M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection

Yuxia Wang, Jonibek Mansurov,* Petar Ivanov,* Jinyan Su,* Artem Shelmanov,* Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, Preslav Nakov
Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
{yuxia.wang, jonibek.mansurov, jinyan.su, artem.shelmanov,
osama.afzal, tarek.mahmoud, alham.fikri, preslav.nakov}@mbzuai.ac.ae
chenxi.whitehouse@gmail.com petar.ivanov256@gmail.com

Abstract

Large language models (LLMs) have demonstrated remarkable capability to generate fluent responses to a wide variety of user queries, but this has also resulted in concerns regarding the potential misuse of such texts in journalism, educational, and academic context. In this work, we aim to develop automatic systems to identify machine-generated text and to detect potential misuse. We first introduce a large-scale benchmark **M4**, which is **m**ulti-generator, **m**ulti-domain, and **m**ulti-lingual corpus for **m**achine-generated text detection. Using the dataset, we experiment with a number of methods and we show that it is challenging for detectors to generalize well on unseen examples if they are either from different domains or are generated by different large language models. In such cases, detectors tend to misclassify machine-generated text as human-written. These results show that the problem is far from solved and there is a lot of room for improvement. We believe that our dataset M4, which covers different generators, domains and languages, will enable future research towards more robust approaches to this pressing societal problem. The M4 dataset is available at https://github.com/mbzuai-nlp/M4

1 Introduction

Large language models (LLMs) are becoming mainstream and easily accessible, ushering in an explosion of machine-generated content over various channels, such as news, social media, question-answering fora, educational, and even academic contexts. Recent LLMs (e.g., ChatGPT and GPT4) have demonstrated to be able to generate remarkably fluent responses to a wide variety of user queries. The articulate nature of such generated text makes LLMs attractive for replacing human labor in many scenarios. However, this has also resulted in concerns regarding the potential misuse of such texts, such as spreading misinformation (e.g., in journalism) and causing disruptions in the education system (e.g., in student essays) (Tang et al., 2023).

Unfortunately, humans perform only slightly better than chance when classifying machine-generated vs. human-written text (Mitchell et al., 2023). Therefore, we aim to develop automatic systems to identify machine-generated text with the aim to mitigate its potential misuse.

There has been some previous effort in detecting machine-generated text. For example, Guo et al. (2023) detect whether a certain text (English and Chinese) is generated by ChatGPT or is human-written across several domains, Shijaku and Canhasi (2023) identified whether TOEFL essays were human-written and generated by ChatGPT over a mini set (126 essays for each). Both these attempts only focused on generations of ChatGPT. The RuATD Shared Task 2022 involved artificial text in Russian over models of machine translation, paraphrase generation, text summarization and text simplification (Shamardina et al., 2022). However, they used generations of models fine-tuned for specific tasks or domains, which is not of interest here. Instead, we pay more attention to zero-shot generations of LLMs, such as the subset of RuATD generated by ruGPT-3. Mitchell et al. (2023) detected generations of GPT-2, OPT-2.7, Neo-2.7, GPT-J, and NeoX, but these LLMs are obsolete since there is GPT-3 and even GPT-4.

Overall, prior work either focused on only one or two particular languages or detected machine-generated text for a specific LLM (e.g. ChatGPT) or a specific domain (e.g., news by Zellers et al. (2019)). We include multiple languages, and various popular LLMs across diverse domains in this work, aiming to develop more general machine-generated text detection approaches.

Our contributions can be summarized as follows:

- We collect **M4**: a large-scale multi-generator,

* Equal contribution.

multi-domain, and multi-lingual corpus for detecting machine-generated text.

- We leverage diverse features including semantic, stylistic, and statistical based on token prediction probability using GPT-2 (GLTR) and features used for news veracity detection (NELA) to distinguish human-written vs. machine-generated text.
- We analyze the evaluation results from various dimensions: (1) the performance of different detectors across different domains given a specific generator, (2) the performance of different detectors across different generative models for a specific domain, and (3) the impact on the performance by interactions of different domains and generators in a multilingual setting.

2 Related Work

The latest efforts in identifying machine-generated text approach this task as a binary classification problem. We categorize the detection strategies into black-box and white-box methods, contingent upon the level of access to the LLM that is suspected to have generated the target text.

2.1 Black-Box Detection

Under black-box detection, classifiers are restricted to API-level access to an LLM (only the text is available). To develop a proficient detector, black-box approaches are generally designed to first extract and select features based on text samples originating from both human and machine-generated sources, and then to train a classification model leveraging relevant features. Therefore, the effectiveness of black-box detection models is heavily dependent on the quality and the diversity of the acquired data.

Related Corpus Recently, a growing body of research has concentrated on amassing responses generated by LLMs and comparing them to human-written texts spanning a wide range of domains. Guo et al. (2023) collected the HC3 dataset, which consists of nearly 40K questions and their corresponding answers from human experts and ChatGPT (English + Chinese), covering a wide range of domains (open-domain, computer science, finance, medicine, law, and psychology). Shijaku and Canhasi (2023) gathered TOEFL essays written by human writers and such generated by ChatGPT (126 essays for each). Both of these studies only focused on generations by ChatGPT. RuATD Shared Task 2022 involved an artificial text in Russian over models of machine translation, paraphrase generation, text summarisation and text simplification (Shamardina et al., 2022). However, they used generations by models fine-tuned for specific tasks or domains, which is not our focus here. We pay more attention to zero-shot generations of LLMs, such as the subset of RuATD generated by ruGPT-3.

In general, previous studies have concentrated on detecting machine-generated text in one or two specific languages, or for a particular LLM such as ChatGPT, or within a specific domain such as news (Zellers et al., 2019). Our work broadens this scope to include multiple languages and a variety of widely-used LLMs across different domains. The goal is to develop a more universal approach to the detection of machine-generated text.

Guo et al. (2023) applied two methods: logistic regression with GLTR features (Gehrmann et al., 2019) and an end-to-end RoBerta classifier, to detect whether a certain text (English and Chinese) is generated by ChatGPT or humans across several domains The underlying assumption is that most LLMs sample from the head of the distribution, thus word ranking information of the language model can be used to distinguish LLM-generated text. Shijaku and Canhasi (2023) detected TOEFL essays using XGBoost with manually extracted 244 lexical and semantic features (e.g. TF-IDF).

There are also widely-used off-the-shelf GPT detectors, such as the OpenAI detection classifier,[1] GPTZero,[2] and ZeroGPT.[3] OpenAI's AI-text Classifier is fine-tuned on the output of a pre-trained language model. They trained their model on samples from multiple sources of both human-written and AI-generated text using text generated by 34 models from five different organizations that deploy language models, including OpenAI. GPTZero is trained on an extensive and varied corpus of text authored by both humans and AI, with a primary focus on English. As a classification model, GPTZero predicts whether a given piece of text, at various textual granularities, including sentence, paragraph, and entire document levels, was generated by a large language model.

