

Check Me If You Can: Detecting ChatGPT-Generated Academic Writing using CheckGPT

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo
EECS/ITS, The University of Kansas, Lawrence, KS, USA
{zyliau, zyao, fli, bluo}@ku.edu

Abstract

With ChatGPT under the spotlight, utilizing large language models (LLMs) for academic writing has drawn a significant amount of discussions and concerns in the community. While substantial research efforts have been stimulated for detecting LLM-Generated Content (LLM-content), most of the attempts are still in the early stage of exploration. In this paper, we present a holistic investigation of detecting LLM-generated academic writing, by providing the dataset, evidence, and algorithms, in order to inspire more community effort to address the concern of LLM academic misuse. We first present GPABenchmark, a benchmarking dataset of 600,000 samples of human-written, GPT-written, GPT-completed, and GPT-polished abstracts of research papers in CS, physics, and humanities and social sciences (HSS). We show that existing open-source and commercial GPT detectors provide unsatisfactory performance on GPABenchmark, especially for GPT-polished text. Moreover, through a user study of 150+ participants, we show that it is highly challenging for human users, including experienced faculty members and researchers, to identify GPT-generated abstracts. We then present CheckGPT, a novel LLM-content detector consisting of a general representation module and an attentive-BiLSTM classification module, which is accurate, transferable, and interpretable. Experimental results show that CheckGPT achieves an average classification accuracy of 98% to 99% for the task-specific discipline-specific detectors and the unified detectors. CheckGPT is also highly transferable that, without tuning, it achieves $\sim 90\%$ accuracy in new domains, such as news articles, while a model tuned with approximately 2,000 samples in the target domain achieves $\sim 98\%$ accuracy. Finally, we demonstrate the explainability insights obtained from CheckGPT to reveal the key behaviors of how LLM generates texts.

1 Introduction

The recently debuted Large Language Model (LLM) - ChatGPT has shown an impressive ability to generate sophisticated texts with human-like language style and quality. While

LLMs/ChatGPT provide an efficient means to retrieve and summarize information, concerns have been raised that the LLM-generated content (LLM-content) can be misused to abuse the trust systems we have. For instance, one of the biggest inappropriate “use cases” in academia is using ChatGPT in cheating or plagiarism [44, 81]. Instead of working with originality, authors use LLMs to compose their articles and obtain interests in a dishonest way. ChatGPT may also be used in other trust systems for fraud and scam purposes, e.g., internet phishing and romance scams [31, 74].

LLM-content detection can be challenging due to the new characters of LLM/ChatGPT. First, like a human conversationalist, the output of LLM has a relevant, organized response with a low level of grammar errors. Second, the sampling mechanism of LLM output ensures that the choice of words is stochastic, therefore, the responses are distinct even with multiple repeated inquiries. Third, the misuse of LLM-content can be stealthy, since users can invoke ChatGPT to polish human writing. Existing plagiarism/LLM detectors perform poorly in detecting GPT-polished text (Section 3.2). Although more academic institutes and publishers have announced policies on the use of LLM-content, they are hard to enforce unless we have a tool to effectively detect LLM-content.

Although people have accumulated a variety of experiences in identifying LLM-content, a holistic view of how LLM-generated output can be distinguishable from human writing is still missing. For example, [33, 50] report that ChatGPT tends to generate output with more objectivity while human-written texts are more subjective; and the language used in ChatGPT is more formal, focused, and fluent. However, based on these qualitative characterizations of subtle language features, the detection performance is relatively poor. Our study shows that human evaluators only achieve $\sim 50\%$ detection accuracy with GPT-generated academic writings. It suggests that the features focusing on the language appearance may not be reliable in particular writing cases (e.g., scientific papers), as human-written research papers also demonstrate objectivity and formality. To bridge this gap, we believe that understanding how ChatGPT writes semantically (e.g., choosing words

and forming sentences) is necessary. In other words, only a *language-model-based detector* can identify a *language-model-based generator*.

To this end, in this paper, we aim to explore (1) the typical scenarios of using/misusing ChatGPT in academic writing, (2) the difficulty of detecting LLM-content in academic writing, and (3) the possibility of developing language-model-based detectors to accurately identify LLM-content. Specifically, we make the following efforts for each aspect of the study:

We identify three typical cases of using or abusing ChatGPT in academic writing, including composing, completing, and polishing. To further reflect the heterogeneous writing styles across disciplines, we pick three representative disciplines for investigation: computer science for technical/engineering writing, (2) physics for science writing, and (3) humanities and social sciences for art writing. Accordingly, we collect and share a dataset of 600,000 human-written and ChatGPT-generated academic abstracts, called *GPABenchmark*. To the best of our knowledge, GPABenchmark is the most comprehensive and publicly available ChatGPT-generated dataset of academic writing by far.

To validate the difficulty of detecting LLM-content in academic writing, we carry out both human- and algorithm-evaluation. We first conduct an extensive field study with human evaluators to assess if they can distinguish LLM-content accurately provided with a mixture of true and false samples. The cohort of 150+ evaluators consisting of university faculties, researchers, and graduate students, proves that the recognition of LLM-content is difficult for visual inspection based on language appearance, with or without individual experiences of writing research articles. In addition, we test multiple state-of-the-art algorithmic detectors on GPABenchmark, e.g., GPTZero, and show that they demonstrate modest to poor performance, especially with GPT-polished text.

Last, as the primary goal of this paper, we develop and evaluate a language-model-based detection framework, named CheckGPT, to explore the possibility of building automation tools for LLM-content detection. Specifically, our proposed approach has the following advantages: (1) without requiring white-box access to the LLM model, using deep learning framework with expressive GPABenchmark dataset, the proposed checkGPT achieves a high accuracy compared to human and state-of-the-art (SOTA) LLM-content detectors. (2) We adopt a model-agnostic setting that our model can be treated as a plugin to most of the pre-trained language models (e.g., BERT), as a result, the number of parameters to be trained can be largely reduced. With light training burden, CheckGPT provides convenience in knowledge update and software deployment. (3) Due to the ability to learn generalized semantic patterns as watermarks of LLM-content, our proposed model demonstrates a good potential for domain transfer. With minimum fine-tuning efforts, our model can quickly pick up the ability to detect LLM-content for new disciplines and new domains. Finally, we conduct comprehen-

sive experiments to support all the design goals and strengths of CheckGPT. In summary, our main contributions are:

- We present a publicly available GPT-generated academic writing dataset - GPABenchmark, a cross-disciplinary corpus consisting of human-written, GPT-written, GPT-completed, and GPT-polished research paper abstracts. GPABenchmark has the potential to be a cornerstone for benchmarking GPT detectors in academia, and a valuable resource to assist the design of new detecting methods.
- We evaluate the SOTA open-source and commercial GPT detectors and show that they provide unsatisfactory performance in academic writing. Meanwhile, with a user study of 150+ participants, we show that even experienced faculty/researchers are unable to distinguish between human-written and GPT-generated academic writing.
- We present CheckGPT, a GPT-generated content detector - a deep learning based and model-agnostic framework with validated benefits of affordability, transferability, and interpretability. We demonstrate the outstanding performance (>98% average accuracy) of CheckGPT on GPABenchmark with extensive experiments. We share CheckGPT at <https://anonymous.4open.science/r/CheckGPT-80B2>.

Ethical Considerations. The user study in Section 4 was reviewed and approved by the Human Research Protection Program at the University of Kansas. All the research paper abstracts collected in Section 3 are open to the public. We invoked ChatGPT’s API (with payment) to collect the GPT-generated abstracts. The GPABenchmark dataset and the CheckGPT tool will be shared with the community.

The academic community is actively discussing how AI writing assistance tools may pose potential challenges to research and education [4, 57, 76]. OpenAI also posted their perspectives on the education-related risks and opportunities [65]. In this paper, we provide a detection tool for LLM-Content. The impact of ChatGPT and other AI writing assistance tools on academic integrity is outside of the scope of the paper.

The rest of the paper is organized as follows: we introduce the background of LLM and survey the related literature in Section 2. We introduce the GPABenchmark dataset and evaluate the open-source and commercial ChatGPT detectors in Section 3, followed by a user study of LLM-content detection in Section 4. We present the technical details of CheckGPT in Section 5 and experimental results in Section 6. Finally, we discuss the prompt engineering and model interpretation issues and conclude the paper in Sections 7 and 8.

2 Background and Related Work

2.1 Large Language Models (LLMs)

LLMs refer to the language models in natural language processing (NLP) trained on massive amounts of data with deep learning frameworks consisting of the ultra-large amount of

parameters. Aiming at modeling the sequences of tokens in human-written texts as unprecedentedly sophisticated probability distributions, LLMs are capable of generating highly sophisticated language outputs which can achieve human-like language style and quality. The most notable examples of LLMs are the OpenAI GPT (Generative Pre-trained Transformer) series models, including ChatGPT, an advanced GPT-3 model trained on 175 billion parameters. One of the most prominent characteristics of LLMs is their significant rise in performance brought by the scaling effect, which can not be observed in small models. For example, work [99] summarizes the emergent abilities of LLMs in three aspects: (1) in-context learning, (2) instruction following, and (3) step-by-step reasoning. In work [93], emergent abilities are validated in major LLMs (e.g., LaMDA [87], GPT-3 [11], Gopher [72]) through a series of few-shot prompted tasks [11], where the performance of accuracy jumps sharply from close-to-zero at under- 10^{22} FLOPs¹ models to 40% at 10^{23} FLOPs model.

ChatGPT is built on top of OpenAI’s GPT-3.5 with fine-tuning through both supervised and reinforcement learning techniques. Benefited from the large-scale autoregressive pre-training based on transformer networks and comprehensive fine-tuning based on reinforcement learning from human feedback (RLHF), ChatGPT is proven to mimic a versatile human conversationalist and succeed in many writing generation tasks, such as creating student essays, legal documents, business pitches, poetry/song lyrics, and programming codes.

2.2 Detection of LLM-Generated Texts

The concern of LLM/ChatGPT misuse has been raised widely in academia because (1) academic integrity violations such as cheating and plagiarism will become *easy-to-conduct* and *hard-to-detect*. With the generative character of GPT, the texts for the same prompt can be very different at each time, but all can present human-like patterns and styles. The GPT-generated outputs with effortless clicks are most likely to pass existing plagiarism checkers predefined by hand-craft rules (e.g., exact match scanning). (2) False and redundant information may flood the publication systems. Although ChatGPT is good at mimicking a human writer and showing plenty of details, the facts in its output can frequently be wrong, especially in STEM subjects. For example, Stack Overflow had to ban LLM-generated posts to ensure that visitors are able to find reliable answers efficiently [80]. With the same consideration, academic conferences started to ban LLM-generated texts (e.g., ICML). Rules are also enforced to clearly state LLM usages in acknowledgment (e.g., Nature and RSC).

The detection of LLM-generated texts can be categorized into white-box and black-box approaches [85]. Requiring full access to the target LLM, white-box approaches implant watermarks into LLM outputs and detect the watermarks to

¹Floating point operations per second (FLOPs) measures the scale of model parameters.

identify machine-generated texts. However, the owners of LLMs are increasingly reluctant to open-source their models, black-box approaches that only gather the output of LLMs have attracted more interest. They can be further categorized into (1) feature-based by examining hand-crafted statistical disparities, linguistic patterns, and fact verification [85]; and (2) model-based by learning another language model, which is good at discriminating linguistic characteristics between human-written and machine-generated texts. Our approach falls into the second category.

