Unmasking Memorization: Assessing Dutch Language Memorization in mT5 Models

Kas Berendsen

A thesis submitted in fulfillment of the requirements for the degree of MSc Applied Data Science

Student Number: 6406394

First Examiner: Antal van den Bosch

Second Examiner: Florian Kunneman

Utrecht University

June 19, 2024

Abstract

This study investigates the memorization of Dutch language content in mT5 models, a multilingual variant of the T5 Transformer-based models. A fill-mask evaluation technique is used to assess memorization and how it varies across different model sizes. Results show that memorization increases with model size up to a certain point. Significant memorization is observed in the 580M and 1.2B sized models, while the smallest 300M and largest 3.7B models are close to baseline generalization performance, minimizing memorization effects. Additionally, the findings reveal that data duplication and context length impact the memorization effect. Moderately duplicated sequences exhibit the highest memorization. Context length similar to pre-training conditions also results in the highest observed memorization and sharply declines when context length is lowered. These findings have implications for model reliability as well as ethical and legal implications, particularly regarding the use of copyrighted training data. This research underscores the need to balance training data and adjust model design to promote generalization and minimize memorization in multilingual models.

Table of contents

1 Introduction

In the field of Natural Language Processing (NLP), memorization within Large Language Models (LLMs) has generated significant attention. Memorization refers to the ability of models to recall specific information from their training data. This phenomenon plays a crucial role in both reliability and generalization, referring to the model's ability to apply learned patterns from its training data to new unseen data. Moreover, memorization also raises ethical and legal questions, specifically in relation to the use of copyrighted material for model training. The recent EU AI Act underlines these concerns, stating that models like these should be transparent in their use of copyrighted material and should therefore disclose the use of this material in generated model output as well as in its training stage (European Parliament, 2023).

This research aims to further investigate the phenomenon of memorization within LLMs, driven by several motivations. First, the assessment of the memorization level in LLMs is essential in order to evaluate their generalization ability. As research of Tirumala et al. (2022) indicates, memorization can either result in overfitting or improve generalization performance, depending on the memorization context and extent. Therefore, the assessment of the memorization level helps in evaluating model robustness and generalization ability when applied to new unseen data, for example generating text after a user prompt.

Second, there are important ethical and legal considerations when it comes to memorization. LLMs are often trained on huge amounts of data obtained through web-scraping. This includes copyrighted content from, for example, news outlets. This process of data collection without the explicit permission of the authors and content owners can result in copyright infringement and concerns regarding data privacy. What is more, as models grow in size in terms of both the parameter count as well as the amount of the training data (Naveed et al., 2024), recent research shows that larger models have the tendency to memorize more data and retain it for longer periods compared to smaller models (Tirumala et al., 2022). This retention could cause unwanted consequences like leakage of personal data or even verbatim reproduction of the training content (Hartmann et al., 2023). Even though it is known that larger models display these characteristics, it remains necessary to assess the exact level of memorization in different contexts to better understand and limit these privacy risks. One of these contexts regards the different languages in multilingual models, as they might display different memorization patterns.

Addressing the multilingual aspect of memorization forms the final motivation for this research. Current research on memorization in LLMs is mainly focused on monolingual English models and datasets. This leaves a gap in understanding memorization dynamics for multilingual models and non-English languages. This study addresses this gap by assessing memorization of the Dutch language within the mT5 model family (Xue et al., 2021), a multilingual variant of the T5 transformer-based models created by Google (Raffel et al., 2019). These models are used for tasks like text summarization and translation.

By presenting insights into the memorization dynamics within the multilingual mT5 models, this study aims to shed light on the robustness and generalization potential of these models. Additionally, this study seeks to address the ethical considerations and implications of content memorization when balancing model performance and the risks associated by training on copyrighted content.

1.1 The mT5 Model Family

This research investigates the mT5 model family, a multilingual variant of T5 transformer-based models developed by Google. The selection of the mT5 models is motivated by several reasons. First, the models are open source and extensive documentation of their architecture and training data is publicly available. This has a clear advantage over closed-sourced models like GPT-4, for which such information is not readily available. Furthermore, the mT5 models are available in multiple sizes[1] enabling the assessment of memorization level in relation to model scale. Finally, the training dataset of mT5 models, known as mC4, is also publicly available (Xue et al., 2021). This makes the extraction of Dutch sequences known to be in the training data possible, facilitating the evaluation of memorization.

