

FindMyText: Robust, Scalable Detection of Text Containment in Large Web-Crawled Corpora

Lars Henry Berge Olsen¹, Pierre Lison¹, Martin Jullum¹ and Mark Anderson¹

¹Norwegian Computing Center, Oslo, Norway.

{lhbolsen, plison, jullum, anderson}@nr.no

<https://github.com/NorskRegnesentral/FindMyText>

<https://findmytext.nr.no>
(password: EMNLP2026)

Abstract

We present FindMyText, an open-source Python package designed to efficiently assess whether a given text appears, in part or in full, within a text corpus. The tool builds on prior techniques for document fingerprinting, but extends them with a novel mechanism to explicitly capture sequences of matching fingerprints. By identifying such chains, the tool can more reliably detect near-verbatim copies of a given text rather than mere textual similarities. This makes FindMyText particularly suited for verifying the presence of copyrighted material in a corpus. Leveraging a distributed, disk-based indexing framework, the system scales to large web-crawled datasets. Using a new benchmark for evaluating text containment methods, we show that FindMyText outperforms alternative approaches across three datasets (ArXiv papers, Wikipedia, and generic web content).

1 Introduction

Large Language Models (LLMs) depend on colossal amounts of data for their pre-training, much of it gathered through web crawling. Understanding the composition of this pre-training data is crucial for multiple NLP tasks, such as data selection and curation (Albalak et al., 2024; Parmar et al., 2024), enhancing model transparency and interpretability (Wang et al., 2023; Chang et al., 2024), and assessing potential copyright infringements or licensing violations (Karamolegkou et al., 2023; Longpre et al., 2024; Scharrenberg and Sun, 2025).

Pinpointing the exact texts that were part of an LLM’s pre-training data is, however, a non-trivial problem. Commercial LLM providers have so far been reluctant to disclose the exact content of their training data, often in fear of lawsuits. This led to the development of various techniques to conduct membership inference attacks on black-box LLMs (Oren et al., 2023; Shi et al., 2023), notably by taking advantage of LLMs’ memorisation abilities (Hartmann et al., 2023; Ahmed et al., 2026).

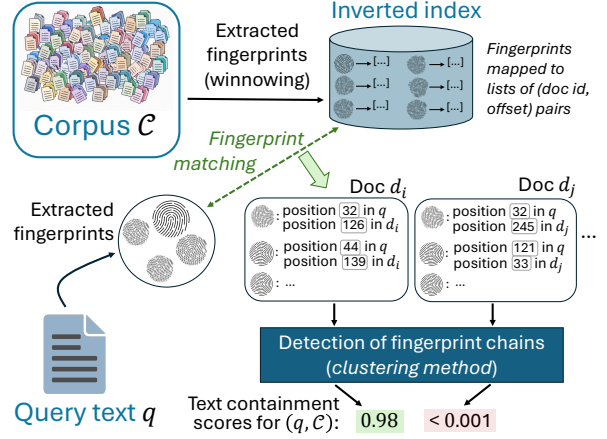


Figure 1: General sketch of the FindMyText approach.

The reliability of those attacks have, however, been called into question (Meeus et al., 2024; Zhang et al., 2024a; Liu et al., 2025), particularly for production LLMs that are carefully designed to avoid generating copyrighted content.

Transparency obligations introduced by recent regulations such as the EU AI Act (European Parliament and Council, 2024) may in the future compel LLM providers to disclose more details about their data sources. Nevertheless, even if the full training corpus were made publicly available, establishing whether a specific text (say, a copyrighted book chapter) is part of that corpus remains technically difficult. A text may indeed be spread across numerous websites, each with its own format, and mixed together with other pieces of content. Furthermore, pre-training datasets typically undergo multiple preprocessing steps, including text extraction (e.g. OCR), boilerplate removal, document segmentation, and various types of text normalisation and reformatting. Due to this extensive curation process, techniques for near-duplicate detection (Manku et al., 2007) may fail to reliably establish whether a text is part of a pre-training corpus.

This paper presents a novel method for the efficient detection of *text containment*, defined as

the task of determining whether an input text is wholly or partially included in a target corpus. This method is implemented in FindMyText, an open-source toolkit which is specifically designed to (1) scale to large corpora and (2) be robust to discrepancies between the text provided by the user and its counterpart in the corpus.

The approach builds upon *document fingerprinting*, a well-established paradigm in information retrieval (Schleimer et al., 2003). However, instead of simply computing the ratio of shared fingerprints, as often the case in near-duplicate detection, FindMyText explicitly identifies *chains of matching fingerprints*. This enables a more precise detection and localisation of shared text fragments – rather than merely estimating document similarity – which is essential for tasks such as copyright infringement detection.

Along with the release¹ of the FindMyText toolkit, this paper makes the following contributions:

1. A novel, chain-based method for locating shared text fragments (Sec. 3.2).
2. A synthetic benchmark to compare text containment methods (Sec. 5.1).
3. The experimental validation of the detection methods in FindMyText using the above benchmark on three corpora (Sec. 5.3).

This paper is structured as follows. The next section reviews prior work on text alignment and near-duplicate detection. Sec. 3 describes the methods used in FindMyText. Sec. 4 outlines the system design and implementation. Sec. 5 presents experimental results on corpora of various sizes and genres, namely Wikipedia, ArXiv, and generic web crawls. Finally, Sec. 6 concludes the paper.

2 Related Work

String alignment Given a query document q and a corpus $\mathcal{C} = \{d_i \text{ for } 1 \leq i \leq |\mathcal{C}|\}$, one straightforward approach to text containment is to use *sequence alignment* algorithms to measure the proximity of q with each text d_i composing the corpus $\mathcal{C}$. Those pairwise alignment algorithms typically rely on dynamic programming and can be divided into *global* and *local* methods.

Global alignment, such as the algorithm of Needleman and Wunsch (1970) employed to compute the edit distance between two strings, aims

to align entire sequences end-to-end. In contrast, local alignment focuses on finding best matching *subsequences*. The best-known local alignment method is the Smith and Waterman (1981) algorithm and its heuristic approximations, notably BLAST (Altschul et al., 1990). Given two texts d_1 and d_2 and a scoring system (specifying a substitution matrix and gap penalties), those methods can be employed to find the two regions in d_1 and d_2 yielding the highest alignment score. As the purpose of FindMyText is to identify whether a *portion* of an input text q can be found in a given corpus $\mathcal{C}$, local alignment stands closest to this paper.

Unfortunately, given their computational complexity – which is quadratic for Smith-Waterman – pairwise alignment algorithms cannot scale to large text corpora. Nevertheless, as shown in Sec. 5.1, they provide a useful *ground truth* to precisely quantify the degree to which two texts share a common subsequence (allowing for minor changes).

