Memorization or Lucky Guesses: Detecting Short Sequences from Copyrighted Dutch News in LLM Output

Joris Veerbeek[1], Kas Berendsen[1], Alessandra Polimeno[2], Antal van den Bosch[1]
[1]Utrecht University, The Netherlands
[2]DANS, The Netherlands
{j.veerbeek,a.p.j.vandenbosch}@uu.nl
kbberendsen@gmail.com, aapolimeno@gmail.com

Abstract

Demonstrating that large language models have memorized copyrighted material is more feasible for high-volume publishers than for smaller outlets whose content appears less frequently online. This study explores how even short, repeated sequences–rather than full articles–can serve as evidence of memorization. Focusing on Dutch news outlets included in the mC4 dataset, we test whether GPT-4 and mT5 reproduce excerpts from thousands of articles, including standardized editorial boilerplate. By comparing results to a post-training baseline and modeling memorization as a survival process, we find that repeated, outlet-specific phrases are significantly more likely to be completed verbatim. The approach provides a means to detect empirical evidence of memorization in cases where full reproduction is unlikely.

Keywords: large language models, memorization, copyrighted content, verbatim reproduction

1. Introduction

In 2023, *The New York Times* filed a lawsuit against OpenAI, alleging copyright infringement for using its content in training large language models (LLMs). The lawsuit included an appendix listing 100 articles that GPT-4 had almost entirely completed verbatim, some spanning multiple paragraphs (New York Times, 2023), which demonstrated a significant instance of memorization.

Memorization at such an extreme length is uncommon in LLMs (Huang et al., 2024), but *The New York Times*' global reach may have amplified the likelihood of its content appearing, possibly multiple times, in training data. The newspaper's articles are frequently republished and excerpted across third-party websites, leading to repeated appearances in multiple contexts and amplifying the likelihood of memorization (Diakopoulos, 2024). Similar patterns have been observed in books (Chang et al., 2023), where frequent online reproduction raises the probability of memorization.

For smaller publishers—especially those in non-English markets—proving that their content has been memorized by AI models is arguably harder. Their material lacks the ubiquity of *The New York Times*, making long reproductions more difficult to obtain.

However, as we argue in this paper, the correct recitation of short sequences—such as five-token fragments—can also reveal that copyrighted content was used in training, even when full articles are not fully or perfectly memorized. This is especially crucial for smaller or niche publishers, who want to obtain indications that their content has indeed been used without permission, even in minor reproductions. Our experiments demonstrate that even short word n-grams can provide strong evidence of a model's exposure to copyrighted material.

Specifically, we examine potential memorization in GPT-4 using data from four Dutch news outlets in the mC4 dataset. Our analysis consists of two complementary approaches. First, we test whether GPT-4 and mT5 can correctly complete a random selection of articles from these outlets. Second, we develop a methodology to identify frequently occurring, but unique, strings in newspaper content and test whether these can be correctly completed by the models.

We compare the results of these two approaches with a baseline set of articles published after the models' training was completed. To assess the significance of memorization, we introduce a survival analysis framework, modeling memorization as a time-to-event process and using survival analysis to compare observed memorization against the post-training baseline. While longer memorized sequences of 10 words or more are rare, our approach reveals systematic retention of highly standardized editorial elements–such as column openings, calls to action, and accountability statements–which show significant deviations from the baseline. Overall, this paper introduces a framework that enables smaller news organizations to systematically test whether their content has been memorized in LLMs.

2. Related work

Memorization in Transformer-based LLMs is typically defined as the ability of an LLM to recite spe-

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 5960–5969
11–16 May 2026.

cific sub-sequences of its training data verbatim or near-verbatim. This recitation must be based on some storage, somewhere in the attention heads and layers, of the sequential information. During training, it is arguably profitable to remember training sequences, as this will reduce training loss during repeated presentations of the training data over epochs. However, remembering long sequences unique to the training data will not lower validation or test loss.

The increasing size of Transformer-based LLMs leaves increasing room within the parameter space to memorize training data (Carlini et al., 2023). Carlini et al. (2021) found that iterative learning in multiple training epochs causes training sequences to be memorized better; they also found that larger models exhibit stronger memorization.

Previous research has explored the verbatim recitation of training data by LLMs using various methods. Two types of genres, viz. news articles and literary works, are often in focus of this type of work, due to the contested copyright violations on particularly these types of data.

2.1. News articles

Carlini et al. (2021) discuss the vulnerability of LLMs to what they refer to as *training data extraction attacks*, demonstrating this by asking GPT-2 models to complete unfinished training data sequences and compare their output to the actual continuations. They were able to obtain hundreds of correctly completed training text sequences, including personal information, even when some of this information was included only in one document within the training data. They report that $1,800$ queries extracted from 100 news articles produced 604 correctly memorized completions, showing a high level of recitation. Nasr et al. 2023 ask ChatGPT (gpt-3.5-turbo) for the continuation of text, also finding that the continuations often reveal memorized content from the training data.