1.1.1 mT5 Architecture and Training Objective

The mT5 model (multilingual T5) architecture is based on the T5 model (Text-to-Text Transfer Transformer) (Xue et al., 2021). mT5 is an encoder-decoder transformer model that follows a unified text-to-text format for generative tasks like summarization and translation for which the model generates text based on text input. One of the important characteristics of the mT5 model is its use of the encoder-decoder architecture, originally proposed by Vaswani et al. (2017). By using this architecture, the mT5 models can effectively deal with several NLP tasks using the same training objective (teacher-forced maximum likelihood) for every task. This means that a single set of hyperparameters can be used for fine-tuning different downstream tasks.

Regarding the pre-training objective of the mT5 models, they are pre-trained on a masked language modeling "span-corruption" objective (Xue et al., 2021). Here, consecutive spans of input tokens are replaced with a mask token, after which the model is trained to reconstruct the masked-out tokens. The prediction of the masked-out tokens will also be the model objective when assessing memorization in this research.

1.2 Definitions of Memorization

Within the context of NLP and LLMs, memorization refers to models recalling specific information from their training data. It entails the ability to reproduce verbatim text or generate content strongly resembling the training data. The exact definition of memorization, however, differs with regards to previous research. Nasr et al. (2023) gather multiple notions on memorization and categorize them into two definitions. The first is *extractable memorization*.

Definition 1: Extractable Memorization. *Given a model with a generation routine Gen, an example x from the training set X is extractably memorized if an adversary (without access to X) can construct a prompt p that makes the model produce x (i.e., $Gen(p) = x$).*

The definition of extractable memorization is most applicable to studies like the work of Nasr et al. (2023), which aims to extract the training data without prior knowledge of such data. In this research, however, the goal is to assess the memorization of the Dutch language within the models using a set of sequences known to be in the training data. Therefore, the second definition provided by Nasr et al. (2023), which follows the notion of memorization introduced by Carlini et al. (2023), is more suitable for this research. Here, memorization is defined as *discoverable memorization*.

[1] Huggingface - mT5 Documentation (Consulted on 14 May 2024)

Definition 2: Discoverable Memorization. *For a model with a generation routine Gen, and an example $[p||x]$ from the training set X that consists of a true prefix p and suffix x, suffix x is discoverably memorized if $Gen(p) = x$.*

Discoverable memorization allows the evaluation of the ability of models to recall specific sequences from the training data when given a true prefix as a prompt. An important difference, in contrast to extractable memorization, is that for discoverable memorization, an adversary needs access to the training data beforehand. Since this research focuses on masked token prediction rather than text completion, a third definition is introduced which alters the definition of discoverable memorization to better fit the methodology used in this study.

Definition 3: Masked Token Memorization. *For a model with a generation routine Gen, and an example x from the training set X, where prompt p is derived from sequence x by replacing consecutive token spans with a mask token, sequence x is masked token memorized if the model can correctly predict the masked tokens based on the unmasked context.*

Mathematically, this can be represented as:
Let $x = (t_1, t_2, ..., t_n)$
Let $p = (t_1, t_2, ..., t_{i-1}, [MASK], t_{i+1}, ..., t_n)$
The sequence x is masked token memorized if $Gen(p) = t_i$

Masked token memorization enables the assessment of memorization within the mT5 model family by leveraging the unmasked context of training sequences. By incorporating masked tokens into sequences, this study aligns its evaluation method with the pre-training objective of the mT5 models.

1.3 Related Work

Related studies have investigated different aspects of memorization in LLMs. An example is the work of Carlini et al. (2023), which quantifies memorization in multiple language model families, including GPT-Neo and T5. This study shows that larger models have the tendency to memorize more data, resulting in a log-linear relationship between model size and memorization. Specifically, Carlini et al. (2023) identify three important properties that impact memorization:

1. **Model Size**: Larger models within a model family memorize significantly more than smaller models.
2. **Data Duplication**: Sequences repeated more often in the training data are more likely to be memorized.
3. **Context Length**: Longer context lengths make it easier to extract memorized sequences.

Carlini et al. (2023) tested the T5 models to verify the memorization patterns found in other model families like GPT-Neo. The authors discovered that while T5 models display memorization, the extent was comparatively lower. This variation suggests that different model architectures and training strategies can influence the level of memorization.

