



The Odyssey of robots.txt Governance: Measuring Convention Implications of Web Bots in Large Language Model Services

Jian Cui*

University of Illinois Urbana-Champaign
Urbana, Illinois, USA
jiancui3@illinois.edu

XiaoFeng Wang

Nanyang Technological University
Singapore, Singapore
xiaofeng.wang@ntu.edu.sg

Mingming Zha*

Indiana University Bloomington
Bloomington, Indiana, USA
mzha@iu.edu

XiaoJing Liao

University of Illinois Urbana-Champaign
Urbana, Illinois, USA
xjliao@illinois.edu

Abstract

Web content is an essential element for large language model (LLM) services, supporting both training and inference processes. To manage the content access of web bots from LLM service vendors (i.e., LLM bots), web content publishers are increasingly incorporated content access rules into robots.txt, a long-established web content management protocol. However, the rise of proprietary LLM bots, such as OpenAI’s *ChatGPT-User* and Google’s *Google-Extended*, has raised concerns about the transparency of web content access and whether these bots adherence to robots.txt rules. However, there is limited understanding of these LLM bots, concerning their impact on web publishers and broader web content governance. To fill this gap, we present a systematic analysis of 18 LLM bots on 582,281 robots.txt files. Our findings reveal a significant increase in robots.txt rules associated with LLM bots, particularly in domains that fall into the finance and news category. Despite the heightened integration, web publishers face challenges in managing robots.txt configurations due to the complexity of the LLM ecosystem and the involvement of third-party brokers. Furthermore, we identified several cases of robots.txt violations, including instances where LLMs memorized web content from restricted domains, and where *ChatGPT-User* ignored robots.txt and accessed restricted content. These results highlight the gaps in the current web content governance and underscore the need for enforceable content management mechanisms to respect web publishers’ intentions and content control.

1 Introduction

Similar to the quote from 2007, “*without web bots, there would be no search engines*,” we are now in an era where “*without web bots, there would be no LLM services*.” Web content, such as text corpora from news websites, images from image hosting platforms, code from developer forums, and even multilingual parallel corpora from Wikipedia, plays vital roles in LLM services, including assembling initial training corpora [74], conducting instruction fine-tuning [94], and enabling Retrieval Augmented Generation [35]. As demand for web content as training data grows, LLM vendors have been increasingly deploying proprietary web bots (i.e., LLM bots) dedicated to crawling public web content. Examples include *GPTBot* and

ChatGPT-User from OpenAI [1, 57], *ClaudeBot* from Anthropic [9], and *Google-Extended* from Google [66].

However, as these LLM bots proliferate, concerns over public web content collection ethics, privacy, and adherence to web content governance protocols—particularly robots.txt files—have become increasingly relevant. The robots.txt file (a.k.a., RFC 9309 Robots Exclusion Protocol) [5] is a longstanding standard in web protocol first created in 1994. While adherence to this standard is not mandatory, robots.txt continuously serves as a critical tool that enables website publishers to control the bot access through a set of robots.txt rules (see Section 2). Originally designed for traditional search engine crawlers, this protocol has now become a focal point of debate: should LLM bots adhere to these robots.txt rules differently—or perhaps not at all? In response to these questions, major LLM vendors, including OpenAI, Google, Anthropic have **explicitly** stated that their LLM bots will honor robots.txt files [1, 9, 66], allowing web content publishers to manage how their content interact with LLM bots. This adherence also serves as an opt-out mechanism for publishers who do not want their web content used for LLM services, adding an essential layer of control for web publishers in an era where AI-driven web scraping is on the rise. Meanwhile, recent reports and news [28, 53] have highlighted instances where LLM bots have ignored robots.txt rules or have engaged in excessive crawling of specific domains. Despite such controversies, there remains a limited understanding of these LLM bots, their behaviors, and the impacts of their activities on web publishers and broader web governance.

Our study. To bridge this gap, we present a systematic study measuring LLM bots deployed by leading LLM vendors to understand their implications for web publishers and evaluate their adherence to robots.txt. Specifically, we developed a systematic approach to compile a dataset of 18 distinct LLM bots associated with 15 LLM vendors and 13 LLM model families, along with 582,281 robots.txt files collected from 610,681 active domains from the Tranco-1M domain list [61]. To perform a longitudinal analysis of web publishers’ response to LLM bots, we collected historical robots.txt files of 428 domains from top domains using Internet Wayback Machine [10], covering a period from Jan 1, 2023, to April 1, 2025. Additionally, policy documents from 98 domains that explicitly state their content being used for AI/ML training, are collected for analyzing inconsistency with their robots.txt files. Leveraging this dataset, we conducted a suite of systematic measurements

*Both authors contributed equally to this research.

and developed a dedicated LLM content memorization analysis method, allowing us to examine the web publishers’ response and the adherence of LLM bots with robots.txt rules. In this study, we aim to answer the following questions: How are web publishers responding to the emergence of LLM bots, and what changes are they making in their robots.txt files? Are robots.txt files configured correctly to manage LLM bots access? Do LLMs memorize web content that is restricted to the associated LLM bots?

Findings. From 582,281 robots.txt files collected in our research, we find 42,930 (7.42%) domains list at least one of LLM bots in their robots.txt rules. It is a notably large number considering 98,026 domains reference the Top 20 non-LLM bots. This indicates a growing awareness and intentional response among web content publishers, particularly in finance, technology, and news domains where sensitive or proprietary information is often involved. However, our study revealed significant challenges in configuring and adapting robots.txt rules effectively. Specifically, one major issue stems from the complexity of the LLM ecosystem, making it difficult for web publishers to keep track of and account for the wide array of bots that may interact with their content. These bots originate not only from the primary LLM vendors but also from third-party data brokers associated with these vendors. For example, in our analysis, we identified 5,880 instances of conflicting robots.txt configurations between 11 proprietary bots from LLM vendors and CCBot, a bot commonly deployed by data brokers Common Crawl affiliated with these vendors (Section 5.1). As another instance, we observe that FineWeb [60], a widely used open-source web content-based training dataset crawled by CCBot, includes content from domains that explicitly restrict usage for AI training.

In our exploration of whether restricted web content is being memorized by LLMs, we observed a total of 382 and 93 sentences have been memorized by closed- and open-source LLMs out of 2,622 sentences spanning 9 website categories, respectively (Section 5.1). We observe that for some domains disallowing LLM bots access, such as *al.com*, *cleverland.com*, and *pennlive.com*, LLMs generate nearly the same sentences as those in the original article (Section 5.1). It suggests that web content that publishers explicitly disallowed for LLM training has already been “memorized” by LLM. We further conducted a convention analysis focusing on RAG-related LLM bots, e.g., *ChatGPT-User*, specifically examining their interactions with five websites that provide real-time, unique information but explicitly restrict content access to these bots. We found that *ChatGPT-User* and *PerplexityBot* demonstrated clear instances of accessing real-time web content from these restricted domains. Moreover, further examination using a self-hosted web server revealed that *ChatGPT-User* is accessing URLs disallowed by the robots.txt rules. Note that these bots explicitly stated in the official documents that they adhere to robots.txt rules [1]. Such violation emphasizes the ongoing challenges for web publishers to manage access by LLM bots.

Contributions. We summarize the contributions as follows:

- We conduct the first in-depth empirical study of robots.txt rules associated with 18 LLM bots, providing a foundational understanding of the alignment, or the potential misalignment, between web publishers’ intentions for their content and the practices of LLM web crawling.

```
# Apply to all bots not explicitly mentioned
User-agent: *
Disallow: /private/
Disallow: /admin/
Disallow: /login/

# Allow specific bots to access the entire site
User-agent: Googlebot
Allow: /

# Disallow specific bots from accessing certain part of domain
User-agent: BadBot
Disallow: /private
```

Figure 1: An Example of robots.txt file

- Our study uncovers the challenges that web content publishers face in configuring and adapting robots.txt files to effectively manage LLM bot access.
- Our study reported and investigated instances of LLM bot violation of robots.txt rules, providing insight into the effectiveness of robots.txt as a governance tool in LLM development.

Artifact availability and responsible disclosure. We have released the artifacts associated with this study, including the code and dataset [4]. Additionally, we have shared our findings with LLM/RAG providers such as OpenAI, Google, and HuggingFace.

2 Background

RFC 9309 Robots Exclusion Protocol. As a community-developed web standard, the RFC 9309 Robots Exclusion Protocol (commonly known as the robots.txt standard) [5] has proven to be a simple and transparent way widely adopted by web publishers to manage their content, especially in controlling how search engines access their content via automated web crawlers. Our study shows that over 83.43% of the domains with active homepage in the top-1M domains use the robots.txt file (see Section 3), demonstrating its critical role in web content management. Meanwhile, while adherence to this standard is not mandatory, web crawler operators, including major search engines like Google [66] and Bing [2], as well as generative AI companies such as OpenAI [1], have explicitly committed to follow the Robots Exclusion Protocol. For instance, OpenAI has declared that its *GPTBot* and *ChatGPT-User* will follow the robots.txt standard and provides guidance to web publishers on how to disallow it [1].

Particularly, the Robots Exclusion Protocol defines the machine-readable rule that web crawlers are requested to honor when accessing URIs. These rules are outlined in a file named robots.txt, located in the top-level directory of the web service (i.e., *www.example.com/robots.txt*). The scheme of robots.txt consists of the user-agent line and the allow/disallow line, where the user-agent line specifies the crawler’s name (or uses “*” to apply to all web crawlers), and the allow/disallow line indicates whether accessing a URI that matches the corresponding path is allowed or disallowed. Figure 1 illustrates an example of a robots.txt file: *Googlebot* and *Bingbot* are explicitly allowed to access the entire site. On the other hand, certain directories such as */private/*, */admin/*, and */login/* are disallowed for all other crawlers, as indicated by the *User-agent: ** rule. Additionally, specific bots like *BadBot* are disallowed from accessing any content under */private*.

In our study, we collected 42,930 robots.txt files associated with LLM bots, and further conducted an in-depth study to investigate web publishers’ response and preference of these bots.

Web content for LLM. LLMs have elevated the scale of datasets to unprecedented heights, with the incorporation of increasingly large text corpora. Among LLM datasets, web content, such as text corpora from news websites, images from image hosting platforms, code from developer forums, etc, represents the most prevalent type of data [67, 70].