2.2. Literary Works

Several research papers have shown that literary works are being used to train LLMs and that portions of their content can be extracted from the model. Karamolegkou et al. (2023) show that LLMs memorize substantial parts of copyrighted text fragments of English literature. They show that this effect increases with an increase in LLM model size. Still, even small models tend to memorize. In their extraction attack study, they use different techniques: prefix probing, where they ask for the continuation of the text from the book; as well as asking for a specific page from the book. Both Karamolegkou et al. (2023) and Chang et al. (2023) found that memorization is tied to overall popularity of the book online.

3. Methodology

3.1. Data

We conduct a series of systematic experiments to examine whether LLMs have memorized content from Dutch news outlets. Dutch provides a suitable test case for studying memorization beyond English: it is a medium-resource language that appears in large multilingual corpora such as mC4, but with substantially lower representation than English. This balance makes it possible to observe how memorization manifests in a language that is present in web-scale datasets yet not dominant within them.[1] Focusing on Dutch media thus allows us to test how memorization behaves beyond the major English-language publishers whose content dominates the web.

Our analysis focuses on four outlets: *Algemeen Dagblad* (*ad.nl*), one of the Netherlands' largest and most widely read newspapers; *de Volkskrant* (*volkskrant.nl*) and *NRC* (*nrc.nl*), both considered leading examples of quality journalism; and *De Groene Amsterdammer* (*groene.nl*), a weekly magazine with a strong analytical and investigative tradition.

These outlets were selected because they represent a broad spectrum of Dutch journalism–ranging from daily to weekly publication, and from general-interest to in-depth reporting–and because they have relatively high representation in publicly available web corpora such as mC4. The research was carried out as a commissioned study, with the aim of providing an empirical basis for discussions about the reuse of journalistic content in AI systems.

To evaluate potential memorization, we compare two datasets: a hypothesized training dataset and a control group.

3.1.1. Hypothesized training dataset

The hypothesized training dataset consists of Dutch-language articles from the mC4 corpus, a large-scale collection of web pages compiled via Common Crawl (Xue et al., 2021). Since this dataset was available before GPT-4's training, some of its content was likely included in the model's training data. However, since the data

[1] According to public statistics for the GPT-3 training corpus, for example, Dutch accounts for approximately 0.34% of the total word count (ranking eighth), compared to 92.65% for English. See: https://github.com/openai/gpt-3/blob/master/dataset_statistics/languages_by_word_count.csv.

sources for GPT-4 have not been publicly released, this is actually not known.

Source	Number of articles
ad.nl	141,985
nrc.nl	594,438
volkskrant.nl	162,668
groene.nl	32,877

Table 1: Number of articles of the four outlets in mC4 dataset

Table 1 provides an overview of the relative share of each outlet within mC4. *NRC* has the largest presence in the corpus due to its publicly accessible archive, whereas *De Groene Amsterdammer* has the fewest articles, reflecting its weekly publication schedule.

3.1.2. Baseline

To distinguish between completions based on memorization and those that result from lucky guesses–instances where a model's output happens to match by chance–we established a control group to serve as a baseline. We mirrored the mC4 collection process as closely as possible, but used a more recent Common Crawl snapshot than mC4, specifically, the `CC-MAIN-2023-23` dump. MC4 was built from all Common Crawl scrapes available as of its release in 2020 (Xue et al., 2021), meaning our snapshot postdates the mC4 collection window. We retrieved all available articles from this snapshot, extracted publication dates, and excluded any published before 2022. One major publisher, `nrc.nl`, had blocked Common Crawl at an earlier stage; for this outlet, we used the MediaCloud API[2] to collect articles from the same period.

We chose 2022 as a cut-off point because, according to OpenAI's documentation at the time, GPT-4's training data extends until September 2021.[3] This creates a temporal buffer between the model's training cut-off and the publication of the articles, thus reducing the likelihood that any of them were memorized. At the same time, the gap is not so large that the model's general world knowledge would no longer apply. This allows us to better isolate memorization from broader familiarity or reasoning.

After collecting the baseline data, we cleaned it according to mC4's standards: stripping boilerplate, filtering out low-quality lines, deduplicating pages, and keeping only Dutch-language text. For our experiments, we sample 500 articles per news outlet.

By including a control group, our study goes beyond typical memorization analyses that primarily identify overlaps with training data. It helps assess whether, and to what extent, LLMs have memorized content from Dutch news outlets, including within small n-grams.

