	τ	News	Wiki	Reddit	Books	Abstracts	Reviews	Poetry	Recipes	Total
R-B GPT2	0.759	1.0%	3.2%	3.7%	4.0%	1.6%	3.1%	15.4%	7.3%	5.0%
R-L GPT2	0.307	1.8%	7.0%	2.4%	2.5%	1.3%	4.0%	15.4%	5.1%	5.0%
R-B C-GPT	0.988	4.4%	4.0%	0.7%	9.0%	0.0%	1.1%	0.5%	18.6%	5.0%
RADAR	0.343	0.2%	1.2%	10.7%	2.0%	4.2%	20.4%	8.6%	0.1%	5.0%
GLTR	0.818	0.4%	0.8%	1.1%	0.0%	0.2%	0.1%	1.8%	33.4%	5.0%
F-DetectGPT	0.920	5.1%	1.6%	1.9%	6.3%	1.8%	2.8%	7.5%	2.2%	3.7%
LLMDet	1.000	10.2%	12.8%	6.1%	2.7%	0.2%	3.9%	3.4%	0.1%	5.0%
Binoculars	0.090	2.8%	5.7%	7.2%	4.0%	5.3%	5.7%	3.8%	5.9%	5.0%
GPTZero	0.068	3.0%	2.0%	3.0%	13.0%	10.0%	5.0%	0.0%	1.0%	4.6%
Originality	0.494	4.0%	13.0%	2.0%	3.0%	4.0%	4.0%	3.0%	7.0%	5.0%
Winston	0.892	0.0%	10.0%	0.0%	13.0%	0.0%	0.0%	4.0%	13.0%	5.0%
ZeroGPT	1.000	29.0%	48.0%	0.0%	1.0%	0.0%	5.0%	0.0%	52.0%	16.9%

Table 8: False Positive Rates of our detectors on the RAID dataset broken up by domain when naively using a single threshold (τ). We see that while the total FPR across all domains still sums to 5%, individual domains see substantial fluctuation in FPR values.

is, we find the threshold for each detector for each domain that results in 5% FPR on that domain (see Table 13).

D Leaderboard and Pypi Package

In order to truly achieve our goal of standardizing detector evaluation, it is important for RAID to not only be sufficiently challenging, but also have a simple, straightforward interface for submission and comparison. As discussed in section 4.2, we solve this problem by releasing a shared leaderboard and a Pypi package to make submitting to the leaderboard easy. The RAID package can be installed by running:

```
$ pip install raid-bench
```

In Figure 9 we show how to use our pypi package to load the RAID test set and run a detector on the texts. After getting the `predictions.json` file, submitting to the leaderboard simply involves making a folder in the repository, filling out metadata, and creating a pull request.

In Figure 8 we show a screenshot of the leaderboard page on our project website. Submissions are open to the public and are automatically accepted and evaluated via pull requests to our git repository. The data used for evaluation on the leaderboard is the 10% test split of the core RAID dataset that is released without labels.

E Dataset Details

E.1 Domains

In Table 9 we report the exact number of documents sampled from each domain. The only two datasets that include post-2021 documents are Wikipedia and Abstracts. The Reviews domain did not have 2,000 documents and so we sampled the maximum amount. For each domain, we provide a detailed list of all human-written sources with metadata along with the RAID dataset in our code repository. A detailed description of the contents of each domain is as follows:

Book summaries (Bamman and Smith, 2013) This dataset contains plot-centric summaries of books along with their titles. We chose this dataset due to the first-person narrative style and because we expect generators and detectors with knowledge of the source material to have an advantage.

BBC News Articles (Greene and Cunningham, 2006) This dataset contains BBC articles with associated titles. The articles are spread out evenly across 5 categories (sport, technology, entertainment, politics, and business). This dataset was chosen since good generation requires factuality and because News is a large area for LLM-based harm.

Poems (Arman, 2020) This dataset contains poems collected from `poemhunter.com` with their titles and genre. The poems are randomly spread out over genres and topics. We hypothesize that LLMs will write generic and repetitive poetry and that this tendency should be detectable.