To assess the memorization in the tested models, Carlini et al. (2023) used the respective training datasets. For the T5 model family, the C4 dataset was used, a large English corpus derived from the public Common Crawl web scrape[2] (Raffel et al., 2019). For the GPT-Neo models, they were evaluated using The Pile, an English 825 GiB text corpus designed for training large-scale models

[2]Common Crawl (Consulted on 14 May 2024)

(Gao et al., 2020). Both these training datasets are primarily focused on English. This limits the insights into multilingual memorization dynamics.

The research by Nasr et al. (2023) further indicates that memorization is present across different models and data settings. Their study shows that models, including both open- and closed-source models, significantly memorize their training data. The authors show this by presenting multiple methods to extract parts of the training data using the models. The findings of Nasr et al. (2023) reinforce the conclusions of Carlini et al. (2023) and emphasize the broader implications of memorization in LLMs. This underlines the need to understand how different models, including mT5, handle memorization.

Hartmann et al. (2023) also provide an extensive survey of memorization in general-purpose LLMs. In contrast to the other research, they highlight the various types of memorization, including verbatim text, facts, ideas and algorithms, writing styles, distributional properties, and alignment goals. Their taxonomy of different kinds of memorization and discussion on the implications of each type for model performance, privacy, security, and copyright are particularly relevant. Despite earlier research on these domains, Hartmann et al. (2023) are the first to consider them in relation to memorization. They emphasize that memorization in LLMs is not just about verbatim text, but also includes more abstract forms of information, which could have both positive and negative implications for each of the mentioned domains. The authors give an example of the memorization of writing styles. This positively influences model performance as the model is able to transfer a certain style to its output when it is for example tasked to write a formal email. However, models can also memorize unwanted writing styles and can be prone to replicating them. For example, if a model is trained on text with grammar mistakes, it could respond with an output containing faulty grammar, negatively impacting performance.

Research on memorization in a multilingual context is limited. However, a notable exception is the work of Jiang et al. (2020), which creates a benchmark to assess factual knowledge retrieval across 23 languages, including Dutch. Their findings indicate that multilingual language models are challenged in factual knowledge retrieval, particularly in low-resource languages. To obtain factual knowledge, the models have to predict correct information using cloze-style prompts. This method differs from the earlier defined masked token memorization in this study in several ways. Assessing masked token memorization involves recalling and reproducing exact text sequences from the training data. Cloze-style prompts, however, regard a type of fill-in-the-blank question to evaluate the model's understanding of context and its ability to recall specific knowledge. The two methods do share the use of a masking strategy, but they differ fundamentally in their objectives and evaluation metrics. Despite these differences, the work by Jiang et al. (2020) underscores the need for further investigation into the memorization dynamics in multilingual models, especially in languages that have been less studied like Dutch.

1.4 Research Questions and Contributions

These findings in previous work form the foundation and inspiration for this research. Unlike prior studies, which primarily used English datasets, this study investigates the memorization of Dutch language content within the multilingual mT5 model family and how this scales across the model sizes.

For this research, the memorization type will be verbatim text. While Hartmann et al. (2023) emphasize that memorization in LLMs encompasses a broad range of types beyond verbatim text, focusing on verbatim memorization in this study is motivated by several reasons.

Firstly, verbatim memorization directly relates to issues of privacy, security, and copyright. Understanding the extent to which models memorize and reproduce exact text can help mitigate risks associated with sensitive or copyrighted information being unintentionally disclosed.

Secondly, verbatim memorization can affect the reliability and trustworthiness of LLMs in real-world applications. By quantifying and analyzing verbatim memorization, this study contributes to developing reliable and more robust language models.

This focus translates to the first research question guiding this study:

Research question 1: *How does the extent of memorization of Dutch language content vary with different model sizes in the mT5 family?*

In an effort to further investigate the memorization dynamics within the mT5 models, this research also analyzes how context length and data duplication affect Dutch memorization. This contributes to a better understanding of memorization dynamics beyond what has been covered in monolingual models. This focus raises the second research question:

Research question 2: *What are the dynamics of Dutch language memorization across different data duplication and context length scenarios?*

In short, this research aims to extend the investigation to the multilingual version of T5, the mT5 model family, and examine how memorization scales across different model sizes within the specific context of Dutch language processing.