2.3 Related Work

Neural Language Model. The neural networks for word probability modeling have been developed since 2000s [7, 15, 84]. The recurrent neural network (RNN) family [7] aggregates the historical contextual information in text and uses the memory of history to predict the next words. Later, word embedding, which aims to learn a low-dimensional distributed representation, was proved to be effective by modeling the context distributions through shallow neural networks, e.g., word2vec [60, 61] and GLOVE [68] have been shown to greatly improve the performance of NLP tasks. Recently, pre-trained language models have been widely used because of the general but effective word representation, which significantly improves the performance on downstream tasks through fine-tuning. For example, based on the self-attention Transformer networks [91], BERT [20] is pre-trained on large-scale corpus by predicting randomly masked words, and recognizing the correct order between two sentences. With the advantage of paralleling computation and memorizing long sequences, more Transformer-based models were developed, such as RoBERTa [51], ELMo [69], GPT-2 [71], and BART [47].

LLM-Content Generation and Detection. Prior works [29, 46, 62, 86, 98] study LLM-content detection for models before ChatGPT, e.g., Grover, GPT-2, and GPT-3. Recently, [27] evaluated 50 ChatGPT-written biomedical research abstracts with human reviewers and a RoBERTa-based classifier. Their findings show that 34% of the abstracts received scores <50% from the classifier, i.e., they are labeled as likely human-written, while four human reviewers correctly identified 68% of the GPT-written abstracts. [10] trained a transformer-based deep learning model to distinguish between AI-generated and human-written essays in a range of different education levels. [33] and [50] conduct comprehensive studies, including human evaluators. [33] analyzes a series of question-answering datasets in both English and Chinese, and [50] targets the essays written by students and English learners. [35] establishes the first machine-generated text benchmark evaluating a number of detection approaches. The detailed comparisons of current GPT detectors are listed in Table 1. Compared with the other approaches, CheckGPT collects and uses a significantly larger dataset, uses a model-agnostic design for higher affordability, upgradability, and flexibility, achieves

very high accuracy, transferability, and interpretability.

Security and Ethics in AIGC Application. AI-generated content (AIGC) has been used in adversarial activities even before LLMs were introduced [23], while the recent release of ChatGPT may have provided the malicious actors with a powerful tool [19, 73]. In particular, the detection of AI-backed social bots, spam, scams, and misinformation has been extensively studied in the literature, e.g., [16, 54, 78, 96], while the rise of LLMs and ChatGPT introduces both new opportunities [21, 28, 36, 37, 40, 92] and challenges [18, 30, 58]. For instance, ChatGPT may be used in scamming or phishing [31, 34, 74] as well as in the defense [13]. While Open AI has enforced internal mechanisms to prohibit the unethical use of ChatGPT, the restrictions could be evaded through prompt engineering (jailbreaking) [48, 49]. Finally, there are also discussions and concerns with ChatGPT’s potential impact on education and research [24, 81, 95], especially on authorship and plagiarism [4, 25, 44, 82]. A small-scale experiment in [27] showed that most of the GPT-generated abstracts were deemed as completely original by a web-based plagiarism detector (<https://plagiarismdetector.net/>).

3 GPABenchmark: GPT Corpus for Academia

3.1 The GPABenchmark Dataset

The state-of-the-art corpora for ChatGPT text classification mainly focus on question-and-answer (Q&A) dialogues [33, 35]. While the Q&A datasets align with the original design goal of ChatGPT as an interactive “Chat” interface, they become insufficient as the usage scenarios of ChatGPT have significantly expanded beyond chat. When ChatGPT is adopted in academic writing, such as quizzes, essays, reports, and even research papers [24, 81], it generates text that is objective, formal, fluent, and focused [33, 50], which is akin to academic writing style, and thus poses challenges to the detectors. In particular: (1) Human-generated conversations often contain subjective opinions, personal biases, and emotional cues. However, such cues are significantly less observed in academic writing, which is generally formal and objective [8, 9, 33, 42]. (2) Grammatical errors and inconsistencies in human-generated texts may serve as meaningful indicators, however, they are less likely to occur in academic writing, which is expected to meet higher standards for fluency, clarity, and grammatical correctness [38, 56]. Also, academic writers typically adopt a comprehensive and organized style [14] akin to the one generated by ChatGPT [33]. (3) Academic abstracts typically delve into domain-specific and highly-specialized topics [42], which lead to a significantly different term distribution from conversational dialogues.

With the unique characteristics of academic writing, a new ChatGPT-generated corpus is necessary in benchmarking GPT detectors and in assisting the design of new detectors. In this paper, we introduce the first large-scale GPT-generated

text corpus for academic writing, namely the *GPABenchmark*. We define three tasks based on the most representative scenarios where LLMs are used/misused in academic writing: users provide a title for text generation, provide a partial draft for completion, or provide a draft for polishing.

- **Task 1. GPT-written full abstracts (GPT-WRI or WRI).** The author gives a title to ChatGPT and asks it to write an abstract from scratch. A sample prompt is: *Please generate an abstract for a research paper titled “Attitude of the Society Towards People with Visual Impairment.”*
- **Task 2. GPT-completed abstracts (GPT-CPL or CPL).** One function that was often advertised for ChatGPT is text completion. When the author provides a few sentences to ChatGPT, it follows the logic in the seed text to complete the rest of the paragraph. We mimic this scenario as follows: for each abstract with s sentences, we provide its first half ($s/2$ sentences) to ChatGPT and ask it to *complete the abstract with w words*, where w is the number of words in the second half of the original abstract. In this way, the generated abstract will have *approximately* the same length (number of words) as the human-written abstract.
- **Task 3. GPT-polished abstracts (GPT-POL or POL).** We provide the entire abstract to ChatGPT for polishing. We adopt a popular prompt from the ChatGPT users’ community: *“This is an abstract of a research paper. Please rewrite for clarity.”* ChatGPT re-writes the text sentence-by-sentence and generates a polished abstract that is usually shorter than the original. Invoking ChatGPT multiple times will generate different results for the same seed abstract.

In the rest of the paper, we will use the term *human-written* abstracts (HUM) to denote abstracts that are completely written by human authors. We use *GPT-Generated* abstract (GPT-GEN) to denote a superset of all three categories of GPT-content as described above: GPT-WRI, -CPL, and -POL.

We collect the published research papers from three disciplines: (1) computer science (CS), (2) physics (PHX), and (3) humanities and social sciences (HSS), which include four typical “soft science” fields: history, philosophy, sociology, and psychology. The ground truth data are collected from arXiv for CS and physics, and from Springer’s SSRN for HSS. We chose these three fields that spread across the “hard science” (math-intensive) and “soft science” disciplines. We avoided mathematics as we observed that their publications often have very short abstracts that may not provide sufficient information to ChatGPT or to the detector. For CS and physics, we only used papers that were posted on or before 2021, to ensure that they were all human-written, as researchers may have adopted GPT-3 to assist their writings before the web-based ChatGPT was released². For each paper, we store an identifier (ID), a title (T), and an abstract (ABS). In about three weeks, we collected 50,000 data samples for each discipline.

²GPT-3 was first released in 06/2020, access to the test release was by invitation-only until 11/2021, when the API was made publicly accessible.

Table 1: Summary of SOTA LLM-content Detectors. Tool: used/evaluated online detection tools. Open: open-sourced.

Study	Approach				Transfer-ability	# Human Evaluators	Domain			Dataset		
	Tool	ML/Stat	Human	Train DL			News	QA	Essay	Research	Size	Open
Grover or GPT-2&3	[29]		●			—		●		●	300	
	[46]		●		●	—		●			90k	●
	[62]	●				—		●	●		—	
	[86]		●		●	—				●	28k	●
	[98]			●	●		*	●			20k	●
ChatGPT	[10]				●	—			●		100k	
	[27]	●		●		—	2			●	100	
	[33]		●	●	●		17	●			125k	●
	[50]	●	●	●	●		43		●		8k	●
	Ours	●	●	●	●	●	155	●		●	>600k	●

* The number of human evaluators is not explicitly provided.

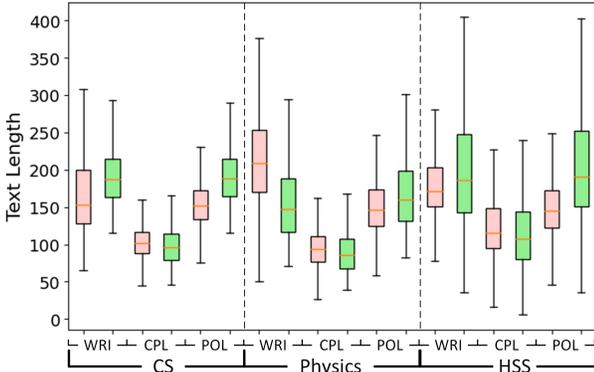


Figure 1: Distribution of abstract lengths (# of words): red: ChatGPT-generated; green: human-written.

We then invoke ChatGPT to generate abstracts in three tasks defined above. ChatGPT is based on OpenAI’s GPT-3.5 family of LLMs, among which gpt-3.5-turbo is considered the most capable and the most updated model. We invoked gpt-3.5-turbo through OpenAI’s API. For each paper in the dataset, we used the prompts described in each task to produce GPT-written, completed, and polished abstracts at the cost of 0.2 cents per 1,000 tokens. Although OpenAI has set a high query rate limit for paid users, it still takes up to several seconds to generate each abstract due to network delay and ChatGPT server overload. In about eight weeks, we generated 150,000 samples in each of the three categories: GPT-written, GPT-completed, and GPT-polished abstracts.

The average lengths of GPT-generated abstracts are 187.3, 109.6, and 152.6 words for Tasks 1, 2, and 3, respectively. The average lengths of human-written abstracts are 183.6, 103.6, and 189.6 words. Note that we only consider the second half of the abstracts in Task 2. The distributions of abstract lengths are shown in Figure 6, where we have some interesting observations: (1) human-written abstracts in physics are shorter than the other disciplines, while GPT-written abstracts in physics are longer than the other disciplines; (2) GPT-written abstracts are longer than human-written abstracts in physics but shorter in CS and HSS; (3) in all three disciplines, GPT-polished abstracts are shorter than human-written abstracts.

3.2 Benchmarking Open-Source and Commercial ChatGPT Detectors

With the GPABenchmark dataset, we evaluate the detection accuracy of three open-source and commercial ChatGPT detectors that are available over the Internet: GPTZero [88], ZeroGPT [1], and OpenAI’s classifier [64]. We are unable to run large-scale experiments due to (1) they do not provide any API, hence, we need to use web scraping in the evaluation and we enforce politeness in web scraping; (2) some of them are slow or enforce rate limits that are low; and (3) some of them charge a fee for the inquiries. For each task (GPT-WRI, -CPL, and -POL), we randomly sample 300 pairs of human-written abstracts and the corresponding GPT-generated abstracts in each discipline (2,400 pairs in total). We feed these abstracts to each detector and summarize their performance as follows. Note that, in Task 2 (GPT-completed abstracts), we only submitted the second half of each abstract to the detectors.

GPTZero [88]. For each text paragraph, GPTZero reports a binary decision of “human-written” or “GPT-generated”. As shown in Table 2 (a), GPTZero demonstrates very high accuracy with human-written abstracts with an average accuracy of 98.1% across all the topics. However, its detection accuracy for GPT-generated abstracts appears to be very low, with an average accuracy of 24.3%. That is, GPTZero has a very strong tendency to classify an input abstract as “human-written”. From Task 1 to Task 3, the detection performance decreases significantly (from 42.5% to 8.1%). That is, when more information is given to ChatGPT, the generated text appears to be more “human-like” in the eyes of GPTZero.