Previous detectors are either end-to-end mod-

[1] https://platform.openai.com/ai-text-classifier
[2] https://gptzero.me/
[3] https://www.zerogpt.com/

els or binary classification models based on extracted features. Features can be categorized into three types: statistical distributions (e.g., GLTR-like word rankings), linguistic patterns (such as vocabulary features, part-of-speech tags, dependency parsing, sentiment analysis, and stylistic features) and fact-verification features (Tang et al., 2023). Classification models can be either traditional algorithms, such as logistic regression, Support Vector Machines, Naïve Bayes, and Decision Trees, or deep neural networks.

2.2 White-Box Detection

Another stream of work focused on zero-shot AI text detection without any additional training overhead (Sadasivan et al., 2023). It can be simply categorized into two clusters depending on whether watermarking is being used. One line of research evaluates the expected per-token log probability of texts and performs thresholding to detect AI-generated texts, such as RankGen (Krishna et al., 2022) and DetectGPT (Mitchell et al., 2023). DetectGPT is a zero-shot LLM text detection method. It leveraged the observation that AI-generated passages tend to lie in the negative curvature of the log probability of texts. Since these approaches rely on a neural network for their detection, they can be vulnerable to adversarial and poisoning attacks. Another line of research aims to watermark AI-generated texts to ease their detection.

2.3 Watermark-Based Detection

Techniques of watermarking are initialized to protect the fair use and the intellectual property of generation models. Such techniques can be used to ease the detection of LLM output text by imprinting specific patterns on them. The core of the watermarking of DNN models is to superimpose secret noise on the protected models. Various approaches range from text-level posthoc lexical substitutions and synonym replacement over generated outputs (Szyller et al., 2021; He et al., 2022), and soft watermarking (Kirchenbauer et al., 2023) using green and red token lists, to hidden-space operations, such as injecting secret signals into the probability vector of the decoding steps for each target token (Zhao et al., 2023).

We focus on black-box methods in this work.

3 Data

We gathered human-written text from various sources across domains such as Wikipedia (the March 2022 version), WikiHow (Koupaee and Wang, 2018), Reddit (ELI5), arXiv, and PeerRead (Kang et al., 2018) for English, Baike and Web question answering (QA) for Chinese, news for Urdu, RuATD (Shamardina et al., 2022) for Russian and news for Indonesian.

For machine generation, we prompt the following multilingual LLMs: ChatGPT, *text-davinci-003*, LLaMa (Touvron et al., 2023), FlanT5 (Chung et al., 2022), Cohere, Dolly-v2, and BLOOMz (Muennighoff et al., 2022). Models are asked to write Wikipedia articles given titles, abstracts based on the title (arXiv), peer reviews based on the title and the abstract (PeerRead), answer questions (e.g., Reddit and Baike/Web QA) and write news briefs based on the title.

3.1 English Corpora

Wikipedia We use the Wikipedia datasets available on HuggingFace[4] and randomly choose 3,000 articles that have a length exceeding 1,000 characters. We then prompt the LLMs to generate Wikipedia articles based on given titles, with the additional requirement that the resulting articles should contain at least 250 words. For generation with Dolly-v2[5], we also specify the minimum number of tokens to be 300.

Reddit ELI5 The ELI5 dataset is a collection of English questions and responses sourced from three specific subreddits, designed to facilitate the process of open-domain, long-form abstractive question answering. These subreddits include r/explainlikeimfive, which focuses on general topics, r/askscience, which is centered around scientific queries, and r/AskHistorians, which deals with historical inquiries. The dataset is intended for situations where the answers to questions need to be of paragraph length or more.

The dataset is available on HuggingFace.[6] Each thread is composed of a question, which includes a title and a body, and the accompanying responses. We preprocessed this dataset to obtain a total of 3,000 examples. For preprocessing, we eliminated any responses with less than 1,000 characters. Next,

[4] https://huggingface.co/datasets/wikipedia

[5] https://huggingface.co/databricks/dolly-v2-12b

[6] https://huggingface.co/datasets/eli5

Source/ Domain	Language	Parallel Data								Total	
		Human	Davinci003	ChatGPT	Cohere	Dolly-v2	BLOOMz	FlanT5	LLaMA	Human	All
Wikipedia	English	3,000	3,000	2,995	2,336	2,702	2,615	0	0	6,458,670	–
Reddit ELI5	English	3,000	3,000	3,000	3,000	0	0	3,000	0	558,669	–
WikiHow	English	3,000	3,000	3,000	3,000	0	3,000	0	0	31,102	–
PeerRead	English	5,798	2,344	2,344	0	0	0	0	2,344	5,798	–
arXiv abstract	English	3,000	3,000	3,000	3,000	0	3,000	3,000	0	2,219,423	–
Baike/Web QA	Chinese	3,000	3,000	3,000	–	–	–	–	–	113,313	119,313
RuATD	Russian	3,000	3,000	3,000	–	–	–	–	–	75,291	81,291
Urdu-news	Urdu	3,000	3,000	3,000	–	–	–	–	–	107,881	113,881
id_newspapers_2018	Indonesian	3,000	3,000	3,000	–	–	–	–	–	499,164	505,164
Arabic-Wikipedia	Arabic	3,000	3,000	3,000	–	–	–	–	–	–	–
Total		32,798	29,344	29,339	11,336	2,702	8,615	6,000	2,344	→122,481	–

Table 1: Statistics information of the dataset. Parallel data: human and machine-generated. Non-parallel human data. Development and test data: 500 examples per source (human or some generator); Training data: 2,000 examples + the non-parallel data.

we filtered out any thread where the title did not end with a question mark, and questions that were accompanied by a body text. We then sorted the responses by the score in descending order. We finally picked the top 1,000 responses for each of the three subreddits.

WikiHow The WikiHow dataset (Koupaee and Wang, 2018) is built from the online WikiHow knowledge base and consists of articles with a title, a headline, which is the concatenation of all bold lines of all paragraphs, and text, which is the concatenation of all paragraphs (except the bold lines). We chose randomly 3,000 articles with lengths of more than 1,000 characters and use their titles and headlines as prompts to produce new machine-generated articles. The original dataset can be downloaded from HugginFace.[7]

PeerRead Reviews We sampled 586 academic papers in fields of NLP and machine learning top-tier conferences released by ReerRead corpus (Kang et al., 2018), with metadata including title, abstract, and multiple human reviews for each paper. Given a paper, we prompt the generative models to generate peer reviews by four different instructions, with (1) and (2) only depending on the title, and another two involving both the title and the abstract.[8] (1) Please write a peer review for the paper of + title; (2) Write a peer review by first describing what problem or question this paper addresses, then strengths and weaknesses, for the paper + title; (3) Please write a peer review for the paper of + title, its main content is as below: + abstract; (4) Write a peer review by first describing what problem or question this paper addresses, then strengths and weaknesses, for the paper + title, its main content is as below: + abstract. This results in $584 \times 4 = 2{,}344$ machine-generated texts for each generator and 5,798 human reviews in total.

Note that peer reviews should be written by qualified reviewers after carefully reviewing the whole paper in real-world scenarios. We generate reviews by large language models in this work just for research purposes.