3.2. Training data extraction attacks

Using this data, we perform a series of training data extraction attacks, where we input randomly cut-off articles into the language models and examine whether their outputs include verbatim continuations of the input text. We conduct these attacks on two models: GPT-4, as a representative of widely-used closed-source language models,[4] and mT5, which was directly trained on mC4 and thus provides a confirmed case of exposure to our target data.

For GPT-4, we used the `openai.ChatCompletion` API with no system prompt, providing only the input text, and set the temperature to zero to maximize deterministic outputs. We used the base GPT-4 version available in the API (`gpt-4-0613`).

For our experiments with mT5 we ran our experiments locally with different model sizes based on the openly available models.[5] We employed a masking strategy, where up to three consecutive tokens were masked in the input text, allowing us to assess whether the model would predict the missing tokens with verbatim accuracy.

3.2.1. Random articles

First, we selected a random sample of $1,000$ articles for each outlet. For each article, we used five randomly chosen cut-off points and provided the preceding text as input to the model, which then generated the remainder of the article until reaching the original length.

[2] https://www.mediacloud.org/

[3] Specifically, we use the `gpt-4-0613` model. According to OpenAI's official documentation (https://archive.is/hhOWK), the training data for this model has a cutoff of September 2021, predating our corpus. While we cannot rule out the possibility that OpenAI updates the training data of versioned models without disclosure, we are not aware of any evidence that this practice occurs.

[4] In a pilot phase of this research, we also experimented with other models such as Google's PaLM, but found that the available API endpoints were chat-oriented and did not support open-ended text completion, making it difficult to conduct extraction attacks in the same manner. This may have changed with more recent API and model updates.

[5] https://github.com/google-research/multilingual-t5

3.2.2. Duplicate texts

Since memorization is more likely to occur with duplicated content (Carlini et al., 2021), we repurposed a method originally designed for deduplicating entire text corpora (Lee et al., 2022) to instead identify repeated strings within the articles of each individual outlet.[6] Rather than applying it across the full corpus, we grouped articles by outlet and detected repetition within each outlet separately, in order to identify strings that are characteristic of a specific outlet. We extracted all duplicate sequences of at least 100 bytes that appeared five or more times within a given outlet.

Source	Duplicate strings ($n \geq 5$)
ad.nl	533
nrc.nl	744
volkskrant.nl	397
groene.nl	25

Table 2: Number of duplicate strings appearing more than five times in each outlet.

In news articles, such repetitions often take the form of boilerplate text, recurring column intros, or standardized calls to action. Table 2 lists the number of duplicate strings we found and tested for each outlet. Notably, the three daily newspapers exhibited a relatively high number of duplicates, whereas *groene.nl*, as a weekly magazine, contained considerably fewer.

For each duplicate string, we iteratively vary the length of the prompt provided to the model. Starting from a minimum of 5 tokens, we incrementally increase the prompt length up to 75% of the total sentence length. This allows us to observe how the model continues generating text when given progressively more context from the original sentence.

3.2.3. Categorization of duplicates

Duplicated content in news articles is not always unique to a single outlet. Some texts are repurposed across multiple outlets, such as copyright statements or widely circulated references, including quotations from religious or historical sources. To better characterize the nature of these duplicates and distinguish patterns of reuse, we develop a typology that categorizes them along two dimensions.

The first dimension distinguishes whether the content is outlet-specific or shared across multiple outlets. The second dimension assesses whether the content is boilerplate, meaning it recurs across different articles within the same outlet. This yields four categories of content:

1. **Editorial Content**: Content that is reused across different contexts, such as movie descriptions published in event listings or descriptions of museums or works of art repurposed from earlier publications.

2. **Editorial Boilerplate**: Content specific to the newspaper that recurs across multiple articles, like standard editorial phrases or recurring sections.

3. **General Boilerplate**: Non-article-specific text, such as JavaScript messages, website code, or references to the website.

4. **General Content**: Quotations from legal texts, the Bible, politicians, or other sources, used in various contexts.

Figure 1 presents an example from each category. We manually classified the 1,699 duplicates identified in the previous step according to this typology. In our analysis, we evaluate both the full set of duplicates and the subset consisting only of editorial-specific duplicates.

3.3. Evaluation

To measure memorization, we employ a strict exact matching approach, which counts the number of consecutive words that match the original, from left to right, until the first divergence occurs. Exact replication provides a rigorous measure of memorization, capturing only those cases where the model reproduces text with complete fidelity. Although this likely underestimates the true extent of memorization–since paraphrased reproductions are not captured–exact matching ensures that any detected overlap is a clear-cut case. This further helps to distinguish genuine memorization from lucky guesses, which become increasingly plausible as LLMs grow more capable of generating fluent and contextually accurate continuations.

