Midterm Project

Guanlin He, Jiamu Bai, Xin Gu

October 7, 2024

1 Introduction

This project builds a classifier that can accurately identify the source LLM used to complete a given text. Given an input text x_i and a corresponding completion x_j, the goal is to identify which language model (LLM) produced x_j. The task is to construct a classifier $f : \mathcal{X} \times \mathcal{X} \to \mathcal{M}$ that maps each input-completion pair (x_i, x_j) to its corresponding model M_j.

We explore the ability of a deep learning-based classifier to distinguish between text completions generated by three GPT-family models, GPT-neo[GBB+20], GPT2 [RWC+19] and GPT2-distilling [SDCW19] model. We also use a more modern model: Qwen2 0.5b [Tea24]. For datasets, we extract sentences from HellaSwag dataset[ZHB+19] and IMDb dataset[MDP+11] and generate text completions from both models with each input sentences, achieving an accuracy of 97 percent.

2 Dataset Curation

2.1 Data Sources

- HellaSwag dataset: HellaSwag dataset consists of sentences and contexts crafted to test a model's ability to perform sentence completion. Each item in the dataset presents a context followed by an incomplete sentence, with possible endings that the model must choose or predict. This structure enables the generation of coherent and contextually appropriate sentence completions.

- IMDb dataset: IMDb dataset[MDP+11] consists of movie reviews written by users. Each review in the dataset provides a coherent piece of text with well-defined contexts, allowing for meaningful completions by the LLMs.

2.2 Dataset preprocessing

- Text Truncation: For IMDB dataset, each review is truncated to its first half, preserving the initial context while removing the latter portion. This truncation allows us to simulate incomplete sentences and paragraphs that

require completion. For HellaSwag dataset, as the dataset has already partitioned into two halves, we directly use first half to generate second half by LLMs.

- LLM Text Completion: The truncated texts are then fed into models, each tasked with filling in the missing half of the review.

2.3 Train/test dataset construction

- Each truncated sentence/paragraph is denoted as x_i. Since we use 4 LLMs to generate different x_j, we label the generated x_j in the following way:
  - Label 0: GPT2-distill
  - Label 1: GPT-neo
  - Label 2: GPT2
  - Label 3: qwen2 0.5b
- Generation setup: when generating x_j, LLMs are only allowed to generate 200 new tokens with a penalty of repetition set to 1.2, top $p = 0.9$ and top $k = 50$.
- For IMDB dataset, each model does text completion for 1000 reviews with train and test split ratio of 8:2. The total training data is 3200 reviews and total test data is 800 reviews, both with evenly distribute of labels.
- For HellaSwag datasets, each model does text completion for 10K reviews, also with train and test split ratio of 8:2. The total training data is 32K reviews and total test data is 8K reviews, both with evenly distribute of labels.

3 Classifier Design

The classifier is designed to distinguish between text completions generated between 4 different LLMs. To achieve this, we employ a BERT-based architecture for feature extraction and classification, leveraging BERT's robust language understanding capabilities to capture the nuances between completions from the 4 models.

3.1 Training Strategy

The classifier is trained using the following training strategy:

- Loss Function: Cross-Entropy Loss to distinguish between 4 classes.
- Batch size: 64. As we used a 40G A100, such batch size reached to limitation.

- Epochs: 3. As we run the training with pretrained weights, 3 epochs are sufficient for good empirical accuracy.
- Data Splitting: The dataset was split into training and test sets with ratio of 8:2.
- Optimizer: Utilized the AdamW optimizer with a learning rate of 5×10^{-5} for fine-tuning the transformer model.
- Mixed Precision Training: Enabled mixed-precision calculations using `torch.no_grad()` during inference to save memory and speed up computations.
- Regularization: Employed techniques like gradient clipping to stabilize training and prevent gradient explosion.

3.2 Model Architecture

Due to the large corpus of IMDb dataset, we use a max token size = 1024. As the regular BERT model only supports 512 max token size, we employ the pretrained longformer-base-4096 [BPC20] to handle larger inputs.

For our main task HellaSwag datasets, we used regular BERT with correspond tokennizer and pretrained weight to do the classification.

4 Experiment and Results

We run the evaluation after 3 epochs of training on HellaSwag datasets and achieve 97 percent of accuracy. If we want to display the result of single example of classification, it would be too many words. Therefore, we would like to present our test result by using t-SNE analysis.

4.1 Why t-SNE

t-SNE is a technique used for reducing high-dimensional data into lower-dimensional representations, usually 2D or 3D. It is specifically designed to preserve local relationships in the data, meaning that points that are close to each other in the high-dimensional space will remain close in the 2D plot. In our case, we feed test data into the BERT model and get the hidden states information from the output. The final layer's hidden state will represent the feature embedding before the activation function. Then we apply the stored all embedding with corresponding label mapping into the 2d space to see how classifier works.