Duplicate detection The field of information retrieval offers several methods for efficiently assessing whether two documents are duplicates or near-duplicates (Manku et al., 2007). If the input document q exists verbatim in a corpus $\mathcal{C}$, exact matching can easily be performed through hashing. However, in most practical cases, the query document q and its counterpart d_i will differ, due to the multiple normalisation, formatting and segmentation steps applied on web-crawled corpora.

In those cases, *near-duplicate detection* methods can be used to measure the similarity between q and a document $d_i \in \mathcal{C}$. A common approach is to extract the k -grams from the two texts (so-called "shingles") and then compute the Jaccard similarity between the two resulting sets. As computing Jaccard similarities with all shingles remains costly, one can use compact probabilistic sketches such as *MinHash* (Broder, 1997), which builds a short document signature by retaining only the minimum hash values of its shingles under multiple hash functions. These signatures can in turn be indexed using *locality-sensitive hashing* (LSH), which groups together candidate texts with high similarity.

Alternatively, one can rely on *winnowing* to select representative fingerprints from a sequence of hashes (Schleimer et al., 2003). Unlike MinHash, winnowing preserves positional information, making it well suited for identifying shared passages. Winnowing constitutes the backbone of FindMyText, as described in the next section.

¹The toolkit is released under an MIT license.

Vector-based approaches Duplicates can also be detected by mapping documents to sparse or dense vector spaces and retrieving texts $d_i \in \mathcal{C}$ close to q under a chosen distance metric. SimHash (Charikar, 2002) uses compact binary embeddings with Hamming distance, while Retsim (Zhang et al., 2024b) learns an embedding space tailored for duplicate detection. SemDeDup (Abbas et al., 2023) relies on embeddings from pre-trained models to capture semantic similarity. Although effective for detecting globally similar documents, these methods are less suitable for detecting local containment when only a small portion of text is shared.

3 Approach

3.1 Preprocessing

Winnowing Given a text d_i , FindMyText first splits it into words, extracts its shingles (k -grams) and converts those into hashes. A text comprising n words will thus result in $n - k + 1$ hashes. To reduce the number of hashes to index, we rely on a *winnowing* mechanism. Given a sequence of hashes $h_1, \dots, h_{n-k+1}$, we slide a window of size w and select, for each window, the minimum hash:

$$m_i = \min\{h_j \mid i \leq j \leq i + w - 1\}.$$

The selected minima are kept as fingerprints, removing duplicates. This greatly reduces the number of hashes to retain while preserving the ability of the fingerprints to identify shared substrings: as shown by Schleimer et al. (2003), any substring of length at least $k + w - 1$ is guaranteed to yield at least one common fingerprint.

Indexing For a given text d_i , winnowing returns a list of fingerprints (hashes) along with their respective positions. Those fingerprints can then be extracted for every document in a given corpus $\mathcal{C}$, and stored in an *inverted index* that maps each hash h_i to a list of (doc_id, pos) pairs, where pos is the position (character-level offset) of the fingerprint in the document. This inverted index allows us to efficiently determine the list of documents of $\mathcal{C}$ in which a given set of k -grams occurs.

3.2 Text containment detection

Let $\mathcal{C}$ be a corpus whose winnowed fingerprints are indexed as described above and q a query document. Our objective is to determine whether there exists a document $d_i \in \mathcal{C}$ that shares some textual content with q . This can be formalised via a *text*

containment score $s(q, d_i)$, which can be computed using two strategies, described below.

Nb. of shared fingerprints Let $\mathcal{F}(\cdot)$ represent the set of fingerprints extracted from a document. A simple measure of text containment is the number of unique fingerprints shared by q and d_i :

$$s_{nb_shared}(q, d_i) = |\mathcal{F}(q) \cap \mathcal{F}(d_i)| \quad (1)$$

Note that, unlike the Jaccard similarity commonly employed for near-duplicate detection (Broder, 1997), the $s_{nb_shared}(q, d_i)$ score is not normalised by the total number of fingerprints extracted from q and d_i , as the goal is here to measure text containment rather than text similarity.

The $s_{nb_shared}(q, d_i)$ scoring function ignores the positions of the fingerprints. However, shared text segments should see their fingerprints occur in the same order and with roughly the same spacing. This can be captured by the next method.

Identification of fingerprint chains Let $h \in \mathcal{F}(q) \cap \mathcal{F}(d_i)$ denote a fingerprint shared between q and d_i . As winnowing preserves positional information, we can access its position $p_q(h)$ in the query document q and its position $p_{d_i}(h)$ in a corpus text d_i . Based on those two character-level positions, we can then define the *position offset* $\delta(h) = p_{d_i}(h) - p_q(h)$ between the two. The shared fingerprints can be represented as points $(p_q(h), \delta(h))$ in a two-dimensional space, as illustrated in Figure 2.

The key intuition of our method is that shared text fragments gives rise to *geometrically coherent structures* in this space. More precisely, a text fragment found in both q and d_i will correspond to a sequence of fingerprints with slowly increasing p_q but nearly constant δ .

Formally, we consider two shared fingerprints h_i and h_j as *connected* if $|p_q(h_i) - p_q(h_j)| \leq \tau_{pos}$ and $|\delta(h_i) - \delta(h_j)| \leq \tau_{off}$, where $\tau_{pos} = 30$ controls the maximum gap between two matches within a reused passage, and $\tau_{off} = 10$ controls how much the relative shift between the two documents may vary (providing robustness to small local insertions or deletions). Connected clusters $\{C_k\}$ are extracted and any cluster with fewer than $\kappa = 5$ fingerprints is discarded. The similarity score is then the size of the largest cluster:

$$s_{chain}(q, d_i) = \max_k |C_k| \quad (2)$$

Note, when h occurs $n_{h,q}$ times in q and n_{h,d_i} times in d_i , all $n_{h,q} \times n_{h,d_i}$ position pairs are included.

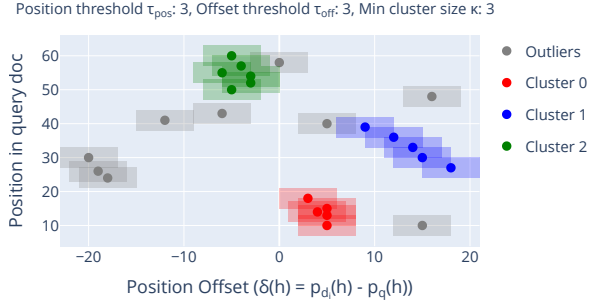


Figure 2: Illustration of the method for identifying fingerprint chains. Clusters of size $\geq \kappa$ are shown in colour, while fingerprints that cannot be connected to a cluster are ignored. The largest cluster has here 6 fingerprints, yielding a score $s_{\text{chain}}(q, d_i) = 6$.