3.4. Survival analysis

To distinguish coincidental overlap from systematic reproduction, we introduce a **survival analysis framework** for studying memorization. A short overlap of four tokens could easily occur by chance, while longer identical stretches are less likely to be coincidental. The central question, however, is whether such overlaps occur systematically more often for texts that were likely included in the training data compared to those that were not.

Survival analysis allows us to test this empirically by modeling the position of the first deviation as a time-to-event process. For each input in the random, duplicate, and baseline subsets, we compute a *max overlap*: the position where the generated

[6] https://github.com/google-research/deduplicate-text-datasets

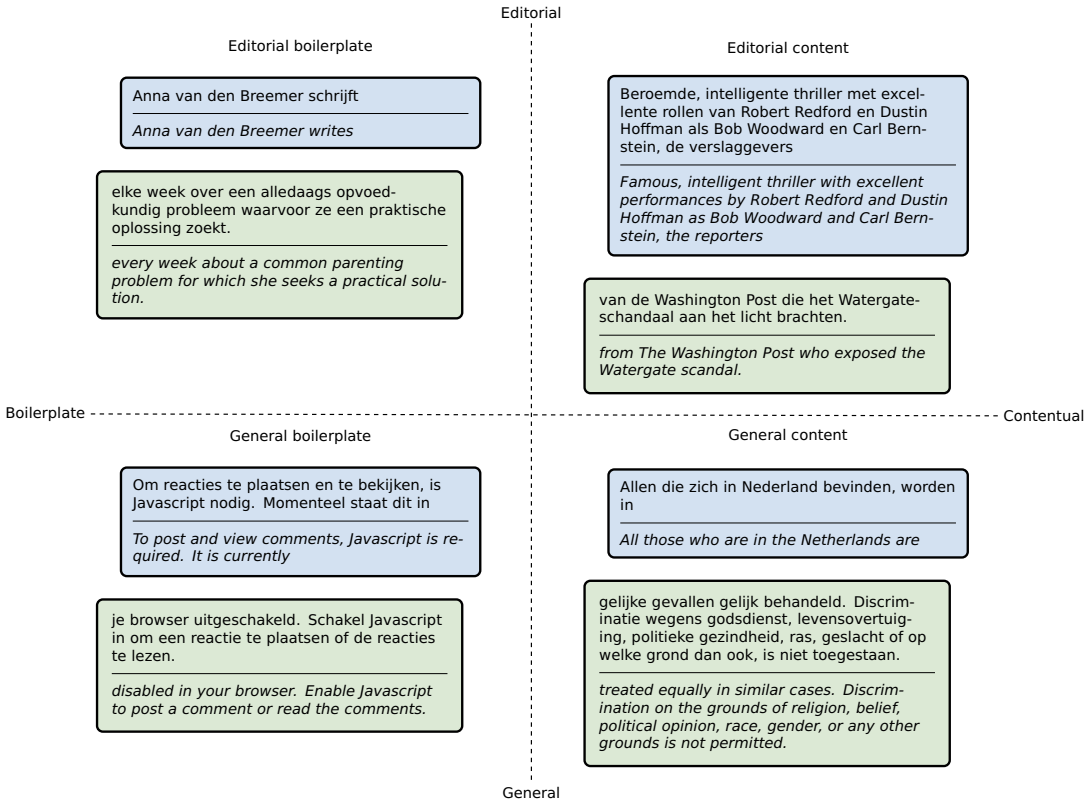

Figure 1: Examples of duplicate strings categorized along two dimensions—editorial versus general, and boilerplate versus contentual—each shown with the corresponding verbatim completions correctly reproduced by GPT-4.

output first diverges from the original article. In this formulation, the "event" is the first deviation, and the "survival time" represents how long the model continues reproducing the original verbatim. Outputs that never deviate are treated as right-censored observations–texts that "survive" without experiencing the event. This approach is especially valuable in cases where long verbatim reproductions are uncommon.

We compare survival patterns across text origins using a log-rank test (Fleming and Harrington, 1981) implemented via `scikit-survival` (Pölsterl, 2020). The test provides a robust, non-parametric way to assess whether survival curves differ significantly between groups, while naturally handling censored observations. From these estimates, we derive the hazard rate of memorization–the probability that a token is copied at a given position, conditional on it not having diverged earlier.

4. Results

The results are organized into three sections. First, we present a summary of the overall findings, focusing on aggregated completions across all outlets. Next, we conduct a detailed analysis of each news outlet individually. Finally, we analyze the results by breaking down different categories of duplicate strings. We report significance at three conventional levels (0.05, 0.01, and 0.001), emphasizing the latter for stronger evidence of deviation from the baseline.