Arxiv Abstract For the Arxiv Abstract dataset, we use a dataset from Kaggle.[9] We select 3,000 abstracts from this dataset that have a minimum length of 1,000 characters. We use the titles of the chosen abstracts in prompts to create machine-generated abstracts.

3.2 Corpora in Other Languages

Chinese QA For the Chinese partition, we sampled 3,000 (question, answer) pairs from Baike and the Web QA corpus,[10] with each answer satisfying the criteria that the length should be more than 100 Chinese characters. We prompt generative models by the combination of a brief title and a detailed description for each question.

Urdu News This dataset is derived from the Urdu News Data 1M, a collection of one million news articles from four distinct categories: Business & Economics, Science & Technology, Entertainment,

[7] https://huggingface.co/datasets/wikihow

[8] We do not consider hallucinations in the context of detecting machine vs. human generations, so we manipulate peer reviews relying on the title and the abstract, instead of the full content of the paper.

[9] https://www.kaggle.com/datasets/Cornell-University/arxiv

[10] https://github.com/brightmart/nlp_chinese_corpus

and Sports. These articles were gathered from four reputable news agencies in Pakistan (Hussain et al., 2021). Each entry in this dataset includes a headline, a category, and a news article text. We selected a sample of 3000 examples from the corpus to generate machine-generated text. To ensure a balanced representation of the four categories in the dataset, we randomly sampled 750 examples from each category. We then used ChatGPT (gpt3.5-turbo) to generate content for the news article based on the provided headline.

Russian RuATD This dataset is sourced from RuATD Shared Task 2022 (Shamardina et al., 2022) devoted to artificial text detection in Russian. The authors of this task prepared vast human and machine-generated corpora from various text generators. We replaced the machine-generated data with new generations from state-of-the-art multilingual LLMs. The human-written texts are collected from publicly available resources across six domains. The list of domains includes normative Russian, social media posts, texts of different historical periods, bureaucratic texts that have a complex discourse structure and various specific named entities, subtitles, and web texts. In particular, for the construction of human-written data, the task organizers used the following sources: (1) diachronic subcorpora of the Russian National Corpus[11], which covers three historical periods of the society and the Modern Russian language ("pre-Soviet", "Soviet", and "post-Soviet"); (2) several social media platforms; (3) top-100 most viewed Russian Wikipedia pages spanning the period of 2016- 2021 according to the PageViews statistics; (4) news articles from the Taiga corpus (Shavrina and Shapovalova, 2017) and the corus library[12]; (5) a corpus of digitilized personal diaries "Prozhito" written during the 20th century (Melnichenko and Tyshkevich, 2017); (6) government documents from the RuREBus shared task (Ivanin et al., 2020).

Indonesian News 2018 This dataset is a collection of Indonesian news articles [13] from 7 different news websites, collected in 2018. For this purpose, we pick news from CNN Indonesia as we found that this source provides the cleanest data. We selected a sample of 3000 examples from the corpus to generate machine-generated text. We generate the artificial news by prompting ChatGPT to write a news article, given the title.

Arabic Wikipedia Similar to the English Wikipedia generation, we randomly choose 3,000 Arabic articles that have a length exceeding 1,000 characters. We then prompt the LLMs to generate Wikipedia articles based on given titles.

3.3 Dataset Statistics

The overall statistics about the data for different tasks and languages are given in Table 1. We collected $\sim$ 122k human–machine parallel data in total, with 101k for English, 9k for Chinese, 9k for Russian, 9k for Urdu, 9k for Indonesian, and 9k for Arabic respectively, in addition to over 10M non-parallel human-written texts.

Train, Dev, and Test Split For all languages, for each domain, given a generator (e.g., ChatGPT), we keep 500$\times$2 (500 examples from human and 500 from the machine-generated text) for development, 500$\times$2 for testing, and we use the rest for training.

4 Detection Models

4.1 RoBERTa Classifier

RoBERTa Classifier is based on the pretrained language model RoBERTa (Liu et al., 2019) and is finetuned to the task of detecting machine-generated text. Using pretrained classifier has been well studied and widely used in previous work (Solaiman et al., 2019; Zellers et al., 2019; Ippolito et al., 2019; Bakhtin et al., 2021; Uchendu et al., 2021). In (Solaiman et al., 2019) finetuned RoBERTa on the output of GPT-2 (Radford et al., 2019) and illustrates its high detection accuracy and the ability to transfer across different decoding strategies.

4.2 XLM-R Classifier

The XLM-R classifier is based on the XLM-RoBERTa model (Conneau et al., 2019), which is a cross-lingual variant of RoBERTa. XLM-RoBERTa is pre-trained on a multilingual corpus, enabling it to effectively model and understand text in multiple languages. The XLM-R-based detectors were fine-tuned using the capabilities provided by the transformers library[14].

[11] https://ruscorpora.ru/old/en/index.html
[12] https://github.com/natasha/corus
[13] https://huggingface.co/datasets/id_newspapers_2018
[14] https://huggingface.co/docs/transformers/model_doc/xlm-roberta

4.3 Logistic Regression with GLTR Features

GLTR (Gehrmann et al., 2019) studied three types of features of an input text. Their major assumption is that to generate fluent and natural-looking text, most decoding strategies sample high probabilities tokens from the head of the distribution. We select two categories of features: (1) the number of tokens in the Top-10, Top-100, Top-1000, and 1000+ ranks from the LM predicted probability distributions (4 features); and (2) Frac(p) distribution over 10 bins ranging from 0.0 to 1.0 (10 features). Frac(p) describes the fraction of probability for the actual word divided by the maximum probability of any word at this position. And then a logistic regression model is trained to perform the classification based on the 14 extracted features.

We do not incorporate the top-10 entropy feature (the entropy along the top 10 results for each word), as the features represented by the entropy distribution over 10 bins differ among inputs. The range of entropy differs between input text, after binning to 10 clusters based on the range, the number of tokens falling into each bin has different meanings for different entropy ranges.

4.4 Stylistic Features

Stylometry The extracted stylometry features (Li et al., 2014) are character-based features such as the number of characters, alphabets, special characters and etc., syntactic features such as the number of punctuation and function words, structural features such as the total number of sentences and word-based features such as the total number of words, average word length, average sentence length and etc.

NELA An updated version of the News Landscape (NELA) features (Horne et al., 2019) is used and they could be broken into six groups. Style – captures the style and structure of the article. Complexity – captures how complex the writing in the article is. Bias – captures the overall bias and subjectivity in the writing. Affect – captures sentiment and emotion used in the text. Moral – is based on Moral Foundation Theory (Graham et al., 2012). The event – captures two concepts: time and location.

The extracted features from both methods are used for binary classification to be determined if the text is written by a machine or a human performed by a Linear Support Vector Machine.

4.5 GPTZero

GPTZero is one of the popular proprietary systems for detecting machine-generated content[15], originating from Princeton University. It was launched in January 2023 and is claimed to gain one million users since then. Due to its proprietary nature, the technical details of GPTZero are not extensively disclosed. The description released by the authors says that the tool checks perplexity and burstiness, and its underlying model was trained on a large, diverse corpus of human-written and AI-generated text, with a focus on English prose. The system can analyze texts ranging from individual sentences to entire documents. The authors claim that it can robustly detect various AI language models, including ChatGPT (Ouyang et al., 2022), GPT-3 (Brown et al., 2020), GPT-2 (Radford et al., 2019), LLaMA (Touvron et al., 2023). Despite GPTZero is a closed system, we believe it is important to include its results in our benchmark as they demonstrate the capabilities and flaws of current commercial systems for the detection of machine-generated content. To conduct our experiments, we used the paid API that returns the probability of text to be generated by a model.

