Detection and Watermarking of LLM-generated text using Contrastive Learning and K-Means Clustering

Ankush Raut
University of Colorado Boulder
ankush.raut@colorado.edu

Abstract

The text generated by Large Language Models (LLMs) across a variety of tasks mimics natural language very closely. Detection of AI-generated text is essential to prevent the rise of plagiarism and risks associated with impersonation. In this work, the performance of DistilBERT in classifying AI-generated text (sourced from Kaggle datasets) correctly will be studied. Additionally, a watermarking strategy will be employed while generating text from Llama-2 to study whether introduction of implicit watermarks can aide in the detection of LLM-generated text. The code is available at Watermarking-LLM-Generated-text.

1 Introduction

Large Language Models (LLMs) like Llama-2, the GPTs, BERT, etc. have been extensively trained on a huge corpus encompassing much of the internet. The effectiveness of the Transformer architecture that LLMs are built on along with the extensiveness of the corpus enables LLMs to mimic human conversation. With such a drastic improvement in language generation and inference comes the possibility of a widespread misuse of LLMs, for plagiarism, impersonation, and identity theft. In order to mitigate the consequences from such activities, it is essential to have methods in place for accurately identifying LLM-generated text. In this work, the efficacy of DistilBERT in the detection of LLM-generated text will be studied using datasets sourced from Kaggle. Following these observations, an implicit watermarking method will be implemented during the generation of text from Llama-2. The classifiers will be tested again on the texts generated with the watermark, to observe if they're able to detect those better.

2 Related Work

Watermarking LLM-generated text is a relatively new research avenue in NLP, considering how the large-scale use of LLMs, specifically for text generation has only started to happen in the recent times. The existing work on watermarking LLM-generated text largely involves token-level watermarking. The sentence-level watermarking system proposed here is inspired by Tsvetkov et al (2023). They make use of locality sensitive hashing (LSH) for sampling sentences with rejection from LLMs. He et al (2024) have theorized a simpler K-Means clustering-based approach for sampling sentences with rejection. Both of these works fine-tune a sentence transformer with a contrastive learning objective by augmenting single-domain texts using the Pegasus paraphraser, to be used for generating sentence embeddings required for sampling with rejection. In this work, cross-domain texts will be utilized. The other methods involved remain similar to the works mentioned above.

3 Methodology

3.1 Classifier

In this work, the distilbert-base-uncased-finetuned-sst-2-english pre-trained weights were fine-tuned for LLM-generated text detection and an initial classification report on the hold-out set was obtained. Then after replacing the LLM-generated texts in the hold-out set with watermarked texts, another classification report was generated. These reports have been compared and contrasted in the Experiments section.

3.2 Contrastive Learning to generate sentence embeddings

The all-MiniLM-L6-v2 sentence transformer was fine-tuned on the hold-out set using a contrastive learning objective. The contrastive loss function used is described below, and was first proposed by Chen et al (2021).

$$\mathcal{L}_{cl} = mean\left((1 - label) \times \|embed_1 - embed_2\|^2 + label \times \|clamp(1 - \|embed_1 - embed_2\|, 0)\|^2\right)$$

This contrastive loss function enforces the sentence embeddings of similar sentences to be similar and those of different sentences to be distant. The dataset preparation for this fine-tuning involves generation of paraphrases of sentences in the dataset using the pegasus_paraphrase pre-trained weights. The true label corresponding to the paraphrases will be 1. Sentences will be randomly sampled from the corpus to create negative examples, the true label corresponding to those being 0.

3.3 K-Means clustering for generating regions

The fine-tuned all-MiniLM-L6-v2 sentence transformer will be used to generate sentence embeddings for all sentences in the hold-out set. K-Means clustering was used to seggregate these embeddings into different clusters. At each generation stage, these clusters will be divided into blocked and valid clusters. The division will be dependent on the cluster assignment of the sentence embedding of the previous sentence.

3.4 Generation with rejection sampling

For each LLM-generated text in the hold-out set, a new generation using Llama-2-7b-chat-hf was initialized using the first sentence. The cluster assignment of the sentence embedding of the previous sentence was used as a random seed for dividing the clusters created in the previous method into valid (25% of the clusters) and blocked regions. If the sentence embedding of the newly generated sentence fell in the rejection region, it was selected, otherwise rejected until 5 sentence generation steps, post which the last trial was accepted.

4 Experiments

4.1 Data

In this work, 2 Kaggle datasets were combined - ArguGPT and The Learning Agency Lab's AI-generated essays data. These datasets, when combined, yield 1375 human-generated essays on different topics and 353 essays generated by different GPT models. 10% of these essays were randomly selected for the hold-out set, 75% of the remaining were put in the training set and the remaining in validation set for early stopping.

4.2 DistilBERT

The weight initialization for this classification was done using the pre-trained weights of distilbert-base-uncased-finetuned-sst-2-english. For fine-tuning, 10 epochs were specified with early stopping upon no improvement of the binary cross-entropy loss on the validation set for 3 consecutive epochs, with a learning rate of 10^{-5}.

The model showed improvement in validation loss until 4 epochs. The output probabilities were converted to discrete values using a threshold averaged over all thresholds that resulted in a true positive rate higher than 0.8 and a false positive rate lower than 0.2. The results are summarized in Table 1.

4.3 Paraphrase generation

The pegasus_paraphrase model was used to generate 2 different paraphrases for each sentence in the hold-out set. The paraphrases were generated with sampling enabled for conditional generation using the aforementioned model, maximum token limit of 30, minimum token limit of 5, 2 beams, and a length penalty of 1.