4.1. Overall results

Figure 2 uses the Kaplan–Meier estimator (Kaplan and Meier, 1958) to track the probability of memorization across the first twenty tokens for the three prompt types. The Kaplan–Meier method provides a non-parametric estimate of the survival function–in this case, the probability that the model continues reproducing the original text without deviation up to a given token. Each drop in the curve represents an observed deviation, while flat segments indicate continued reproduction. Initially, all curves follow a similar pattern, with roughly a 40% chance of correctly completing the first word. By the third token, however, the curves for duplicate and random strings begin to diverge from the baseline.

A log-rank test confirms a significant difference between duplicate strings and the baseline ($\chi^2(1) = 10.88, p < 0.001$). For random articles, the probability of memorization is slightly higher overall, but this difference is not statistically significant

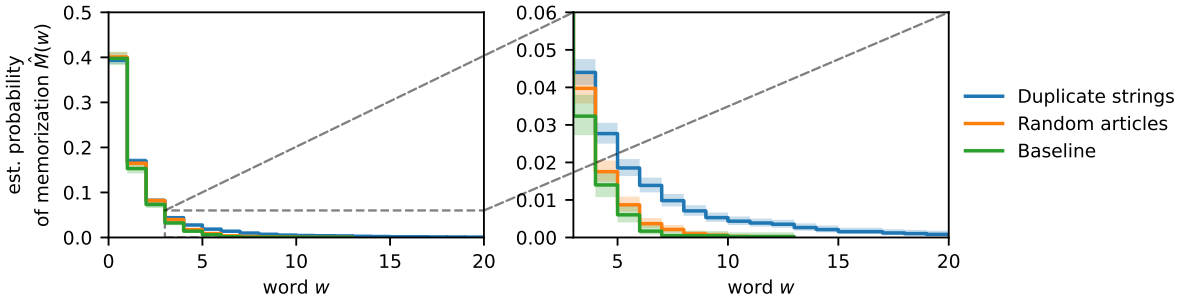

Figure 2: Estimated probability of memorization $M_i(w)$ as a function of text length in words. The curves represent Kaplan–Meier survival estimates, where each step shows the probability that the model continues reproducing the original text verbatim up to length w before diverging. The left panel shows the full range up to 20 words, while the right panel displays probabilities from three words onward. Results are shown for duplicate strings (blue), random article segments (orange), and the post-training baseline (green). Duplicate strings show higher survival probabilities for short n-grams, which indicate a greater likelihood of verbatim reproduction.

($\chi^2(1) = 3.28$, $p = 0.07$). Notably, after approximately 20 tokens, verbatim completion probabilities drop to zero for nearly all prompt types.

4.2. Results by outlet

While duplicate strings are the only prompt type to show clear evidence of news content memorization in GPT-4, per-outlet analysis (Figure 3) reveals notable variation across newspapers.

For *ad.nl*, duplicates show significantly higher memorization rates than the baseline ($\chi^2(1) = 6.90$, $p < 0.01$), whereas random articles show no significant deviation ($\chi^2(1) = 0.26$, $p = 0.61$). related pattern emerges for *nrc.nl*: random articles yield the highest verbatim completion probability for sequences shorter than four tokens, but duplicate strings surpass them at five tokens and beyond ($\chi^2(1) = 9.76$, $p < 0.01$), while no significant differences are observed for random articles ($\chi^2(1) = 2.39$, $p = 0.13$).

In contrast, results for *volkskrant.nl* and *groene.nl* show no significant differences between duplicates, random articles, and the baseline. For *volkskrant.nl*, $\chi^2(1) = 1.23$, $p = 0.26$ (random) and $\chi^2(1) = 0.03$, $p = 0.85$ (duplicates). For *groene.nl*, $\chi^2(1) = 0.0002$, $p = 0.99$ (random) and $\chi^2(1) = 0.58$, $p = 0.44$ (duplicates). As Figure 3 shows, limited duplicates introduce substantial uncertainty.

4.3. Break-down of duplicates

A more detailed analysis of duplicate texts reveals notable differences in their typology across newspapers (Figure 5). For *groene.nl* and *www.ad.nl*, duplicates are primarily editorial boilerplate, whereas duplicates from *nrc.nl* and *volkskrant.nl* are predominantly editorial content.

When distinguishing editorial boilerplate from general boilerplate, general content, and other editorial material, a clear deviation from the baseline becomes apparent across all outlets. The estimated probabilities indicate that the chances of correctly completing a five-gram sequence are roughly twice as high for editorial boilerplate compared to the baseline (Figure 4).

Log-rank tests confirm these patterns. Significant differences in the memorization rate between the baseline and editorial boilerplate are observed for *ad.nl* ($\chi^2(1) = 7.86$, $p < 0.01$), *nrc.nl* ($\chi^2(1) = 4.44$, $p < 0.05$), and *volkskrant.nl* ($\chi^2(1) = 5.73$, $p < 0.05$). However, for *groene.nl*, which had the smallest number of boilerplate texts, these differences are not statistically significant ($\chi^2(1) = 0.76$, $p = 0.38$).