4 Implementation

FindMyText is implemented in Python. Its use proceeds in two stages:

Stage 1 Given a corpus $\mathcal{C}$, the user builds an inverted index $\mathcal{I}_{\mathcal{C}}$ of winnowed fingerprints over its documents – or selects one of the already compiled indexes at their disposal.

Stage 2 The resulting index is then queried with a document q to determine whether, and to what extent, its contents can be found in $\mathcal{C}$.

The extraction of winnowed fingerprints is implemented in Numba to maximise efficiency, reaching a throughput per CPU core of approximately 150 ms. per million characters². For our experiments, the size k of k -grams was set to 4 (ignoring punctuation), and the window w to 6.

The construction of the index $\mathcal{I}_{\mathcal{C}}$ can be distributed on many machines, each being responsible for extracting fingerprints on a subset of the full document collection. Those partial results are then merged in a single inverted index in which each fingerprint is mapped to a list of (doc_id, pos) pairs. To scale to very large corpora, FindMyText adopts a disk-based storage scheme built on memory-mapped arrays, allowing it to handle indexes that exceed the available memory while still providing efficient query performance. The lookup for shared fingerprints between q and documents indexed in $\mathcal{I}_{\mathcal{C}}$ is vectorised, thereby retrieving potential matches in a single operation.

Along with the Python package, we provide a web interface for querying the indexes used in our experiments and visualising detected overlaps; see Appendix A.

²As measured on an AMD Epyc Turin (2.7-5.7 GHz).

5 Experiments

We evaluate FindMyText on three corpora spanning a range of text lengths and domains. For each corpus, we construct a synthetic benchmark consisting of either (1) texts that contain a segment drawn from the corpus and (2) texts that share no (substantial) segment with it. We then assess the performance of various detection methods on these benchmarks.

5.1 Synthetic benchmark

Text containment can be trivially evaluated by sampling documents either from, or outside, the given corpus $\mathcal{C}$. However, such a setup fails to reflect more challenging and realistic scenarios, where the query q and its counterpart $d_i \in \mathcal{C}$ have undergone various transformations. For example, either document may be mixed together with other pieces of content, or the two may differ due to formatting or text standardisation procedures.

Oracle To capture those more challenging scenarios, we construct a benchmark deliberately designed to provide “difficult” instances for text containment methods. To determine whether a pair of texts share a common text segment, we use a local aligner based on Smith-Waterman as oracle (Smith and Waterman, 1981). More precisely, if the aligner returns a common region with a score larger than a given threshold T , we consider the pair as positive, and negative otherwise. It is worth stressing that, unlike exact string matching, local alignment algorithms such as Smith–Waterman tolerate small discrepancies between the texts through penalties for character substitutions and gaps.

Using this oracle, the instances of the benchmark are generated with the following approach:

Positive instances To generate a positive example, we first sample a document d_i from $\mathcal{C}$ and then select a consecutive fragment $d'_i \subset d_i$ of that document. This fragment is then subjected to a sequence of editing operations designed to replicate the transformations commonly introduced during the pre-processing of web-crawled corpora, including de-hyphenisation, case normalisation, boilerplate removal, and the standardisation of characters such as quotation marks, ligatures, and dashes. At each step, the modified fragment d'_i is compared back with the original document d_i using the aligner. Those edits are repeated until we obtain a fragment d'_i that is right above a given threshold T_{pos} . Finally, the fragment d'_i is concatenated with some

other text content that is not part of $\mathcal{C}$, thereby giving the final text d_i'' . The pair (d_i'', d_i) is a positive example as the two documents share by construction a common text region according to the oracle, although this overlap is partially obscured by minor edits and the addition of unrelated content.

Negative instance To create “difficult” negative examples, we also sample a document d_i from $\mathcal{C}$. We then seek to construct an equivalent text d_i' that is semantically similar to d_i , but does *not* constitute a case of text containment according to the oracle. More specifically, we initialise $d_i' \leftarrow d_i$, and then iteratively apply one of those three operations:

- We split d_i' in chunks (respecting sentence boundaries) and permute their order.
- We select a chunk from d_i' , use an LLM to paraphrase it³, and replace it in d_i' .
- We insert a short boilerplate (page headers/footers, web navigation links, etc.).

The operations on d_i' are repeated until we reach a text whose oracle score drops below a minimum threshold T_{neg} . The pair (d_i', d_i) is a negative example as the two texts no longer contain a common text fragment of sufficient length, although they remain semantically similar.

The thresholds on the alignment scores are set to respectively $T_{pos} = 1000$ and $T_{neg} = 100$. An alignment score > 1000 corresponds in practice to a shared segment of around half a page, although the exact length may depend on the type of edits between the two documents. Similarly, an alignment score < 100 corresponds to the absence of any shared segment longer than 1-2 sentences. See the Appendix C for detailed examples.

5.2 Methods

We evaluate the following detection methods:

BM25 This baseline relies on the Okapi BM25 algorithm to find the documents in $\mathcal{C}$ closest to the query document q . After retrieving the BM25 scores of the closest documents, the baseline marks the instance as positive if at least one score is above a given threshold, and negative otherwise.

Dense retrieval One can also detect near-duplicates by deriving document embeddings for

both q and d_i and measuring their proximity according to a given distance metric (Abbas et al., 2023). In practice, those dense vectors can be computed with text embedding models, and then indexed through frameworks for approximate vector search. We use here document vectors computed with Qwen3-Embedding-4B⁴ and indexed with FAISS using HNSW (Douze et al., 2024).

Number of shared fingerprints As mentioned in Sec. 3.2, one simple way to detect text containment is to count the fingerprints shared by the query q and a document d_i . This count can be efficiently performed through the inverted index.

Chain-based method The final method is the approach from Sec. 3.2, which seeks to explicitly capture the occurrence of fingerprint chains based on positional information.

Hyper-parameters – such as the τ_{pos} and τ_{off} thresholds of the chain-based approach – are calibrated empirically based on a validation set synthesised following the procedure of Sec. 5.1.

5.3 Evaluation details

We use the synthetic benchmarking procedure outlined in Sec. 5.1 to generate, for each corpus, 1K positive and 1K negative instances.

Corpora The three corpora employed in the experiments are (1) Wikipedia, (2) ArXiv papers and (3) generic content crawled from the web. Wikipedia articles are extracted from a dataset of 1M English-language articles crawled on September 2025. The ArXiv papers correspond to papers uploaded under CC BY, CC BY-SA, and CC0 licenses on September 2024 and included in the Common Pile (Kandpal et al., 2025). Finally, the generic web content is drawn from the English part of the HPLT dataset v3.0 (Burchell et al., 2025). We filter out texts below 2K characters or above 100K characters, yielding a set of 381K articles from Wikipedia, 245K full-text papers from ArXiv, and 50.7M crawled content from HPLT.

Metrics We evaluate the performance of each method using the following metrics:

- **AUC-ROC**, which measures how well the method distinguishes positive from negative examples across all score thresholds.