ZeroGPT [1]. For each input text snippet, ZeroGPT reports a decision from nine different labels (Appendix A.1). We map them to integer scores in the range of [0, 8], where 0 indicates human-written and 8 indicates AI/GPT-generated. We also use a threshold of 4.0 on each score to generate a binary decision of “human-written” and “GPT-generated” for each test (please refer to Appendix A.1 for more discussions on this decision threshold). We present ZeroGPT’s average detection accuracy for each task and each discipline in Table 2 (b1), and the average score for each experiment in Table 2 (b2). ZeroGPT’s detection accuracy for human-written abstracts is

Table 2: Performance of open-source and commercial GPT detectors. Values in red: detection accuracy <50%, or average score on the wrong side of the decision threshold.

	T1. GPT-WRI			T2. GPT-CPL			T3. GPT-POL		
	CS	PHX	HSS	CS	PHX	HSS	CS	PHX	HSS
(a) Classification accuracy (in %) of GPTZero.									
GPT	30.3	25.3	72.0	17.0	6.0	43.7	1.7	2.3	20.3
Human	99.3	99.7	100	99.7	99.7	94.3	99.7	95.7	95.7
(b1) Detection accuracy (in %) of ZeroGPT									
GPT	67.4	68.4	92.3	25.3	10	62.4	3.3	2.7	24.7
Human	100	98.4	95	99.7	99.7	94.7	98.3	98.6	92.7
(b2) Average score reported by ZeroGPT. 0:human, 8:GPT									
GPT	5.43	5.39	7.41	2.26	0.97	4.97	0.35	0.29	2.15
Human	0.09	0.13	0.52	0.08	0.04	0.47	0.20	0.14	0.64
(c.1) Detection accuracy (in %) of OpenAI’s detector									
GPT	80.7	70	63	63.7	23.7	27.3	6.3	4.3	6
Human	51.0	69.7	84.0	35.3	59.7	79.6	50.7	69.0	88.0
(c.2) Average score reported by OpenAI. 0:human, 4:GPT									
GPT	3.11	2.89	2.72	2.70	2.12	2.04	1.75	1.59	1.52
Human	1.42	1.17	0.59	1.71	1.35	0.68	1.38	1.14	0.52

close to 100% in CS and physics, and slightly lower (~95%) in humanities and social sciences (HSS). Its accuracy with fully GPT-written abstracts is also high, especially for HSS (92.3%). However, the detection accuracy for GPT-completed and GPT-polished abstracts in CS and physics appears to be very low (in the range of [5%, 25.3%]), while the accuracy for HSS appears to be relatively higher. While ZeroGPT claims a detection accuracy of 98%, it appears to be less effective in academic writing. Similar to GPTZero, ZeroGPT also has a tendency to classify GPT-generated text as human-written.

OpenAI’s Classifier [64]. For each input text snippet, OpenAI’s own classifier generates a decision out of five classes that are mapped to integer scores in [0, 4], where 0 indicates “very unlikely AI-generated”, 2 means “unclear if it is AI-generated”, and 4 indicates “likely AI-generated” (Appendix A.2). We use a threshold of 2 to generate a binary decision for each test. Note that a classification of “unclear if it is AI-generated” (2) is considered wrong for both GPT-generated and human-written inputs. We present OpenAI’s classification accuracy in Table 2 (c1) and the average scores in Table 2 (c2). OpenAI’s classifier shows slightly different patterns from GPTZero and ZeroGPT. It demonstrates moderate performance in classifying abstracts that are fully written by humans or GPT. However, its accuracy for GPT-completed and GPT-polished abstracts appears inadequate (but slightly better than GPTZero and ZeroGPT). We also noticed that this classifier is very sensitive to the length of text. While it requires a minimum of 1,000 characters for each input text snippet, a shorter input (e.g., input in Task 2 GPT-CPL) is more likely to yield a wrong or “unclear” decision.

4 User Study: Identification of Human-Written and GPT-Generated Abstracts

With all the news reports and online/informal discussions that human users are unable to distinguish ChatGPT-generated text from man-written text, we investigate this problem through a user study in a relatively well-defined domain – the research publication. We aim to answer three research questions: (1) Could (experienced) researchers distinguish between human-written and GPT-written/polished research papers/abstracts? (2) Does prior experiences with reading and writing research papers contribute to the capability of identifying GPT-generated papers? (3) Does the researchers’ capability in identifying GPT-generated text vary by discipline?

We designed a questionnaire as follows³: On the landing page, an IRB information statement is displayed to the participants, who will then select their “most familiar discipline” among CS, Physics, and Humanities & Social Sciences (HSS). The main questionnaire page first asks the participants to provide basic background information: role (faculty, researcher, or student), whether they have published research papers (yes or no), and their self-claimed familiarity with research papers (expert, knowledgeable, somewhat familiar, or no familiarity). Our suggested rubric for “expert” is “have published 10+ papers OR read 100+ papers”. Three abstracts are then displayed to the user, who is asked to annotate each as “human-written” or “GPT-generated/polished”. Each question is randomly sampled from human-written or GPT-generated abstracts from Task 1 and Task 3. For abstracts in Task 3, we display the following hint: “This abstract was completely written by humans OR written by humans and then polished by ChatGPT.”

We distributed questionnaires to faculty members, researchers, and graduate students in the Department of EECS, Department of Physics, and College of Liberal Arts at our University. Physics faculty members also shared the questionnaire with collaborators in a research organization in Europe. In approximately four weeks, we received 155 responses with 465 annotated abstracts. The overall accuracy, defined as the proportion of correctly identified abstracts out of all abstracts, was 48.82%, which is slightly worse than random guesses. The detailed statistics of the responses are shown in Table 3. From the responses, we have the following observations:

- It is extremely challenging for human users to distinguish between human-written and GPT-generated paper abstracts. Only 21 users correctly identified all three abstracts. If all participants were making random selections, 19.38 users would have scored 3 correct selections. That is, the top performers are only slightly better than random guesses.
- Participants have the tendency to annotate all abstracts as “human-written”. 57.33% of human-written abstracts were correctly labeled as “human-written”, while 59.66% of GPT-

³This user study was reviewed and approved by the Human Research Protection Program at the University of Kansas (STUDY00150100).

Table 3: Detailed results of the user study to identify GPT-generated paper abstracts: Pat.: number of participants; Abs.: number of annotated abstracts; Cor.: number of correct annotations; Acc.: accuracy; GPT: accuracy for GPT-generated abstracts; Man: accuracy for human-written abstracts.

Category	Par.	Abs.	Cor.	Acc.	Man	GPT
Role						
Faculty	44	132	65	49.2%	58.6%	41.9%
Researchers	30	90	45	50.0%	58.2%	37.1%
Students	81	243	117	48.1%	56.3%	40.3%
Discipline						
CS	57	171	86	50.3%	59.0%	43.0%
Physics	48	144	77	53.5%	65.1%	37.7%
HSS	50	150	64	42.7%	46.5%	39.2%
Self-claimed Familiarity with Research Papers						
Expert	52	156	80	51.3%	60.6%	43.5%
Knowledgeable	56	168	80	47.6%	57.3%	34.7%
Somewhat	39	117	57	48.7%	56.0%	43.3%
No familiarity	8	24	10	41.7%	46.7%	33.3%
Published papers?						
Yes	106	318	155	48.7%	58.1%	39.2%
No	49	147	72	49.0%	55.6%	42.7%

generated abstracts were mistakenly labeled as “human-written”. The result confirms the public opinion that ChatGPT achieves human-like language style and quality.

- Users are better at identifying fully GPT-written abstracts with an accuracy of 43.81%, while they perform worse with GPT-polished abstracts with an accuracy of 37.5%. In both cases, the accuracy is still lower than random guesses.
- Users’ self-claimed expertise appears to slightly affect their capability to identify human-written and GPT-generated abstracts. For example, participants with “No familiarity” with papers performed worse than the others. However, most of the differences are *not* statistically significant.
- Users are better at identifying GPT-generated abstracts in physics. They are significantly worse at identifying GPT-generated abstracts in humanity and social sciences.

5 CheckGPT: An Accurate Detector for ChatGPT-generated Academic Writing

5.1 The System Model and Assumptions

Our objective is to build a classifier, CheckGPT, to determine whether a given text is generated by ChatGPT. We denote our classifier as \mathcal{H} , and the classification problem can be formulated as:

$$\hat{y} = \mathcal{H}(s) \quad (1)$$

$$\operatorname{argmin}_{\theta} \mathcal{L}(y, \hat{y}) \quad (2)$$

where s represents an unstructured text snippet (i.e., paper abstract). Given s , $\mathcal{H}(s)$ outputs the probability distribution \hat{y} considering label space $\{‘h’, ‘g’\}$, where ‘h’ indicates human-written text and ‘g’ indicates ChatGPT-generated text. The goal is to find an optimal set of parameter θ for classifier \mathcal{H} , so that the loss function \mathcal{L} measuring the distance between prediction \hat{y} and observation y is minimized.

In his paper, we consider a *black-box defender*, who only has access to the observed samples. However, she has no insider knowledge of the LLM which generates these samples, including weights, structures, and gradients. This is a realistic assumption, considering OpenAI has not open-sourced LLM since the GPT-3.5 family. It is worth noting that all the models and datasets used in CheckGPT are publicly available.

Based on the nature of the task and the defender’s goals, we further make these assumptions: (1) *Moderate Data Availability*. We do not assume the defender’s privileged access to or abusive use of ChatGPT. The training and testing samples are collected strictly following OpenAI’s policy. With the rate limit and pricing, an ordinary user cannot have massive amounts (tens of millions to billions) of samples. Hence, the defender focuses on a more concise and domain-specific task. (2) *Affordability*. We do not assume the defender’s access to excessive computing power, which is only affordable to large organizations. We aim to develop a lightweight solution that smaller entities could conveniently obtain and deploy in a daily operational environment. And (3) *Privacy-preserving Local Deployment*. The end users may not share their data with a detection service provider due to concerns such as privacy, intellectual property, or policy, e.g., student essays or papers under review. Therefore, the detector should be easily transferred to a new domain using a small amount of data from the target domain and affordable computation resources.

5.2 The Baseline Approaches

We employ the non-deep learning methods as our baseline approach. The raw texts are first transformed into vector representations using the Term Frequency-Inverse Document Frequency (TF-IDF) model. Due to the vast vocabulary of the dataset, PCA is applied to reduce the feature space to 100 dimensions. We adopt three machine learning models to distinguish between human-written and GPT-generated abstracts: Gaussian Naive Bayes (GNB), support vector machine (SVM), and random forest (RF). For each task and discipline, the models are trained with 35,000 human-written and GPT-generated samples, respectively, and tested with 15,000 samples from each class, i.e., a 70/30 train-test split ratio.

The classification performance of the baseline models is shown in Table 4. We have the following observations: (1) While NB has been widely used in text analysis [63, 75], it performs poorly across all the tasks. This indicates that the assumptions (e.g., Gaussian Distribution and Independence) may not hold in this problem. (2) For Task 1, SVM and RF

Table 4: The baseline approach: classification accuracy (in %) for each classifier on each task and each discipline dataset.

Classifier	T1. GPT-WRI			T2. GPT-CPL			T3. GPT-POL		
	CS	PHX	HSS	CS	PHX	HSS	CS	PHX	HSS
GNB	53.3	51.9	52.2	52.1	51.7	50.8	50.2	50.5	51.4
SVM	96.4	98.2	74.8	86.4	90.5	66.5	57.8	75.9	53.3
RF	96.5	98.2	84.8	87.4	90.5	72.9	52.0	72.7	58.7

perform well in distinguishing GPT-written abstracts in CS and physics. However, their performance decreases to approximately 80% for HSS. A possible explanation is that the abstracts fully written by ChatGPT demonstrate *unique lexical features* that are easily distinguishable with linear classifiers. (3) For Task 2, the accuracy of SVM and RF drops to 86% for Computer Science, 90% for Physics, and around 70 % for humanity and social sciences. (4) The GPT-polished abstracts in Task 3 are more challenging to detect. The performance of the SVM and RF decreases sharply to 52% and 58% for CS and HSS, respectively, and to 70% for physics.