For the general editorial content, this pattern is less clear. For *ad.nl*, there is still a significant difference ($\chi^2(1) = 5.59$, $p < 0.05$), but there is not for *volkskrant.nl* ($\chi^2(1) = 3.08$, $p = 0.07$). For *nrc.nl*, the baseline actually exceeds the general content verbatim completions ($\chi^2(1) = 33.86, p < 0.001$).

5. Masked token string prediction in mT5

To obtain a confirmation of the broad outcomes of our study, we performed an additional experiment with mT5 (Xue et al., 2021), an encoder-decoder Transformer architecture known to be pre-trained on mC4. The mC4 corpus was, in fact, presented in the paper that introduced the mT5 architecture, a multi-lingual variant of the encoder-decoder T5 architecture (Raffel et al., 2020). Differently sized

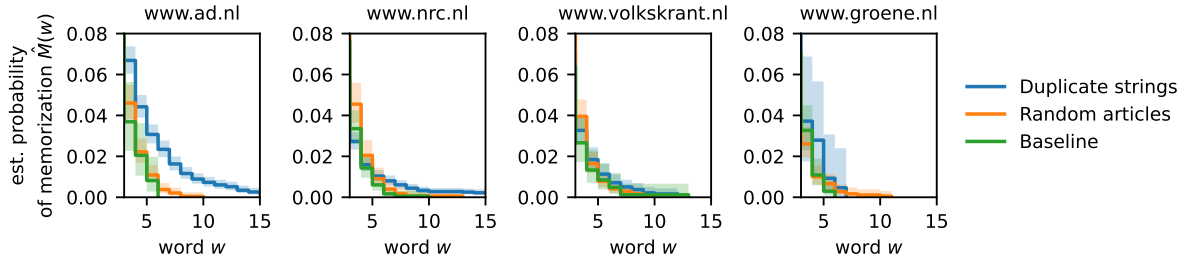

Figure 3: Verbatim completion probabilities for duplicate strings (blue), random article segments (orange), and a baseline of articles outside GPT-4's training period, split separately for each outlet.

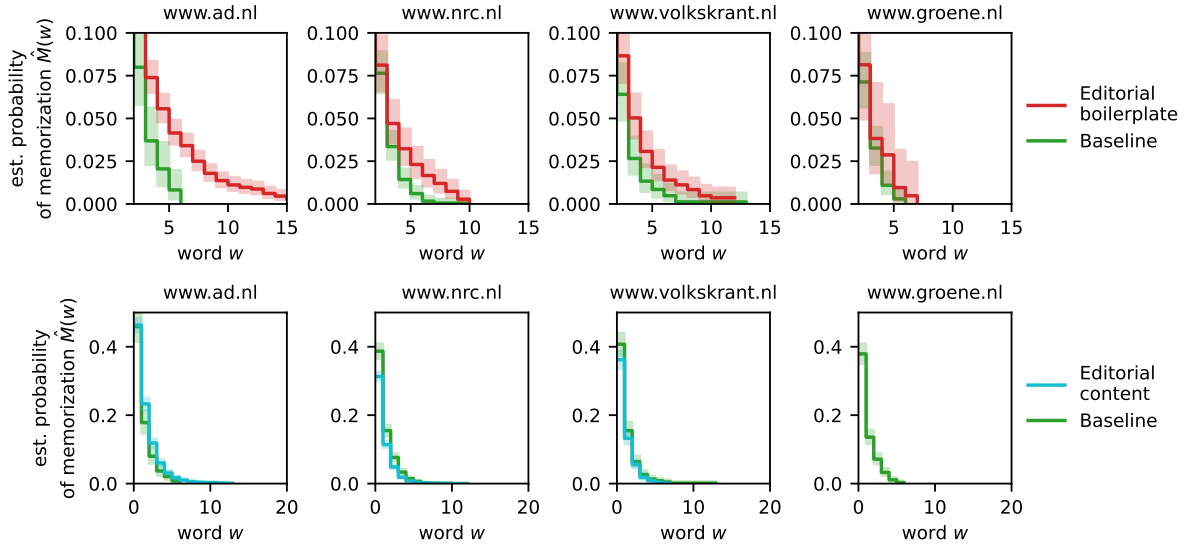

Figure 4: Differences in verbatim completion between editorial boilerplate duplicates (top row), editorial content (bottom row), and the post-training baseline, across the four outlets.

mT5 models are publicly available.[7] The encoder part of mT5 is trained on a "span corruption" objective, i.e. on the prediction of masked token spans of different lengths. In order to replicate our study with GPT-4 we tested the memorization capacities of mT5 on this precise task: how well is mT5 capable of correctly predicting masked token spans that were part of the training data, versus token spans that were not?