5 Experiment and Results

In this section, we conduct experiments in three settings: (1) same machine-text generator, cross-domain evaluation; (2) same domain, cross-generator evaluation; and (3) cross-lingual, cross-generator evaluation in Section 5.4. We additionally use the popular off-the-shelf detector API: GPTZero and simply refer to it as a zero-shot setting relative to our unreleased benchmark.

5.1 Same-Generator, Cross-Domain

Given a specific machine-text generator, such as ChatGPT and *davinci-003*, we train the detection model using training data from one domain and evaluate the model on the test set from the same domain (in-domain) and other domains (out-of-domain). Results by generator ChatGPT are shown in Table 2 and GPT3.5-davinci-003 in Table 3.

5.1.1 ChatGPT

RoBERTa For the RoBERTa detector, Table 2 shows that the best accuracy (bold numbers) for all domains is close to 1.0, with most of them occurring under in-domain evaluation, except for the

[15] https://gptzero.me/

Test → / Train ↓	Wikipedia Acc	Wikipedia Prec	Wikipedia Recall	Wikipedia F1	WikiHow Acc	WikiHow Prec	WikiHow Recall	WikiHow F1	Reddit ELI5 Acc	Reddit ELI5 Prec	Reddit ELI5 Recall	Reddit ELI5 F1	arXiv Acc	arXiv Prec	arXiv Recall	arXiv F1	PeerRead Acc	PeerRead Prec	PeerRead Recall	PeerRead F1
RoBERTa																				
Wikipedia	**0.997**	**0.994**	**1.000**	**0.997**	0.482	0.050	0.002	0.004	0.487	0.067	0.002	0.004	0.556	0.983	0.114	0.204	0.607	0.000	0.000	0.000
WikiHow	0.183	0.099	0.078	0.087	**0.997**	**0.998**	**0.996**	**0.997**	0.893	0.873	0.920	0.896	0.969	0.942	**1.000**	0.970	0.844	0.613	0.967	0.750
Reddit ELI5	0.791	0.707	0.994	0.826	0.824	0.802	0.860	0.830	0.897	0.829	**1.000**	0.907	0.995	0.998	0.992	0.995	0.806	0.557	0.967	0.707
arXiv	0.915	0.857	0.996	0.921	0.757	0.967	0.532	0.686	**0.959**	0.977	0.940	**0.958**	**1.000**	**1.000**	**1.000**	**1.000**	0.524	0.338	**1.000**	0.505
PeerRead	0.582	0.646	0.362	0.464	0.660	**0.988**	0.324	0.488	0.751	**1.000**	0.502	0.668	0.990	1.000	0.980	0.990	**0.980**	**0.925**	**1.000**	**0.961**
LR-GLTR																				
Wikipedia	**0.974**	**0.976**	0.972	**0.974**	0.785	0.878	0.662	0.755	0.862	0.785	**0.998**	0.879	0.944	0.983	0.904	0.942	0.709	0.672	0.816	0.737
WikiHow	0.913	0.873	0.966	0.917	**0.924**	**0.921**	0.928	**0.924**	0.938	**0.966**	0.908	0.936	0.904	0.998	0.810	0.894	0.841	0.875	0.796	0.834
Reddit ELI5	0.960	0.949	0.972	0.960	0.900	0.903	0.896	0.900	**0.954**	0.927	0.986	**0.955**	0.917	**1.000**	0.834	0.909	0.789	0.792	0.784	0.788
arXiv	0.925	0.873	**0.994**	0.930	0.873	0.825	0.946	0.882	0.848	0.768	**0.998**	0.868	**0.963**	0.964	**0.962**	**0.963**	0.770	0.701	**0.942**	0.804
PeerRead	0.889	0.821	**0.994**	0.900	0.712	0.639	**0.976**	0.772	0.845	0.767	0.992	0.865	0.894	0.988	0.798	0.883	**0.942**	**0.991**	0.892	**0.939**
Stylistic																				
Wikipedia	**0.974**	**0.976**	**0.972**	**0.974**	0.562	0.738	0.192	0.305	0.747	0.784	0.682	0.729	0.968	0.970	0.966	0.968	0.865	0.875	0.852	0.863
WikiHow	0.590	0.566	0.776	0.654	**0.957**	**0.977**	**0.936**	**0.956**	0.593	0.612	0.508	0.555	0.466	0.474	0.628	0.540	0.619	0.628	0.584	0.605
Reddit ELI5	0.889	0.912	0.861	0.886	0.497	0.483	0.084	0.143	**0.923**	**0.892**	**0.962**	**0.926**	0.893	0.973	0.808	0.883	0.807	0.863	0.730	0.791
arXiv	0.737	0.681	0.893	0.773	0.550	0.624	0.252	0.359	0.706	0.824	0.524	0.641	**1.000**	**1.000**	**1.000**	**1.000**	0.876	0.840	0.930	0.883
PeerRead	0.642	0.671	0.560	0.610	0.512	0.773	0.034	0.065	0.593	0.927	0.202	0.332	0.776	0.963	0.574	0.719	**0.996**	**1.000**	**0.991**	**0.996**
NELA																				
Wikipedia	**0.956**	**0.967**	**0.943**	**0.955**	0.769	0.731	0.852	0.787	0.760	0.709	0.882	0.786	0.771	0.691	**0.982**	0.811	0.737	0.664	0.959	0.785
WikiHow	0.654	0.611	0.844	0.709	**0.956**	**0.960**	**0.952**	**0.956**	0.690	0.928	0.412	0.571	0.786	0.850	0.694	0.764	0.885	0.962	0.802	0.875
Reddit	0.875	0.887	0.859	0.873	0.545	0.737	0.140	0.235	**0.931**	0.901	**0.968**	**0.933**	0.783	0.702	**0.982**	0.819	0.906	0.843	**0.997**	0.913
arXiv	0.739	0.755	0.709	0.731	0.637	0.627	0.678	0.651	0.692	0.866	0.454	0.596	**0.972**	**0.970**	0.974	**0.972**	0.847	0.922	0.759	0.833
PeerRead	0.605	0.635	0.493	0.555	0.535	0.830	0.088	0.159	0.585	**1.000**	0.170	0.291	0.840	0.881	0.786	0.831	**0.984**	**0.994**	0.974	**0.984**

Table 2: **Same-generator, cross-domain experiments: train on a single domain of ChatGPT vs Human and test across domains.** Evaluation accuracy (Acc), precision (Prec), recall and F1 scores(%) with respect to machine generations across four detectors.