In summary, while the baseline approach lacks capabilities in some tasks, it still significantly outperforms human evaluators in Section 4. As the TF-IDF model only captures the lexical features, the success of SVM and RF in Task 1 implies that ChatGPT writes with different term distributions than humans, especially in physics, where human authors may use more math/technical terms than ChatGPT. However, the detection accuracy decreases significantly in Task 3, implying that GPT-polished abstracts tend to adopt the vocabulary and term distribution from the seed human-written abstracts.

5.3 The CheckGPT Framework

Preliminaries. The Bidirectional Encoder Representations from Transformers (BERT) [20] family of models, including but not limited to BERT itself and RoBERTa, have shown extraordinary capabilities in a wide range of NLP tasks. RoBERTa (Robustly Optimized BERT approach) [51], is the state-of-the-art member of this family built upon BERT by Meta. Models like RoBERTa are pre-trained on a massive corpus from diverse disciplines. Such extensive training allows them to capture and represent various linguistic patterns, syntactic structures, and semantic relationships in the texts. Its tokenization and encoding enable the transformation of raw data into effective representations, which can be used for downstream tasks. In this work, we utilize the pre-trained RoBERTa to preprocess the text data. The pre-training of the RoBERTa utilizes a masked language modeling (MLM) objective, which can be formalized as:

$$\mathcal{L}_{\text{MLM}} = -\mathbb{E}_{\mathbf{s} \sim \mathcal{D}_3} \log P(m|\mathbf{s}) \quad (3)$$

where \mathcal{D}_3 is the corpus, \mathbf{s} denotes an input sequence, and m is a masked token. The representations extracted by RoBERTa

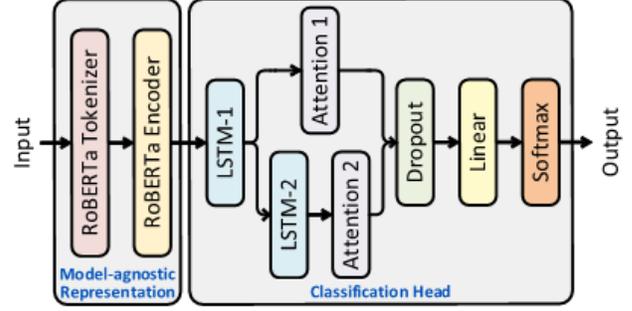


Figure 2: The architecture of the CheckGPT model.

serve as the inputs of our downstream classifier, a Long-Short-Term Memory (LSTM) network [39]. LSTM is a variant of Recurrent Neural Networks (RNNs) that has gained incredible success in natural language processing by handling sequential information. LSTM mitigates the gradient vanishing problem and improves model performance over long sequences by incorporating the gating mechanism, which enables it to effectively and selectively retain or update information.

Input Representation. Our CheckGPT includes two stages: representation, and classification. As shown in Figure 2, CheckGPT uses a model-agnostic design for text representation, so that any tokenization and embedding model could be employed in this framework. This design achieves higher affordability, upgradability, and flexibility since (1) adopting standardized and powerful embedding models save lots of effort and computations compared with training a large embedding model, which is usually beyond the capability of regular users or organizations; (2) a plugin design allows future upgrades by seamlessly accommodating new representation models; and (3) the lightweight classification head is easier to be tuned when new data or domains are added. In our proof-of-concept prototype of CheckGPT, the first stage is completed using the tokenizer and encoders of RoBERTa-large⁴. For tokenization, the pre-trained RoBERTa-large enforces a limit of 512 tokens. The tokenization can be formalized as:

$$\mathbf{X} = \text{BPE}(\mathbf{s}) = \{x_i\}_{i=1}^n \quad (4)$$

where \mathbf{X} denotes a sequence of length n consisting of individual tokens x_i , and BPE refers to the byte-level pairing encoding utilized by RoBERTa.

For the embedding layer, the RoBERTa uses embeddings of size 1024 to represent each token. In this way, our texts are transformed into contextualized representations with a shape of $n \times 1024$. The encoding can be formalized as:

$$\mathbf{E} = \text{TransformerEncoder}(\mathbf{X}) = \{e_i\}_{i=1}^n, e_i \in \mathbb{R}^{1024} \quad (5)$$

where \mathbf{E} denotes a sequence consisting of individual embedding e_i .

⁴<https://github.com/facebookresearch/fairseq/tree/main/examples/roberta>

LSTM Classification. The embeddings are finally fed into the LSTM classifier f_θ . Our classifier consists of two subsequent bi-directional LSTM layers. Each of these layers has a hidden state with a size of 256 and is followed by an attention layer [6]. The outputs of the two layers are concatenated, and then followed by a dropout layer with a dropout rate of $p = 0.5$, and finally fed into a dense layer. The dense layer gives two output values. Each softmaxed value represents the probability of belonging to each of the two classes: “GPT-generated” (y_g) or “Human-generated” (y_h). In details, LSTM classifier $f_\theta(\mathbf{E})$ is shown as follows:

$$\begin{aligned} h_1 &= \text{LSTM}_1(\mathbf{E}), \quad r_1 = \text{ATTN}_1(h_1) \\ h_2 &= \text{LSTM}_2(h_1), \quad r_2 = \text{ATTN}_2(h_2) \\ (\hat{y}_g, \hat{y}_h) &= \text{Softmax}(\text{FC}(\text{Dropout}(r_1 \oplus r_2))) \end{aligned} \quad (6)$$

Model Training. The classifier f_θ with parameter θ is optimized independently with the RoBERTa frozen during the training. We adopt an AdamW optimizer [52], a CosineAnnealing learning rate scheduler [53], and a gradient scaler for efficient mixed-precision training [59]. We employ cross-entropy loss for binary classification. Given the model’s predicted probabilities $\hat{y} = (\hat{y}_h, \hat{y}_c)$ and one-hot encoded ground truth $y = (y_h, y_c)$, the loss of a data sample is calculated as:

$$\mathcal{L}(\theta) = -[y_c \log(\hat{y}_c) + y_h \log(\hat{y}_h)] \quad (7)$$

Design Choices and Discussions. One of the alternative approaches is directly applying RoBERTa by adding a RobertaClassificationHead [41]. However, our experiments with a two-layer linear head (1M parameters) incur a 6-7% lower accuracy compared to CheckGPT. This can be attributed to LSTM’s capability to track the sequential dependencies over long periods in the text sequences [97].

Another alternative approach is to fine-tune the entire pre-trained model [66, 67], i.e., the model adopts the pre-trained parameters as a warm start and gets retrained on the new dataset. The CheckGPT design has several advantages compared with tuning the entire model: (1) CheckGPT will significantly reduce the parameters for training to save both time and computing resources. Considering the parameters of large language models ranging from 66M (DistilledBERT [77]) to 355M (RoBERTa-large [51]) and 1750M (GPT-3 [11]), our model only has 4M parameters (during training). The drop in model size also reduces the risks of over-fitting, especially when the dataset used for fine-tuning is small [5, 89]. (2) Without requiring fine-tuning, our framework is model-agnostic which can be compatible with various representation approaches (e.g., BERT, BART). As a result, CheckGPT is a lightweight detector and can be used with almost any publicly available pre-trained language models. As a tool for academia, this quality makes CheckGPT friendly to deploy and easy to customize. (3) By freezing the LLM with well-crafted parameters gained from extensive training, we retain the meta-knowledge to the greatest extent, which is expected to improve

Table 5: CheckGPT’s classification accuracy (in %) for each task and each discipline: TP rate, TN rate, overall accuracy.

	T1. GPT-WRI			T2. GPT-CPL			T3. GPT-POL		
	CS	PHX	HSS	CS	PHX	HSS	CS	PHX	HSS
TPR	99.96	100.0	99.94	99.21	97.71	98.58	98.43	98.93	98.45
TNR	99.98	99.99	99.92	99.43	98.14	99.00	98.37	99.22	98.49
Acc	99.97	99.99	99.93	99.32	97.93	98.79	98.39	99.08	98.47

CheckGPT’s transferability when dealing with samples from new domains (will be demonstrate in Section 6.3).

6 Experiments

6.1 Settings and Metrics

We implement CheckGPT with PyTorch 1.13.1 in Python 3.9.1 on Ubuntu 22.04. The pre-trained RoBERTa is adopted from [41]. All the experiments were conducted on an Nvidia 2080Ti GPU and an Intel i9-9900k CPU. We use GPABenchmark for most of the experiments. CheckGPT is trained with an initial learning rate of $2e-4$, a batch size of 256, and an early-stop strategy to finish training when the validation loss does not improve for a predefined number of epochs.

When we consider CheckGPT as GPT-generated content detector, the *true positive* rate ($TPR = \frac{TP}{TP+FN}$) is the proportion of correctly detected GPT-generated abstracts out of all GPT-generated abstracts, i.e., the accuracy in classifying GPT-generated text. The *true negative* rate ($TNR = \frac{TN}{TN+FP}$), is the proportion of correctly identified human-written abstracts out of all human-written abstracts, i.e. the accuracy in classifying human-written text. The overall classification accuracy of CheckGPT is defined as the proportion of correctly classified samples over all the testing samples: $Acc = \frac{TP+TN}{TP+FP+TN+FN}$.

6.2 Task- and Discipline-specific Classifiers

The task-specific and discipline-specific CheckGPT classifiers are trained at an average speed of 120s for each epoch of 80,000 training samples, while the average testing speed is 0.03s per sample. We report the classification accuracy in Table 5. Each task/discipline is trained with 80% of the samples (40,000 GPT-generated and 40,000 human-written samples) and tested with the remaining 20%. CheckGPT achieves very high performance in all cases. In particular, abstracts fully written by ChatGPT are the easiest to detect, with overall classification accuracy higher than 99.9% in all three disciplines. In Task 2, we only use the second half of the abstracts, i.e., the sentences written by GPT, in the classifier. The slightly lower classification accuracy could be explained by (1) with more seed information, ChatGPT generates higher quality writing with more domain-specific knowledge, and (2) the samples are shorter (half abstracts) than the ones in Task 1

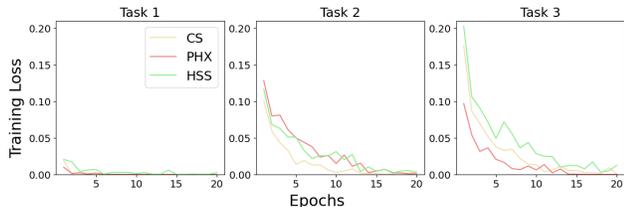


Figure 3: Training loss of the task-specific and discipline-specific classifiers.

(full abstracts) so that they provide less information to the classifier. Finally, the average classification accuracy of Task 3, which appeared to be the most difficult task for the open-source and commercial detectors (Section 3.2) and the baseline approaches (Section 5.2), is also in the range of [98%, 99%]. In conclusion, the task-specific discipline-specific classifiers demonstrate outstanding performance in distinguishing human-written and GPT-generated paper abstracts, even for tasks that pose great difficulties for the other SOTA detectors.