4.2 t-SNE result for 4 LLMs classification on HellaSwag datasets

- **Label 0:** GPT2-distill

- **Label 1:** GPT-neo
- **Label 2:** GPT2
- **Label 3:** Qwen-2 0.5B

The t-SNE result is shown in Figure 1. The distinct clustering observed in the plot indicates that the classifier has effectively learned to separate the outputs generated by each of these four LLMs. Each color represents a different LLM, and the clear boundaries between the clusters highlight the model's ability to distinguish between the classes.

In-Depth Analysis.

- **Well-Defined Clusters:** The embeddings form four distinct clusters with minimal overlap, suggesting that the classifier's learned feature space has successfully separated the data points based on the LLM source. This clear separation indicates that the model has captured unique characteristics or patterns in the text generated by each LLM.
- **Separation Between Classes:** The noticeable distance between clusters implies that the representations for each LLM are significantly different, allowing the classifier to differentiate them with high accuracy. The minimal overlap between clusters reduces the likelihood of misclassification.
- **Consistency in Clustering:** The even distribution of points within each cluster shows that the classifier consistently maps similar outputs to nearby locations in the feature space, reinforcing the model's effectiveness in learning distinguishable representations.
- **Interesting observation** One of the interesting observation is that even after training, the bert classifier model cannot clearly distinguish between gpt2 and gpt2-distall version. Maybe a interesting discussion would be classfy output of LLM model and their smaller weight version.

4.3 More results: IMDb Experiment

Following the initial success in classifying outputs from four different language models (LLMs) using the HellaSwag dataset, we extend our investigation to a more focused experiment using the IMDB dataset with GPT-2 and GPT-neo. This section delves into the rationale behind this experiment and the new strategies we employed, particularly the use of a larger dataset, increased token size, and a more advanced classifier.

Figure 1: HellaSwag dataset 4 LLMs classifier result.

4.4 Motivation for the IMDB Experiment

The decision to transition from the 4-class classification on the HellaSwag dataset to a 2-class classification on the IMDB dataset using GPT-2 and GPT-neo was guided by several key factors:

- **Larger Token Size and Corpus:** The IMDB dataset offers a more extensive and sentiment-rich corpus compared to the HellaSwag dataset. With movie reviews often consisting of detailed expressions of opinion, the text lengths are significantly longer, requiring models capable of handling larger token sizes. In this experiment, we increased the maximum token size to 1024 tokens to accommodate these longer sequences, enabling a more nuanced analysis of the models' text generation capabilities.

- **Advanced Classifier Architecture:** To leverage the larger input size and extract more complex features, we employed the Longformer model with support for sequences up to 1024 tokens. The Longformer's attention mechanism is specifically designed to efficiently handle long texts, making it an ideal choice for capturing the intricate details present in the IMDB dataset. This shift from previous approaches emphasizes the use of more sophisticated architecture to achieve higher accuracy in distinguishing between the two LLMs.

- **Exploring Domain-Specific Characteristics:** While the HellaSwag dataset contained diverse scenarios and context-based tasks, the IMDB dataset is narrowly focused on sentiment analysis. This focused context allows for a deeper exploration of how each LLM, particularly GPT-2 and

GPT-neo, handles sentiment-rich language. It also provides an opportunity to analyze the stylistic and linguistic differences of the LLMs in generating opinionated content.

4.5 Experiment Design and Approach

The IMDB dataset was specifically chosen for this experiment due to its larger corpus size and the natural language variability present in movie reviews. By increasing the maximum token size, we were able to capture more context and detail within each review, leading to richer embeddings that could potentially highlight finer distinctions between the outputs of GPT-2 and GPT-neo.

Key Improvements in Experiment Design:

- **Use of Longformer with 1024 Tokens:** Unlike traditional models that are limited to processing 512 tokens, the Longformer model's ability to handle sequences up to 1024 tokens significantly enhanced the feature extraction process. This allowed the classifier to better understand and differentiate between the generative styles of GPT-2 and GPT-neo over longer and more complex reviews.
- **Handling Larger Texts Efficiently:** The Longformer's sparse attention mechanism reduces the computational complexity typically associated with large text processing, enabling us to efficiently classify long-form reviews in the IMDB dataset. This is a key advantage over previous models, which struggled with the memory constraints of large sequences.
- **Comparison to Prior Approaches:** The focus on a more advanced classifier and a dataset with a larger token size is a direct improvement over the methods used in the HellaSwag experiment. The shift to a higher-capacity model reflects the need to tackle the increased linguistic complexity and sentiment analysis demands of the IMDB dataset.