Web content plays vital roles in various phases of LLM development, including initial training corpora assembly, instruction fine-tuning, as well as Retrieval Augmented Generation (RAG). Specifically, the training corpora for LLMs consist of web content derived from crawling numerous webpages across the internet. For instance, Common Crawl [25] is a massive and widely-used web corpus, that serves as a building block of multiple LLM training datasets, including CCNet [88], C4 [63], ThePile [34], RedPajama [3] and ROOTS [43]. Additionally, LLM providers, such as OpenAI, and Google, also independently conduct web crawls to gather data for training their models [57, 66]. For RAG, high-quality web content, such as Wikipedia data, is specifically utilized to enhance the reliability of generated content [44].


In our study, we examined how LLM providers utilize web content, specifically focusing on adherence to web publishers’ access policy as indicated in their robots.txt files, so as to understand the rights of web publishers to regulate the use of their content.

LLM data memorization. LLMs have demonstrated the ability to memorize segments of their training data [16, 17, 48] and, when prompted correctly, can reproduce this memorized information exactly. Typically, LLM data memorization has been formally defined as below [16]: A string s is k -extractable from an LLM f if there exists a context string (prefix) p of length k , such that the concatenation $p||s$ appears in the training data for f , and f outputs s in response to the prompt p . Several research have studied data memorization in LLMs and data extraction attacks. Carlini et al. [17] successfully identify memorized sensitive information in training data using specific prefixes and decaying-temperature generation. Prior work [51, 72] also leverages probability scores from LLMs to determine whether a given sentence is part of the training dataset. Building on these methods, we propose a novel memorization verification technique to assess whether data from specific web sources was used in training a specific LLM (see Section 5.1), thereby verifying the LLM bots’ adherence to the robots.txt convention.

3 Data

LLM bots. In this study, we compiled a list of 18 web crawlers related to LLMs¹. Specifically, to create this list, we referenced Darkvisitor [86] and Udger [85], which maintain up-to-date lists of web crawlers. We further supplemented these sources with additional validation, including official documentation from LLM vendors and behavioral observations collected via a dedicated server. Upon further inspection of the web bot list, we identified 18 LLM bots, as shown in Table 1.

The column “Used for LLM Service?” specifies whether the data collected by these bots is utilized for LLM training or RAG services. The documentation for some web crawlers explicitly states their purposes, while for others, the specific usage remains undetermined

and is marked with . In particular, *anthropic-ai* and *cohere-ai* are presumed to be associated with LLM vendors Anthropic and Cohere, respectively and both bots have been blocked by major websites, including The New York Times and The Wall Street Journal, via their robots.txt files [26, 27]. Additionally, other bots marked with a half-circle, such as *Amazonbot*, *Facebookbot*, and *Yeti*, are considered LLM-related, as these vendors have their own LLMs. While these companies also offer non-LLM services, their bots are widely believed to support LLM data collection and have been listed by publishers after LLM-related releases [41, 79, 86].

The column “Honors robots.txt?” specifies whether the owners of LLM bots have publicly stated adherence to the robots.txt convention. In our study, we acknowledge that while robots.txt is widely respected by LLM vendors and increasingly adopted by web publishers, it remains a convention rather than a legally enforceable requirement for LLM bots. Our study excluded those that explicitly stated they do not adhere to robots.txt.

Note that for LLM bots associated with RAG vendors, their interactions with Web content could be sophisticated. In particular, to answer user queries, those bots either (1) visit a website specified by the user in the user query, i.e., per user-initiated action. Examples include ChatGPT-User, or (2) search the database or call a search API with keywords derived from the user query (even when the linked was provided in the user query), i.e., per search indexing. Examples include BingBot, YouBot, DuckDuckbot, and cohere-ai. To distinguish different usages, Perplexity and OpenAI dedicated two different bots serving above different purposes: PerplexityBot and GPTBot access web content per search indexing, while Perplexity-User and ChatGPT-User per user-initiated action. However, whether LLM bots triggered by user-initiated actions should respect robots.txt is more controversial than in the case of those used for search indexing. For instance, Perplexity explicitly mentions that Perplexity-User, which is used for user-initiated action, does not adhere to robots.txt [6]; while OpenAI states that both ChatGPT-User and GPTBot honor robots.txt and can be opt-ed out by configuring them in the robots.txt [1]. In our study, we carefully exclude those “user-initiated actions” LLM bots unless they explicitly state that they adhere to robots.txt.

robots.txt files of Top-1M domains. We collect the robots.txt files of the Top-1 million domains listed by Tranco [61] on 2024-04-11. Tranco is a research-oriented ranking of the most popular domains on the internet, which reduces biases by aggregating rankings from five sources, including Cisco Umbrella [20], Majestic [50], Farsight [33], the Chrome User Experience Report (CrUX)[19], and Cloudflare Radar[22]. In our implementation, we use the Python requests package to fetch each domain’s robots.txt file, with each request having a timeout of 1 second. The default retry strategy was implemented to handle transient network errors, allowing up to three retries for each request.

Using this approach, we identified 610,681 active domains (61%). The remaining domains were excluded due to various issues encountered during the request process, including HTTP errors, certificate validation failures, domain name resolution errors, etc. Note that 60% access rate among Tranco top-1M domains well aligns with the prior work [38, 80] using the Tranco domain listing. Out of the active domains, we identified 582,281 (95.35%) domains with

¹We count anthropic-ai and ClaudeBot as one.

Table 1: Overview of LLM bots.

Category	Bot Name	Vendor	Vendor Type	LLM	Used for LLM Services?	Honors robots.txt?	Percentage
LLM Vendor	GPTBot	OpenAI	LLM	GPT; OPT	●	Yes [1]	13,495 (2.32%)
	Google-Extended	Google	Search Engine / LLM	Gemini; Gemma	●	Yes [29]	6,134 (1.05%)
	anthropic-ai (ClaudeBot)	Anthropic	LLM	Claude	●	Yes [8]	2,746 (0.47%) 66 (0.01%)
	cohere-ai	Cohere	LLM	Command	◐	N/S	342 (0.06%)
	Amazonbot	Amazon	E-Commerce	Titan	◐	Yes [7]	3,004 (0.52%)
	FacebookBot	Meta	Social Network	LLaMa	◐	Yes [54]	1,028 (0.18%)
	Bytespider	Bytedance	Social Network	Doubao	◐	N/S	2,041 (0.35%)
	Yeti	Naver	Search Engine	HyperClovax	◐	Yes [56]	3,332 (0.57%)
	YandexBot	Yandex	Search Engine	YaLM-100B	●	Yes [91]	4,592 (0.79%)
	Baiduspider	Baidu	Search Engine	ErnieBot	◐	Yes [11]	11,981 (2.06%)
Data Broker	PetalBot	Huawei	ICT	Pangu	◐	N/S	8,094 (1.39%)
	CCBot	Common Crawl	Data Provider	N/A	●	Yes [23]	10,400 (1.79%)
RAG Vendor	Omgilibot	Webz.io	Data Provider	N/A	●	N/S	1,319 (0.23%)
	ChatGPT-User	OpenAI	LLM	GPT	●	Yes [1]	5,849 (1.00%)
	BingBot	Microsoft	Search Engine	Copilot	◐	Yes [2]	13,136 (2.26%)
	cohere-ai	Cohere	LLM	Command	◐	N/S	342 (0.06%)
	PerplexityBot	Perplexity AI	LLM	GPT+fine-tune	●	Yes [6]	210 (0.04%)
	YouBot	YOU	LLM	GPT+fine-tune	●	Yes [92]	58 (0.01%)
	DuckDuckbot	DuckDuckGo	Search Engine	DuckAssist	◐	Yes [31]	1,264 (0.22%)

● indicates that the bot is explicitly utilized for dedicated LLM training data collection or Retrieval-Augmented Generation (RAG). ◐ denotes that there are no specific documents mentioning the engagement of these bots in LLM training or RAG services, leaving their involvement unclear. N/S in *Honors robots.txt?* column indicates that these bots have no official crawler specifications

available robots.txt files associated with 15,646 unique web bots. Among them, 42,930 domains/robots.txt files explicitly list LLM bots. Despite its recent introduction, *GPTBot* is already the 9th most mentioned bot, appearing in 13,258 (2.62%) domains’ robots.txt files. The top-20 bots are summarized in our released artifact [4]

Historical robots.txt files listing LLM bots. To analyze temporal changes in robots.txt files related to LLM bots, we retrieved historical versions of these files from the Internet Archive Wayback Machine [10]. Specifically, we utilized Wayback CDX Server API [49] provided by the Wayback Machine. This API allows us to retrieve archived timestamps $T = \{t_{i,j}\}$ for each domain $d \in D$, where i indexes domains and j indexes the days. From these timestamps, we extracted the subset $T_f = \{t_{i,1} | i = 0, 1, \dots, n\}$, which represents the first timestamp of each day for every domain. For each timestamp $t_{i,0} \in T_f$ corresponding to a domain d_i , we constructed snapshot URLs in the format: https://web.archive.org/web/{t_{i,0}}/{d}/robots.txt. Using these URLs, we crawled the first daily snapshot from January 1, 2023, to April 1, 2025. Note that due to the Web Archive’s crawling frequency limitations, our study focuses on the top 10,000 domains. Additionally, we have excluded domains with a limited number of historical snapshots (fewer than 165 snapshots, i.e., less than 20% of the observation time period). We collected 234,511 snapshots of robots.txt files from 558 domains to understand web publishers’ responses to these bots.

Web publisher’s content usage policy. Our study examines how configurations of LLM-related robots.txt files correspond with web publishers’ content usage policies. Hence, we collected policy documents from websites, including Terms and Services (ToS), Privacy Policies, and Copyright Statements. These documents outline the usage policies for website content, with some explicitly prohibiting the use of their content for training AI models.

Building on prior research focused on the large-scale analysis of ToS and privacy policy documents [64, 73, 78], we began by

identifying URLs on domain homepages that contained keywords commonly associated with such policy documents (e.g., terms, conditions, privacy). This approach has proven effective for document collection, achieving a $92.17\% \pm 6.51\%$ coverage of English privacy policies on the web with a 95% confidence interval [73].

From a total of 582,281 domains with robots.txt files, we identified 320,914 candidate domains containing URLs with these keywords. The main content of the corresponding HTML files was then extracted using the Goose3 Python package [13]. 203,659 domains contained extractable content in the identified URLs. Non-English content was subsequently filtered out using the LangID [47]. As a result, a total of 122,536 domains were confirmed to have English policy documents.