³This operation was performed with Qwen2-7B-Instruct with chunks of ≈ 400 characters and a temperature = 0.8.

⁴To avoid expensive computations, we use precomputed document embeddings for Wikipedia articles available at https://huggingface.co/datasets/maknee/wikipedia_qwen_4b.

Corpus → Approach ↓	Wikipedia			ArXiv			HPLT		
	AUC-ROC	P@R=90%	P@R=99%	AUC-ROC	P@R=90%	P@R=99%	AUC-ROC	P@R=90%	P@R=99%
BM25	.066	.474	.497	.569	.648	.637	.581	.546	.535
Dense Retrieval	.000	.473	.497	/	/	/	/	/	/
Shared fingerprints	.868	.804	.758	.335	.520	.506	.907	.792	.717
chain-based approach	.998	.998	.997	.991	.989	.980	.999	1.00	1.00

Table 1: Results on the three corpora for the task of deciding whether a query document q shares a text fragment with a corpus $\mathcal{C}$. Evaluation metrics are AUC-ROC and Precision@Recall at two recall levels (0.9 and 0.99). For each corpus, a benchmark comprising 1K negative and 1K positive instances were generated (see Sec. 5.1). The chain-based method employs $\tau_{\text{pos}} = 30$ and $\tau_{\text{off}} = 10$ as maximum gaps between fingerprints. Results for the dense retriever are only available for the Wikipedia articles due to the availability of precomputed embeddings for those texts. As all methods are deterministic, the results were obtained with one run.

- **Precision@Recall**, which expresses the precision at the score threshold required to achieve the specified recall level (0.9 or 0.99).

5.4 Results

Table 1 presents the experimental results. We observe that the two baselines (BM25 and dense retrieval with an LM-based embedding model) are unable to disentangle the positive and negative examples, and even misled by the adversarial design of the benchmark, thereby leading to very low AUC scores. Indeed, the negative examples were deliberately constructed to be *similar* to one document d_i in a corpus, but without exhibiting any shared text fragments – a distinction that the baselines are unable to capture. This behaviour is expected, as both methods operate by retrieving semantically or lexically similar documents rather than identifying the presence of long shared text spans.

The approach based on the number of shared fingerprints performs better, but fails as a reliable indicator of text containment, particularly for long texts such as ArXiv papers. As counting shared fingerprints does not use any positional information, it cannot distinguish true containment from mere similarity. In contrast, the main detection method implemented in FindMyText, which relies on the identification of fingerprint chains through clustering, discriminates much more effectively between positive and negative examples, achieving an AUC-ROC around 0.99.

The runtime performance of FindMyText is mainly influenced by the index size. The response time was around 10 ms. for Wikipedia, 40 ms. for ArXiv papers, and 450 ms. for the much larger HPLT data, which included over 50M texts.⁵

⁵Runtime measured on an AMD Epyc Turin machine with 64 cores and 160 GB of RAM.

6 Conclusion

We presented FindMyText, a Python toolkit designed to determine whether a text is (partly or fully) included in a large corpus. The toolkit relies on the construction of an inverted index of document fingerprints that can be efficiently extracted through winnowing. Given a new query q , the toolkit searches the index for matching fingerprints. Those are then grouped using a clustering approach to detect chains of consecutive fingerprints, providing evidence of shared text fragments.

The toolkit is designed to be robust to small differences between the query text and the documents included in the corpus. This robustness is particularly important for LLM pre-training corpora, which typically undergo extensive cleaning and curation. “Cleaned” texts will often differ from their original version in terms of formatting, text normalization, or segmentation, while still containing largely the same content.

The purpose of FindMyText is to detect text *containment* rather than semantic similarity. One of its primary use cases is to determine whether the corpus used for pre-training an LLM includes copyrighted material or content that violates licensing conditions. In this context, the goal is to identify reused passages and partial copies, rather than documents that merely express similar ideas or convey comparable information. As such, text containment is closely aligned with the legal notion of *substantial similarity*, which concerns the reproduction of protected expression rather than the underlying ideas themselves (Asay, 2022).

Future work on FindMyText will focus on releasing precompiled fingerprint indexes for established pre-training corpora such as CommonCrawl.

References

- Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*.
- Ahmed Ahmed, A Feder Cooper, Koyejo Sanmi, and Liang Percy. 2026. [Extracting books from production language models](#). *arXiv [cs.CL]*.
- Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Hae-won Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A Survey on Data Selection for Language Models. *Transactions on Machine Learning Research*.
- Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. 1990. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410.
- Clark D Asay. 2022. An empirical study of copyright’s substantial similarity test. *UC Irvine Law Review*, 13(1).
- Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29.
- Laurie Burchell, Ona De Gibert Bonet, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Pinzhen Chen, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajič, Jindřich Helcl, Erik Henriksson, Mateusz Klimaszewski, Ville Komulainen, Andrey Kutuzov, Joona Kytöniemi, Veronika Laippala, Petter Mæhlum, Bhavitvya Malik, and 16 others. 2025. [An Expanded Massive Multilingual Dataset for High-Performance Language Technologies \(HPLT\)](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17452–17485, Vienna, Austria. Association for Computational Linguistics.
- Tyler A Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. 2024. Scalable Influence and Fact Tracing for Large Language Model Pretraining. In *The Thirteenth International Conference on Learning Representations*.
- Moses S Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 380–388.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. *arXiv [cs.LG]*.
- European Parliament and Council. 2024. Regulation (EU) 2024/1689... (Artificial Intelligence Act).
- Valentin Hartmann, Anshuman Suri, Vincent Bind-schaedler, David Evans, Shruti Tople, and Robert West. 2023. [SoK: Memorization in general-purpose large Language Models](#). *arXiv [cs.CL]*.
- Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A Feder Cooper, Aviya Skowron, Shayne Longpre, Lintang Sutawika, Alon Albalak, Zhenlin Xu, Guilherme Penedo, Loubna Ben, Elie Bakouch, John David, Honglu Fan, and 8 others. 2025. The Common Pile v0.1: An 8TB Dataset of Public Domain and Openly Licensed Text. *arXiv preprint*.
- Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. 2023. [Copyright Violations and Large Language Models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412, Singapore. Association for Computational Linguistics.
- Ken Liu, Christopher A Choquette-Choo, Matthew Jagielski, Peter Kairouz, Sanmi Koyejo, Percy Liang, and Nicolas Papernot. 2025. Language Models May Verbatim Complete Text They Were Not Explicitly Trained On. In *Forty-second International Conference on Machine Learning*.
- Shayne Longpre, Nikhil Singh, Manuel Cherep, Kushagra Tiwary, Joanna Materzynska, William Brannon, Robert Mahari, Naana Obeng-Marnu, Manan Dey, Mohammed Hamdy, Nayan Saxena, Ahmad Mustafa Anis, Emad A Alghamdi, Vu Minh Chien, Da Yin, Kun Qian, L I Yizhi, Minnie Liang, An Dinh, and 24 others. 2024. Bridging the Data Provenance Gap Across Text, Speech, and Video. In *The Thirteenth International Conference on Learning Representations*.
- Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. 2007. Detecting near-duplicates for web crawling. In *Proceedings of the 16th international conference on World Wide Web*, pages 141–150.
- Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. 2024. SoK: Membership Inference Attacks on LLMs are rushing nowhere (and how to fix it). *arXiv [cs.CL]*.
- S B Needleman and C D Wunsch. 1970. [A general method applicable to the search for similarities in the amino acid sequence of two proteins](#). *Journal of Molecular Biology*, 48(3):443–453.
- Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2023. Proving Test Set Contamination in Black-Box Language Models. In *The Twelfth International Conference on Learning Representations*.
- Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Bo Liu, Aastha Jhunjhunwala, Zhilin Wang, Mostafa Patwary, Mohammad Shoeybi, and Bryan Catanzaro.

