the model, resulting in 99 texts closely aligned with the original dataset in topics, style, and tone. Thus, the AI-generated dataset mirrored the human dataset's topics' proportion, ensuring a balanced and faithful representation of the original data. A bar plot visualized the consistent topic distribution across both datasets.

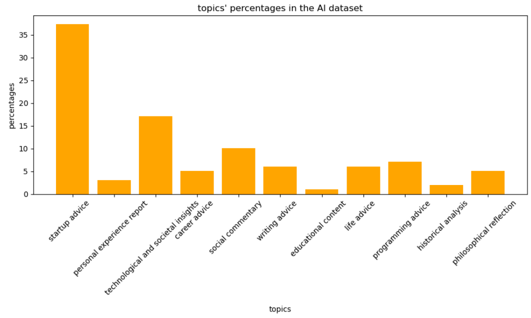

Figure 1: Distribution of topics in the AI-generated dataset, with their relative frequencies

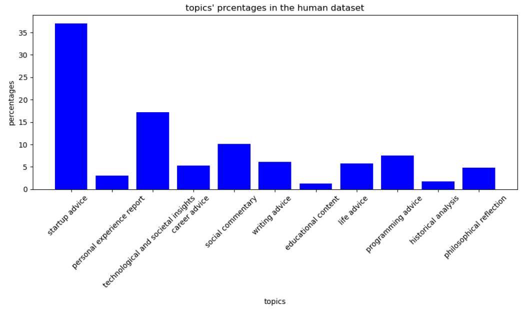

Figure 2: Distribution of topics in the human-generated dataset, with their relative frequencies.

The lexicon analysis provided a qualitative validation of dataset similarity by identifying the 100 most frequent lemmas after normalization, tokenization, and lemmatization, revealing a 39% overlap between the most frequent lemmas of the datasets. This overlap, illustrated through bar plots comparing lemmas' distributions, highlighted shared key terms such as *startup*, *founder*, *company*, *writing*, *language*, *work*, *idea*, *user*, and *people*. These keywords align with Paul Graham's identity as a startup founder, investor, computer scientist, philosophy major, and writer. Additionally, the frequent use of the personal pronoun *I* in both datasets reflects Graham's autobiographical and dairy-like narrative style. These findings suggest that the AI-generated texts retained, at least, the primary focus and thematic essence of Graham's work.

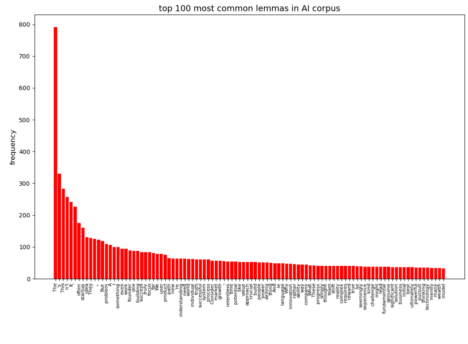

Figure 3: Top 100 most frequent lemmas in the AI corpus, ranked by absolute frequency.

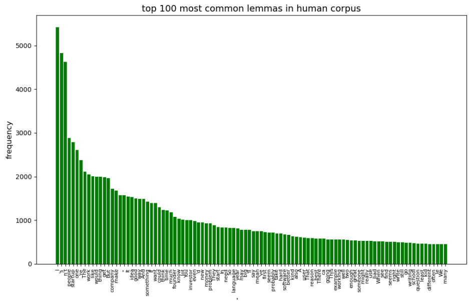

Figure 4: Top 100 most frequent lemmas in the human corpus, ranked by absolute frequency.

The two datasets were also enriched with the column **"**AI_or_human**"**, the annotated ground truth labeled as *AI* or *Human*, essential for evaluating the classification performance. The datasets were then unified maintaining topics' proportions, with 99 entries each for human and AI-generated texts; classification results for both prompting approaches were added under separate columns after performing the classification tasks. Furthermore, minor datasets in JSON format were created to store the lists of tokens and lemmas from the linguistic pre-processing module, preserving list objects for lexicon analysis; also classification results and performance metrics are stored in JSON files.