Test → / Train ↓	Wikipedia Acc	Wikipedia Prec	Wikipedia Recall	Wikipedia F1	WikiHow Acc	WikiHow Prec	WikiHow Recall	WikiHow F1	Reddit ELI5 Acc	Reddit ELI5 Prec	Reddit ELI5 Recall	Reddit ELI5 F1	arXiv Acc	arXiv Prec	arXiv Recall	arXiv F1	PeerRead Acc	PeerRead Prec	PeerRead Recall	PeerRead F1
RoBERTa																				
Wikipedia	**0.996**	**0.994**	0.998	**0.996**	0.478	0.176	0.012	0.022	0.490	0.083	0.002	0.004	0.748	0.925	0.540	0.682	0.567	0.000	0.000	0.000
WikiHow	0.464	0.480	0.874	0.620	**0.994**	0.990	**0.998**	**0.994**	0.586	0.547	0.998	0.707	0.950	0.950	0.950	0.950	0.317	0.262	**1.000**	0.416
Reddit ELI5	0.428	0.424	0.402	0.413	0.881	0.879	0.884	0.881	**0.936**	0.887	**1.000**	**0.940**	0.524	**1.000**	0.048	0.092	0.912	0.749	0.962	0.842
arXiv	0.555	0.529	**1.000**	0.692	0.553	0.529	0.968	0.684	0.544	0.523	0.998	0.686	**0.994**	0.998	**0.990**	**0.994**	0.263	0.248	**1.000**	0.397
PeerRead	0.516	0.944	0.034	0.066	0.502	**1.000**	0.004	0.008	0.519	**1.000**	0.038	0.073	0.533	**1.000**	0.066	0.124	**0.987**	**0.947**	**1.000**	**0.973**
LR-GLTR																				
Wikipedia	**0.903**	**0.893**	0.916	**0.904**	0.735	0.683	**0.876**	0.768	0.682	0.613	**0.990**	0.757	0.715	0.852	0.520	0.646	0.727	0.647	0.998	0.785
WikiHow	0.882	0.839	**0.946**	0.889	**0.796**	**0.774**	0.836	**0.804**	0.777	0.695	0.988	0.816	0.726	0.849	0.550	0.667	0.760	0.676	**1.000**	0.806
Reddit ELI5	0.867	0.835	0.914	0.873	0.760	0.727	0.832	0.776	**0.885**	0.829	0.970	**0.894**	0.534	**0.905**	0.076	0.140	0.902	0.844	0.986	0.910
arXiv	0.471	0.061	0.004	0.008	0.502	0.529	0.036	0.067	0.451	0.344	0.108	0.164	**0.852**	0.845	**0.862**	**0.853**	0.712	0.639	0.972	0.771
PeerRead	0.845	0.832	0.864	0.848	0.735	0.730	0.746	0.738	0.863	**0.858**	0.870	0.864	0.502	0.625	0.010	0.020	**0.946**	**0.996**	0.896	**0.943**
Stylistic																				
Wikipedia	**0.965**	**0.962**	**0.968**	**0.965**	0.666	0.695	0.592	0.639	0.670	0.680	0.642	0.660	0.767	0.918	0.586	0.716	0.795	0.766	0.849	0.806
WikiHow	0.633	0.583	0.930	0.717	**0.939**	**0.945**	**0.932**	**0.939**	0.654	0.625	0.772	0.691	0.578	0.549	0.874	0.674	0.730	0.651	0.988	0.785
Reddit ELI5	0.806	0.836	0.762	0.797	0.640	0.716	0.464	0.563	**0.920**	**0.886**	**0.964**	**0.923**	0.561	0.670	0.240	0.353	0.779	0.802	0.741	0.770
arXiv	0.635	0.811	0.352	0.491	0.491	0.467	0.128	0.201	0.598	0.634	0.464	0.536	**0.974**	**0.972**	**0.976**	**0.974**	0.897	0.835	0.988	0.905
PeerRead	0.607	0.636	0.500	0.560	0.494	0.417	0.030	0.056	0.550	0.708	0.170	0.274	0.663	0.767	0.468	0.581	**0.993**	**0.991**	**0.994**	**0.993**
NELA																				
Wikipedia	**0.925**	**0.931**	**0.918**	**0.924**	0.701	0.638	**0.928**	0.756	0.720	0.664	0.892	0.761	0.472	0.468	0.416	0.441	0.600	0.580	0.727	0.645
WikiHow	0.682	0.641	0.828	0.723	**0.895**	**0.902**	0.886	**0.894**	0.811	0.869	0.732	0.795	0.508	0.509	0.442	0.473	0.826	0.780	0.907	0.839
Reddit ELI5	0.800	0.835	0.748	0.789	0.706	0.899	0.464	0.612	**0.932**	0.911	**0.958**	**0.934**	0.425	0.387	0.256	0.308	0.863	0.836	0.904	0.869
arXiv	0.485	0.059	0.002	0.004	0.510	0.692	0.036	0.068	0.459	0.044	0.004	0.007	**0.885**	**0.889**	**0.880**	**0.884**	0.763	0.882	0.608	0.719
PeerRead	0.480	0.292	0.028	0.051	0.503	0.600	0.018	0.035	0.520	**0.955**	0.042	0.080	0.562	0.645	0.276	0.387	**0.978**	**0.997**	**0.959**	**0.978**

Table 3: **Same-generator, cross-domain experiments: train on a single domain of davinci-003 vs Human and test across domains.** Evaluation accuracy (Acc), precision (Prec), recall and F1 scores(%) with respect to machine generations across four detectors.

Reddit ELI5 dataset (training on arXiv obtains the best performance). In out-of-domain detection, we find that training on Reddit ELI5 achieves better scores, leading to an accuracy greater than 0.79 for all domains. Training on Wikipedia leads to the worst out-of-domain accuracy.

LR-GLTR Accuracy and F1 score show a similar trend that they consistently perform the best when training and test data are from the same domain, while precision and recall do not follow the same tendency, especially recall. The highest recall on the test set is always obtained when the detector is trained in another domain. The best recall on Wikipedia, arXiv and PeerRead are all obtained when the model is trained over arXiv. Wikihow and Reddit's highest recall are from detectors trained

on PeerRead and Wikipedia, respectively.

Stylistic Similar to two detection methods above, for the detector using stylistic features, best scores over four metrics are all obtained when training and testing within the same domain. However, out-of-domain performance is much worse, even closing to a random guess (accuracy is around 0.5), particularly when testing on WikiHow and Reddit ELI5.

NELA Despite NELA features are initiated for checking news article factuality, they perform robust for detecting machine-generated text, showing competitive accuracy and F1 score across in-domain evaluations. This may be attributed to that machine-generated text is prone to consist of hallucinations. Additionally, they are more robust in out-of-domain detection, compared with stylistic features.

5.1.2 GPT3.5-davinci-003

RoBERTa For text generated by *davinci-003*, the RoBERTa classifier shows a similar trend to ChatGPT generations: in-domain detection achieves accuracy that is greater than 0.93 for most domains. Reddit ELI5 is harder than other domains. Training on Reddit ELI5 leads to the best out-of-domain performance while training on arXiv leads to the worst. Moreover, we find that accuracy of Wikipedia when training using data from other domains is around 0.5, namely a random guess in out-of-domain detection for Wikipedia.