2 Data and Methods

2.1 Data

The mT5 models are trained on a multilingual dataset called mC4 (Xue et al., 2021). This dataset includes text in 101 languages sourced from the Common Crawl. It is important to note that not all languages are represented equally in this dataset. English is the most dominant language, making up 5.67% of the dataset. Dutch is the ninth most dominant language, accounting for 1.98% of the dataset. Despite the English dominance, Dutch is still represented relatively well. Also, the mT5 models use a language sampling exponent to better balance the representation of high- and low-resource languages during training. This provides a sufficient basis for assessing the memorization capabilities of the mT5 models in the Dutch language.

2.1.1 Extracting Dutch Sequences From mC4 Dataset

Dutch sequences are extracted from the mC4 dataset. Specifically, strings from articles in Dutch newspapers ('De Volkskrant', 'NRC', 'Het Algemeen Dagblad', 'Het Financieele Dagblad', and 'De Groene Amsterdammer') are found in the training data. The newspapers are detected using their URL, which can be matched in mC4. Next, the amount of duplication of these sequences in the training data is assessed. This results in a dataset containing sequences that are either present only once or multiple times in mC4. For the duplicated sequences, the duplication count is then saved. In this study, the resulting dataset is referred to as 'pre-training data' and contains 2788 sequences.

2.1.2 Baseline Data

Baseline data is also retrieved in order to validate the results of this research. Importantly, the baseline data has to be similar to the pre-training data to ensure minimal bias. Therefore, sequences of articles covering multiple topics from some of the same Dutch newspapers as the pre-training data ('De Volkskrant', 'NRC', and 'Het Algemeen Dagblad') are retrieved from their respective websites. Importantly, all baseline data is publicly available as the first few sentences before a paywall are retrieved. Also, all baseline data has been published in 2024. This ensures that none of the baseline sequences were seen during mT5 model training in 2021 (Xue et al., 2021).

In another effort to minimize bias, the token length distribution of the baseline data is adjusted to match the pre-training dataset. This is done by truncating baseline sequences on a sentence level using stratified sampling. Specifically, sequences are sampled in proportion to the target token length distribution of the pre-training data. Then, each sequence is truncated at the nearest sentence boundary to ensure natural language flow. The resulting baseline (Figure 5.2) and original pre-training (Figure 5.1) token length distributions can be found in the appendix.

2.2 Methods

To investigate the memorization of Dutch sequences within the mT5 model family, the following question needs to be answered: *How does the performance of mT5 models in predicting masked tokens, as measured through fill-mask evaluation, vary across different model sizes, using masked Dutch sequences extracted from the training data?*

The selection of the fill-mask evaluation is motivated by the fact that the mT5 model family is trained with this fill-mask objective. As a result, the models do not need to be trained on a downstream task, like summarizing content, and can be used as-is with this fill-mask objective in mind. Omitting downstream training prevents possible bias in the memorization performance of the models.

A 15% token masking strategy with an average noise span length of 3 tokens, consistent with pre-training parameters (Xue et al., 2021), is employed in this research. The masking percentage parameter is only altered for the evaluation of different context lengths. This strategy ensures that the evaluation is aligned with the model's pre-training conditions, ensuring a valid assessment of its memorization capabilities.

The code for the data pre-processing and analyses is published on Github[1].

2.2.1 Fill-Mask Evaluation

To assess the memorization capabilities of the mT5 models, the fill-mask evaluation technique is used. This technique involves providing the model with input prompts containing masked tokens and tasking it with predicting the masked tokens based on its pre-training. Prompts p derived from Dutch news sequences are extracted from the training data. 15% of the tokens in each prompt are then masked before presenting it to the model. When evaluating multiple context lengths, sequences are also masked on a 20%, 25%, and 30% level. Depending on the number of tokens to be masked, masking could occur at multiple positions in the sequence, with an average of 3 consecutive masked tokens per position and at least one unmasked token between the positions. The model's ability to accurately predict the masked tokens will serve as a proxy for its memorization performance.