2024. [Data, Data Everywhere: A Guide for Pre-training Dataset Construction](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10695, Miami, Florida, USA. Association for Computational Linguistics.
- Noah Scharrenberg and Chang Sun. 2025. [Copyright infringement by large language models in the EU: Misalignment, safeguards, and the path forward](#). In *Proceedings of the Natural Legal Language Processing Workshop 2025*, pages 125–134, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Saul Schleimer, Daniel S Wilkerson, and Alex Aiken. 2003. [Winnowing: local algorithms for document fingerprinting](#). In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, New York, NY, USA. ACM.
- Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting Pretraining Data from Large Language Models. *arXiv [cs.CL]*.
- T F Smith and M S Waterman. 1981. [Identification of common molecular subsequences](#). *Journal of Molecular Biology*, 147(1):195–197.
- Jingtian Wang, Xinyang Lu, Zitong Zhao, Zhongxiang Dai, Chuan-Sheng Foo, See-Kiong Ng, and Bryan Kian Hsiang Low. 2023. Source attribution for large language model-generated data. *arXiv [cs.LG]*.
- Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. 2024a. Membership inference attacks cannot prove that a model was trained on your data. *arXiv [cs.LG]*.
- Marina Zhang, Owen Vallis, Aysegul Bumin, Tanay Vakharia, and Elie Bursztein. 2024b. Retsim: Resilient and efficient text similarity. In *International Conference on Learning Representations*, volume 2024, pages 6177–6195.

A Web interface

The FindMyText web interface is available at <https://findmytext.nr.no>, with access password **EMNLP2026**. Figure 3 shows a screenshot of this interface, which is made available together with the FindMyText toolkit. It allows users to query the precompiled indexes used in our experiments, namely Wikipedia, ArXiv papers, and HPLT web-crawled content. The user first selects one of those corpora, and then provides a query text q either by pasting their own text, loading one of the predefined examples, searching the indexed documents for a source text to inspect, or adding corpus-specific filler text. This filler text can be used on its own as a negative example, or appended to a matching passage to create a more challenging query in which the relevant overlap is embedded in unrelated content.

The interface implements the two fingerprint-based methods evaluated in Sec. 5: the number of shared fingerprints and the chain-based method introduced in Sec. 3.2. For the chain-based method, the user may also adjust the main clustering parameters, τ_{pos} , τ_{off} , and κ , making it possible to inspect the sensitivity of the clustering step.

After running a query, the interface reports the best matching document for each selected method, together with its score and a short ranking of candidate documents. When metadata is available, the matched documents are shown with human-readable titles and links to the original source or an archived version. The interface can also highlight the matching passages in the query text, either for the largest fingerprint chain, for all detected chains, or for all shared fingerprints. This visualisation makes it possible to inspect whether a high score corresponds to a coherent reused passage, as targeted by the chain-based method, or to more scattered fingerprint overlap.

The screenshot displays the FindMyText web interface. At the top, it says "findmytext.nr.no" and provides a brief description: "Check whether a piece of text — or a substantial part of it — appears in a large document corpus, using document fingerprinting." It also includes a link to the source code on GitHub.

The main section is titled "FindMyText: Robust, Scalable Detection of Text Containment in Large Web-Crawled Corpora" and lists the authors: Lars Henry Berge Olsen, Pierre Lison, Martin Jullum, and Mark Anderson, along with their affiliations and email addresses.

The "How it works" section outlines the steps: 1. Pick a corpus to check against. 2. Add your text — type or paste it, load a pre-made example, or add the corpus-specific generic filler. 3. Choose one or more detection algorithms and run the check. 4. Optional: once you have results, highlight the matching passages to see exactly which parts overlap. It also mentions that fingerprints are extracted from the text and compared against the corpus.

The "Access password" section shows a password field with the text "*****". Below it, a note states: "Required for corpus search, detection, and highlighting."

The "1 - Corpus" section shows a dropdown menu with "arXiv papers" selected. Below it, a note says: "Scientific papers from arXiv (open-licensed subset). Try pasting an abstract, or load one of the examples. Want to search the corpus yourself? [Browse the dataset](#)." There is also a search bar for finding articles in the corpus.

The "2 - Your text" section has a dropdown menu with "LLMs — training tips for the Transformer model" selected. Below it, a note says: "Three ways to fill the box: search the corpus above to find a source it contains, then open it and copy a real passage in; load one of our pre-made examples; or add the corpus-specific generic filler — on its own as a neutral, unrelated passage, or appended to text already in the box (a pre-made example or your own)." There is a text area with a pre-made example about neural machine translation. Below the text area, it says "785 characters".

The "3 - Detection algorithm(s)" section has two checkboxes: "Position-aware match — our method (recommended)" and "Shared fingerprints — baseline". Below these, there are settings for the "Position-aware method settings": "Position threshold (τ_{pos})" set to 30, "Offset threshold (τ_{off})" set to 10, and "Min. cluster size" set to 5. There are also links to "Reset to defaults" and "Run detection".

The "Result" section shows two columns: "Shared fingerprints (baseline)" with a score of 31 (0.88 × a sentence) and "Position-aware match (our method)" with a score of 30 (0.75 × a sentence). Both columns show the best match: "Training Tips for the Transformer Model". Below the results, it says "Searched arXiv papers · 111 fingerprints from your text · 11.0s".

The "Matching passages in your text" section has a note: "Highlight which parts of your text overlap the selected match. Pick one or more styles — they can be shown together." There is a dropdown menu with "Training Tips for the Transformer Model — score 31" selected. Below it, there are three checkboxes: "Largest aligned passage — position-aware method's top cluster (what its score rewards).", "All aligned passages — every position-aware cluster, one colour each (catches reordered passages).", and "All shared fingerprints — baseline overlapping passages." There is also a "Show highlighting" button.

