Assessing AI Detectors in Identifying AI-Generated Code: Implications for Education

Authors

Wei Hung Pan	Ming Jie Chok	Jonathan Wong Leong Shan
Yung Xin Shin	Yeong Shian Poon	Zhou Yang
Chun Yong Chong	David Lo	Mei Kuan Lim

Abstract

This abstract introduces the artifact associated with the ICSE '24 paper titled "Assessing AI Detectors in Identifying AI-Generated Code: Implications for Education." The study delves into an empirical examination of the LLM's attempts to circumvent detection by AIGC Detectors. The methodology entails code generation in response to targeted queries using diverse variants. Our primary goal is to attain the Available and Reusable badges. The abstract further offers comprehensive technical details about each artifact component and elucidates its utility for prospective research endeavors.

Folder structure

The project directory should contain the following structure:

Folder/File	Description
GPTZero	Folder that contains all files required to setup and run GPTZero
DectectGPT	Folder that contains all files required to setup and run DectectGPT
GLTR	Folder that contains all files required to setup and run GLTR
Sapling	Folder that contains all files required to setup and run Sapling
Gpt2outputdetector	Folder that contains all files required to setup and run GPT-2 Output Detector
VariantData	Folder that contains all variant data with the AIGC, there should be 13 variant file

_metrics_performance.ipynb	Python jupyter notebook that is used to calculate the performance of the AIGC Detector
.gitattributes	For upload large files like ".pt"
STATUS.md	For mentions the badges we are applying
LICNSE	For describing the distribution rights

```
├── ./DetectGPT                # All files for setup and run DetectGPT
│   └── README.md             # Guidance on setup the DetectGPT
├── ./GLTR                     # All files for setup and run GLTR
│   └── README.md             # Guidance on setup the GLTR
├── ./Gpt2outputdetector       # All files for setup and run Gpt2outputdetector
│   └── README.md             # Guidance on setup the Gpt2outputdetector
├── ./GPTZero                  # All files for setup and run GPTZero
│   └── README.md             # Guidance on setup the GPTZero
├── ./Sapling                  # All files for setup and run Sapling
│   └── README.md             # Guidance on setup the Sapling
├── ./VariantData              # Stores all the variants data with the AIGC
├── _metrics_performance.ipynb   # Calculate the Performance of AIGC Detector
└── README.md                  # Paper Details and structure of the folders
```

We have provided the 13 variants of AI-Generated Content (AIGC) in data folder. Each variant will be using the same set of human-written code, hence, you will only have to execute the code detection for human-written code once.

README file are provided for each AIGC Detector, also known as Code Detection Model (CDM), to provide guidance on how to setup the AIGC Detector. Then, you can execute the code detection file for each AIGC Detector and get the results for the corresponding variant.

After retrieving all the results from the AIGC Detector, execute the code in _metrics_performance.ipynb to get the performance of the AIGC Detector. You are recommended to combine all results csv file into one file, with one header and 13 variant of detector classification results.

Step Instructions

1. Review the '**README.md'** file located at the root of folder, which comprehensively outlines the details of associated research papers and provides a clear structure for the folders.

2. Select a specific AIGC Detector folder and follow the contents of the ‘**README.md’** file located within that particular folder.
3. After retrieving results from the AIGC Detector, execute the code provided in the **'_metrics_performance.ipynb'** notebook to assess the performance of the AIGC Detector.
    a. It is recommended to consolidate all result CSV files into a single file with a unified header, encompassing the 13 variants of detector classification results.