To evaluate the performance of the model in predicting the masked tokens, the (normalized) Levenshtein distance metric is used. This metric measures the similarity between the model's predictions and the sequence's ground truth. This provides a quantitative assessment of the accuracy of the model's memorization capabilities. The distance is measured on a character level, comparing individual characters of the combined tokens within each masking position. Additionally, the Levenshtein distance is normalized based on the length of the longest combined input tokens, ensuring that the metric is scaled appropriately and accounts for variations in token lengths between the model's predictions and the ground truth. Importantly, the distances are measured independently for each masking position within a sequence. Consequently, if a sequence contains multiple masked positions, the normalized Levenshtein distances for the tokens within each masking position are averaged to obtain an overall distance for the sequence. See Figure 2.1 for an example of the fill-mask evaluation of a single sequence.

Different model sizes within the mT5 family are selected for evaluation to examine how memorization scales with model size. These include the small ($\approx$ 300M parameters), base (580M), large (1.2B), and XL (3.7B) variants. Notably, the parameter counts of the mT5 models are higher compared to their corresponding monolingual T5 model variants due to the larger multilingual vocabulary used in mT5 (Xue et al., 2021).

[1] Github Repository

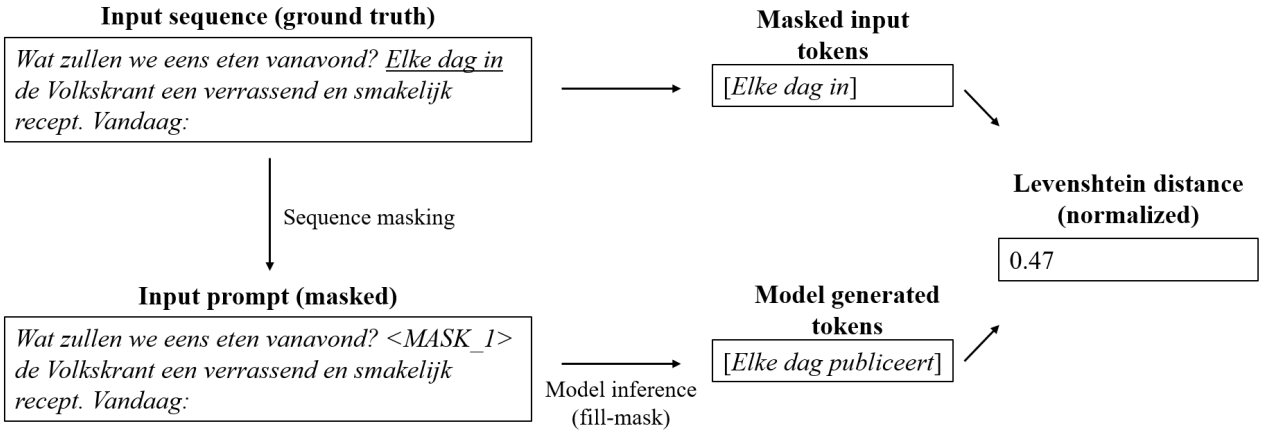


Figure 2.1: An example input sequence that is masked at a single position, demonstrates the fill-mask evaluation method. If multiple positions are masked, the (normalized) Levenshtein distances are calculated for each position, after which the distances are averaged.

2.2.2 Assessing Memorization

After the fill-mask evaluation, the memorization level for the various model sizes is assessed. This is done by calculating the fraction of zero (average) Levenshtein distances after the fill-mask evaluation of the pre-training sequences. This is conducted for each model size to obtain results relating memorization level to model scale. This fraction will be referred to as 'memorized fraction' in this study.

2.2.3 Data Duplication

The impact of data duplication on memorization is assessed by categorizing sequences based on their frequency of occurrence in the training data. The duplication count for every sequence is extracted. For all duplication counts, the memorization performance is evaluated for the multiple model sizes using the fill-mask method described earlier. The results are then analyzed to determine the dynamics between sequence duplication and memorization across varying model sizes.

By categorizing sequences based on duplication frequency, this research seeks to understand whether models are more likely to memorize sequences that appear more frequently in the Dutch training data. This analysis is crucial for identifying patterns in how models handle repeated information and whether increased duplication leads to higher memorization rates. Evaluating multiple model sizes allows for insights into how model capacity influences the memorization of duplicated sequences.

2.2.4 Context Lengths

To analyze how context length affects memorization, sequences are masked at varying percentages: 15%, 20%, 25%, and 30%. As 15% is the original pre-training setting, higher percentages are increasingly less like the training conditions. The context length refers to the proportion of the sequence that remains unmasked. For each context length, the performance using the multiple model sizes is assessed. This investigation aims to determine the optimal conditions under which the models memorize best and how this memorization scales with model size.