Recipes (Bień et al., 2020) This dataset consists of recipes and their dish names. Recipes are a combination of a list of ingredients and a numbered list of steps. This dataset is difficult because it requires significant common sense reasoning, which

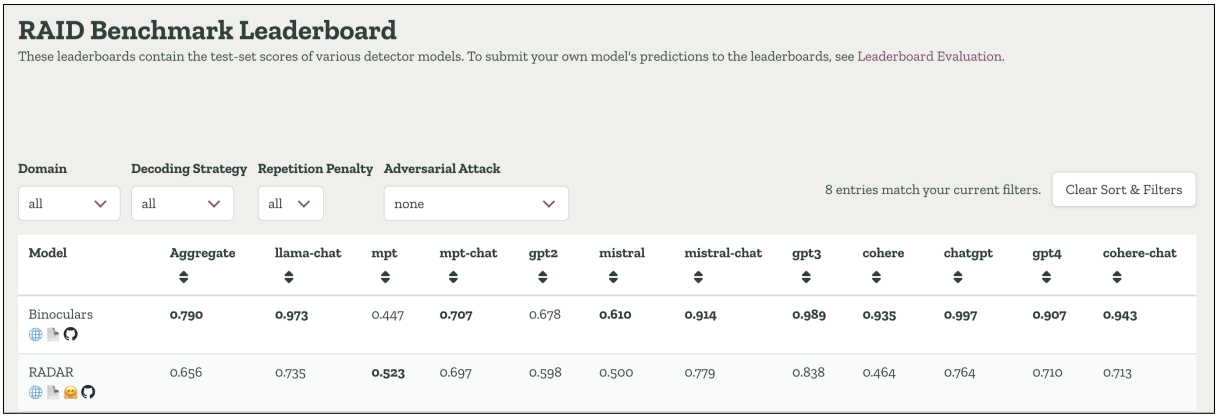

Figure 8: Screenshot of the RAID leaderboard accessible at https://raid-bench.xyz/leaderboard

```python
from raid import run_detection
from raid.utils import load_data

# Define your detector function
def my_detector(texts: list[str]) -> list[float]:
    pass

# Load the RAID test data
test_df = load_data(split='test')

# Run your detector on the dataset
predictions = run_detection(my_detector, test_df)

# Write predictions to a JSON file
with open('predictions.json') as f:
    json.dump(predictions, f)
```

Figure 9: A example showing how to use the RAID Pypi package to evaluate a detector on the dataset and submit it to the leaderboard.

is difficult for models.

Reddit Posts (Völske et al., 2017) This dataset contains reddit posts and their titles. We hypothesize that such data will be challenging to detect due to the first-person and informal style.

Movie Reviews (Maas et al., 2011) This dataset contains movie reviews from IMDb along with the names of the movies. The formality of the reviews are varied and this tests model's ability to recall details from movies as well as generate and detect opinionated text.

Wikipedia (Aaditya Bhat, 2023) This dataset contains introductions to various Wikipedia articles. This dataset is challenging as it tests the models ability to accurately recall facts relating to specific historical events.

Python Code (Raychev et al., 2016) This dataset contains python solutions to coding problems and the associated problem title. We include this as an initial foray into the detection of AI-generated code.

Czech News (Boháček et al., 2022) This domain consists of Czech language news articles. The topics and sources are diverse sampling from over 45 publications including mainstream journalistic websites, tabloids and independent news outlets in the Czech Republic.

German News (Schabus et al., 2017) This domain consists of German language news articles from DER STANDARD, an Austrian daily broadsheet newspaper. Articles are fairly political in nature covering topics such as the European migrant crisis, the 2016 Austrian presidential elections and the Syrian Civil War. We expect this dataset to test models' ability to generate opinionated political content in another language.

Paper Abstracts (Paul and Rakshit, 2021) This is a dataset of abstracts scraped from ArXiv together with paper titles. For this dataset and this dataset only, we filter the data such that only papers from 2023 or later are present in the data. This allows us to rule out the possibility that our models have memorized this text.