2.2 Experimental Setup

As we said, the study employed Google's generative model *gemini-1.5-fast,* selected for smaller relative size (8B parameters), its extensive context window (1M tokens), and low API costs, making it well-suited for low-resource applications and accommodating long labeled examples in the prompt. Each blog post was presented to the model individually by iterating through the datasets' column containing the posts. Initially, zero-shot learning was implemented, leveraging the model's pre-trained capabilities without any labeled examples. Subsequently, few-shot learning was applied, incorporating three randomly retrieved labeled examples of human- and AI-generated texts into the prompt. To ensure consistent classification, the model was set to a temperature of 0, and the desired output format was explicitly and stated within the prompt. Results were then recorded in the mixed dataset under corresponding columns (*classification_[prompting approach])*, with outputs labeled as either "AI" or "human." Testing on a mixed dataset served the need to simulate real-world scenarios.

2.3 Performance metrics

To evaluate the model's performance, accuracy, precision, recall, and F1-score were calculated for both zero-shot and few-shot learning approaches, providing a comprehensive assessment of its detection ability. To compute these metrics, we had to identify true positives (TP), representing AI-generated content correctly classified as "AI"; true negatives (TN), representing human-generated content correctly classified as "human"; false positives (FP), representing human content incorrectly classified as "AI"; and false negatives (FN), representing AI content incorrectly classified as "human". We computed these outcomes by iterating through the mixed dataset and comparing the model's predictions (classification results) with the ground truth labels to identify TP, TN, FP, and FN. The counts for these variables under both learning approaches were visualized using bar charts. Additionally, performance metrics were presented through bar plots, offering a clear visualization of the comparative performances.

2.4 Recap of packages used

- **pandas:** for loading CSV and JSON files into dataframes and performing structured data handling.
- **csv, json:** to save and manage data in CSV and JSON formats.
- **random:** for sampling content examples in prompt design, ensuring variability and reducing biases during synthetic dataset generation and few-shot classification tasks.
- **time:** to add delays between API calls of the model, ensuring compliance with rate limits.
- **google.generativeai:** to configure and interact with Google's generative model gemini-1.5-fast
- **matplotlib:** to create graphs.
- **urllib, bs4.BeautifulSoup:** for web scraping blog posts.
- **nltk** (**nltk.corpus.stopwords**, **nltk.tokenize.word_tokenize**, **nltk.stem.WordNetLemmatizer**) and **string:** for linguistic processing, such as normalization, tokenization, stop word and punctuation removal, and lemmatization.
- **collections.Counter:** for counting and ranking lemmas during lexicon analysis.

3. Results

The results reveal a stark contrast in performance, with few-shot learning demonstrating notable improvements across all metrics compared to zero-shot learning. Zero-shot learning struggled significantly, classifying all texts as human and failing to identify any AI-generated content. While few-shot learning performed better, it still falls short of being a reliable detection method, aligning with our initial hypothesis.

3.1 Classification results

- Zero-shot learning correctly classified all 99 human-generated texts as human (true negatives) but failed to detect any AI-generated texts, resulting in 99 false negatives.

- Few-shot learning correctly classified all human-generated texts (true negatives = 99) and detected 55 AI-generated texts (true positives = 55), with 44 false negatives and no false positives.

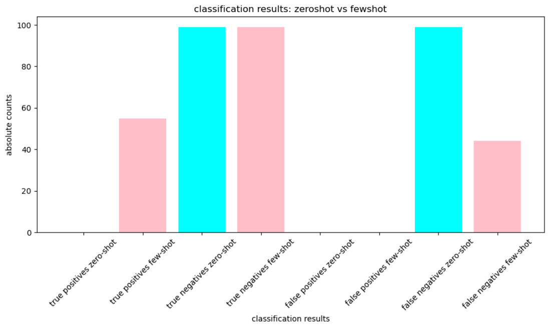

Figure 5: Classification results (true positives, true negatives, false positives, false negatives) for zero-shot and few-shot approaches.