To generate our testing data, we created a set of $2,788$ strings from the aforementioned Dutch news websites as occurring in mC4. We selected $1,018$ unique strings occurring only once, and $1,770$ strings occurring multiple times, as in our earlier experiments. An additional set of 300 unique strings were extracted from articles from the same news websites from 2024 (i.e., not in mC4) to assess baseline performance on unseen data. Strings were randomly split once in a prompt part (of minimally 5 tokens) and a completion part.

To generate instances to be presented as input to the mT5 models, we employed a 15% token masking strategy (i.e., on average 15% of all tokens are masked), with an average masked token span length of three consecutive tokens, consistent with pretraining parameters (Xue et al., 2021). If a sentence contains more than one mask, at least one unmasked token is left between the masks.

Figure 6 shows the increasing percentage of correctly predicted masked token spans by increasingly sized mT5 models on the unique and duplicate test items, and the baseline items not in mC4. While prediction accuracies remain relatively low, with just over 5% as the best score attained by the largest model on duplicate strings, there appears to be a reasonably persistent memorization effect with strings that occur often in mC4. On the other hand, unique strings occurring only once in mC4 are predicted less accurately. The largest model (3.7B parameters) predicts baseline token spans more accurately than unique patterns contained in its training data. Although we cannot explain the latter difference, the other results are broadly in

[7] https://huggingface.co/docs/transformers/en/model_doc/mt5

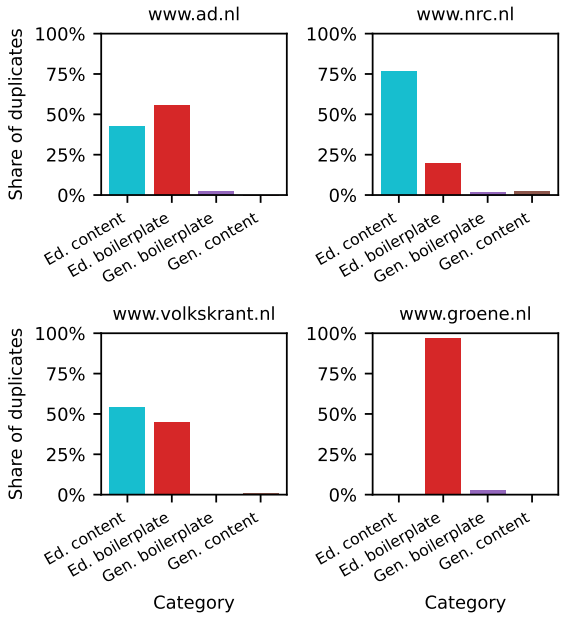

Figure 5: Duplicate type distribution by outlet

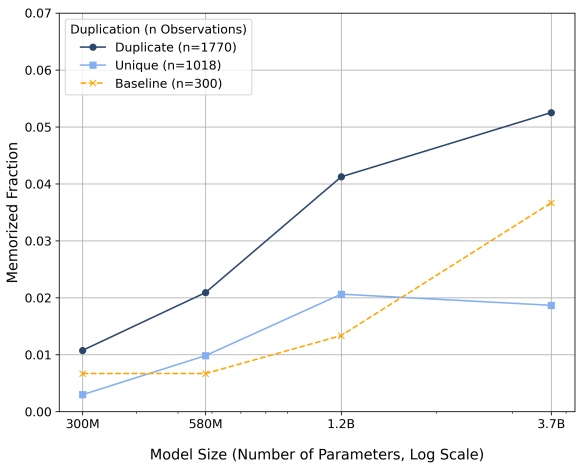

Figure 6: Masked token span prediction accuracy in the mT5 model family (various sizes) on mC4 test data (unique and duplicate strings) and baseline data not in mC4.

line with our earlier findings; they are indicative of a memorization effect that increases with model size and that persists compared to the also improving baseline performance. However, the effect appears to be present only in strings that were repeatedly present in the training data.

6. Discussion

Overall, the results demonstrate that duplicate strings exhibit higher memorization probabilities than the baseline, both in GPT-4 and, to a lesser extent, in mT5, with significant deviations observed in specific outlets such as *ad.nl* and *nrc.nl*. Editorial boilerplate content, in particular, showed the strongest memorization effects. In contrast, smaller datasets, like *groene.nl*, did not show significant differences, likely due to limited sample sizes.