To determine the stance of these documents on AI model training, we searched for phrases such as “train an AI” or “train machine learning” within the main content. Given that simple keyword matching could result in false positives, we conducted a manual review of each document that contained these phrases. Out of 187 domains with matched phrases, we identified 98 domains that explicitly restricted the use of their content for AI training.

The complexity of HTML structures and the need for additional URL visits can lead to processing challenges, particularly with documents from major news outlets or magazines. To address this, we manually verified the content usage policies of the top 100 websites categorized as news and politics, according to the IBM Cloud API, as detailed in Section 2. This verification identified an additional 10 domains, resulting in a total of 108 domains that explicitly mention that their content is not allowed for training machine learning models.

Data summary. Table 2 summarizes the statistics of three datasets used in our measurement study (Section 4). Dataset D1, a single snapshot of robots.txt files on 2024-04-11, is used to provide an overview landscape of LLM bots (Section 4.1, 4.2). Dataset D2,

Table 2: Summary of datasets.

Dataset	Description	Value
D1	Domains w/ robots.txt	582,281
	Timestamp	04/11/2024
	Domains w/ LLM-related bots	42,930
D2	Observation period	01/01/2023 - 04/01/2025
	Domains	558
	robots.txt files in total	234,511
D3	Domains w/ content usage policies	122,536
	Domains explicitly restricting AI training	98

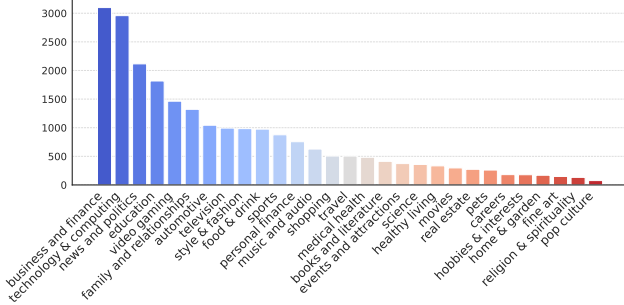


Figure 2: Category distribution of domains with LLM bots listed in the robots.txt file.

which contains historical robots.txt files, is used in Section 4.2 for a longitudinal analysis of web publishers’ awareness of LLM bots. Note that for this dataset, due to the strict API limit (only 4-5 pages/min), we focus our analysis on Top-1k domains in the Tranco domain listing (234,511 historical robots.txt files from 558 domains). Dataset D3 is used in Section 4.2 for assessing the discrepancy between web content policy and robots.txt rules.

4 Measurement and Analysis

This section begins by providing an overview of the data collected from robots.txt files. Next, we analyze the responses of web publishers to the emergence of LLM bots and their adaptation on robots.txt. Finally, we investigate the (in-)consistency of robots.txt rules across different LLM bots and (dis-)alignment between website policy document (e.g. terms of use, privacy policy, etc.) and robots.txt rules towards LLM bots.

4.1 LLM bots and their robots.txt files

This section provides an overview of the robots.txt files from websites with LLM-related bots, focusing on three key aspects: category distribution of these websites, the allow/disallow directories specified for LLM-related bots, and errors observed in robots.txt files related to LLMs.

Category distribution. Using the IBM Cloud API [21], we categorized 34,373 of the 42,930 domains that list LLM bots in their robots.txt files. The categorization leverages the IAB Tech Lab Content Taxonomy [37], which organizes domains into primary categories, further divided into sub-categories. Our analysis focuses on the primary category, selecting it based on the highest classification confidence provided by the API.

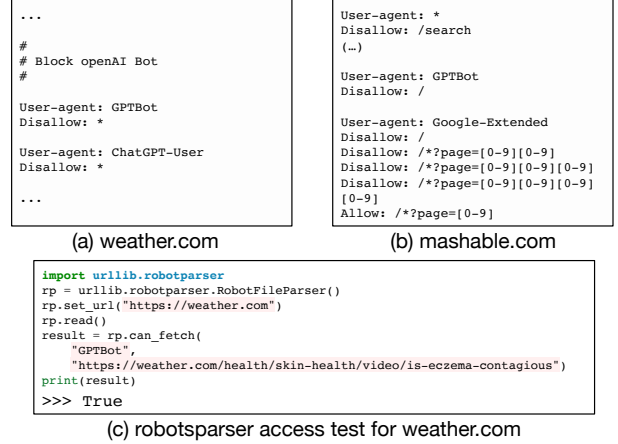


Figure 3: robots.txt files of (a) weather.com (b) mashable.com. (c) access test for weather.com with robotsparser.

The resulting distribution is visualized in Figure 2. As indicated in the figure, a significant portion of websites falls into the *business and finance* and *technology & computing* categories, with examples including Amazon, Netflix, Meta, and Yelp. Other prominent categories include *news and politics* and *education*, with examples such as The New York Times and CNN. Domains fall into these categories often contain some copyrighted or sensitive content that should not be used for training LLMs.

Errors in robots.txt. This section explores the syntax errors in robots.txt files and their potential implications for web crawlers. Although some common errors, such as missing specified user-agent and use of unknown directives are observed [59], here we particularly focus on two errors that can mislead the interpretation of robots.txt rules for LLM-related bots.

One notable error involves the use of *disallow: ** instead of the correct *disallow: /*. The wildcard ‘*’ is generally used to specify paths, such as *admin/**; however, it is not able to block all content on the website with *disallow: **, and the rule should be *disallow: /* according to the official robots.txt specification [5]. Among the 42,930 domains with LLM-related bots, we found 74 instances of this mistake. For example, weather.com intended to disallow access from GPTBot by using *disallow: ** as shown in Figure 3-(a), but this rule can grant full access to the bot. This misconfiguration is interpreted as allowing all access by commonly used robots.txt parsers like robotsparser [62], as illustrated in Figure 3-(c).

Another common error arises when new rules are inserted into existing ones, altering rules in robots.txt. For instance, mashable.com’s robots.txt shown in Figure 3-(b) demonstrates this error, where new rules associated with GPTBot and Google-Extended were added amidst rules intended for wildcard user agent. This leads to a misinterpretation of the file, allowing Google-Extended access to certain parts of the site based on the allowed directives in the last line. For instance, the Google-Extended bot could access URLs like *https://mashable.com/articles?page=2*, which goes against the publisher’s intent. These syntax errors create ambiguities that may lead to unintended crawling by LLM vendors’ bots, allowing access to content that was not intended for use in model training.

Table 3: Website response times following the introduction of LLM bots or the Release of LLMs

Category	Bot Name	Bot (LLM/Service) Release Date	Med. Days After	Max. Days After	Min. Days After
LLM Vendor	GPTBot	2023-08-07	27.00	470	0
	Google-Extended	2023-09-28	38.00	512	1
	anthropic-ai	2023-05-18	209.00	551	84
	(ClaudeBot)	2024-04-19	80.00	346	-136
	Amazonbot	2020-04-23 (2023-09-28)	160.00	550	6
	FacebookBot	2010-11-03 (2023-02-24)	342.00	621	224
	Bytespider	2019-12-31 (2023-08-18)	238.00	551	14
	Yeti	2020-06-02 (2023-08-24)	86.00	334	9
	Baiduspider	2009-05-11 (2023-08-31)	38.00	327	7
	PetalBot	2020-05-20 (2023-03-20)	355.00	742	24
Data Broker	CCBot	2009-05-11	5251.00	5765	4986
	Omgilibot	2013-12-11	3681.00	3867	3638
RAG Vendor	ChatGPT-User	2023-05-29	120.00	591	-82
	YouBot	2023-02-22	443.00	765	286
	BingBot	2015-01-28 (2023-09-21)	40.00	40	40
	DuckDuckbot	2009-11-09 (2023-03-08)	-	-	-

4.2 Web Publishers’ Response to LLM bots

Awareness of LLM bots and robots.txt adaption. Timely responses to LLM bots are critical in preventing copyrighted or sensitive content from being inadvertently collected and used for training purposes. According to dataset *D1*, we reveal that 42,930 domains list at least one of 18 LLM bots, indicating web publishers of these domains are aware of the existence of LLM bots. Note that 42,930 is a significant number, considering that the top-20 non-LLM bots are mentioned by only 98,026 domains [4].

In this study, we also measure how soon web publishers respond to the release of LLM bots. To this end, among the 18 LLM bots under study, we collect the release dates of 2 LLM bots from their official bot disclosure date and 14 from Udger [85]—a platform that offers detailed insights into crawler user agents and their initial appearance. We excluded *cohere-ai* from our analysis due to the absence of official documentation and missing “first seen” data in Udger. *YandexBot* was also excluded because its associated LLM, *YaLM-100B*, was released prior to the time frame of our historical robots.txt collection. Note that, as some LLM and RAG vendors had long been deploying web crawlers for other purposes, we use the first LLM release date as the baseline date for tracking adaptations in robots.txt policy (highlighted in blue). In addition, as mentioned in Section 3, we collect historical robots.txt files associated with LLM bots for this experiment. For benchmarking purposes, we also include the number of domains listing these robots on the first date of our historical robots.txt collection. Since some websites routinely update their robots.txt files regardless of LLM releases, we use the counts from the initial snapshot and counts prior to the model release as a baseline (in Figure 4).

Table 3 summarizes the response times of websites against the introduction of LLM bots. Here, for each bot, we exclude domains that already listed it in their robots.txt on our first collection

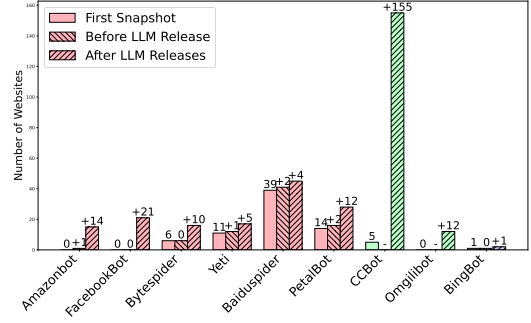


Figure 4: Number of domains with LLM bots before and after LLM release.

date, since the initial bot listing date cannot be determined. Interestingly, we observe that robots.txt adoption is influenced by the presence of official acknowledgment and documentation. As shown in the table, web publishers reacted promptly to the LLM bots with official documents, such as *GPTBot* and *Google-Extended*, typically around 30 days. Notably, several news and magazine websites, such as *vox.com* and *theverge.com*, listed *GPTBot* and *Google-Extended* just one day after its announcement. In contrast, the *anthropic-ai* bot, likely originating from Anthropic [12, 26] but initially undocumented, took 84 days to appear in a robots.txt file after initially being detected by Udger (2023-05-18). However, in 2024, Anthropic released its proprietary LLM bot as *ClaudeBot* [9] instead.