3.2 Performance metrics:

Few-shot learning significantly outperformed zero-shot learning:

- Accuracy: few-shot learning reached 77.8%, compared to 50% for zero-shot learning.
- Precision: few-shot learning achieved a perfect score of 1.0, while zero-shot learning scored 0.
- Recall: few-shot learning attained 55.6%, whereas zero-shot learning failed to detect any AI-generated texts (0 recall).
- F1-score: few-shot learning recorded 71.4%, with zero-shot learning again scoring 0.

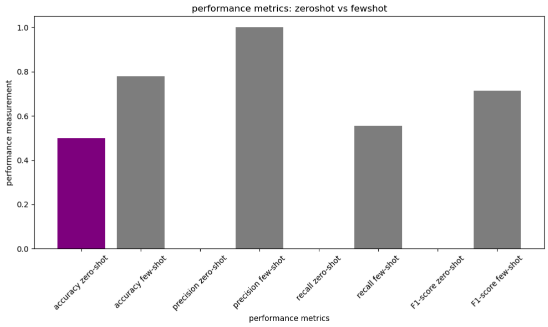

Figure 6: Performance metrics (accuracy, precision, recall, F1-score) comparison between zero-shot and few-shot learning.

3.3 Discussion of results

The preliminary findings confirm our hypothesis that few-shot learning outperforms zero-shot learning. In fact, the zero-shot approach was completely unable to detect AI-generated content, classifying all texts as human (precision = 0, recall = 0). Although its accuracy stands at 0.5, this should not be mistaken for balanced performance; rather, it reflects the model's tendency to default all classifications to "human." By contrast, few-shot learning was able to identify just over half of the AI texts and maintained perfect precision (1.0), ensuring that anything labeled as AI was, in fact, AI—an essential feature when mislabeling human content could have significant consequences. However, the moderate recall (55.6%) illustrates a limited sensitivity to AI-generated texts, signaling the need for further refinement to bolster recall without sacrificing precision. Overall, even a single round of few-shot learning proves insufficiently robust for consistent AI text detection, highlighting a need for more advanced methods.

4. Conclusion and future directions

Few-shot learning has demonstrated a clear advantage over zero-shot methods in detecting AI-generated text, though its moderate recall suggests that a limited number of labeled examples is insufficient to fully address the detection challenge. To refine and generalize these findings, several strategies can be pursued. Repeating the classification task across multiple trials would yield more reliable performance estimates. Testing the model on larger, more diverse datasets, varied authors, multimodal content, and different numbers of labeled examples per prompt, would provide a more comprehensive evaluation of its adaptability and generalization capabilities. Additionally, conducting a deeper analysis of the linguistic features in classified and misclassified texts could uncover the specific attributes the model uses (or could use) to inform its decisions.

References

1. Vaswani A, et al. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017.
2. Radford, Alec and Karthik Narasimhan. Improving Language Understanding by Generative Pre-Training. 2018.
3. B. Brown, et al. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020.
4. Edwards, A. and José Camacho-Collados. "Language Models for Text Classification: Is In-Context Learning Enough?". 2024.
5. Anglin, K. L., & Ventura, C. Automatic Text Classification With Large Language Models: A Review of openai for Zero- and Few-Shot Classification. *Journal of Educational and Behavioral Statistics, 2024*.
6. *Zero-shot learning vs few-shot learning vs fine-tuning*. Medium.
7. *Pandas documentation.*
8. *CSV documentation.*
9. *Random module documentation.*
10. *Time documentation.*
11. *Google.generativeai module documentation*.
12. *Matplotlib documentation.*
13. *Bs4 documentation*.
14. *NLTK documentation.*
15. *Collections documentation*.
16. Jupyter notebooks of the lectures.

Project's repository

All the data, code and figures can be accessed visiting the following GitHub repository.