4.6 t-SNE Result for IMDB Dataset Classification

Figure 2 presents the t-SNE visualization of the test data embeddings for the IMDB dataset classification between GPT-2 and GPT-neo. The use of a larger token size and a more advanced classifier led to a clear separation between the clusters, reflecting the model's capability to distinguish between the two LLMs with high precision.

Observations:

- **Enhanced Clustering due to Larger Token Context:** The clusters generated in the t-SNE visualization are distinctly separated, indicating that the increased token size and the use of the Longformer have enabled the model to leverage more context, resulting in better-defined feature representations for each LLM.

- **Minimal Overlap Suggests Robust Classification:** The distinct separation between GPT-2 and GPT-neo clusters highlights the classifier's effectiveness in identifying nuanced differences between the two models, even when dealing with lengthy and sentiment-laden reviews.
- **Improved Decision Boundaries:** The use of an advanced architecture like the Longformer not only allowed us to process larger texts but also helped create sharper decision boundaries, as evidenced by the clear distinction in the embedding space between the two classes.

4.7 Implications and Future Directions

The results of the IMDB experiment provide strong evidence that increasing the maximum token size and using a more advanced classifier can significantly enhance model performance in distinguishing between different LLMs. This experiment has broader implications for NLP applications that involve long-form text, such as document classification and content generation.

Key Takeaways:

- The ability to process longer sequences up to 1024 tokens with the Longformer allowed us to exploit the full potential of the IMDB dataset, capturing deeper semantic and contextual features that traditional models might miss.
- The success of this approach suggests that using domain-specific datasets like IMDB, coupled with advanced models like the Longformer, can lead to more precise classifications and deeper insights into LLM behavior.

5 Related Work and Future Work Discussion

There has been significant research aimed at distinguishing between machine-generated text and human-written content, as well as differentiating the outputs of various language models. For instance, GLTR [GSR19] employs statistical techniques to detect generated text by analyzing the predictability of individual tokens, providing insights into the subtle variations in text produced by different language models. The work by [Bro20] emphasizes the unique text generation capabilities of GPT-3 compared to its predecessors, indicating that these differences in generated content can be utilized to identify the specific LLM responsible. It is clear that each LLM exhibits distinct inference and sentence generation capabilities, which lead to variations in their output. These differences can be attributed to factors such as the model's architecture, the training data it was exposed to, or its underlying inference logic.

A recent workshop paper [RMPT24] presents an approach closely related to black-box LLM classification, where the authors leverage the varying question-answering abilities of LLMs to distinguish between them. This method aligns

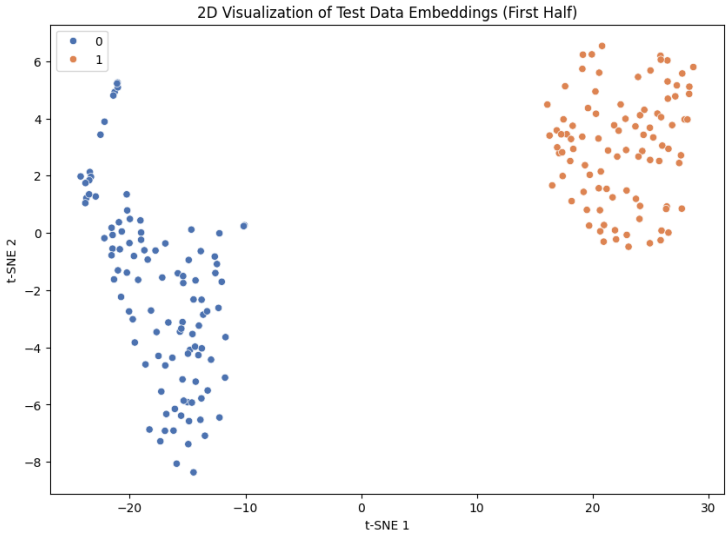

Figure 2: t-SNE visualization of test data embeddings for IMDB dataset classification between GPT-2 (label 0) and GPT-neo (label 1).

with our efforts but highlights the challenges associated with the high computational resources and significant time requirements needed for such tasks. While such approaches offer valuable empirical insights, they are often criticized for their resource-intensive nature and are sometimes viewed as more exploratory than fundamentally innovative.

References

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

[Bro20] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[GBB+20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[GSR19] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*, 2019.

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis, 2011. Dataset available at: https://ai.stanford.edu/~amaas/data/sentiment/.

[RMPT24] Gurvan Richardeau, Erwan Le Merrer, Camilla Penzo, and Gilles Trédan. The 20 questions game to distinguish large language models. *arXiv preprint arXiv:2409.10338*, 2024.

[RWC+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[SDCW19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^{2}Workshop*, 2019.

[Tea24] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[ZHB+19] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.