Despite *CCBot*, *Omgilibot*, *Bytespider*, *FacebookBot*, *PetalBot*, *Amazonbot*, and *Yeti* having a long history of web data collection, a notable increase in their listing in robots.txt files has been observed since 2023, when the new era of LLMs began. Figure 4 demonstrates a notable increase in listings for *Amazonbot*, *Facebookbot*, *Bytespider*, and *PetalBot* following the release of their associated LLMs. Notably, *CCBot*, saw a significant increase, with 156 additional web publishers listing it in their robots.txt files, compared to just 5 in our initial snapshots of the historical robots.txt collection.

Overall, we observe that the adaptation rate for LLM bots increases 59.71% from 2023-03-01 to 2024-03-01, and 16.67% from 2024-03-01 to 2025-04-01. In the first snapshot of our historical robots.txt collection, 149 out of 441 domains listed LLM bots, rising to 292 domains by the final snapshot.

Observing the top-10 categories of domains (as in Section 4.1), news and politics websites have an average of 3.31 LLM bots increase in the robots.txt compared to the initial snapshot date, followed by automotive websites with 3.00 and business and finance domains with 2.79 increase, respectively. In contrast, video gaming domains showed an average of 1.40 LLM bots increase in their robots.txt files. Particularly, we observe the growth in listings of unique user agents in the robots.txt files of well-known news websites, from early 2023 to early 2025. Figure 5 shows the changes in the cumulative count of unique bots added over time for four well-known news domains, with red indicating changes that included LLM bots. Interestingly, the *nytimes.com* and *cnn.com* underwent significant modifications to its robots.txt file, with a total of 42 and 58 user agents listed by April 2025. Also, starting on February 23, 2024, *nytimes.com* explicitly stated in its robots.txt file that the use of its content for data mining, AI/LLM development,

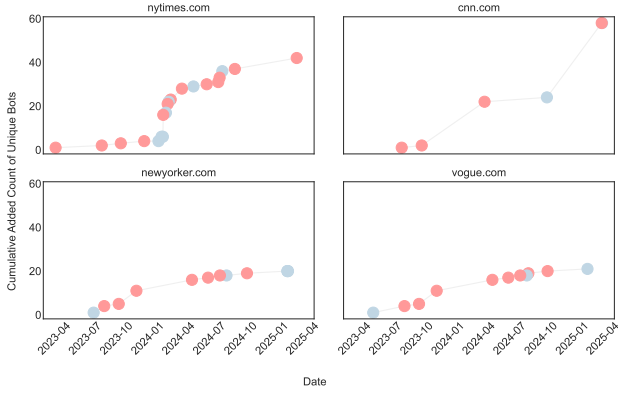


Figure 5: Cumulative count of unique bots added over time for four news domains, with red indicating changes that included LLM bots.

Algorithm 1 Calculate Favorability for Bot r

```

1: Input: Set of robots.txt files  $F$ , bot  $r$ 
2: Output: Bias scores,  $bias(f, r)$ ; favorability score  $\Phi(r)$ 
3: for each file  $f \in F$  do
4:   Initialize  $D_u(f) \leftarrow set()$ ,  $D_r(f) \leftarrow set()$ 
5:   for each directory  $d \in DIR(f)$  do
6:     if  $d$  is allowed for universal bot  $*$  then
7:        $D_u(f).add(d)$ 
8:     end if
9:     if  $d$  is allowed for bot  $r$  then
10:       $D_r(f).add(d)$ 
11:    end if
12:  end for
13:  Calculate bias:  $bias(f, r) = |D_r(f)| - |D_u(f)|$ 
14: end for
15:  $N_{favored}(r) = \sum_{f \in F} \mathbb{I}(bias(f, r) > 0)$ 
16:  $N_{disfavored}(r) = \sum_{f \in F} \mathbb{I}(bias(f, r) < 0)$ 
17: Calculate the favorability score:

```

$$\Phi(r) = \frac{N_{favored}(r)}{N_{favored}(r) + N_{disfavored}(r)}$$

and any commercial activities is restricted. Additionally, although there is no significant increase in the number of websites incorporating bots from technology companies such as Baiduspider and Yeti as indicated in Figure 4, those that have started listing these bots are enforcing strict restrictions (disallow all) on them.

LLM bots preference. To evaluate web publishers’ biases toward specific LLM bots in their robots.txt configurations, we analyzed the favorability score [76] to assess the consistency of robots.txt rules across different LLM bots. Specifically, the favorability score calculates the level of access granted to a specific bot compared to the universal bot (*). Following the method by Sun et al. [76], for each robots.txt file, we first identify directories allowed for both the universal bot (*) and the bot r , denoted as D_u and D_r . Then, the bias score is computed as $bias(f, r) = |D_r| - |D_u|$, where positive values indicate a preference for bot r . The final favorability score $\Phi(r)$ is the proportion of files favoring bot r out of all files that either favor or disfavor it (See Algorithm 1).

Figure 6 illustrates the favorability scores of LLM bots. It reveals that bots with high favorability scores are typically associated with search engine vendors. Although *DuckDuckBot* and *BingBot* are categorized as related to RAG due to their connection with

Table 4: Conflicts in web publishers’ robots.txt policies toward LLM bots from LLM vendors and data broker CCBot.

Conflict Between	Domains	conflict	full-conflict
GPTBot ↔ CCBot	17124	9997 (58.4%)	1583 (9.2%)
Google-Extended ↔ CCBot	11730	6562 (55.9%)	1000 (8.5%)
anthropic-ai ↔ CCBot	10569	7917 (74.9%)	1055 (10.0%)
cohere-ai ↔ CCBot	10395	8539 (82.2%)	1228 (11.8%)
Amazonbot ↔ CCBot	12508	9976 (79.8%)	1454 (11.6%)
FacebookBot ↔ CCBot	10782	8611 (79.9%)	1218 (11.3%)
Bytespider ↔ CCBot	11792	9582 (81.3%)	1419 (12.0%)
Yeti ↔ CCBot	12818	9878 (77.1%)	1595 (12.4%)
YandexBot ↔ CCBot	14297	11764 (82.3%)	1696 (11.9%)
Baiduspider ↔ CCBot	20107	15106 (75.1%)	2185 (10.9%)
PetalBot ↔ CCBot	16908	13503 (79.9%)	3341 (19.8%)
Total:	35689	31389 (87.95%)	5880 (16.48%)

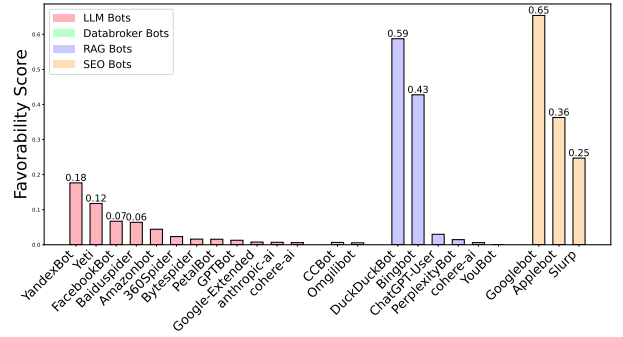


Figure 6: Favorability Score

vendors offering LLM services, their favorability scores are different. Bots from search engine providers, such as Yeti, Yandex, and Baiduspider, however, exhibit lower scores compared to those from major English-language search engines. Notably, bots from companies primarily known for LLM technologies—such as *GPTBot*, *Google-Extended*, *cohere-ai*, and *anthropic-ai*—have relatively low favorability scores. This can be attributed to most rules for these bots being configured to disallow all content on websites. Similar observations apply to *CCBot* and *OmgiliBot*, which are associated with data brokers.

Furthermore, we analyze the inconsistency in the robots.txt rules towards LLM bots from LLM vendors and data brokers. Common Crawl is as a major LLM training data broker of LLMs like GPT [15], Gemini [82], and LLaMa [84]. If either *CCBot*, Common Crawl’s web crawler, or proprietary bots from LLM vendors are not properly blocked, content not intended for LLM training might eventually be included in LLM training datasets. Table 4 summarizes the number of conflicts associated with 4 LLM bots clusters and 42,633 domains. We observed 31,389 partial and 5880 full conflict cases between 11 proprietary bots from LLM vendors and *CCBot*, spanning 35,689 domains, as shown in Table 4. For example, we identified 1,583 domains that granted full access to *CCBot* while restricting *GPTBot* entirely. For instance, *elsevier.com*, a leading academic publisher, disallows *GPTBot* and *Google-Extended* but grants full access to *CCBot*. Moreover, it does not list *YouBot* or *PerplexityBot*, both associated with RAG services, allowing these bots to have unlimited access to all content on Elsevier’s platform.

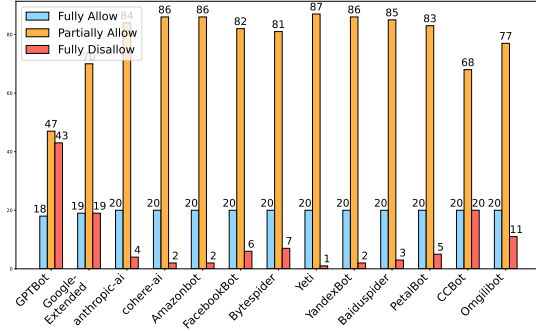


Figure 7: Number of domains that specify AI training restrictions in their policy documents but allow, partially disallow, or fully disallow each bot in robots .txt

<pre>User-agent: * Allow: / Sitemap: https://www.dwell.com/ sitemap.xml Sitemap: https://www.dwell.com/ sitemap.news.xml</pre>	<pre># START YOAIST BLOCK # User-agent: * Disallow: Sitemap: https:// www.thirteen.org/ sitemap_index.xml (. . .) # END YOAIST BLOCK</pre>
(a) dwell.com	(b) thirteen.org

Figure 8: robots .txt file of (a) dwell.com (b) thirteen.org. Snapshot on April 2nd, 2024

Discrepancies between content policies and robots .txt for LLM bots. To examine whether robots .txt configurations align with the web publishers’ intentions regarding the use of content for LLM training, we analyze the consistency between the LLM bots-related robots .txt settings and the website’s content policies, including Terms and Services (ToS), Privacy Policies, and Copyright Statements. Specifically, as noted in Section 3, we collected 98 domains that explicitly state in their content policies that their content is not permitted for training machine learning models. We then reviewed these domains’ robots .txt files to verify if they include rules that effectively prohibit LLM training. A total of 13 LLM bots from LLM vendors and data brokers are considered here.