In Figure 3, we show the training loss of the task-specific discipline-specific classifiers. Models in Task 1 quickly learned some simple features, e.g., lexical features, and achieved satisfactory performance, while Tasks 2 and 3 are clearly more difficult. In most cases, HSS is more challenging than CS, while physics is the easiest. That is, ChatGPT does a better job mimicking human-written style in soft sciences. A different pattern is observed in Task 2, as the short samples in physics introduce additional challenges to the classifier.

Finally, we visualize the distribution of the human-written and GPT-generated abstracts in the feature space. We randomly select 2,000 CS abstracts from each task and each label, extract their vector representations right from the last linear layer of the classification head, and employ the t-Distributed Stochastic Neighbor Embedding (t-SNE) [90] to visualize the data points in a 3-dimensional space, as shown in Figure 4. As we can see, the GPT-written abstracts in Figure 4 (a) are highly clustered, which implies that the vocabulary, the writing style, and the semantic features of these abstracts are very consistent, so that they are easily distinguishable from the human-written abstracts, which appear to be more diverse in the feature space. The GPT-completed samples in Figure 4 (b) are shown to be significantly more diverse than the GPT-written samples. While they are also closer to the human-written samples in the feature space, there is still a clear separation between the two classes of samples. Finally, the GPT-polished samples in Figure 4 (c) appear to be scattered, where some data points appear to be blended with the human-written samples in a 3-dimensional space, which also demonstrates the difficulty in separating these two classes.

6.3 Transferability across Tasks/Disciplines

We evaluate CheckGPT’s capability in handling cross-task and cross-disciplinary testing samples. First, we use the nine

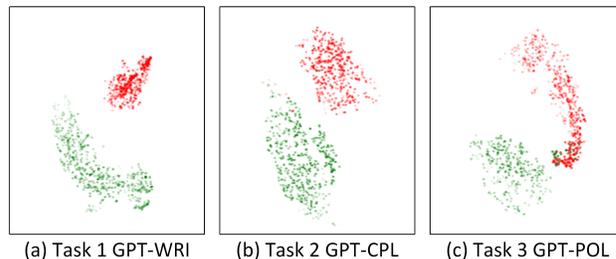


Figure 4: Feature space distribution of human-written (green) and GPT-generated (red) abstracts.

basic models trained in Section 6.2 to evaluate testing samples from other tasks and disciplines, *without model fine-tuning*. In Figure 5 (a), each value demonstrates the classification accuracy (in %) using the model from the task/discipline denoted on the x-axis against testing samples (10,000 GPT-generated and 10,000 human-written) from the task/discipline indicated on the y-axis. For example, when we use the model trained with Task 3 (GPT-polished) physics data (denoted as 3P in the figure) to evaluate the CS testing samples from Task 2 (2C), the classification accuracy is 88% (row 6 column 8 of Fig. 5 (a)). From the figure, we observe the following:

- CheckGPT is adaptable *across disciplines*. It demonstrates solid performance (mostly >90%) when we test a model with samples from different disciplines in the same task.
- CheckGPT appears to be less adaptable across tasks. In particular, the models trained in Task 1 demonstrate low performance with testing samples from the other tasks, while the models from Task 2 are also incapable of handling Task 3 (GPT-POL) data. Note that the models always give high TN rates (close to 100%), hence, a classification accuracy of $\sim 70\%$ implies a TP rate of only $\sim 40\%$.
- The models trained in Task 3 demonstrates solid performance with testing samples from Tasks 1 and 2. It implies that Task 3 could be the most difficult task, and the models have learned subtle but inherent features of AIGC.

We then fine-tune the final linear layer of each model with a small amount of data from the target domain, i.e., 1%, 5%, and 10% of the target dataset. We report the classification accuracy of the tuned models in Figure 5 (b) to (d). As shown, tuning the model with as few as 1% of data (500 human-written and 500 GPT-generated samples) increases the classification accuracy to >90% in 62 out of 81 experiments. If we increase the fine-tuning data to 5,000 samples in each label, the majority of the tuned models (55/81) achieve a classification accuracy of >95%. Moreover, we still observe similar patterns of transferability as we have observed from Figure 5 (a).

The Unified Classifiers. We train a cross-discipline classifier for each task with the training samples from all three disciplines in the same task. We test each classifier with the testing samples in each discipline and report the true positive and true negative rates in Table 6 (a). Finally, we train a unified

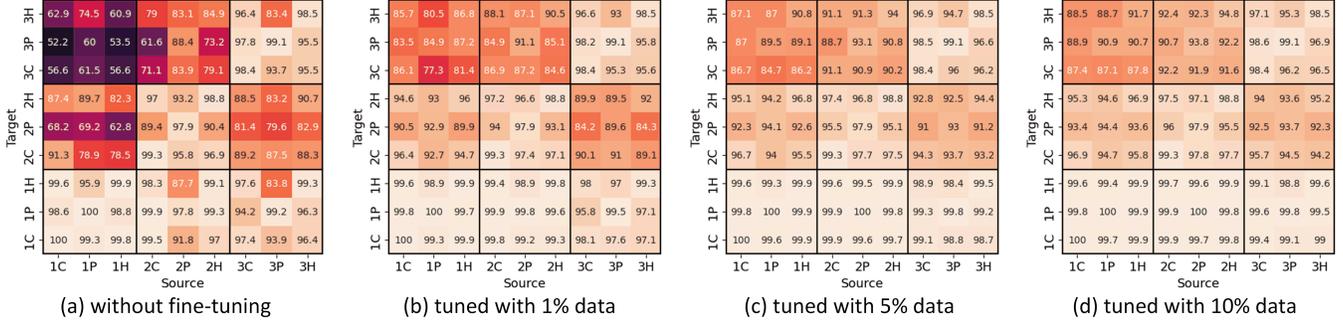


Figure 5: CheckGPT’s transferability across disciplines and tasks: (a) without fine-tuning, (b)-(d): tuned with 1%, 5%, and 10% data from the target domain, respectively. 1C: Task 1 GPT-WRI CS data; 2P: Task 2 GPT-CPL+physics; 3H: GPT-POL+HSS.

Table 6: Classification accuracy (in %) of the unified classifiers.

	T1. GPT-WRI			T2. GPT-CPL			T3. GPT-POL		
	CS	PHX	HSS	CS	PHX	HSS	CS	PHX	HSS
(a) Task-specific Cross-disciplinary Classifiers									
TPR	99.98	100	99.94	99.35	97.47	98.40	98.93	99.40	98.75
TNR	99.98	99.99	99.93	99.22	98.29	99.23	98.44	99.43	98.83
TPR		99.97			98.41			99.03	
TNR		99.97			98.91			98.90	
(b) Cross-task, Cross-disciplinary Classifier									
TPR	100	100	99.97	99.36	97.46	98.88	98.20	99.08	98.56
TNR	99.23	99.73	99.67	98.80	98.63	98.80	99.23	99.84	99.68
TPR					98.95				
TNR					99.30				

cross-task cross-discipline classifier for all the tasks and disciplines. The true positive and true negative rates are reported in Table 6 (b). In summary, both unified classifiers (a) and (b) perform well in the testing of each task-discipline experiment. Specifically, the unified training helps in boosting the performance of individual tasks, such as the difficult GPT-POL in HSS. These experiments further suggest the feasibility of developing general detecting algorithms in academia.

6.4 Transferability to New Domains

To evaluate CheckGPT with GPT-generated text from other domains, we have collected the following datasets:

- **Wikipedia Abstracts [Wiki]**. We randomly select 1,500 samples from the Wikipedia articles corpus [12]. The dataset contains the first introductory section of Wiki articles. We revise the ChatGPT prompts to avoid terms such as “research” and “paper”. For example, we use the prompt “Please generate a brief introduction of...” in Task 1.
- **Essays**. We use two types of essays from the Hewlett Foundation Automated Essay Scoring dataset [26]: [Essay-C] Essay set 1 contains 1,785 essays of 350 words on average. We adopt the original prompt from the dataset in Task 1:

Table 7: CheckGPT’s classification accuracy (in %) for other datasets.

dataset	w/o fine-tuning			w/ fine-tuning		
	Task 1	Task 2	Task 3	Task 1	Task 2	Task 3
Wiki	99.63	97.66	93.82	99.93	100.00	99.60
Essay-C	77.18	99.87	93.76	98.25	100.00	99.74
Essay-P	83.14	95.90	95.77	96.51	99.48	99.39
BBC	79.66	98.41	90.89	97.44	99.31	98.62

“Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you.”, and design prompts for Tasks 2 and 3 accordingly. [Essay-P] Essay set 7 contains 1,730 stories about patience. We refer to the original prompts from the dataset to design ChatGPT prompts e.g., “write a story in your own way about patience” is used in Task 1. We remove essays that are shorter than 70 words.

- **BBC News Article Dataset [BBC]**. The dataset contains 1,454 BBC news articles from 2004 to 2005 in five topical areas: business, entertainment, politics, sport, and technology [32]. We use prompts to emphasize “news articles” to ChatGPT, e.g., “Please generate a news article titled ...”.

We employ the task-specific cross-discipline classifiers introduced in Section 6.3 on the four datasets and report the classification accuracy in Table 7. As shown in the table, CheckGPT demonstrates solid performance in detecting GPT-generated text content in domains other than academic writings, especially in Tasks 2 and 3. Moreover, we use 50% of the data from each new domain to fine-tune the last linear layer of the model and evaluate with the remaining 50% of the samples. As shown in Table 7, the fine-tuned models achieve very high classification accuracy in the new domains.

6.5 Transferability to New Models

OpenAI released GPT-4 on March 14, 2023, while Google released its LLM-based chatbot, Bard, a week later. They only provide web-based access and they enforce highly restricted rate limits, hence, we test them in small-scale experiments.

Table 8: CheckGPT’s true positive rate for Bard and GPT-4.

Bard				GPT-4			
Task 1	Task 2	Task 3	Unified	Task 1	Task 2	Task 3	Unified
50/50	53/53	44/51	138/154	53/53	53/53	53/53	157/159

We invoke Bard and GPT-4 with the same methods in Section 3.1 to generate AI-written, AI-completed, and AI-polished text for 53 randomly selected CS abstracts. Bard returned error messages for 3 abstracts in Task 1 and 2 abstracts in Task 3. We use the task-specific classifiers and the unified classifier to evaluate all the AI-generated abstracts, and show the true positive rates in Table 8. CheckGPT achieves 100% accuracy in 5 experiments. However, the detection accuracy for Bard-polished text (Task 3) is relatively low, at 86.3%. Further investigation shows that Bard makes very small changes in some polishing tasks. It only changes a few words, e.g., the tenses of verbs, while the sentence structures are mostly preserved. Therefore, classifying such abstracts as human-written appears to be a reasonable decision.

6.6 Use of ChatGPT in arXiv Papers

With the popularity of LLMs/ChatGPT, we raise the pivotal question: How many authors are (possibly) using ChatGPT to write or polish real-world research papers? In a pilot study, we collect 1,000 abstracts from arXiv in each month spanning from June 2022 to May 2023. We evaluate each abstract with the unified cross-task, cross-disciplinary classifier and show the ratio of identified GPT-generated abstracts in Figure 6. We observe the following from the results: (1) there is a significant increase in the usage of ChatGPT in papers/abstracts posted on arXiv, with a peak of 13.54% and 14.02% in April and May 2023. (2) The nearly exponential growth started in December 2022, which is consistent with ChatGPT’s initial release date of 11/30/2022. (3) CheckGPT also reported 2.8% ~ 3.7% of the abstracts posted before November 2022 as GPT-generated. This could be explained by CheckGPT’s 1% false positive rate, which could increase slightly since arXiv covers a wider spectrum of disciplines. Moreover, it is also possible that LLMs such as GPT-3 might be used by a small number of early adopters in the research community.