E.2 Generative Models

In Table 10 we list the exact generative models used in our project along with their unique identifiers. All open-source models were run using the HuggingFace `transformers` library (Wolf et al., 2020) and all closed-source models were run using

Dataset	Genre	Size
(Paul and Rakshit, 2021)	Abstracts	1966
(Bamman and Smith, 2013)	Books	1981
(Raychev et al., 2016)	Code	920
(Greene and Cunningham, 2006)	News	1980
(Arman, 2020)	Poetry	1971
(Bień et al., 2020)	Recipes	1972
(Völske et al., 2017)	Reddit	1979
(Maas et al., 2011)	Reviews	1143
(Aaditya Bhat, 2023)	Wiki	1979
(Boháček et al., 2022)	Czech	1965
(Schabus et al., 2017)	German	1970

Table 9: The number of articles sampled from each domain with their corresponding sources

Model	Identifier
GPT-2 (Radford et al., 2019)	`gpt2-xl`
MPT (+ *Chat*) (MosaicML, 2023)	`mpt-30b` `mpt-30b-chat`
Mistral (+ *Chat*) (Jiang et al., 2023)	`Mistral-7B-v0.1` `Mistral-7B-Instruct-v0.1`
LLaMA Chat (Touvron et al., 2023)	`Llama-2-70b-chat-hf`
Cohere (+ *Chat*) (Cohere, 2024)	`command (co.generate())` `command (co.chat())`
GPT-3 (Ouyang et al., 2022)	`text-davinci-002`
ChatGPT (OpenAI, 2022)	`gpt-3.5-turbo-0613`
GPT-4 (OpenAI, 2023)	`gpt-4-0613`

Table 10: The generative models used in our project

the proprietary APIs from Cohere[7] and OpenAI[8]. The following is a detailed list of the generative models used in the project.

GPT2 XL 1.5B (Radford et al., 2019) is a decoder-only model trained on the WebText dataset. This dataset consists of a collection of all documents that were linked from reddit posts or comments that had at least 3 or more upvotes. Released in February of 2019 and having 1.5B parameters, GPT2 is the predecessor of GPT3 and GPT4 and the most powerful open-source OpenAI model.

GPT3 (Ouyang et al., 2022) is a closed-source language model released by OpenAI on November 29th, 2022. The model was allegedly trained with a variety of data including the Common Crawl (filtered), WebText2, and Wikipedia datasets (Brown et al., 2020) but exact composition of the training dataset is unknown. It is the first model shown to work well with prompts and has shown great zero- and few-shot capabilities. In this study, we use the `text-davinci-002` model. We queried the model from November 1st to November 2nd 2023. Unfortunately, as of January 4th 2024, this model is no longer available for use on the OpenAI API. This is unfortunate as it prevents us from expanding the domains in future releases. We encourage researchers to keep this in mind when using OpenAI models for their research projects.

ChatGPT (OpenAI, 2022) is a version of GPT3 fine-tuned using Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022). We use the June 13th 2023 checkpoint of the model (`gpt-3.5-turbo-0613`). Although the number of parameters is unknown, ChatGPT demonstrates outstanding capability in language and code generation.

GPT4 (OpenAI, 2023) is the latest iteration of OpenAI's GPT family of models and is one of the largest and most powerful language models available to date. In this study, we use the `gpt-4-0613` checkpoint of the model through the `ChatCompletion` interface[9]. We queried the model from November 1st to November 2nd 2023.

LLaMA 2 70B (Touvron et al., 2023) is a decoder-only model trained by Meta (Facebook) and is the second model in the LLaMA series. Released on July 18th 2023, LLaMA 2 is the successor to the original LLaMA model which was trained on webpages from CommonCrawl, multilingual Wikipedia, books from Project Gutenberg, and QAs from Stack Exchange. The composition of LLaMA 2's training data is not known but it has shown impressive performance on many open-source evaluations and is widely considered competitive with the open-source state-of-the-art.

Mistral 7B (Jiang et al., 2023) is a decoder-only model trained by Mistral and is the first model released by the company. Released on September 27, 2023, Mistral 7B outperforms LLaMA 2 13B across various benchmarks at half the size. While model weights are open-source, the training data

[7] `https://docs.cohere.com/reference/about`
[8] `https://platform.openai.com/docs/introduction`
[9] `https://platform.openai.com/docs/guides/chat`