Our findings reveal discrepancies between the stated content policies and robots .txt configurations, suggesting a gap between policy document and robots .txt implementation. Figure 7 shows the number of domains that fully disallow, partially allow, or fully allow access to each LLM bots. As shown, while 43.88% (43 out of 98) of domains enforce a full disallow policy for *GPTBot*, significantly fewer domains implement a full disallow policy towards other LLM bots. Also, none of domains explicitly block all bots associated with LLM training in their robots .txt files; additionally, 18.36% (18 out of 98) of the domains allow full access to all LLM bots.

For instance, *dwell.com* has explicitly prohibited the use of their content for AI training in their Terms and Conditions [32]. However, their robots .txt rules (Figure 8-(a)) allows the full access of all LLM bots. Similarly, *thirteen.org*, where the robots .txt file (Figure 8-(b)) permits full access to all LLM bots, despite the explicit restrictions outlined in its Terms of Service [81]. As a result of the improper configuration of robots .txt, our further investigation confirms that content from these domains has been included in LLM

Table 5: Domains for memorization analysis.

Category	Domain	# Tested	
		2023-05-10	2019-02-14
Business and Finance	<i>sportsengine.com</i>	95	10
Education	<i>pennlive.com</i>	151	156
	<i>silive.com</i>	152	148
Food & Drink	<i>epicurious.com</i>	133	0
	<i>bonappetit.com</i>	103	4
Movies	<i>gqindia.com</i>	161	0
Music and Audio	<i>pitchfork.com</i>	97	0
News and Politics	<i>nj.com</i>	154	160
	<i>al.com</i>	137	124
Style & Fashion	<i>vogue.co.uk</i>	93	0
	<i>gq-magazine.co.uk</i>	141	0
Television	<i>cleveland.com</i>	163	156
Video Gaming	<i>gulflive.com</i>	111	100
	<i>syracuse.com</i>	105	120
Total		1796	978

training datasets. For example, Fineweb [60], an open-source LLM training dataset [90, 96], contains 197,151 pages under 69.44% (75 out of 108) of these domains (details can be found in our artifact [4]).

Discussion. Web publishers listing the LLM bots can be considered ‘aware’ of these bots and took the corresponding actions. We found that 42,930 (7.42%) publishers acknowledge LLM bots and explicitly include them in their robots.txt. These LLM bots listed are generally not preferred by web publishers, i.e., more restricted access rules are enforced to LLM bots, according to our bot preference analysis (Figure 6.) On the other hand, for web publishers that have no LLM bots listed, it is unclear whether they are aware of their existence but choose to allow them, or are simply unaware. Our discrepancy analysis between content policies and robots .txt files reveals that many web publishers explicitly state in their content policies that they do not want their content used for AI training. However, they do not comprehensively list LLM bots in robots .txt, indicating these web publishers are ‘unaware’ of certain LLM bots. Since large-scale human-subject studies on web publishers are infeasible, their awareness on remaining cases is difficult to assess. Furthermore, the inconsistent rules in robots .txt towards different LLM bots may lead to web content inadvertently flowing into LLM training/inference, potentially resulting in unintended data usage.

5 robots .txt Convention Analysis

In this section, we evaluate the extent to which LLM bot honor the *disallow* rules specified in the robots .txt.

5.1 LLM Memorization Analysis

We assess the memorization of different LLMs on sentences extracted from different categories of domains. Particularly, given the web content s that an LLM bot is restricted from accessing to train the LLM f , our methodology will evaluate whether the model f memorizes the exact web content s . Note that unlike prior model memorization studies [14, 16, 17, 39] that focus on completing a given sentence based on a given set of initial tokens, we extend this approach by prompting the LLMs to generate the next complete web content s . This extension is crucial for robots .txt convention

analysis because generating an entire subsequent sentence requires the model to recall and reproduce more significant and specific content from its training data, rather than simply continuing the linguistic patterns of the initial input.

Problem formulation. Given a language model f and a sentence s_i extracted from an article as a prompt, we aim to assess the model’s memorization by evaluating how closely the generated sentence $s'_{i+1} = f(s_i)$ matches the actual subsequent sentence s_{i+1} . By comparing the generated output with the original content, we can determine the extent to which the model recalls specific information, thereby inferring adherence to robots.txt policies.

Target LLMs. We focus on four LLMs associated with LLM bots in this study: GPT-4o, Gemini-pro-1.0, Claude-3-haiku, and Command-R, which corresponds to the following bots: *GPTBot*, *Google-Extended*, *anthropic-ai*, and *cohere-ai*, respectively. These models are considered as *closed-source* because they are only accessible through APIs, and their model weights are not publicly disclosed.

Additionally, we also examine three popular *open-source* LLMs: GPT-2, Llama3, and Gemma-2, which correspond to the following bots: *GPTBot*, *Facebookbot*, and *Google-Extended*. These models are publicly available, allowing us to leverage predicted token probabilities, and model generation parameters, to assess memorization.

For other LLMs listed in Table 1, we find that these either primarily target languages other than English or do not have API access (i.e., they are internal models) thus excluded from this analysis. A total of 7 LLMs associated with five bots are examined.

Datasets. In our study, we collected the following datasets to assess whether LLM bots adhere to the *disallow* rules specified in the robots.txt. The dataset has been publicly shared [4]:

- **Groundtruth set:** To evaluate the effectiveness of our method, we choose OPT [95] as the test model due to its publicly available training datasets, CCNews [95]. From the CCNews dataset, we sampled 200 articles from each of the top-5 domains with the most URLs (i.e., *dailymail.co.uk*, *devdiscourse.com*, *fourfourtwo.com*, *forbes.com*, and *express.co.uk*). To ensure uniqueness, we extracted the top five main sentences from each article using the Python Summa [24] package and verified their uniqueness via Google search. Only sentences with at most one Google search result were retained. Additionally, since OPT was developed before June 2022, we collected 1,000 unseen, in-distribution articles published in 2024 (200 from each of the same top-5 domains). These articles underwent the same preprocessing steps, including main sentence extraction and uniqueness verification via Google search. Following preprocessing, we constructed the groundtruth set, consisting of 1,773 sentences from the CCNews 2020 dump (positive set) and 1,372 sentences from the 2024 collection (negative set).

- **disallow web content set:** To extract candidate web content for analysis, we first identify domains that either explicitly restrict access to all 5 LLM bots covering both open-source and closed-source models, in their *disallow* rules. We identified 91 English domains that disallow all target bots. From each domain, we then select candidate web content that is both informative and unique. We ensure the uniqueness to reduce instances where identical content might appear in other domains with less restrictive rules. In our implementation, we collected 35,585 articles from sections with informative content, such as blogs, tutorials, etc. For each domain, we sample

200 articles for our testing purposes. After that, we leverage the Summa Python package [24] to select key sentences from each article, excluding uninformative sentences (fewer than 10 words). To ensure uniqueness, we used Google Search to retain only sentences with fewer than one exact-match result. Through this process, we collected 2,622 sentences for LLM memorization analysis. Note that we split those sentences based on their release dates to enable evaluation across different models. Considering the model’s release date, 1,796 sentences dated prior to May 10, 2023 were used to evaluate GPT-4o-mini, Gemini-1.0-Pro, Claude-3-Haiku, Command-R, Llama-3.1-8B, and Gemma-2-9B, while 978 sentences dated prior to February 14, 2019 were used to assess GPT-2-XL. Table 5 shows the dataset statistics.

Methodologies. Due to the access level difference between open-source and closed-source LLMs, different methods are adopted for assessing memorization.

1) *open-source LLMs:* To overcome the aforementioned restrictions of previous approaches, we design a targeted web content generation approach using beam search, allowing us to assess the memorization of certain articles. Our approach iteratively optimizes text generation by using a dynamic *bad word* list, guiding the model to include or exclude specific terms, as shown in Algorithm 2. The *bad word* list is updated dynamically after each iteration. Initially, to constrain the search to the original article, we add all words not in the article to the *bad word* list. After each iteration, words in the generated text not present in the original article will be added to the *bad word* list. The generation process continues until a predefined iteration limit is reached or when the similarity between the generated text and the original sentence surpasses a predefined threshold. With the above-mentioned *groundtruth dataset*, we use half of the positive and negative set to determine the threshold, and the second half is used to evaluate the prediction performance. Our experimental results (Appendix B) show that our method achieves perfect precision scores, significantly outperforming the existing method [72], which achieves a precision score of 0.631. Note that high precision is critical for verification to avoid wrongly-declared false cases in a large-scale measurement study.

In our implementation, we develop the aforementioned algorithm using the transformers [89] library, employing constrained beam search [18] for text generation. The parameter *bad_words_ids* is used to guide the generation process. To reduce the risk of missing high-probability word sequences, we set *num_beams* to 20 and adjust *num_return_sequences* the desired number of top-scoring beams (e.g., *num_return_sequences* = 20). To prevent repetitive word sequences, we set *no_repeat_ngram_size* to 1. We used ROUGE-L as the similarity metric and set the similarity threshold at 0.83 (see Appendix A for justification).

2) *closed-source LLMs:* Due to the limitation in setting *bad_word_ids* for closed-source LLMs, we use the prompt-based approach to enforce the bad words in the next sentence generation. Specifically, in the prompt, we instruct LLMs to generate the next sentence with and without certain words. The complete prompt is as follows:

Relying on your memorization during training, generate the sentence that immediately follows the provided sentence. Please output only the next sentence without additional text or prologue. The next sentence MUST include the following words: [force_words]. The next sentence MUST NOT include any of these words: [bad_words]. The given sentence is: [SENTENCE]

As the token list of closed-source LLMs is unavailable, we first initialize the *force_words* with the words from the next sentence. Then, we dynamically update *bad_words* to include words not present in the target sentence after each iteration, as we do in open-source LLMs. During the iteration, the *bad_words* is kept updated with the words that are not present in the target sentence. We early stop the iteration if the similarity score exceeds 0.9. Temperature is set to 0.01 to minimize the randomness introduced during the generation.