7 Discussions

7.1 Prompt Engineering

Prompts are used to provide instructions to conversational LLMs to customize/refine their responses. Research efforts on prompt engineering aim to guide or improve the design of ChatGPT prompts [22, 94]. We identify several popular prompt designs that have been adopted and discussed in the community [2, 3, 43, 70] and employ each of them to polish 5,000 abstracts randomly selected across all disciplines.

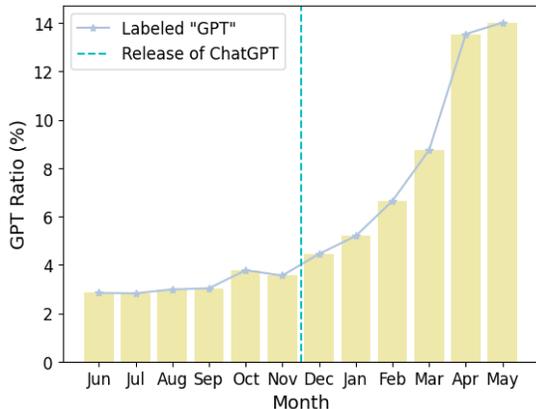


Figure 6: Detecting the use of ChatGPT in papers on arXiv.

Table 9: CheckGPT’s true positive rate on different prompts.

Prompt	1	2	3	4	5
w/o tuning	90.07%	93.57%	95.15%	97.35%	91.78%
w/ tuning	97.24%	98.72%	98.03%	98.37%	97.46%

Prompt 1: Rewrite the following abstract of a research paper in first-person, clear, and academic language

Prompt 2: Please act as an expert paper editor and revise the abstract section of the paper from the perspective of a paper reviewer to make it more fluent and elegant. Here are the specific requirements: [See Appendix A.3 for full prompt]

Prompt 3: Write a polished and refined version of the following abstract of a research paper to improve its overall quality and readability.

Prompt 4: I want you to act as an academic researcher. You will be responsible for rewriting the abstract of a research paper for clarity. Here is the original abstract of the paper:

Prompt 5: I want you to act as an academic paper writer. You will be responsible for rewriting a paper abstract. Your task is to improve the writing and clarity of the abstract. Here is the original abstract of the paper:

We employ the cross-disciplinary classifier for Task 3 to evaluate the polished abstracts and report the true positive rates in Table 9. We can observe a slight decrease in CheckGPT’s classification accuracy when different prompts are used. However, in all the cases, CheckGPT’s detection accuracy is still higher than 90%. Moreover, when the last linear layer of the classifier is tuned with 1,500 human-written samples and 1,500 GPT-polished samples (30% of the data), the classification accuracy reached 97% to 98%.

7.2 Model Interpretation

Besides the accuracy, the transparency of CheckGPT model is also important. The interpretation of CheckGPT not only helps us understand the rationale behind a specific decision, but also provides discerning insights to distinguish AI-generated from human-written texts. Therefore, to investigate this, we em-

ploy two methods: Integrated Gradients [45, 83] and Shapley Values [17, 55]. They represent two different angles: model-specific and model-agnostic explainability

- **Integrated Gradients.** This method assigns the importance to each value by the gradients compared with the baselines along the path. The baselines are the inputs that induce a “neutral” decision. We utilize the implementation in [45] and apply it to our models.
- **Shapley Values.** Originally introduced in [79] and recently applied to machine learning interpretation, a Shapley Value quantifies the impact of each feature by perturbing the input value and seeing how the change of input contributes to prediction. We adopt the implementation in [55].

Word-level Analysis. We first apply these methods at word-level to measure the contribution of each word toward the decisions. As the example shown in Fig. 7 using Integrated Gradients, the word “landscape” and “automated” are identified as the most significant features for Task 1 and Task 3 respectively. The feature saliency is almost uniformly distributed across the entire paragraph in Task 2.

Fig 8 shows the comparisons of GPT-generated abstract and human-written abstract explained by Shapley Values. The words “on” and “data” are the most supportive features leading to a decision of “human-written”. Words of metadiscourses are the most important features in Task 1 and Task 3. Reporting verbs like “explore”, “aim”, “discuss” and “examine” are mostly adopted by the “GPT-written” style for describing intentions. The transitional phases that guide the readers, like “However”, “Overall” and “Ultimately”, are also significant features for GPT writing, especially Task 2.

Our attempts at the word-level experiments produce relatively *uninformative* findings. The significance assigned to each individual word is usually insufficient for human users to draw useful conclusions. The limitation of the methods is due to the sophistication of the LLMs which capture complex semantic and linguistic features. Thus, the word-level interpretations are inadequate for our analysis.

Sentence-level Analysis. While independent words do not show a sufficient power of explainability, the corporative semantic patterns captured within sentences have the potential to give a more comprehensive insight. Language comprehension relies heavily on context, nuance, and syntactic structures, which are far more informative beyond interpreting individual words. Furthermore, the LLMs like ChatGPT, typically build their task to generate coherent and sentence-level responses. Thus, a sentence-level analysis has been conducted in the hope of a better-quality interpretation.

In Fig 9, we extend our analysis to sentence-level interpretations using Shapley Values because of its coherent output. The abstracts are parsed into sentences, which become the new units of features. From the results, we find that sentence-level analysis provides more meaningful and consistent insights for identifying GPT-generated texts. First, we find that the

supporting sentences for human texts or GPT texts locate differently in an abstract, which means that the GPT writing style for different presentation goals contains distinguishable and unique patterns for detection. Second, we can see that ChatGPT frequently starts the abstract with a declarative statement like “This paper proposes” to emphasize the focus of the paper. It shows that the particular ways of presenting ideas consist another character of GPT’s “footprint” in writing. Last, in the last sentence of the abstracts, ChatGPT usually tries to use a conclusive statement to summarize the findings or contributions of the paper, which is also widely observed in regular ChatGPT conversations. This “habit” of summarization which is designed for Q&A tasks reveals the third pattern uniquely carried by GPT even when it is writing abstracts. Additional examples of our interpretations are given in Appendix.

In summary, comparing the interpretations derived from word- and sentence-level results, we find that the complex linguistic and presentation patterns can be better expressed by sentence-level features. However, we have to admit that it also trades off the granularity and thus currently can not provide results in finer details (e.g., patterns of wording and phrasing). These interpretation experiments demonstrate that there are no explicit or dominant indicators that can be easily captured for GPT writing recognition. The finding emphasizes the necessity for applying sophisticated and automated tools, like deep learning techniques, to perform effective detection for complicated and subtle semantic features.

8 Conclusion and Futureworks

In this paper, we first introduce a benchmarking dataset, namely GPABenchmark, for LLM-content detection in academia. GPABenchmark contains 600,000 samples of human-written, GPT-written, GPT-completed, and GPT-polished abstracts of research papers in three disciplines. Second, we show that the existing online ChatGPT detectors fall short in detection accuracy. A user study of 150+ participants finds that human users are incapable of identifying GPT-generated text. Finally, we present CheckGPT, a deep learning-based detector for GPT-generated academic writing. With extensive experiments, we show that CheckGPT is highly accurate with additional advantages in affordability, flexibility, transferability, and explainability.

It is our future plan to further investigate GPT-generated text from different disciplines and prompts. The prompt engineering problem is highly challenging due to the complexity of the LLMs, the mostly black-box nature of ChatGPT, and the community’s very limited understanding of the operational mechanisms behind ChatGPT prompts. We also plan to investigate how the users may manipulate the prompts or re-edit the GPT-generated text to escape the detectors. Post-processing may present a significant challenge, as knowledgeable users with insights into either the detector or ChatGPT may purposefully revise GPT-generated text to evade detection.

References

- [1] ZeroGPT: AI Text Detector. Available at: <https://www.zerogpt.com>, January 2023.
- [2] Fatih Kadir Akın et al. Awesome ChatGPT Prompts. Available at: <https://github.com/f/awesome-chatgpt-prompts>, 2023.
- [3] Kevin Amiri. A collection of ChatGPT, GPT-3.5, GPT-4 prompts. Available at: <https://github.com/kevinamiri/Instructgpt-prompts>, 2023.
- [4] Brent A Anders. Is using chatgpt cheating, plagiarism, both, neither, or forward thinking? *Patterns*, 4(3), 2023.
- [5] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*, 2019.
- [6] Christos Baziotis, Athanasios Nikolaos, Pinelopi Papalampidi, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, and Alexandros Potamianos. NTUA-SLP at SemEval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive RNNs. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 613–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [7] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.
- [8] Douglas Biber. *Variation across speech and writing*. Cambridge University Press, 1991.
- [9] Douglas Biber and Bethany Gray. Challenging stereotypes about academic writing: Complexity, elaboration, explicitness. *Journal of English for Academic Purposes*, 9(1):2–20, 2010.
- [10] Arend Groot Bleumink and Aaron Shikhule. Keeping ai honest in education: Identifying gpt-generated text. 2023.
- [11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [12] Martin Brümmer, Milan Dojchinovski, and Sebastian Hellmann. Dbpedia abstracts: a large-scale, open, multilingual nlp training corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3339–3343, 2016.
- [13] Enrico Cambiaso and Luca Caviglione. Scamming the scammers: Using chatgpt to reply mails for wasting time and resources. *arXiv preprint arXiv:2303.13521*, 2023.
- [14] Margaret Cargill and Patrick O'Connor. *Writing scientific research articles: Strategy and steps*. John Wiley & Sons, 2021.
- [15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [16] Stefano Cresci. A decade of social bot detection. *Communications of the ACM*, 63(10):72–83, 2020.
- [17] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE, 2016.
- [18] Luigi De Angelis, Francesco Baglivo, Guglielmo Arzilli, Gaetano Pierpaolo Privitera, Paolo Ferragina, Alberto Eugenio Tozzi, and Caterina Rizzo. Chatgpt and the rise of large language models: the new ai-driven infodemic threat in public health. *Frontiers in Public Health*, 11:1567, 2023.
- [19] Erik Derner and Kristina Batistič. Beyond the safeguards: Exploring the security risks of chatgpt. *arXiv preprint arXiv:2305.08005*, 2023.
- [20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [21] David Dukić, Dominik Keča, and Dominik Stipić. Are you human? detecting bots on twitter using bert. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 631–636. IEEE, 2020.
- [22] Sabit Ekin. Prompt engineering for chatgpt: A quick guide to techniques, tips, and best practices. 2023.
- [23] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.
- [24] Mehmet Firat. What chatgpt means for universities: Perceptions of scholars and students. *Journal of Applied Learning and Teaching*, 6(1), 2023.