LR-GLTR Similar to results under ChatGPT, the highest accuracy and F1 score are achieved in the in-domain setting. Compared with detection results over ChatGPT-generated text, the logistic regression model with GLTR features shows lower accuracy, precision, recall and F1 score over *davinci-003* generations across five domains. This implies that for the LR-GLTR detector, patterns of ChatGPT generations are easier to learn. This may be attributed to the fact that ChatGPT is tuned better than *davinci-003*, and fits more into the GLTR assumption.

Stylistic Compared to text generated by ChatGPT, it is harder to distinguish human-written and machine-generated text from GPT3.5 *davinci-003* based on tylistic features. Specifically, when training text is from different domains, detection is mostly as poor as random guesses.

NELA Leveraging NELA features, it is challenging to detect text generated by *davinci-003* when the model has not seen in-domain data during training, especially when training on arXiv and PeerRead, recall over Wikipedia, WikiHow and Reddit are close to 0.0. This is possibly because the textual style of academic papers is much different from other three domains.

5.1.3 Take-away

RoBERTa performs the best under in-domain evaluation among four detectors, but the worst in the out-of-domain evaluation. This may result from overfitting specific domain data during training. For all detectors, in-domain results are consistently better than the out-of-domain ones.

Compared to the average performance on ChatGPT and *davinci-003* across domains, the accuracy on ChatGPT is higher than that on *davinci-003*. This indicates that ChatGPT may leave more distinctive signals in their generated text than *davinci-003*, allowing for the detector to learn or select suitable features to distinguish between machine-generated and human-written texts.

5.2 Same-Domain, Cross-Generator

Given a specific domain, we train the detector using the training data of one generator and evaluate the detector on the test set from the same generator and other generators. Results on the domain of arXiv are shown in Table 4 and Wikipedia in Table 5.

5.2.1 arXiv

RoBERTa In Table 4, it shows that fine-tuning on either ChatGPT, davinci, and Cohere achieve good performance when detecting texts generated by the other two generators. BLOOMz, on the other hand, is harder to be detected if trained on the output of ChatGPT, davinci, or Cohere. In addition, the RoBERTa classifier fine-tuned on BLOOMz is good at generalising to detect the output of all four generative models.

LR-GLTR Akin to the trend of cross-domain evaluation, training and testing within the same generator always show the best accuracy and F1 score. Performance drops significantly when training and test data are generated from different language models, because of different distributions between the outputs of different generators, while there is an exception appearing between ChatGPT and Cohere. In both arXiv and the following Wikipedia, training on Cohere, and accuracy in Cohere and ChatGPT

Test → / Train ↓	ChatGPT Acc	ChatGPT Prec	ChatGPT Recall	ChatGPT F1	davinci Acc	davinci Prec	davinci Recall	davinci F1	Cohere Acc	Cohere Prec	Cohere Recall	Cohere F1	BLOOMz Acc	BLOOMz Prec	BLOOMz Recall	BLOOMz F1
RoBERTa																
ChatGPT	**0.997**	**0.994**	**1.000**	**0.997**	**0.997**	0.994	**1.000**	**0.997**	0.994	**0.998**	0.990	0.994	0.777	**1.000**	0.554	0.713
davinci	0.996	0.992	**1.000**	0.996	0.995	0.992	0.998	0.995	0.994	**0.998**	0.990	0.994	0.814	0.997	0.630	0.772
Cohere	**0.997**	0.994	**1.000**	**0.997**	0.996	0.994	0.998	0.996	**0.996**	**0.998**	**0.994**	**0.996**	0.826	0.997	0.654	0.790
BLOOMz	0.993	0.988	0.998	0.993	0.993	**0.998**	**0.998**	0.993	0.990	0.988	0.992	0.990	**0.981**	0.988	**0.974**	**0.981**
LR-GLTR																
ChatGPT	0.963	**0.964**	0.962	0.963	0.653	0.901	0.344	0.498	0.969	**0.964**	0.974	0.969	0.655	0.906	0.346	0.501
davinci	0.812	0.839	0.772	0.804	**0.852**	0.845	**0.862**	**0.853**	0.785	0.829	0.718	0.77	0.737	0.808	0.622	0.703
Cohere	**0.968**	**0.964**	**0.972**	**0.968**	0.660	**0.904**	0.358	0.513	**0.970**	**0.964**	**0.976**	**0.970**	0.615	**0.881**	0.266	0.409
BLOOMz	0.892	0.877	0.912	0.894	0.712	0.808	0.556	0.659	0.795	0.849	0.718	0.778	**0.872**	0.872	**0.872**	**0.872**
Stylistic																
ChatGPT	**1.000**	**1.000**	**1.000**	**1.000**	0.710	**1.000**	0.420	0.592	0.877	**1.000**	0.754	0.860	0.624	**1.000**	0.248	0.397
davinci	0.973	0.974	0.972	0.973	**0.974**	0.972	**0.976**	**0.974**	0.828	0.963	0.682	0.799	0.871	0.967	0.768	0.856
Cohere	0.976	0.994	0.958	0.976	0.838	0.997	0.678	0.807	**0.988**	0.994	**0.982**	**0.988**	0.655	0.981	0.316	0.478
BLOOMz	0.634	0.953	0.282	0.435	0.760	0.974	0.534	0.690	0.555	0.899	0.124	0.218	**0.985**	0.986	**0.984**	**0.985**
NELA																
ChatGPT	**0.972**	**0.970**	**0.974**	**0.972**	0.520	0.692	0.072	0.130	0.642	0.913	0.314	0.467	0.488	0.167	0.006	0.012
davinci	0.483	0.412	0.080	0.134	**0.885**	**0.889**	**0.880**	**0.884**	0.458	0.208	0.030	0.052	0.730	0.834	0.574	0.680
Cohere	0.701	0.888	0.460	0.606	0.494	0.446	0.050	0.090	**0.939**	**0.942**	**0.936**	**0.939**	0.471	0.208	0.073	0.089
BLOOMz	0.486	0.111	0.004	0.008	0.555	0.816	0.142	0.242	0.487	0.158	0.006	0.012	**0.969**	**0.968**	**0.970**	**0.969**

Table 4: **Same-domain, cross-generator experiments: train and test on arXiv (single machine-text generator vs human).** Evaluation accuracy (Acc), precision (Prec), recall and F1 scores(%) with respect to the machine generations across four detectors.