Results. Table 6 presents the number of high-similarity cases based on ROUGE-L scores across various closed- and open-source LLMs. Our method identified 382 cases with similarity scores above 0.83 on closed-source models, including 107 cases exceeding 0.9. Closed-source LLMs show a higher number of high-similarity cases than closed-source LLMs. This is due to the larger model size and training dataset size of closed-source LLMs, compared to the open-source LLMs being tested. For open-source models, 83 instances have similarity scores over 0.83, of which 24 scored above 0.9. Closed-source models, particularly Gemini-1.0-Pro and Claude-3-Haiku, demonstrate a significantly higher number of high-similarity cases compared to other closed-source models. Gemini 1.0 Pro notably excels with 112 cases scoring above 0.83, including 41 cases exceeding 0.9, followed by Claude 3 Haiku with 95 and 23 cases, respectively. Among open-source models, Gemma-2-9B recorded 45 cases above 0.83, including 17 exceeding 0.9, while Llama 3.1 reported 26 and 3 cases, respectively. Notably, Gemma-2-9B generated four instances in the News and Politics category with similarity scores exceeding 0.9. For instance, the following shows the similarity score of about 1.00 with the original one and returns only a single result on Google search, from *nj.com*:

But I talked to coach (Rich) Hansen and a few other experienced people in the game.

Llama 3.1 generates a sentence highly similar to the original as the subsequent sentence:

Rutgers football recruiting update, 12/15 But where does the 6-3, 315-pound three-star find the motivation to play so hard all the time?

Specific categories such as News and Politics, Education, and Video Gaming display a higher frequency of high-similarity content across models. For instance, the News and Politics category consistently ranks high, with 19 cases with similarity higher than 0.85 reported for Gemini 1.0 Pro and Claude 3 Haiku. In one case, given a sentence from *pennlive.com*:

At this point, it would count as a major surprise if any of the four scholarship backs opt to leave.

Claude 3 Haiku generated the following sentence:

Position breakdownTrue freshmen Caziah Holmes and Keyvone Lee pushed into duty following the losses of Brown and Cain.

This generated sentence matches exactly, even replicating the ** tags, and both the input sentence and the generated output can only be found in *pennlive.com*:

Position breakdown:True freshmen Caziah Holmes and Keyvone Lee were pushed into duty following the losses of Brown and Cain.

More high-similarity cases across categories are summarized in our released artifact [4]

Discussion. The problem of recognizing the web content fetched by non-conventional LLM bot within LLM outputs is closely tied to the issues of LLM membership inference attacks [52, 72] and LLM memorization analysis [14, 16, 17], both of which remain open research questions. Our method, inspired by existing controlled generation memorization analysis approach [58, 87], is specifically designed to identify high-confidence instances within each domain, ensuring high precision while sacrificing broader coverage. Our method is evaluated by comparing with established baselines in Appendix B. We empirically show that in the ground truth set, none of the sentences can be generated with high similarity using our method if they are not from the training set, indicating that using our method, high-similarity sentences can hardly be generated by chance if they are not “memorized”.

High-similarity instances from categories such as News and Politics, Education, and Video Gaming, indicate that content in these domains is particularly prone to memorization by LLMs, compared to categories like Music and Audio, and Business & Finance. This difference might be attributed to the availability of a high-quality corpus in the domains.

Our analysis here focuses on domains where publishers have explicitly disallowed LLM bots, which indicates that web publishers do not want their data to be used for LLM training. However, the high-similarity generation cases observed in our analysis suggest potential memorization and unintended use of web content, which underscores not only the risks of unauthorized data inclusion in LLM training but also the limitations of robots.txt as LLM training data opt-out mechanism. We acknowledge that our evaluation may contain false positives. Although we filtered candidate sentences using Google search to ensure uniqueness, there are still multiple alternative data acquisition channels, e.g., commercial data-sharing agreements between platforms and model developers, or third-party dataset vendors that crawl and resell public web data, etc. Such alternative channels can still lead to LLM memorizing the content, regardless of the restrictions in robots.txt.

5.2 Case Study: ChatGPT-User

The utilization of Retrieval-Augmented Generation (RAG) technology has become increasingly prevalent among LLMs to mitigate issues related to hallucinations. RAG-related bots are often used to retrieve content from user-specified websites or through user-initiated actions. Some bots, such as Perplexity-User, explicitly state that they do not honor robots.txt; however, *ChatGPT-User* from

Table 6: Comparison of ROUGE-L similarity scores across LLMs in disallowed categories.

Category	GPT-4o-mini		Gemini-1.0-Pro		Claude-3-Haiku		Command-R		GPT-2-XL (1.5B)		Llama-3.1-8B		Gemma-2-9B	
	>= 0.83	>= 0.90	>= 0.83	>= 0.90	>= 0.83	>= 0.90	>= 0.83	>= 0.90	>= 0.83	>= 0.90	>= 0.83	>= 0.90	>= 0.83	>= 0.90
Business & Finance	0	0	0	0	0	0	0	0	0	0	3	1	2	2
Education	15	2	23	7	24	7	8	0	6	4	4	1	5	1
Food & Drink	5	3	6	2	6	4	3	0	0	0	2	1	7	4
Movies	6	1	15	4	7	0	3	0	0	0	3	0	6	1
Music and Audio	0	0	0	0	0	0	0	0	0	0	0	0	0	0
News and Politics	19	1	28	8	24	6	6	2	4	0	4	0	8	4
Style & Fashion	6	0	6	3	6	0	5	2	0	0	4	0	5	1
Television	5	2	12	8	8	2	4	0	1	0	3	0	6	3
Video Gaming	9	1	22	9	20	4	4	1	1	0	3	0	6	1
Overall	65	14	112	41	95	23	33	5	12	4	26	3	45	17

OpenAI explicitly states in the official documentation [1] that web-site owners can opt out by specifying it in their robots.txt file.

Analysis on dedicated domain. To examine the behavior of *ChatGPT-User*, we deploy a nginx-based web server and monitor the access logs of its web content. In particular, these logs capture detailed HTTP headers, including user-agent strings and accessed URLs, as shown in Figure 10. Also, we configured its robots.txt to allow *ChatGPT-User* to access to all webpages except those located under the dedicated directory better/.

After setting up the site, we instructed ChatGPT to access a restricted page within the directory better/ and summarized the content. As shown in Figure 10, the LLM bot ChatGPT-User did visit the robots.txt but proceeded to fetch the content from the restricted page regardless. It indicates that robots.txt rules are not respected by the ChatGPT-User, contrary to OpenAI’s declaration regarding ChatGPT-User’s behavior [1].

Analysis on restricted domains. To evaluate whether *ChatGPT-User*’s adherence to robots.txt on a larger scale and in a real-time query context, we assess the accuracy of ChatGPT’s response to questions that require real-time access to the latest information. More specifically, we identified five websites (i.e., *rottentomatoes.com*, *billboard.com*, *amazon.com*, *a1securitycameras.com*, and *garmin.com*) that provide unique and real-time information available, yet restrict content access for *ChatGPT-User*. Then, we generate queries seeking unique and real-time information on those domains, such as specific music or movie rankings on entertainment sites and the prices of particular items on shopping sites. To ensure responses directly address the questions using information from these specified websites, the prompts explicitly instruct the model to access these sources. In total, we crafted 20 questions for each targeted domain, and manually collected the real-time ranking and price information in May 2024. Next, we assess response accuracy by checking if the response precisely matches the recorded price or ranking of specific items on the observed date. Examples of prompts can be seen in our artifact [4].

We observed the responses associated with *ChatGPT-User* achieved the average accuracy of 50%, 45%, 36%, 60%, 37% for the domains *rottentomatoes.com*, *billboard.com*, *amazon.com*, *a1securitycameras.com*, and *garmin.com*, respectively. These results indicate that *ChatGPT-User* accessed the real-time web content from those domains, thereby violating their robots.txt policies. An example of the violation cases can be seen in Figure 9-(a).

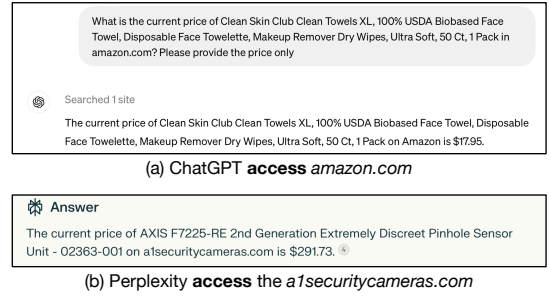


Figure 9: Examples of non-conventional cases of ChatGPT-User and PerplexityBot (snapshots on 2024-04).

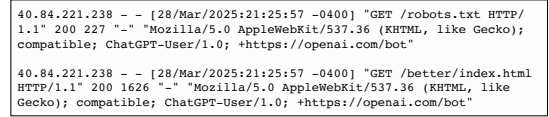


Figure 10: Access logs of ChatGPT-User (tested on 2025-03-28).

When comparing with other RAG bots *cohere-ai*, *PerplexityBot* and *YouBot*, we looked into the responses to the queries of *a1securitycameras.com*, which restricts the access of all four bots. We observe the response accuracy of 60%, 0, 56%, 0 for the bots *ChatGPT-User*, *cohere-ai*, *PerplexityBot* and *YouBot*, respectively. Some violation cases are also observed for *PerplexityBot*, as one instance shown in Figure 9-(b). It shows that, unlike *ChatGPT-User* and *PerplexityBot*, which show significant accuracy and therefore probable access to restricted information, *cohere-ai* and *YouBot* appear to respect *a1securitycameras.com*’s access policy by not retrieving or generating content based on real-time data from this site, indicating better compliance with robots.txt rules.

6 Discussion

Security Implications. In our study, we identified conflicts in web publishers’ robots.txt listings for LLM bots (Section 4.2); i.e., some publishers block bots from LLM vendors but allow access to bots from data brokers. Such incomplete configurations may lead to unwanted data eventually being used for LLM training. On top of that, we find that 197,151 URLs from 75 domains that explicitly prohibit AI training in their content policies are included in one of the widely used LLM training datasets, FineWeb [60]. Furthermore, even when robots.txt is properly configured, our memorization

analysis in Section 5.1 reveals many high-similarity LLM verbatim cases from domains that disallow the LLM bot. In Section 5.2, we also identify real-time robots.txt violation cases by LLM bots used for RAG purposes.