Below the checkboxes, there is a note: "Overlap with 'Training Tips for the Transformer Model' — Position-aware — largest aligned passage: 80.3% · Position-aware — all aligned passages: 80.3% (1 cluster) · Shared fingerprints (baseline) — all overlapping passages: 81.1%".

At the bottom, there is a text area with a pre-made example about neural machine translation, which is highlighted in green. Below the text area, it says "A research demo from Norsk Regnesentral (NR). [GitHub](#)".

Figure 3: FindMyText web interface.

Corpus → Approach ↓	Wikipedia			ArXiv			HPLT		
	AUC-ROC	P@R=90%	P@R=99%	AUC-ROC	P@R=90%	P@R=99%	AUC-ROC	P@R=90%	P@R=99%
Shared fingerprints	.564	0.597	0.558	.122	.500	.500	.843	.629	.584
Chain-based approach	.996	.997	.990	.982	.968	.932	1.0	1.00	1.00

Table 2: Results on the three corpora when using thresholds on the Smith-Waterman score of respectively 500 for the positive instances and 100 for the negative ones to generate the instances of the benchmark. All experimental settings are otherwise identical to Table 1.

Corpus → Approach ↓	Wikipedia			ArXiv			HPLT		
	AUC-ROC	P@R=90%	P@R=99%	AUC-ROC	P@R=90%	P@R=99%	AUC-ROC	P@R=90%	P@R=99%
Shared fingerprints	.983	0.966	0.949	.719	.651	.534	.972	.943	.920
Chain-based approach	.999	.999	.998	.993	.992	.989	.999	1.00	1.00

Table 3: Results on the three corpora when using thresholds on the Smith-Waterman score of respectively 2000 for the positive instances and 100 for the negative ones to generate the instances of the benchmark. All experimental settings are otherwise identical to Table 1.

B Additional results

The results from Table 1 were obtained on the basis of a benchmark where the positive instances were designed such as to have a Smith-Waterman score of at least 1000 between the query document q and a document $d_i \in \mathcal{C}$. This score reflects the presence of a shared text fragment (allowing for some small discrepancies) between the two documents of around half a page.

We can create alternative benchmarks in which this minimal score for positive instances is modified. Lower scores lead to a more challenging benchmark in which positive instances only share a small common part (and thus a small number of common fingerprints), while higher scores create instances with longer common fragments.

Tables 2 and 3 illustrate the results obtained with a minimum Smith-Waterman score of respectively 500 and 2000. We can observe that the baseline method relying on the number of common fingerprints increasingly struggles when the length of the shared text goes down, especially for long documents such as ArXiv papers. Unsurprisingly, longer fragments are easier to detect.

C Benchmark examples

We provide below two examples of texts in the synthetic benchmark generated for the Wikipedia corpus.

Example 1 is a positive edited in which a text fragment (in green) can be found in both d_i and d_i'' , although with some discrepancies between the two (underlined in the text) – for instance, "fi" is transformed into a fi ligature, and some words are hyphenated or have changed their casing. In addition, unrelated content is inserted at the beginning of the text. According to the oracle, the Smith-Waterman score of (d_i, d_i'') is 1007, which classifies the pair as a positive example.

In contrast, Example 2 is a negative instance in which the pair (d_i, d_i') does not have a shared text fragment. The edited d_i' is created by applying a series of permutations between text chunks, paraphrasing with an LLM, and inserting short boilerplate texts. The document d_i' remains relatively close to the original d_i , as it covers the same topic and has many shared words and N-grams. However, the edit operations have led to a document that, according to the oracle, no longer contains a shared text fragment, with a Smith-Waterman score of only 79.

Example 1: Positive instance in Wikipedia benchmark

Initial text d_i (from Wikipedia):

"101 Mitigations" is the fifteenth episode of the thirtieth season of the American animated television series The Simpsons, and the 654th episode overall. The episode was directed by Mark Kirkland with a story by Rob LaZebnik and teleplay by Brian Kelley and Dan Vebber. It aired in the United States on Fox on March 3, 2019.

Plot

Homer, Bart, Lisa and Maggie are eating at The Gilded Truffle, where they used a misprinted coupon good for 100% off. Meanwhile, with the saved money, Marge gets a Swedish massage. Outside the restaurant, Homer warns the kids to not hustle to get ahead in life. The valet parking attendant Raphael hands Homer the keys to the wrong car, a fancy 1957 seafoam Cadillac Eldorado Biarritz convertible. They drive away with it, having fun on the road. When they go back to the restaurant, Comic Book Guy accuses them of stealing his father's car, and when he notices his mint condition copy of Radioactive Man #1 was damaged, he presses charges and Homer is arrested. In court, Judge Snyder finds Homer guilty, even after a touching apology letter written by Lisa.

With two weeks before Homer's sentencing hearing, Marge goes to The Android's Dungeon & Baseball Card Shop to negotiate with Comic Book Guy, who says the issue is about getting respect. At home, Lisa discovers sentencing mitigation videos, including one employed by Mr. Burns for one of his crimes against Springfield, portraying him in a more sympathetic light as a product of a neglected and bullied childhood. The family's efforts to produce a mitigation video for Homer fall short but Lisa is able to splice together the work using her Final Cut Pro skills. In court, Snyder is initially open to setting Homer free but Comic Book Guy delivers an impassioned courtroom speech, the best Snyder has ever heard. He will deliver his verdict the next day.

Lisa finds a replacement Radioactive Man #1 online in a nearby Ogdenville comic book shop, though it fails to quell Comic Book Guy's grudge against Homer. However, he notices Homer's 1975 season one Welcome Back, Kotter keychain, a "precious totem" to Homer, the only gift his father ever gave him. To make Homer feel how he felt with his beloved car and comic, Comic Book Guy smashes the keychain with Thor's hammer Mjölnir, then agrees to drop the charges. He also befriends Homer and invites him to come to Comic-Con with him.

In the epilogue, Bart is shown in detention at school being supervised by Principal Skinner. Bart shows Skinner a prank sentencing mitigation video with Milhouse extolling Bart's transformation, while the video plays Bart escapes from the room.

Production

Filmmaker Guillermo del Toro guest starred as himself, directing a video about Mr. Burns. Del Toro previously directed the opening sequence of the twenty-fifth season episode "Treehouse of Horror XXIV."

Reception

"101 Mitigations" scored a 0.8 rating with a 4 share and was watched by 2.25 million people, making The Simpsons Fox's highest rated show of the night. Tony Sokol of Den of Geek gave the episode 4 out of 5 stars, stating "What did we learn from this episode? Certainly not the intended lesson that moments of pure joy always have consequences, it is German is earth's closest language to Klingon. The episode is funny and revelatory, though not always tummy rumbling funny. '101 Mitigations' contains a good mix of the clever and the silly, with a moral compass set on cruise control."