- [25] Annette Flanagin, Kirsten Bibbins-Domingo, Michael Berkwits, and Stacy L Christiansen. Nonhuman “authors” and implications for the integrity of scientific publication and medical knowledge. *Jama*, 329(8):637–639, 2023.
- [26] The Hewlett Foundation. The Hewlett Foundation: Automated Essay Scoring. Kaggle, available at: <https://www.kaggle.com/c/asap-aes>, 2012.
- [27] Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. Comparing scientific abstracts generated by chatgpt to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *bioRxiv*, pages 2022–12, 2022.
- [28] Andres Garcia-Silva, Cristian Berrio, and José Manuel Gómez-Pérez. An empirical study on pre-trained embeddings and language models for bot detection. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 148–155, 2019.
- [29] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 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, 2019.
- [30] Kacper T Gradonm. Electric sheep on the pastures of disinformation and targeted phishing campaigns: The security implications of chatgpt. *IEEE Security & Privacy*, 21(3):58–61, 2023.
- [31] Dijana Vukovic Grbic and Igor Dujlovic. Social engineering with chatgpt. In *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE, 2023.
- [32] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML’06)*, pages 377–384. ACM Press, 2006.
- [33] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023.
- [34] Julian Hazell. Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*, 2023.
- [35] Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*, 2023.
- [36] Maryam Heidari and James H Jones. Using bert to extract topic-independent sentiment features for social media bot detection. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0542–0547. IEEE, 2020.
- [37] Maryam Heidari, Samira Zad, Parisa Hajibabae, Masoud Malekzadeh, SeyyedPooya HekmatiAthar, Ozlem Uzuner, and James H Jones. Bert model for fake news detection based on social bot activities in the covid-19 pandemic. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0103–0109. IEEE, 2021.
- [38] Eli Hinkel. *Teaching academic ESL writing: Practical techniques in vocabulary and grammar*. Routledge, 2003.
- [39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [40] Emma Hoes, Sacha Altay, and Juan Bermeo. Using chatgpt to fight misinformation: Chatgpt nails 72% of 12,000 verified claims. 2023.
- [41] Huggingface. RoBERTa. Available at: https://huggingface.co/docs/transformers/model_doc/roberta.
- [42] Ken Hyland. Stance and engagement: A model of interaction in academic discourse. *Discourse studies*, 7(2):173–192, 2005.
- [43] Ashish Jaiswal. Smart ChatGPT Prompts. Available at: <https://github.com/asheshcric/smart-chatgpt-prompts>, 2023.
- [44] Mohammad Khalil and Erkan Er. Will chatgpt get you caught? rethinking of plagiarism detection. *arXiv preprint arXiv:2302.04335*, 2023.
- [45] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.
- [46] Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. Artificial text detection

- via examining the topology of attention maps. In *ACL Anthology*, 2021.
- [47] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.
- [48] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*, 2023.
- [49] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.
- [50] Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models. *arXiv preprint arXiv:2304.07666*, 2023.
- [51] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [52] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- [53] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.
- [54] Zhuoran Lu, Patrick Li, Weilong Wang, and Ming Yin. The effects of ai-based credibility indicators on the detection and spread of misinformation under social influence. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–27, 2022.
- [55] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.
- [56] Tony Lynch and Kenneth Anderson. Grammar for academic writing. *English Language Teaching Centre*, pages 1–6, 2013.
- [57] Kamil Malinka, Martin Perešín, Anton Firc, Ondřej Hujňák, and Filip Januš. On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree? *arXiv preprint arXiv:2303.11146*, 2023.
- [58] Steve Mansfield-Devine. Weaponising chatgpt. *Network Security*, 2023(4), 2023.
- [59] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *International Conference on Learning Representations*.
- [60] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [61] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.
- [62] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*, 2023.
- [63] Tom Michael Mitchell et al. *Machine learning*, volume 1. McGraw-hill New York, 2007.
- [64] OpenAI. AI Text Classifier. Available at: <https://platform.openai.com/ai-text-classifier>, 2023.
- [65] OpenAI. Educator considerations for ChatGPT. Available at: <https://platform.openai.com/docs/chatgpt-education>, 2023.
- [66] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
- [67] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. Finetuning RoBERTa on a custom classification task. Available at: https://github.com/facebookresearch/fairseq/blob/main/examples/roberta/README_custom_classification.md, 2020.
- [68] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.
- [69] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237. ACL, 2018.

- [70] PlexPt. Awesome ChatGPT Prompts. Available at: <https://github.com/PlexPt/awesome-chatgpt-prompts>, 2023.
- [71] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [72] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.
- [73] Karen Renaud, Merrill Warkentin, and George Westerman. From chatgpt to hackgpt: Meeting the cybersecurity threat of generative ai. *MIT Sloan Management Review*, 64(3):1–4, 2023.
- [74] Sayak Saha Roy, Krishna Vamsi Naragam, and Shirin Nilizadeh. Generating phishing attacks using chatgpt. *arXiv preprint arXiv:2305.05133*, 2023.
- [75] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*, volume 62, pages 98–105. Citeseer, 1998.
- [76] Malik Sallam. Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, volume 11, page 887. MDPI, 2023.
- [77] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
- [78] Wajiha Shahid, Bahman Jamshidi, Saqib Hakak, Haruna Isah, Wazir Zada Khan, Muhammad Khurram Khan, and Kim-Kwang Raymond Choo. Detecting and mitigating the dissemination of fake news: Challenges and future research opportunities. *IEEE Transactions on Computational Social Systems*, 2022.
- [79] LS SHAPLEY. A value for n-person games. *Contributions to the Theory of Games*, pages 307–317, 1953.
- [80] StackOverflow. Use of ChatGPT generated text for content on Stack Overflow is temporarily banned. Available at: <https://meta.stackoverflow.com/questions/421831/temporary-policy-chatgpt-is-banned>, 2023.
- [81] Chris Stokel-Walker. Ai bot chatgpt writes smart essays-should academics worry? *Nature*, 2022.
- [82] Chris Stokel-Walker. Chatgpt listed as author on research papers: many scientists disapprove. *Nature*, 2023.
- [83] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.
- [84] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.
- [85] Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*, 2023.
- [86] Panagiotis C Theocharopoulos, Panagiotis Anagnostou, Anastasia Tsoukala, Spiros V Georgakopoulos, Sotiris K Tasoulis, and Vassilis P Plagianakos. Detection of fake generated scientific abstracts. *arXiv preprint arXiv:2304.06148*, 2023.
- [87] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.
- [88] Edward Tian. GPTZero. Available at: <https://gptzero.me/>, 2023.
- [89] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, 2020.
- [90] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [91] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [92] Hong Wang, Xuan Luo, Weizhi Wang, and Xifeng Yan. Bot or human? detecting chatgpt imposters with a single question. *arXiv preprint arXiv:2305.06424*, 2023.
- [93] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

- [94] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.
- [95] Jurgen Willems. Chatgpt at universities—the least of our concerns. *Available at SSRN 4334162*, 2023.
- [96] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1096–1103, 2020.
- [97] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.
- [98] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.
- [99] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A Additional Experimental Results

A.1 Benchmarking ZeroGPT

For each input text snippet, ZeroGPT [1] returns one of the nine possible decisions. We assign an integer score of [0, 8] as follows:

0. Your text is Human written
1. Your text is Most Likely Human written
2. Your text is Most Likely Human written, may include parts generated by AI/GPT
3. Your text is Likely Human written, may include parts generated by AI/GPT
4. Your text contains mixed signals, with some parts generated by AI/GPT
5. Your text is Likely generated by AI/GPT
6. Your text is Most Likely AI/GPT generated
7. Most of Your text is AI/GPT Generated
8. Your text is AI/GPT Generated

The distribution of the scores for each task and each discipline is shown in Table 10. For instance, for GPT-polished abstracts (Task 3) in CS, 88.3% were annotated as “human written” by ZeroGPT, while 4.7% were annotated as “Most likely human written”.

When we converted the 9-point scores to binary decisions of “GPT”/“Human”, a threshold of 4 was used. While we can also make the case that categories 2, 3, 4 should be categorized as “GPT” in Task 3, since the decision statements indicate that they “may include parts generated by AI/GPT,” which is the case for Task 3. However, changing the decision threshold will not significantly change the observations and conclusions in Section 3.2, since only a very small portion of the samples in Task 3 were annotated with those three labels, as shown in Table 10. For Tasks 1 and 2, the text snippets we sent to ZeroGPT were completely written by ChatGPT, hence, a threshold of 4 is the most reasonable choice.

A.2 Benchmarking Open-AI’s Text Classifier

For each input text snippet, the OpenAI text classifier [64] returns a decision from one of the five classes. We map them to an integer score of [0, 4] as follows:

0. The classifier considers the text to be very unlikely AI-generated.
1. The classifier considers the text to be unlikely AI-generated.
2. The classifier considers the text to be unclear if it is AI-generated.
3. The classifier considers the text to be possibly AI-generated.
4. The classifier considers the text to be likely AI-generated.

The distribution of the scores for each task and each discipline is shown in Table 11. For instance, for human-written CS

Table 10: Distribution of detection score generated by the ZeroGPT: 0: human-written; 8: GPT-generated. The largest score category for each experiment is shown in bold.

	T1. GPT-ERI			T2. GPT-CPL			T3. GPT-POL		
	CS	PHX	HSS	CS	PHX	HSS	CS	PHX	HSS
(a) Score distribution (in %) of GPT-generated abstracts.									
0	16.7	21	1.7	52.7	75.7	18	88.3	93	52
1	4.7	3	2	1.3	1	1	4.7	2	6.7
2	6	5	2	13.3	11.3	13.7	2	1	8.3
3	2.7	1	2	6.7	0.7	3	1.3	0.7	5.3
4	2.7	1.7	0	0.7	1.3	2	0.3	0.7	3
5	4.3	0.7	0.3	5.3	2	7.7	1.3	0	6.7
6	4.7	5.7	2.7	4	3.3	7.3	1	0.7	5
7	8.7	17.3	3.3	3.3	1	9.7	0	0.3	3.3
8	49.7	44.7	86	12.7	3.7	37.7	1	1.7	9.7
(b) Score distribution (in %) of human-written abstracts.									
0	93.7	97.7	79	96.3	98.7	87.3	92	96	79
1	3.3	0	9	0.7	0	0.7	4	1	6.7
2	3	0.7	5	2.7	1	5.7	2	1.3	5
3	0	0	2	0	0	1	0.3	0.3	2
4	0	0.3	1.7	0	0	1.3	0.3	0	1.7
5	0	0	1	0	0.3	1	0.3	0.3	2
6	0	0	1.3	0	0	1	0	0	2
7	0	0.3	0.7	0.3	0	0.3	0	0.3	0.3
8	0	1	0.3	0	0	1.7	1	0.7	1.3

Table 11: Distribution of detection score generated by the OpenAI text classifier: 0: very unlikely AI-generated; 2: unclear if it is AI-generated; 4: likely AI-generated. The largest score category for each experiment is shown in bold.

	T1. GPT-ERI			T2. GPT-CPL			T3. GPT-POL		
	CS	PHX	HSS	CS	PHX	HSS	CS	PHX	HSS
(a) Score distribution (in %) of GPT-generated abstracts.									
0	0	0	0	0	0	8	4	5	11.3
1	0.3	0.3	3.3	1.3	12.3	11	23.7	35.3	31.3
2	19	29.7	33.7	35	64	53.7	66	55.3	51.3
3	50	50.7	51	56	22.7	24	6.3	4	6
4	30.7	19.3	12	7.7	1	3.3	0	0.3	0
(b) Score distribution (in %) of human-written abstracts.									
0	11	15.7	60	4.3	7.7	56.3	12.7	18	62.0
1	40	54	24	31	52	23.3	38	51	26
2	45.3	28.3	14	54	38	16.7	48.3	29.7	10.7
3	3.7	2	1.3	10.7	2	3.3	1	1.3	1
4	0	0	0.7	0	0.3	0.3	0	0	0.3

abstracts, 11% are classified as “very unlikely AI-generated”, 40% are classified as “unlikely AI-generated”, 45.3% are classified as “unclear if it is AI-generated”, and the remaining 3.7% are classified as “possibly AI-generated”.