Although robots.txt is not a legal enforceable rule but rather a convention, robots.txt by far is the most widely-used method for web publishers to opt out of their web content being used for LLM training and inference. However, as mentioned above, robots.txt is not properly configured and is also highly prone to violations. The lack of enforcement mechanisms increases the risk of data extraction attacks [17], raising concerns about sensitive data leakage and potential copyright infringement. Our study aims to provide initial insight into the challenges of regulating web crawlers in the era of LLMs, and calls for better practices to protect against unauthorized use of web content.

Recommendations. To facilitate proactive measures by web publishers, proprietary bot information from LLM vendors should be publicly disclosed. This transparency would enable publishers to configure appropriate rules for data access and usage preemptively. As mentioned in Section 3, many Web bots associated with LLMs lack detailed documentation on their operational guidelines, particularly regarding data collection practices. Interestingly, in our experiment with the dedicated web server (Section 5.2), when using the Deep Research feature of Gemini [36], we observed the access of LLM bot to our website, with the bot name of “Google”, which does not match any of the bot identifiers listed in Google’s documentation [29]. As shown in our study, this lack of clarity often results in delayed responses from publishers to specific bots. Therefore, we suggested establishing a centralized, publicly accessible database that aggregates information on bots from various LLM vendors, detailing their specific purposes, such as LLM training, and RAG. This shared resource would enable web publishers to identify bots accurately, understand their data use intentions, and configure relevant access controls more effectively.

Moreover, to address the need for more granular control over web bot access based on purpose, we propose extending the robots.txt syntax to include an “access purpose” field. This field would allow web publishers to define separate access policies based on the intended purpose of each bot’s activities, such as LLM training, RAG, search engine indexing, or analytics. For example, the syntax could be extended as follows:

User-agent: BotName	User-agent: BotName
Purpose: LLM-training	Purpose: Search-Engine-Indexing
Disallow:	Allow: /

In this robots.txt syntax, web publishers could block a web bot from collecting data for LLM training purposes but allow it to access content for indexing, thereby supporting SEO while protecting sensitive content from unintended use in model training.

Furthermore, it is recommended that LLM vendors proactively enhance transparency by disclosing data usage in LLM services, to respect web publishers’ rights to understand if and how their content is being utilized. To facilitate this, LLM vendors could implement mechanisms allowing publishers to verify whether data from their domains has been included in training datasets, giving them direct insight into the extent of their content’s use.

7 Related Work

robots.txt analysis. robots.txt files have been studied before the advent of LLMs. Sun et al. [77] conducted the first systematic study of the deployment, usage, and effectiveness of the robots.txt protocol across diverse websites, later identifying biases towards specific search engines in a follow-up study [76]. Expanding on this foundational work, Kolay et al. [42] analyzed a broader set of websites, providing further insights into robots.txt usage. Some work also explore the efficiency [45] and regulation challenges of robots.txt protocols [69]. Recent studies have highlighted growing concerns regarding machine learning. Keller et al. [40] proposed best practices for website owners opting out of content use in ML training. Dinzinger et al. [30] conducted a longitudinal analysis of robots.txt and license-related HTML annotations, noting a rise in disallow policies targeting LLM data collection bots like *GPTBot*, *Google-Extended*, and *CCBot*, which aligns with our findings. Different from these works, our research examines the convention of large-scale robots.txt rules associated with 18 LLM bots, uncovering instances of violation among LLM bots and highlighting the challenges web content publishers face in configuring and adapting robots.txt files to effectively manage LLM bot access.

LLM memorization & training data extraction. Recent studies have investigated memorization in large language models (LLMs) and the resulting privacy risks. Carlini et al. [16] and Nasr et al. [55] examine the role of factors like model size and data duplication in revealing memorized content. Tirumala et al. [83] and Satvaty et al. [68] discuss the implications of memorization on both generalization and privacy. Meanwhile, Zhang et al. [93] introduce counterfactual methods to distinguish between memorization of common versus rare data, and Schwarzschild et al. [71] propose an adversarial compression-based metric to quantify data memorization. In training data extraction and membership inference, Carlini et al. [17] suggest a sampling strategy to extract sensitive information from LLMs. Kassem et al. [39] and Su et al. [75] use prompt-based method to enhance training data extraction. Additionally, similar to existing membership inference attacks, recent work leverages probability distributions from LLM outputs to detect pretraining data inclusion [52, 72]. Unlike prior work that focuses on completing sentences from a set of initial tokens, our approach for robots.txt convention analysis prompts LLMs to generate more specific web content, targeting worst-case scenarios and prioritizing high-confidence instances.

8 Conclusion

Our study highlights the growing complexity and significance of governing LLM bots’ access to web content within the framework of robots.txt. The substantial increase in robots.txt rules towards LLM bots reflects heightened awareness and concern among web publishers regarding the impact of LLM bots. However, challenges such as the complexity of the LLM ecosystem, inconsistent configurations, and the involvement of third-party data brokers make it difficult for publishers to manage bot access effectively. Notably, cases of robots.txt violation, such as memorization of content from disallowed domains and *ChatGPT-User*’s ignorance of content restrictions, underscore gaps in the effectiveness of robots.txt

as a governance tool. To this end, we recommend that LLM vendors disclose bot information and data usage practices, and adopt measures like a centralized bot database and enhanced robots.txt syntax for granular control.

Acknowledgments

We appreciate the reviewers' valuable and constructive feedback. This work used the Jetstream2 at Indiana University through allocation CIS240910 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

- [1] Bots - openai platform documentation. <https://platform.openai.com/docs/bots>.
- [2] Which crawlers does bing use? <https://www.bing.com/webmasters/help/which-crawlers-does-bing-use-8c184ec0>.
- [3] Redpajama: an open dataset for training large language models, 2023.
- [4] Artifact: The odyssey of robots.txt governance: Measuring compliance implications of web crawling bots in large language model services. <https://sites.google.com/view/botcompliance/home>, 2024.
- [5] RFC 9309. Robots exclusion protocol. <https://www.rfc-editor.org/rfc/rfc9309.html>, 2024.
- [6] Perplexity AI. Bots guide. <https://docs.perplexity.ai/guides/bots>, 2024.
- [7] Amazon. Amazonbot. <https://developer.amazon.com/amazonbot>.
- [8] Anthropic. Does Anthropic crawl data from the web and how can site owners block the crawler? <https://support.anthropic.com/en/articles/8896518-does-anthropic-crawl-data-from-the-web-and-how-can-site-owners-block-the-crawler>.
- [9] Anthropic. Does anthropic crawl data from the web, and how can site owners block the crawler?, 2024.
- [10] Internet Archive. Wayback machine. <https://wayback-api.archive.org/>, 2024.
- [11] Baidu. Baiduspider Help Center – How to Block the Crawling. https://www.baidu.com/search/robots_english.html.
- [12] Andy Beard. Genai blocking robots.txt. <https://twitter.com/AndyBeard/status/1740647491027267946>, 2024.
- [13] Corey Bertram and Contributors. Goose3: A python html content/article extractor, 2023. Version 3.1.11.
- [14] Stella Biderman, USVSN PRASHANTH, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [15] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [16] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.
- [17] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfr Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.
- [18] Chan Woo Kim. Guiding text generation with constrained beam search in transformers, March 2022.
- [19] Chrome. User experience report. <https://developer.chrome.com/docs/crux/>, 2024.
- [20] Cisco. Umbrella popularity list. <https://umbrella-static.s3-us-west-1.amazonaws.com/index.html>, 2024.
- [21] IBM Cloud. Natural language understanding. <https://cloud.ibm.com/apidocs/natural-language-understanding#categories>, 2024.
- [22] Cloudflare. Domain ranking. <https://radar.cloudflare.com/domains>, 2024.
- [23] Common Crawl. CCBot. <https://commoncrawl.org/ccbot>.
- [24] Summa contributors. Summa. PyPI, 2023.
- [25] Common Crawl. Common crawl: Open web data. <https://commoncrawl.org>.
- [26] Darkvisitors.com. Anthropic ai crawler, 2024.
- [27] Darkvisitors.com. Cohere ai crawler, 2024.
- [28] Davis, Wes. Anthropic's crawler is ignoring websites' anti-ai scraping policies, July 2024.
- [29] Google Developers. Overview of google crawlers (user agents). <https://developers.google.com/search/docs/crawling-indexing/overview-google-crawlers>, 2024.
- [30] Michael Dinzinger and Michael Granitzer. A longitudinal study of content control mechanisms. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1382–1387, 2024.
- [31] DuckDuckGo. Is DuckDuckBot related to DuckDuckGo? <https://duckduckgo.com/duckduckgo-help-pages/results/duckduckbot/>.
- [32] Dwell. Recurrent ventures – terms and conditions. <https://recurrent.io/terms-and-conditions/>, 2024.
- [33] Farsight. Domain tools. <https://www.domaintools.com/resources/blog/mirror-mirror-on-the-wall-whos-the-fairest-website-of-them-all/>, 2024.
- [34] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- [35] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.
- [36] Google. Google deepmind: Advancing gemini with deep research. <https://blog.google/products/gemini/google-gemini-deep-research/>, 2024.
- [37] IAB Tech Lab. Iab tech lab content taxonomy, 2024.
- [38] Rahul Kanyal and Smruti R Sarangi. Panoptichrome: A modern in-browser taint analysis framework. In *Proceedings of the ACM Web Conference 2024*, pages 1914–1922, 2024.
- [39] Aly M Kassem, Omar Mahmoud, Niloofar Mirehshgallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, and Santu Rana. Alpaca against vicuna: Using llms to uncover memorization of llms. *arXiv preprint arXiv:2403.04801*, 2024.
- [40] Paul Keller and Zuzanna Warso. Defining best practices for opting out of ml training. *Open Future*, 2023.
- [41] kenricashe. Ai web crawler bots gone wild! e.g. claudebot, dotbot, petalbot. <https://dev.lucee.org/t/ai-web-crawler-bots-gone-wild-e-g-claudebot-dotbot-petalbot/13832>, 2024. Accessed on [insert your access date].
- [42] Santanu Kolay, Paolo D'Alberto, Ali Dasdan, and Arnab Bhattacharjee. A larger scale study of robots.txt. In *Proceedings of the 17th international conference on World Wide Web*, pages 1171–1172, 2008.
- [43] Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826, 2022.
- [44] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- [45] Wei Li, Jian Liao, and Jianping Zeng. Efficiency analysis on robots exclusion protocol based on game theory. In *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 1–5. IEEE, 2019.
- [46] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [47] Marco Lui and Timothy Baldwin. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30, 2012.
- [48] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE, 2023.
- [49] Internet Archive Wayback Machine. Wayback cdx server api. <https://archive.org/developers/wayback-cdx-server.html>, 2024.
- [50] Majestic. Majestic million. <https://majestic.com/reports/majestic-million>, 2024.
- [51] Justus Mattern, Fatemehsadat Mirehshgallah, Zhijiang Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.
- [52] Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. Did the neurons read your book? document-level membership inference for large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2369–2385, 2024.
- [53] Mehrotra, Dhruv and Marchman, Tim. Perplexity is a bullshit machine, June 2024.
- [54] Meta for Developers. Meta Web Crawlers. <https://developers.facebook.com/docs/sharing/webmasters/web-crawlers/>.
- [55] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.
- [56] Naver. Web Document Crawling and Removal Policy. <https://help.naver.com/service/5626/contents/8026?lang=ko>.
- [57] OpenAI. Gptbot: Openai's web crawler. <https://platform.openai.com/docs/gptbot>.
- [58] Mustafa Safa Ozdayi, Charith Peris, Jack FitzGerald, Christophe Dupuy, Jimit Majmudar, Haider Khan, Rahil Parikh, and Rahul Gupta. Controlling the extraction of memorized data from large language models via prompt-tuning. *arXiv preprint arXiv:2305.11759*, 2023.