Dennis Perkins of The A.V. Club gave the episode a C+, stating "For Homer to finally understand the pain his wacky weekly nonsense causes to another person could be a loaded moment, dramatically. But the episode fudges it. The brisk running time—truncated more by del Toro's time-consuming but attention-grabbing cameo—leaves Homer and CBG's rapprochement hanging unsatisfactorily, pawned off on the joke that Homer regards CBG's invitation to Comic-Con as barely preferable to prison. The Simpsons has room for its characters' signature misbehaviors to be deconstructed in a meaningful way. It's a shame '101 Mitigations' doesn't."

Edited text d_i'' (insertion of unrelated content + small edits):

NACHI Bearing 6908N design bearings have two stamped window-type steel cages, an inner ring without flanges and a guide ring centred on the inner ring. The CC design is indicated by the designation suffix C or CC. Large CC design bearings with the designation suffix EC or ECC have an optimized internal design for increased load carrying capacity.

NACHI Bearing 6908N design bearings have two stamped window-type steel cages, an inner ring without flanges and a guide ring centred on the inner ring ($d \leq 65$ mm) or on the cages ($d > 65$ mm). 55 bar rises 30.09 0mph NEE dew-point 54 sunrise 7:09 set 6:58 Autumn New Moon (Blood) More mulch. Two more loads. 3 cubic yards. 4.5 total. This is a lotta mulch, 3 trailer fulls. The good news is the trailer ... Continue reading

Monthly Archives: September 2008 60 bar rises 30.07 2mph N dew-point 59 sunrise 7:06 set 7:00 Autumn Waning Crescent of the Harvest Moon rise 5:12 set 6:05 Today and tomorrow will be full gardening days. There are bulbs to plant: daffodils, hyacinths, snow drops, ... Continue reading

Plants in position. This is a fruit tree with a guild of plants that will support it. Guilds are a permaculture concept that I will explain later. Paula Westmoreland, a principal in Ecological Gardens, at Day 3 start. More trees ... Continue reading

Near end of day 1. Leveling and moving soil. At the end of day 2 we had mounds of soil and plants placed. At the beginning of day 3 Paula Westmoreland gives direction to the crew. Kate cut more carpet ... Continue reading

63 bar falls 30.15 0mph NNE dew-point 60 sunrise 7:04 set 7:04 Waning Crescent of the Harvest Moon rise 2:44 set 5:23 What a guy night and day. Last night I relieved Kate after the trailer blew its tire. ... Continue reading

Today I have had many opportunities to learn about melding with the movement of the universe. I missed the first lesson when the dumptruck driver put the load of compost in the middle of the truck gate. I could have ... Continue reading

65 bar rises 30.02 0mph NW dew-point 64 sunrise 7:03 set 7:07 Last Quarter Harvest Moon rise 12:10 set 4:24 Pouring rain. Thunder and lightning. Good for the crops, but if it lasts into tomorrow, not so good for site ... Continue reading

66 bar steady 30.00 0mph WNW dew-point 65 sunrise 7:02 set 7:07 Autumn Last Quarter of the Harvest Moon rise 12:10 set 4:34 Orchard schematic from same orientation as photograph below. The large circles are trees, the smaller crenallated ... Continue reading

The last day of our site prep. Raking pea gravel, taking up the chain link laid down to protect the yard and moving paving blocks. Tomorrow I plant bulbs. Then, on Wednesday get compost. Thursday and Friday plant and install. ... Continue reading

62 bar rises 30.11 2mph N dew-point 54 sunrise 6:59 set 7:13 Waning Gibbous Harvest Moon rise 9:58 set 1:37 Starving The Beast By Jennifer Moses Washington Post Tuesday, November 29, 2005; Page A21 BATON ROUGE, La. — "A primary ... Continue reading

At home, Lisa discovers sentencing mitigation videos, including one employed by Mr. Burns for one of his crimes against Springfield, 4/8 portraying him in a more sympathetic light as a product of a neglected and bullied childhood. The family's efforts to produce a mitigation video for Homer fall short but Lisa is able To splice together the work using her Fi- nal Cut pro skills. In court, Snyder is initially open to setting Homer free but Comic Book Guy delivers an impassioned courtroom speech, the best Snyder has ever 9/9 heard. He will deliver his verdict the next day. Lisa finds a replacement Radioactive Man #1 online in a nearby Ogdenville comic book shop, though it fails to quell Comic Book Guy's grudge against Homer. How-ever, he notices Homer's 1975 season one Welcome Back, Kotter keychain, a "precious totem" to Homer, the only gift his fath-er ever gave him. To make Homer feel how he felt with his be-loved car and COMIC, Comic Book Guy smashes the keychain with Thor's hammer Mjölnir, then agrees to drop the charges.

Example 2: Negative instance in Wikipedia benchmark

Initial text d_i (from Wikipedia):

"101 Mitigations" is the fifteenth episode of the thirtieth season of the American animated television series *The Simpsons*, and the 654th episode overall. The episode was directed by Mark Kirkland with a story by Rob LaZebnik and teleplay by Brian Kelley and Dan Vebber. It aired in the United States on Fox on March 3, 2019.

Plot

Homer, Bart, Lisa and Maggie are eating at The Gilded Truffle, where they used a misprinted coupon good for 100% off. Meanwhile, with the saved money, Marge gets a Swedish massage. Outside the restaurant, Homer warns the kids to not hustle to get ahead in life. The valet parking attendant Raphael hands Homer the keys to the wrong car, a fancy 1957 seafoam Cadillac Eldorado Biarritz convertible. They drive away with it, having fun on the road. When they go back to the restaurant, Comic Book Guy accuses them of stealing his father's car, and when he notices his mint condition copy of *Radioactive Man* #1 was damaged, he presses charges and Homer is arrested. In court, Judge Snyder finds Homer guilty, even after a touching apology letter written by Lisa.

With two weeks before Homer's sentencing hearing, Marge goes to The Android's Dungeon & Baseball Card Shop to negotiate with Comic Book Guy, who says the issue is about getting respect. At home, Lisa discovers sentencing mitigation videos, including one employed by Mr. Burns for one of his crimes against Springfield, portraying him in a more sympathetic light as a product of a neglected and bullied childhood. The family's efforts to produce a mitigation video for Homer fall short but Lisa is able to splice together the work using her Final Cut Pro skills. In court, Snyder is initially open to setting Homer free but Comic Book Guy delivers an impassioned courtroom speech, the best Snyder has ever heard. He will deliver his verdict the next day.

Lisa finds a replacement *Radioactive Man* #1 online in a nearby Ogdenville comic book shop, though it fails to quell Comic Book Guy's grudge against Homer. However, he notices Homer's 1975 season one *Welcome Back, Kotter* keychain, a "precious totem" to Homer, the only gift his father ever gave him. To make Homer feel how he felt with his beloved car and comic, Comic Book Guy smashes the keychain with Thor's hammer Mjölhnir, then agrees to drop the charges. He also befriends Homer and invites him to come to Comic-Con with him.