A.3 Prompts

The complete Prompt 2 in Sec 7.1 is as follows:

Please act as an expert paper editor and revise the abstract section of the paper from the perspective of a paper reviewer to make it more fluent and elegant. Here are the specific requirements:

1. *Enable readers to quickly grasp the main points or essence of the paper.*
2. *Allow readers to understand the important information, analysis, and arguments throughout the entire paper.*
3. *Help readers remember the key points of the paper.*
4. *Please clearly state the innovative aspects of your model and methods in the abstract, emphasizing your contributions.*
5. *Use concise and clear language to describe your methods and results, making it easier for reviewers to understand the paper.*

Here is the original abstract section of the paper:

B Examples of Model Interpretation

In this section, we present several examples of model interpretation in Fig 7. Fig 8 and Fig 9.

#s Computational law is an emerging branch of legal studies and information concerned with the mechanization of legal reasoning . Machine learning can be described a technique by which an automated response is generated based on input data . Legal experts spend lot of time dealing with complicated legal tasks that require effective analysis , reasoning and decision making . Machine learning could be said to replicate human behavior in this sense . This paper debates the benefits , limitations and implications of computational analysis and artificial intelligence in representations of the law . #/s

(a) Human-written text.

#s This paper aims to analyze the benefits , demerits , and criticisms of the revolution of computational analysis and artificial intelligence in law . The paper will provide an overview of how technology has changed the legal landscape and discuss how computational analysis and artificial intelligence have impacted legal processes and practices . The benefits of the revolution of computational analysis and artificial intelligence in law include increased efficiency , accuracy , and cost savings . However , there are also several demerits , including concerns about bias and job displacement . Additionally , the paper will explore criticisms of the use of computational analysis and artificial intelligence in law , including ethical concerns , issues of transparency , and the potential for unintended consequences . Overall , this paper aims to provide a comprehensive analysis of the impact of computational analysis and artificial intelligence on the legal profession and to stimulate further discussion and research in this area . #/s

(b) GPT-written text in Task 1.

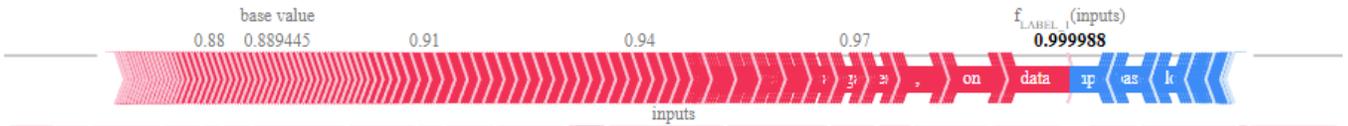
#s In recent years , there has been growing interest in combining the fields of computational law and machine learning to create more efficient and accurate legal decision - making processes . This paper explores the potential benefits and challenges of this integration , as well as the ethical and regulatory considerations that must be taken into account . Ultimately , it argues that while there are significant obstacles to be overcome , computational law and machine learning have the potential to revolutionize the legal system and improve access to justice for all . #/s

(c) GPT-completed text in Task 2.

#s This paper discusses a new area of legal studies called computational law , which focuses on using technology to automate legal reasoning . Machine learning is a specific tool used in computational law that generates automated responses based on input data . Legal experts often spend a great deal of time analyzing complex legal tasks , requiring effective analysis , reasoning , and decision - making skills . Machine learning has the potential to replicate human behavior in these areas . The paper examines the advantages , disadvantages , and consequences of using machine learning and artificial intelligence in computational analysis of the law . #/s

(d) GPT-polished text in Task 3.

Figure 7: Word importance using Integrated Gradients. A case of HSS written by humans and ChatGPT in three different tasks. Green regions indicate positive contributions to the corresponding label, and red ones indicate negative contributions.



Computational law is an emerging branch of legal studies and information concerned with the mechanization of legal reasoning. Machine learning can be described a technique by which an automated response is generated based on input data. Legal experts spend lot of time dealing with complicated legal tasks that require effective analysis, reasoning and decision making. Machine learning could be said to replicate human behavior in this sense. This paper debates the benefits, limitations and implications of computational analysis and artificial intelligence in representations of the law.

(a) Human-written text.



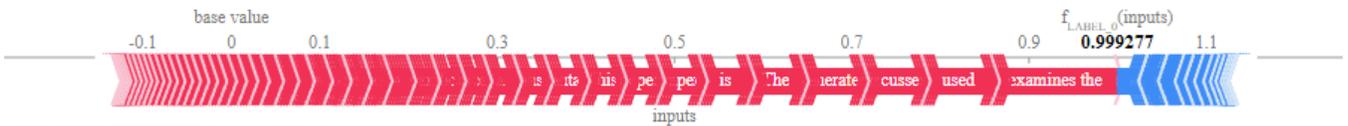
This paper aims to analyze the benefits, demerits, and criticisms of the revolution of computational analysis and artificial intelligence in law. The paper will provide an overview of how technology has changed the legal landscape and discuss how computational analysis and artificial intelligence have impacted legal processes and practices. The benefits of the revolution of computational analysis and artificial intelligence in law include increased efficiency, accuracy, and cost savings. However, there are also several demerits, including concerns about bias and job displacement. Additionally, the paper will explore criticisms of the use of computational analysis and artificial intelligence in law, including ethical concerns, issues of transparency, and the potential for unintended consequences. Overall, this paper aims to provide a comprehensive analysis of the impact of computational analysis and artificial intelligence on the legal profession and to stimulate further discussion and research in this area.

(b) GPT-written text in Task 1.



In recent years, there has been growing interest in combining the fields of computational law and machine learning to create more efficient and accurate legal decision-making processes. This paper explores the potential benefits and challenges of this integration, as well as the ethical and regulatory considerations that must be taken into account. Ultimately, it argues that while there are significant obstacles to be overcome, computational law and machine learning have the potential to revolutionize the legal system and improve access to justice for all.

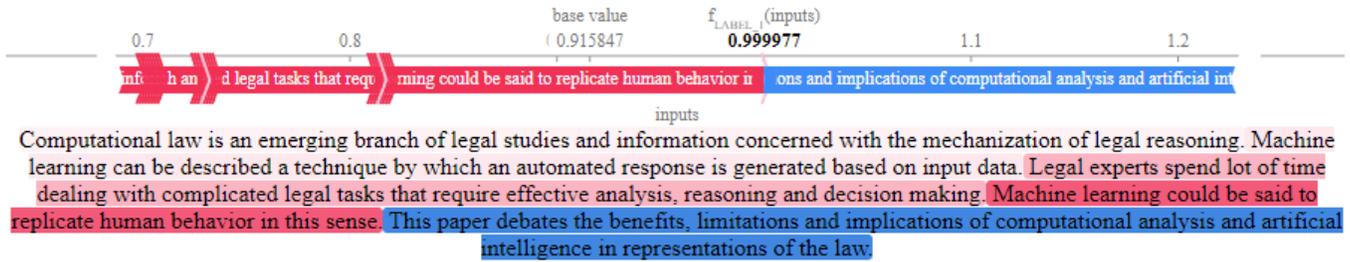
(c) GPT-completed text in Task 2.



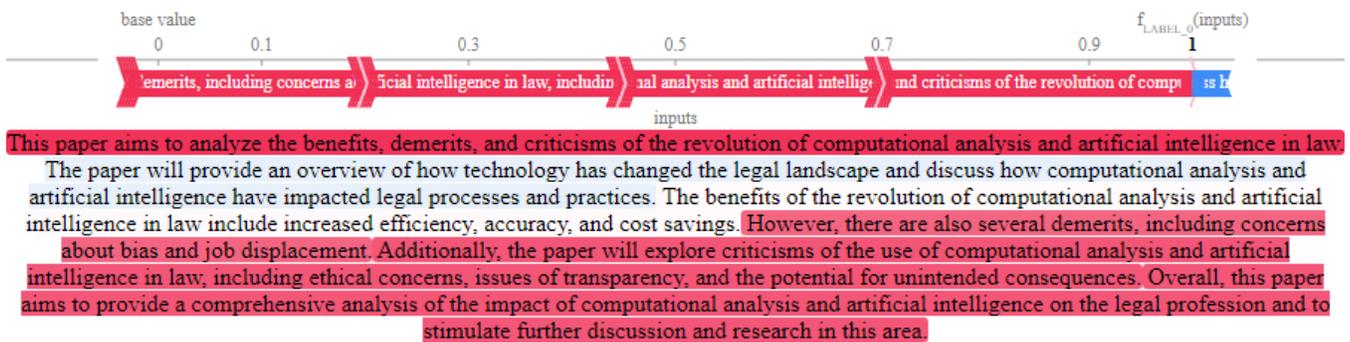
This paper discusses a new area of legal studies called computational law, which focuses on using technology to automate legal reasoning. Machine learning is a specific tool used in computational law that generates automated responses based on input data. Legal experts often spend a great deal of time analyzing complex legal tasks, requiring effective analysis, reasoning, and decision-making skills. Machine learning has the potential to replicate human behavior in these areas. The paper examines the advantages, disadvantages, and consequences of using machine learning and artificial intelligence in computational analysis of the law.

(d) GPT-polished text in Task 3.

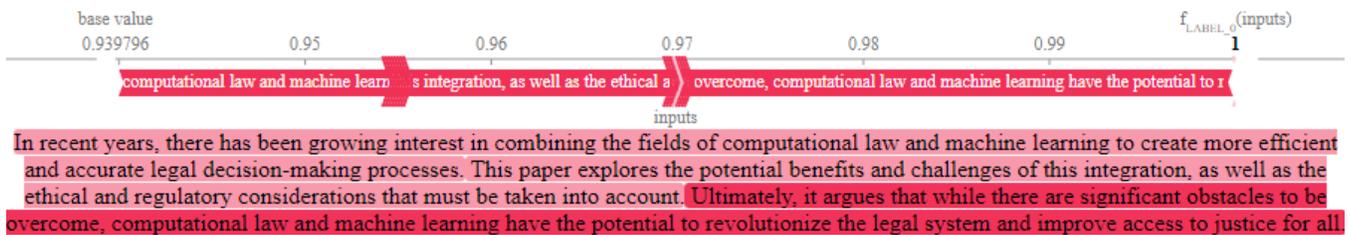
Figure 8: Word importance using Shapley Values. A case of HSS written by humans and ChatGPT in three different tasks. Red regions indicate positive contributions to the label of a particular text, while blue ones indicate negative contributions.



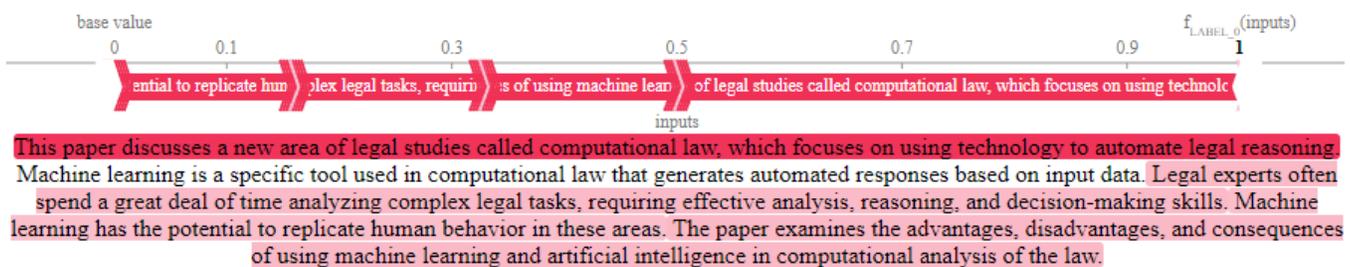
(a) Human-written text.



(b) GPT-written text in Task 1.



(c) GPT-completed text in Task 2.



(d) GPT-polished text in Task 3.

Figure 9: Sentence importance using Shapley Values. A case of HSS written by humans and GPT in three different tasks. Red regions indicate positive contributions to the label of a particular text, while blue ones indicate negative contributions.