Test → / Train ↓	ChatGPT Acc	ChatGPT Prec	ChatGPT Recall	ChatGPT F1	davinci Acc	davinci Prec	davinci Recall	davinci F1	Cohere Acc	Cohere Prec	Cohere Recall	Cohere F1	BLOOMz Acc	BLOOMz Prec	BLOOMz Recall	BLOOMz F1
RoBERTa																
ChatGPT	**1.000**	**1.000**	**1.000**	**1.000**	0.974	**1.000**	0.948	0.973	0.990	**1.000**	0.980	0.990	0.995	**1.000**	0.990	0.995
davinci	0.999	0.998	**1.000**	0.999	**0.993**	0.998	**0.988**	**0.993**	0.997	0.998	**0.996**	0.997	0.999	0.998	**1.000**	0.999
Cohere	**1.000**	**1.000**	**1.000**	**1.000**	0.977	**1.000**	0.954	0.976	**0.998**	**1.000**	0.996	**0.998**	**1.000**	**1.000**	**1.000**	**1.000**
BLOOMz	**1.000**	**1.000**	**1.000**	**1.000**	0.977	**1.000**	0.954	0.976	0.998	**1.000**	0.996	0.998	**1.000**	**1.000**	**1.000**	**1.000**
LR-GLTR																
ChatGPT	**0.974**	**0.976**	0.972	**0.974**	0.850	**0.968**	0.724	0.828	0.929	**0.978**	0.878	0.925	0.813	0.757	0.922	0.831
davinci	0.941	0.900	**0.992**	0.944	**0.903**	0.893	**0.916**	**0.904**	0.905	0.895	0.918	0.906	0.775	0.700	**0.962**	0.810
Cohere	0.965	0.951	0.980	0.966	0.850	0.938	0.750	0.833	**0.951**	0.952	**0.950**	**0.951**	0.756	0.713	0.856	0.778
BLOOMz	0.699	0.950	0.420	0.583	0.664	0.941	0.350	0.510	0.550	0.879	0.116	0.205	**0.910**	**0.894**	0.930	**0.912**
Stylistic																
ChatGPT	**0.974**	**0.976**	**0.972**	**0.974**	0.933	**0.974**	0.890	0.930	0.876	**0.977**	0.771	0.862	0.637	0.732	0.434	0.545
davinci	0.967	0.962	**0.972**	0.967	**0.965**	0.962	**0.968**	**0.965**	0.905	0.966	0.839	0.898	0.670	0.782	0.472	0.588
Cohere	0.904	0.957	0.846	0.898	0.822	0.947	0.682	0.793	**0.942**	0.935	**0.949**	**0.942**	0.698	0.734	0.623	0.674
BLOOMz	0.537	0.849	0.091	0.164	0.537	0.849	0.090	0.163	0.540	0.846	0.098	0.176	**0.952**	**0.940**	**0.966**	**0.953**
NELA																
ChatGPT	**0.956**	**0.967**	0.943	**0.955**	0.910	**0.962**	0.854	0.905	0.781	0.948	0.595	0.731	0.502	0.537	0.036	0.067
davinci	0.946	0.935	**0.960**	0.947	**0.925**	0.931	**0.918**	**0.924**	0.875	0.920	0.821	0.868	0.489	0.382	0.034	0.063
Cohere	0.800	0.916	0.661	0.768	0.748	0.900	0.558	0.689	**0.938**	**0.940**	**0.935**	**0.937**	0.472	0.143	0.011	0.021
BLOOMz	0.494	0.200	0.004	0.008	0.492	0.082	0.043	0.053	0.496	0.072	0.081	0.006	**0.960**	**0.959**	**0.961**	**0.960**

Table 5: **Same-domain, cross-generator experiments: train and test on Wikipedia (single machine-text generator vs human).** evaluation accuracy (Acc), precision (Prec), recall and F1 scores(%) with respect to machine generations across four detectors.

are comparable, and similar high accuracy occurs when training on ChatGPT and testing on Cohere. This somewhat suggests that ChatGPT and Cohere share some patterns in terms of GLTR.

Stylistic Table 4 shows that when the detector is trained on ChatGPT, precision across all generators are 1.0. This means that the model does not make false positive predictions. These results contradict the established trend that training and testing within the same generator always show the best performance of the model.

NELA Majority scores of the detector using NELA features are around and lower than 0.5. This indicate they are not suitable to be used detecting arXiv text, which is not surprising because the NELA features are specifically designed for news articles. While the performance when training and testing within the same generator still remains re-

markable, with accuracy being around 0.9.

5.2.2 Wikipedia

RoBERTa Unlike the arXiv dataset, Fine-tuning a RoBERTa classifier performs well on the Wikipedia dataset for almost all generators, even on texts generated by BLOOMz. It might be because that RoBERTa had been trained on Wikipedia in pre-training stage.

LR-GLTR BLOOMz shows the lowest cross-generator accuracy and F1 score. Specifically, it shows low recall (<0.5) when training on BLOOMz and testing on other generators. Low recall (<0.5) is also observed in the domain of arXiv when training on other generators and testing on BLOOMz. This means that there are many false negative examples, namely, many machine-generated texts are misclassified as human-written ones. These indicate that the distribution of BLOOMz outputs are very different from other generators.

Stylistic Akin to arXiv text, in Wikipedia text, training on ChatGPT-generated data obtains the best precision across generators except for BLOOMz.

NELA From the accuracy, precision, recall and F1 score by ChatGPT, davinci and Cohere generators, we find that the NELA features extracted from Wikipedia perform relatively well in comparison to the arXiv, while the BLOOMz generator shows as poor results as arXiv.

5.2.3 Take-away

Compared between the four detectors, RoBERTa performs the best in cross-generator experiments, and the accuracy of the other three detectors using more interpretable features is close to each other. For all detectors, the performance when training and testing data from the same language model is always better than the performance where training and test data are from different generators.

For all detectors in both arXiv and Wikipedia, results on BLOOMz test sets in cross-generator evaluations are mostly lower than those of other large language models. We speculate that generations from BLOOMz are significantly different from other language models.

Compared between arXiv and Wikipedia, the former is overall harder than the latter.

5.3 Zero-shot Evaluation: GPTZero

While for some domains, GPTZero cannot be considered as a zero-shot detector, as it was trained on some collection of human- or machine-generated texts, particular data it was trained on is unknown. Therefore, for some domains it can be considered zero-shot. The results of the GPTZero evaluation are presented in Table 6. The values for some pairs of generation models and datasets are missing due to the limited time constraints. From these results, we can see large discrepancy of results among different generators and domains. According to performance on ArXiv and WikiHow, GPTZero cannot reliably detect text written by BLOOMz. The recall of detecting machine-generated text is close to 0. Among all domains, ArXiv appears to be the most difficult for the detector: all scores for it are significantly lower than for other domains. We also note that similarly to BLOOMz, texts generated by GPT-3.5 for ArXiv are almost not detected. However, GPTZero demonstrates good performance for general purpose domains such as Wikipedia, Wikihow, and Reddit. The highest F1 scores are obtained for ChatGPT: 0.931 on Wikipedia and 0.916 on Reddit. We assume that this could be due to GPTZero was specifically trained for ChatGPT and these domains. Obtained results show that zero-shot detection for novel domains and generators might be a difficult task.

5.4 Multilingual Evaluation

We conducted a multilingual evaluation to assess the performance of XLM-R as a detection model across different languages and generative models. We evaluated XLM-R on detecting English, Chinese and Russian texts that were generated by two generative models: either ChatGPT or davinci-003. For the English set, we combined English datasets from different domains: Wikipedia, Wikihow, Reddit ELI5, arXiv and PeerRead. For the English test set, we merged 500 samples of machine-generated texts and 500 human texts from each domain. The distribution for the English validation set followed the same pattern, while the remaining samples were utilized for training purposes.