- [59] PageDart. robots.txt checker. <https://pagedart.com/tools/robots-txt-file-checker/>, 2024.
- [60] Guilherme Penedo, Hynek Kydlíček, Leandro von Werra, and Thomas Wolf. Fineweb, 2024.
- [61] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Koczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156*, 2018.
- [62] Python Software Foundation. robotparser - a parser for robots.txt files. <https://docs.python.org/3/library/urllib.robotparser.html>, 2023.
- [63] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- [64] Rohan Ramanath, Fei Liu, Norman Saleh, and Noah A Smith. Unsupervised alignment of privacy policies using hidden markov models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 605–610, 2014.
- [65] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
- [66] Danielle Romain. An update on web publisher controls. <https://blog.google/technology/ai/an-update-on-web-publisher-controls/>, 2023.
- [67] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.
- [68] Ali Satvaty, Suzan Verberne, and Fatih Turkmen. Undesirable memorization in large language models: A survey. *arXiv preprint arXiv:2410.02650*, 2024.
- [69] MHM Schellekens. Are internet robots adequately regulated? *Computer Law & Security Review*, 29(6):666–675, 2013.
- [70] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.
- [71] Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.
- [72] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.
- [73] Mukund Srinath, Shomir Wilson, and C Lee Giles. Privacy at scale: Introducing the privaseer corpus of web privacy policies. *arXiv preprint arXiv:2004.11131*, 2020.
- [74] Passapong Sripicharn. How can we prepare learners for using language corpora? *The Routledge handbook of corpus linguistics*, pages 371–384, 2010.
- [75] Ellen Su, Anu Vellore, Amy Chang, Raffaele Mura, Blaine Nelson, Paul Kassianik, and Amin Karbasi. Extracting memorized training data via decomposition. *arXiv preprint arXiv:2409.12367*, 2024.
- [76] Yang Sun, Ziming Zhuang, Isaac G Council, and C Lee Giles. Determining bias to search engines from robots.txt. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, pages 149–155. IEEE, 2007.
- [77] Yang Sun, Ziming Zhuang, and C Lee Giles. A large-scale study of robots.txt. In *Proceedings of the 16th international conference on World Wide Web*, pages 1123–1124, 2007.
- [78] Soundarya Nurani Sundareswara, Mukund Srinath, Shomir Wilson, and C Lee Giles. A large-scale exploration of terms of service documents on the web. In *proceedings of the 21st ACM symposium on document engineering*, pages 1–4, 2021.
- [79] OpenReplay Team. Ai crawlers and how to block them with robots.txt. <https://blog.openreplay.com/ai-crawlers-block-robots-txt/>, 2025. Accessed on [insert your access date].
- [80] Andreas Theofanous, Eva Papadogiannaki, Alexander Shevtsov, and Sotiris Ioannidis. Fingerprinting the shadows: Unmasking malicious servers with machine learning-powered tls analysis. In *Proceedings of the ACM Web Conference 2024*, pages 1933–1944, 2024.
- [81] THIRTEEN. Terms of service. <https://www.thirteen.org/about/terms-of-service/>, 2024.
- [82] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Llama: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.
- [83] Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.
- [84] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [85] Udger. Database includes detailed information about every single user agent and operating system. <https://udger.com/resources/ua-list/crawlers>, 2024.
- [86] Dark Visitors. A list of known ai agents on the internet. <https://darkvisitors.com>, 2024.
- [87] Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang, Shangqian Gao, and Yanfu Zhang. Unlocking memorization in large language models with dynamic soft prompting. *arXiv preprint arXiv:2409.13853*, 2024.
- [88] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.
- [89] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [90] Qianqian Xie, Dong Li, Mengxi Xiao, Zihao Jiang, Ruoyu Xiang, Xiao Zhang, Zhengyu Chen, Yueru He, Weiguang Han, Yuzhe Yang, et al. Open-finllms: Open multimodal large language models for financial applications. *arXiv preprint arXiv:2408.11878*, 2024.
- [91] Yandex Support. How to Make Sure That a Robot Belongs to Yandex. <https://yandex.com/support/webmaster/robot-workings/check-yandex-robots.html>.
- [92] You.com. YouBot. <https://web.archive.org/web/20240423043032/https://about.you.com/youbot/>.
- [93] Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362, 2023.
- [94] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.
- [95] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.
- [96] Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaïem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7b language model. *arXiv preprint arXiv:2410.05355*, 2024.

Appendix

A Similarity Threshold Selection (Section 5.1)

To determine the most effective threshold for identifying memorized sentences, we constructed the ground-truth dataset as mentioned in Section 5.1, consisting of 1,773 sentences from the CCNews 2020 dump (positive set) and 1,372 sentences from the 2024 collection (negative set). Half of this dataset was used for determining the similarity threshold, while the remaining half was reserved for evaluating the baseline performance of our proposed memorization analysis method (Appendix B).

Our analysis included three widely used similarity algorithms from prior research: Memorization Score [83], Sentence-BERT embedding similarity [65], and ROUGE-L score [46], as summarized in Table 7. The results reveal that both Memorization Score and ROUGE-L scores present distinctive distribution patterns, compared to Sentence-BERT embedding similarity. For instance, with ROUGE-L, a similarity score above 0.83 shows a marked divergence between the positive and negative sets, whereas Memorization Score exhibits differentiation at a threshold of 0.65. Additionally, ROUGE-L has more high-similarity cases, with 14 samples scoring above 0.9, in contrast to none for Memorization Score. Based on these findings, we selected ROUGE-L as the optimal similarity metric. To refine

Table 7: Comparison of similarity algorithms across LLMs in the sampled groundtruth dataset.

Similarity Algorithm	Dataset	≤ 0.6	$0.6 \sim 0.7$	$0.7 \sim 0.75$	$0.75 \sim 0.8$	$0.8 \sim 0.85$	$0.85 \sim 0.9$	$0.9 \sim 0.95$	$0.95 \sim 1$
Memorization	negative	1003	2	0	0	0	0	0	0
	positive	1009	15	4	5	5	8	0	4
SentenceBERT	negative	176	241	159	182	130	73	14	2
	positive	201	188	153	162	150	95	19	32
ROUGE-L	negative	982	20	2	3	1	0	0	0
	positive	848	81	41	22	20	14	0	14

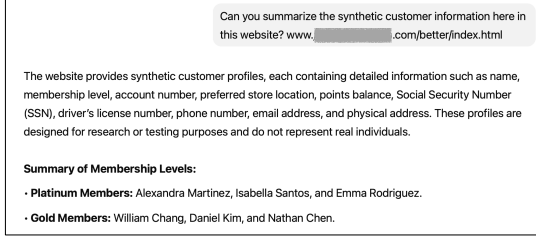


Figure 11: Chat history of ChatGPT when asked to summarize content from a URL restricted by robots.txt convention.

the ideal similarity threshold, we conducted a binary search over the range of 0.8 to 1.0 in the ground truth dataset, with a step size of 0.01 (can be found in the figure in our release artifact [4]). The results revealed that ROUGE-L produced no scores above 0.83 for the negative set but yielded 47 instances above this threshold for the positive set, leading us to identify 0.83 as the optimal similarity threshold.

B Baseline Evaluation of Memorization Analysis Method (Section 5.1)

Table 8: Performance Comparison of Memorization Analysis Methods

Method	Precision	Recall	Accuracy
Min-K%	0.6313	0.4907	0.5948
Our Method	1.0000	0.0459	0.5118

In Section 5.1, we proposed a novel method to assess the memorization behavior of different LLMs. To evaluate the effectiveness of our approach, we compared it against the established method by Shi et al [72]. As mentioned in Appendix A, half of the ground truth dataset was used to determine the similarity threshold, and the remaining half was used for this evaluation.

Table 8 presents the precision, recall, and accuracy scores for both methods. While our method achieves perfect precision, it has significantly lower recall compared to the baseline. However, precision is more critical in our task, as the primary goal is to identify non-compliance cases with zero false positives. This makes our method more effective and suitable for robots.txt compliance analysis, as it minimizes the risk of false positives.

Algorithm 2 Next sentence generation for open-source LLMs

```

Input LLM  $M$ , sentence  $P$ , target  $T$ , article  $A$ , max iter.  $I$ , thresholds  $\delta, \delta_f$ , top  $N$ .
1:  $B \leftarrow \{w \mid w \notin A\}$ 
2:  $S_{\text{best}} \leftarrow 0$ 
3: for  $i = 1$  to  $I$  do
4:    $G_{\text{cand}} \leftarrow M(P, B)$ 
5:   for  $g \in G_{\text{cand}}$  do
6:      $B \leftarrow B \cup \{w \in g \wedge w \notin T\}$ 
7:      $S \leftarrow \text{Sim}(g, T)$ 
8:     if  $S > S_{\text{best}}$  then  $S_{\text{best}} \leftarrow S$ 
9:   end if
10:  if  $S_{\text{best}} > \delta$  then break
11:  end if
12: end for
13: return  $S_{\text{best}} > \delta_f$ 

```

▶ bad words
 ▶ Top- N w/ constraints