At the same time, we did not find strong evidence for structural memorization in random articles. However, inspecting the examples revealed some suspicious cases of completions that hint at possible memorization, often involving highly specific phrases or verbatim reproductions that appeared unlikely to be generated without prior exposure to the text. It is likely that the relatively small sample size (1,000 articles) was insufficient to demonstrate this effect on a larger scale.

Overall, these findings align with prior research showing that repetitiveness in input data enhances memorization in LLMs (Chang et al., 2023). Notably, editorial boilerplate content appears to occupy a unique position between repetitiveness and uniqueness: it is frequent within the specific context of the newspaper outlet, but not outside of it.

This makes it an effective indicator for memorization and offers publishers a potential method for auditing model outputs for privacy or copyright concerns. For smaller outlets, this is particularly relevant: even without large archives or extensive duplication, recurring editorial phrases can serve as a practical signal of whether their material appears in model training data. Rather than searching for full reproductions, such publishers could focus on distinctive short sequences–like boilerplate or column openings–that are both characteristic and frequent enough to reveal traces of memorization.

At the same time, while our analysis offers an effective approach for outlets that lack the reach of an international news organization, our findings suggest that a certain publication volume is necessary for memorization to occur. In the case of *groene.nl*–the smallest outlet in our study, with 32,877 articles and only 25 duplicates–we found no evidence of memorization. This underscores the need for future research to develop robust methods for detecting memorization in low-volume outlets, such as weekly magazines and personal blogs. One promising direction would be to move beyond the strict string matching employed in this paper and instead detect lightly paraphrased or near-verbatim reproductions.

Finally, while our study focused on Dutch material, the overall method is language-independent and can be applied to outlets in other languages, including English, provided they have sufficient representation in Common Crawl-based datasets. Future research could systematically examine both the effect of outlet volume and the role of language prominence in influencing memorization patterns.

7. Conclusion

In this paper, we explored memorization in Dutch news content by LLMs. Unlike findings related to high-profile outlets like *The New York Times*, we observed that long, verbatim memorization of random Dutch news articles is rare. In our tests on GPT-4 and mT5, randomly selected articles from the presumed training data did not show consistent differences from the baseline.

To better distinguish memorization from coincidental overlap, we introduced a survival analysis framework that models verbatim continuation as a time-to-event process. This approach provides a more robust and interpretable way to capture subtle memorization effects. Applying this method to frequently recurring text fragments–such as editorial boilerplate–revealed systematic deviations from the baseline, suggesting that these short, standardized sequences are disproportionately likely to be retained by LLMs. Overall, our findings demonstrate how survival-based modeling of memorization offers a practical way for smaller publishers to assess whether traces of their content appear in large-scale training datasets.

Limitations

The main experiment described in our work relies on a consumer subscription to the OpenAI API, consuming in the order of tens of thousands of API requests, with an unknown ecological footprint. It also relies on the availability of a completion request option that allows unfinished text sequences to be presented as input, after which a completion is returned. Open LLMs under the full control of the user will have that functionality, but commercial LLM providers may choose at any point to limit API request types and refuse completion prompts or requests. Replication of this work is dependent on this availability.

Extending our experiments with English as well as languages with a smaller presence in mC4 would offer broader views on the effect of language subcorpus size on memorization.

Ethics Statement

The work in this paper is a deliberate attempt at collecting evidence of the inclusion of copyrighted data in the mC4 corpus. We are aware we have been using copies of this contentious dataset, which is available from the Hugging Face platform.[8] Our intention is to broaden the collected evidence of the memorization and recitation capabilities of LLMs of training data segments, e.g. on more languages, so that these findings may become available to stakeholders interested in copyright infringements.

[8] https://huggingface.co/datasets/allenai/c4

Generative AI Disclosure Statement

Generative AI (GPT-4o via ChatGPT) was used to refine the language of this paper. Additionally, version o1 was employed to identify broader inconsistencies in the text.

References

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to chatgpt/gpt-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327.

Nick Diakopoulos. 2024. Fair use, copyright, and the challenge of memorization in the nyt vs. openai. https://medium.com/generative-ai-in-the-newsroom/fair-use-copyright-and-the-challenge-of-memorization-in-the-nyt-vs-openai-7f6c0a13f703. Published in Generative AI in the Newsroom, Medium, February 1, 2024.

Thomas R Fleming and David P Harrington. 1981. A class of hypothesis tests for one and two sample censored survival data. *Communications in Statistics-Theory and Methods*, 10(8):763–794.

Jing Huang, Diyi Yang, and Christopher Potts. 2024. Demystifying verbatim memorization in large language models. *arXiv preprint arXiv:2407.17817*.

Edward L Kaplan and Paul Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481.

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.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *CoRR*.

New York Times. 2023. One hundred examples of GPT-4 memorizing content from the New York Times. Document 1-68, Exhibit J.

Sebastian Pölsterl. 2020. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.