1 Introduction

As large language model (LLM) usage has become widespread, having reliable methods for the detection of LLM-generated content has become necessary. Many of the current methods for detection, such as watermarking are not robust against obfuscation methods, such as paraphrasing [2]. This is likely to become more of a problem over time, especially as the outputs of these LLMs become more and more human-like. Recent research has shown retrieval methods to be more reliable at detecting LLM usage, even when documents have been paraphrased [1]. Here we attempt to construct a pipeline for the detection of potential plagiarism using AI, by utilising retrieval and plagiarism detection methods. Using content similarity allows us to report on the similarity between documents, rather than giving a simple "yes/no" answer as to whether a suspicious document has been plagiarised.

2 The Pipeline

The pipeline itself involves generating prompt data based on the MS Marco Question answering dataset, and loading it into an indexer. Then, prompts are paraphrased before being used to search the database and test retrieval. More specific details on how to run the pipeline can be found in the README of the project's repository.

3 Software

LLaMA was used to generate responses to the QnA dataset, for use in testing and to serve as a sample database of AI-generated responses. The default 7B model weights were used, with no additional training. Note that LLaMA did not produce output for a small part of the dataset (<20 entries).

DIPPER [1], which is a semantic-level paraphraser, was used to generate a set of perturbed documents used as a test set for retrieving the corresponding entries.

Between LLaMA and DIPPER, there were several entries that were exceedingly short, and thus not useful for retrieval. Many of these were a result of simple questions in the MS Marco set, so any queries with paraphrased or original documents less than 50 characters long were omitted. Note however that some of these were due to unusual outputs from DIPPER. These entries are included along with the rest of the provided files.

ATIRE[1] was used for indexing and retrieval of documents, and for generating document/collection dictionaries for kl divergence query generation. Generally, the default settings were also used for this.

Sherlock[2] was used for content similarity reports for retrieved documents. Sherlock is a combination of sig and comp which were software written by Rob Pike. The original source for Sherlock is not available, so instead the link to the GitHub repository from where the software was obtained is included.

Details of files produced at all points in this pipeline can be found in the README of the project repository.

4 Results

Based on preliminary results, ATIRE managed to retrieve the correct document within the top 10 hits for ≈93.63% of the queries tested, and correctly retrieved 79.59% of queries as the top-1 hit. It's possible that this result could be improved by altering some parameters within ATIRE's ranking function—it may be worth testing this in future.

For correctly identified documents, Sherlock reported on average a 17.75% similarity between the original and paraphrased document. Comparatively, it reported an average document similarity of 0.96% between the paraphrased document and incorrectly identified documents. More comprehensive testing is needed to determine the general efficacy of content similarity detection software.

[1] `https://github.com/andrewtrotman/ATIRE`

[2] `https://github.com/diogocabral/sherlock`

References

[1] Krishna, K., Song, Y., Karpinska, M., Wieting, J., and Iyyer, M. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defence. https://arxiv.org/abs/2303.13408, 2023.

[2] Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., and Feizi, S. Can AI-generated text be reliably detected? https://arxiv.org/abs/2303.11156, 2023.