4.4 Contrastive Learning

The all-MiniLM-L6-v2 sentence transformer was fine-tuned using the contrastive learning loss function. To setup this fine-tuning, the original sentences in the hold-out set were merged with the paraphrases. The true label corresponding to these pairs would be 1. Corresponding to each sentence in the original hold-out set, 3 distinct sentences were randomly sampled and paired from the rest of the corpus. The true label corresponding to these pairs would be 0. The final data for contrastive learning uniquely yielded 13759 negative pairs and 6927 positive pairs. 25% of this data was separated into a validation set for early stopping. For fine-tuning, 50 epochs were specified with early stopping upon no improvement of the contrastive loss on the validation set for 5 consecutive epochs, with a learning rate of 10^{-5}.

The model showed improvement in validation loss until 5 epochs, upon which it was stopped after the next 5 epochs and saved for generating sentence embeddings.

Table 1: Initial classifier results on the hold-out set

	Precision	Recall	F1-score	Support
Human-generated	0.99	1	0.99	138
AI-generated	1	0.94	0.97	35
Accuracy			0.92	173
Macro average	0.96	0.81	0.86	173
Weighted average	0.93	0.92	0.92	173

4.5 Generation with rejection sampling

4.5.1 Cluster generation

Sentence embeddings were obtained using the fine-tuned all-MiniLM-L6-v2 sentence transformer for all the sentences in the hold-out set. K-Means clustering was employed to generate 8 cluster centroids with a silhouette score of 0.05.

4.5.2 Sampling essay length from range

The Llama-2-7b-chat-hf pre-trained weights were initialized for causal language modeling, i.e. generating the next token conditioned on the past tokens only. The first version of watermarked essay generation involved randomly sampling the number of sentences to be generated from the integer range of the number of sentences in the essays in the hold-out set (4 to 94). The first sentence of each essay in the hold-out set was used to initialize the causal generation. The cluster assignment of the immediate previous sentence at each stage was used as a random seed to divide the 8 clusters into valid (2 clusters) and blocked regions (6 clusters).

The sentence generation was done by prompting the Llama-2 causal language model using the prompt "Generate only one sentence following the given sentence:<sentence>". If the sentence embedding of the generated sentence fell in the valid region it was kept in the essay, otherwise rejected and resampled until 5 tries, upon which the latest generated sentence was kept. The AI-generated essays in the hold-out set were replaced with their counterparts in the watermarked essays. The classification report for this version of the hold-out set is summarized in Table 2.

4.5.3 Sampling essay length from a normal distribution

It was observed that the watermarked essays were much longer than the original AI-generated essays in the hold-out set, which can be attributed the essay length sampling criterion in the first version. The second version of watermarked essay generation involved randomly sampling the number of sentences to be generated from a normal distribution with the average number of sentences per AI-generated essay as the mean and the standard deviation of those as the standard deviation. The classification report for this version of the hold-out set is summarized in Table 3.

5 Results and Conclusion

Initially, the fine-tuned classifier does really well at predicting LLM-generated text for the given dataset, achieving a recall of 94% on the hold-out set in detecting AI-generated essays. The classifier recall drops when the first version of watermarked essays is introduced. This can be attributed to the fact that the average sentence length in the first version of watermarked essays is 71, while the average length of the AI-generated essays in the hold-out set is 18. Similar discrepancy was observed in the standard deviation. From this observation, it can be hypothesized that the longer the AI-generated essay, the harder it is for the encoder-based classifier to detect it.

The classifier accuracy improves significantly upon introducing smaller watermarked essays following the same length distribution as that of the original AI-generated essays. However, it is still lower than the recall on the original AI-generated essays. Moreover, the silhouette score of the clustering algorithm is extremely low, indicating the presence of overlapping clusters. The watermarking method that was experimented in this work made detection of AI-generated essays even harder for the DistilBERT classifier. This can be attributed to the low silhouette score in clustering and the small size of the dataset for fine-tuning the sentence transformer.

Table 2: Classifier results on the watermarked hold-out set version 1

	Precision	**Recall**	**F1-score**	**Support**
Human-generated	0.91	1	0.96	138
AI-generated	1	0.63	0.77	35
Accuracy			0.92	173
Macro average	0.96	0.81	0.86	173
Weighted average	0.93	0.92	0.92	173

Table 3: Classifier results on the watermarked hold-out set version 2

	Precision	**Recall**	**F1-score**	**Support**
Human-generated	0.95	1	0.98	138
AI-generated	1	0.8	0.89	35
Accuracy			0.96	173
Macro average	0.98	0.9	0.93	173
Weighted average	0.96	0.96	0.96	173

6 Future Work

The methodologies presented in this work can be extended across multiple avenues to further study the efficacy of the experimented watermarking design. A larger dataset of AI-generated texts from multiple LLMs will be better suited to making accurate inferences regarding these methods. Additionally, a larger hold-out set would mean a more comprehensive contrastive learning fine-tuning of the sentence transformer. The K-Means clustering would also improve as a result of this. Additionally, other region-generation methods can be explored. During the generation with rejection sampling stage, larger trial limits can be explored, along with different parameters for causal generation. These extensions of the presented work will give us a better understanding of its efficacy.

7 References

1. SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation; Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, Yulia Tsvetkov (2023)

2. k-SemStamp: A Clustering-Based Semantic Watermark for Detection of Machine-Generated Text; Abe Bohan Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, Tianxing He (2024)

3. SimCSE: Simple Contrastive Learning of Sentence Embeddings; Tianyu Gao, Xingcheng Yao, Danqi Chen (2021)

4. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization Jingqing Zhang, Yao Zhao, Mohammad Saleh, Peter J. Liu (2019)

5. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf (2019)

6. Llama 2: Open Foundation and Fine-Tuned Chat Models Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, Thomas Scialom (2023)