Similarly, we employed the same approach for the Chinese and Russian datasets. We used 500 samples of machine-generated texts and 500 samples of human-written texts for both the test set and the validation set in each language. The remaining samples were allocated for training the

	ArXiv			Reddit			WikiHow			Wikipedia		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
BLOOMz	1.000	0.004	0.008	–	–	–	0.000	0.000	0.000	0.833	0.020	0.039
ChatGPT	1.000	0.262	0.415	0.975	0.864	0.916	0.835	0.494	0.621	0.998	0.872	0.931
davinci-003	1.000	0.002	0.004	0.965	0.604	0.743	–	–	–	1.000	0.538	0.700
Cohere	1.000	0.186	0.314	–	–	–	–	–	–	1.000	0.690	0.817
Dolly v.2	–	–	–	–	–	–	–	–	–	1.000	0.294	0.454
FLAN-T5	1.000	0.330	0.496	–	–	–	–	–	–	–	–	–

Table 6: Evaluation results of the zero-shot detection with GPTZero. Metrics are given in respect to the class of machine-generated content.

Generator →		davinci-003						ChatGPT					
↓	Test Domain →	All domain (en)		Baike/Web QA (zh)		RuATD (ru)		All domain (en)		Baike/Web QA (zh)		RuATD (ru)	
	Train Domain ↓	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
davinci-003	All domains (en)	**0.962**	**0.963**	0.777	0.817	0.619	0.693	**0.963**	**0.964**	0.779	0.819	0.679	0.754
	Baike/Web QA (zh)	0.816	0.827	0.990	0.990	0.537	0.613	0.867	0.880	0.943	0.940	0.611	0.698
	RuATD (ru)	0.626	0.723	0.482	0.596	**0.956**	**0.957**	0.615	0.713	0.584	0.699	0.907	0.901
	All	0.960	0.961	**0.991**	**0.991**	0.936	0.938	0.961	0.962	**0.981**	**0.981**	**0.939**	**0.939**
ChatGPT	All domains (en)	0.902	0.894	**0.944**	**0.943**	0.585	0.573	**0.988**	**0.988**	0.975	0.975	0.784	0.812
	Baike/Web QA (zh)	0.753	0.782	0.853	0.828	0.562	0.640	0.793	0.824	**0.994**	**0.994**	0.616	0.700
	RuATD (ru)	0.524	0.675	0.814	0.780	**0.886**	0.876	0.530	0.680	0.974	0.974	**0.976**	**0.976**
	All	**0.908**	**0.903**	0.943	0.941	0.882	**0.877**	0.980	0.980	0.989	0.989	0.955	0.957

Table 7: **Cross-language experiments.** Accuracy (Acc) and F1 scores (for machine-generated class) based on XLM-R over test sets across different languages.

models. Additionally, we combined all the training data from the English, Chinese, and Russian sets and evaluated the performance of the detector on each respective test set to observe the impact of this combined training data on the model's performance. The final results of the evaluation are presented in Table 7.

The results indicate that XLM-R performs best when detecting texts in the same language as the training set, especially when those texts are generated by the same model. Moreover, the performance of XLM-R, which was trained on texts generated by davinci-003, is more robust in detecting texts generated by another model compared to XLM-R trained on texts generated by ChatGPT. For instance, the average decrease in accuracy for XLM-R trained on all davinci-003 texts between detecting texts generated by davinci-003 and ChatGPT is 0.002, whereas, for XLM-R trained on all ChatGPT texts, the decrease is 0.06. Furthermore, training XLM-R on all datasets enhances its average performance across all test sets. Also, it is observed that XLM-R encounters some challenges in detecting text on language when it has not been trained on it. For example, the detector is facing problems in detecting Russian text when it was not trained on it. However, in certain experiments, XLM-R demonstrates good results in detecting Chinese text even when trained on English data, and vice versa.

6 Conclusion and Future Work

We collected a large-scale multi-generator, multi-domain, and multi-lingual corpus for machine-generated text detection. We further experimented with this corpus performing a number of cross-domain, cross-generator, and zero-shot experiments across five detectors. We found that it is challenging for detectors to distinguish machine-generated from human-written text if the model has never seen such data during training, which is either from different domains or generated by different large language models. This mostly results in low recall (lots of false negative examples), i.e., the models are prone to classify machine-generated text as human-written due to their coherent and fluent format. These results show that the problem is far from solved and there is a lot of room for improvement.

All supervised detectors perform well under in-domain and same-generator evaluation. RoBERTa obtains the highest accuracy at the cost of the worse cross-domain performance. BLOOMz shows significantly-distinctive generative distributions from other large language models. We will collect more data from LLMs such as Dolly-v2, FlanT5 and LLaMA, and other languages such as Japanese, Bulgarian and German in the future work, towards multilingual, generalized and robust detection of machine-generated text.

References

Anton Bakhtin, Yuntian Deng, Sam Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. 2021. Residual energy-based models for text. *The Journal of Machine Learning Research*, 22(1):1840–1880.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Jesse Graham, Jonathan Haidt, Sena Koleva, Matt Motyl, Ravi Iyer, Sean Wojcik, and Peter Ditto. 2012. Moral foundations theory: The pragmatic validity of moral pluralism. *Advances in Experimental Social Psychology*, 47.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *CoRR*, abs/2301.07597.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. 2022. Protecting intellectual property of language generation apis with lexical watermark. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10758–10766. AAAI Press.

Benjamin D Horne, Jeppe Nørregaard, and Sibel Adali. 2019. Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–23.

Khalid Hussain, Nimra Mughal, Irfan Ali, Saif Hassan, and Sher Muhammad Daudpota. 2021. Urdu news dataset 1m.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

VA Ivanin, EL Artemova, TV Batura, VV Ivanov, VV Sarkisyan, EV Tutubalina, and IM Smurov. 2020. RuREbus-2020 shared task: Russian relation extraction for business. In *Computational Linguistics and Intellectual Technologies*, pages 416–431.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *CoRR*, abs/2301.10226.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. RankGen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 199–232, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jenny S. Li, John V. Monaco, Li-Chiou Chen, and Charles C. Tappert. 2014. Authorship authentication using short messages from social networking sites. In *11th IEEE International Conference on e-Business Engineering, ICEBE 2014, Guangzhou, China, November 5-7, 2014*, pages 314–319. IEEE Computer Society.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Michail Melnichenko and Natalia Tyshkevich. 2017. Prozhito from manuscript to corpus. *ISTORIYA*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023.

Detectgpt: Zero-shot machine-generated text detection using probability curvature. *CoRR*, abs/2301.11305.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning. *CoRR*, abs/2211.01786.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *CoRR*, abs/2303.11156.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Cherniavskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *CoRR*, abs/2206.01583.

Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning: "taiga" syntax tree corpus and parser. *Proceedings of the Corpora*, pages 78–84.

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N. Asokan. 2021. DAWN: dynamic adversarial watermarking of neural networks. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pages 4417–4425. ACM.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9051–9062.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023. Protecting language generation models via invisible watermarking. *CoRR*, abs/2302.03162.