In the epilogue, Bart is shown in detention at school being supervised by Principal Skinner. Bart shows Skinner a prank sentencing mitigation video with Milhouse extolling Bart's transformation, while the video plays Bart escapes from the room.

Production

Filmmaker Guillermo del Toro guest starred as himself, directing a video about Mr. Burns. Del Toro previously directed the opening sequence of the twenty-fifth season episode "Treehouse of Horror XXIV."

Reception

"101 Mitigations" scored a 0.8 rating with a 4 share and was watched by 2.25 million people, making *The Simpsons* Fox's highest rated show of the night.

Tony Sokol of Den of Geek gave the episode 4 out of 5 stars, stating "What did we learn from this episode? Certainly not the intended lesson that moments of pure joy always have consequences, it is German is earth's closest language to Klingon. The episode is funny and revelatory, though not always tummy rumbling funny. '101 Mitigations' contains a good mix of the clever and the silly, with a moral compass set on cruise control."

Dennis Perkins of The A.V. Club gave the episode a C+, stating "For Homer to finally understand the pain his wacky weekly nonsense causes to another person could be a loaded moment, dramatically. But the episode fudges it. The brisk running time—truncated more by del Toro's time-consuming but attention-grabbing cameo—leaves Homer and CBG's rapprochement hanging unsatisfactorily, pawned off on the joke that Homer regards CBG's invitation to Comic-Con as barely preferable to prison. The *Simpsons* has room for its characters' signature misbehaviors to be deconstructed in a meaningful way. It's a shame '101 Mitigations' doesn't."

Edited text d'_i (permutations, paraphrasing, small edits):

"Number One Hundred And One Countermeasures" serves as the fifteenth installment of the fortieth cycle in the American animated television series, *The Simpsons*, and stands as the six hundred fifty-fourth episode in total. Nevertheless, he spots Homer's 1975 season one *Welcome Back, Kotter* trinket, a "cherished emblem" to Homer, the solitary present his parent ever bestowed upon him. Regrettably, "101 Countermeasures" fails to meet expectations. The segment was helmed by Mark Kirkland, who co-authored the narrative with Rob LaZebnik, alongside teleplay contributions from Brian Kelley and Dan Vebber. The launch took place in America on the Fox channel, precisely on March 3, 2019. The storyline progresses chiefly with Homer, Bart, Lisa, and Maggie savoring dishes at The Gilded Truffle, capitalizing on a mistakenly printed voucher for considerable discounts. Simultaneously, Marge employs the leftover money for a rejuvenating Swedish massage. Beyond the eatery, Homer instructs the offspring to avoid rushing to progress in their lives. The valet service operator, Raphael, mistakenly presents Homer with the keys to an incorrect vehicle - an opulent 1957 seafoam-colored Cadillac Eldorado Biarritz convertible. They proceed with it, relishing their journey. The clan's endeavors to create a remediation movie for Homer prove insufficient, yet Lisa succeeds in amalgamating the components utilizing her expertise with Final Cut Pro. In order to evoke Homer's sentiment associated with his cherished automobile and graphic novel, Comic Book Guy employs Thor's mace Mjölhnir to destroy the keychain, subsequently consenting to discontinue the allegations. Moreover, he fosters rapport with Homer and proffers an enticement for him to join on an excursion to the comic convention. Thereafter, Bart is depicted in an appendix serving time in school suspension, overseen by the principal, Skinner. Bart delivers a jovial penance video to Skinner, during which Milhouse acknowledges Bart's metamorphosis, yet as it's showcased, Bart engineers an evasion from the vicinity. The exhibition was orchestrated by visiting actor, Luis Guzmán, who portrayed himself, particularly emphasizing Mr. Smith. Guzmán had previously handled the preliminary segment of the television episode, © 2023 Content Authority. All rights are protected. The final episode, "Jungle of Horror XXIV," is highlighted. The "101 Adjustments" critique was given a rating of nil. 8, determined by opinions from four observers, and was observed by precisely two individuals. This production was seen by a grand total of 25 million people, establishing *The Simpsons* as Fox's most acclaimed show for that specific night. Chris Smith from *Empire Magazine* rated the show 8 out of 10, pondering "What advantages did this episode offer us?" "Most likely not the supposed lesson that fleeting pleasures invariably entail repercussions, it is German, Earth's nearest language to Esperanto. The sequence is amusing and insightful, yet doesn't consistently provoke laughter. "101 Defensive Tactics" integrates an eclectic blend of humor and inventive elements, underpinned by an unwavering ethical standpoint. Dennis Perkins, a constituent of the trio, representing three out of four members, assessed the segment with a grade of C-, noting "A pivotal moment may emerge when Homer realizes the emotional repercussions of his irregular weekly activities on other characters, narratively speaking. Regrettably, the initiative fails to uphold its assurance. The swift duration, curtailed even more by del Toro's protracted yet captivating guest appearance, results in an unsatisfactory resolution for Homer and CBG's reconciliation, with their truce clumsily disposed of via a jest suggesting Homer finds CBG's Comic-Con invitation marginally more tolerable than incarceration. The animated series, *The Simpsons*, allows for the dissection of its characters' distinctive rebellious antics in a substantial manner. Within the courtroom, Snyder initially contemplates releasing Homer. However, Comic Book Guy compels the judge with a persuasive speech, the most compelling he has encountered. A decision will be announced the following day. Lisa locates a substitute *Radioactive Man* #1 online in a nearby Ogdenville comic store, yet it fails to assuage Comic Book Guy's animosity towards Homer. Upon returning to the eatery, the Comic Book Guy accuses them of appropriating his parent's conveyance. Upon realizing his pristine copy of *Radioactive Man* #1 had sustained damage, he initiates legal action, culminating in Homer's incarceration. Recently, within the legal chamber, the document underwent modifications. Judge Snyder emphasizes Homer's accountability, notwithstanding a passionate apology letter dated 8/9 authored by Lisa, disregarding the 5/6 evidence. Approximately Next Page > two weeks ahead of Homer's scheduled court appearance, Marge pays a visit to The Android's Dungeon & Baseball Card Emporium with Comic Book Guy, who claims that the fundamental aspect pertains to preserving dignity. © 2018 Enterprise Label. Comprehensive permissions reserved. Upon reaching her dwelling, Lisa discovers documentation for diminution of punishment, inclusive of an artifact employed by individual named Mr. Burns for one of his offenses against Springfield, presenting him in a more empathetic perspective as a consequence of an unattended and tormented youth. Citations Linked resources *The Simpson* series (30th installment) segments 2019 United States TV content Comic-related Television broadcasts