Importantly, this research uses a masking strategy in contrast to text completion in related work on other models like GPT. This results in a difference in the context available to the models. Memorization assessment using text completion only gives models left-sided context. The use of masking in this study causes the mT5 models to be presented with both left- and right-sided context. This difference could mean that the mT5 models are presented with more information, potentially improving their performance. This should be taken into account when comparing the findings of this study to others using text completion.

2.3 Baseline Comparison

When assessing memorization in various model sizes, the results are compared to the baseline performance which is obtained by using the same methods on the baseline data. As the models have not seen the baseline data during training, the baseline performance will be a proxy for their generalization performance. Consequently, the difference between the fraction of zero Levenshtein distances in the pre-training data and the baseline data will be the 'true' memorized fractions of the models, accounting for the balance between memorization and generalization in the model output.

3 Results and Analysis

Various facets of Dutch language memorization in mT5 models are analyzed. For this section, the focus will be on the memorized fraction, the fraction of zero Levenshtein distances, obtained by applying the fill-mask evaluation method using various model sizes. A distribution of all the resulting Levenshtein distances can be found in the appendix (Figure 5.3).

3.1 Memorization in Various Model Sizes

The first analysis compares the memorization of pre-training data with baseline data across different model sizes, as shown in Figure 3.1. Most apparent is that the memorized fraction increases with model size when the models are applied to the pre-training sequences. The same is true for the baseline data but to a lesser extent. Notably, the baseline performance does significantly improve for the 3.7B model. As the baseline performance is a proxy for generalization performance, this indicates that the 3.7B model generalizes far better when compared to the smaller model sizes.

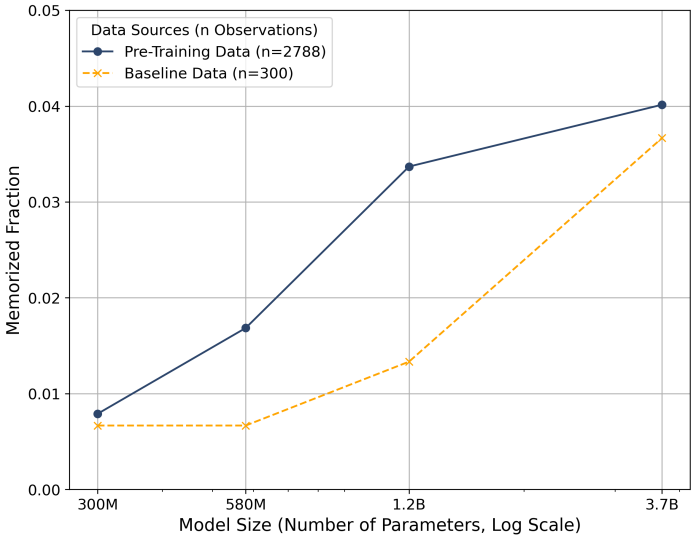


Figure 3.1: Memorization in the mT5 model family. These results use a 15% token masking strategy, resulting in an 85% context length. This context length is consistent with the pre-training objective parameters of the models.

As described, the fraction difference between the pre-training and baseline data will indicate the 'true' memorization effect of the models, accounting for baseline generalization performance. Based on previous research, like the work of Carlini et al. (2023), it is hypothesized that memorization increases with model size and that model performance on pre-training data will show higher memorization compared to the baseline data. To test this hypothesis and obtain the statistical significance of the observed fraction differences, a one-sided bootstrap test is conducted. The bootstrapping method is selected as the baseline data sample has fewer observations (n=300) compared to the pre-training

Model Size	Fraction Difference	One-Sided P-Value	Interpretation
300M	0.001	0.324	Not significant
580M	0.010	0.017	Significant
1.2B	0.020	0.003	Significant
3.7B	0.003	0.311	Not significant

Table 3.1: One-sided bootstrap hypothesis test of pre-training data versus baseline data. The p-values represent the probability of observing a memorized fraction difference between the pre-training and baseline data less than or equal to zero. If this probability is small (5% level), this hypothesis is rejected indicating that the pre-trained memorized fraction is significantly greater when compared to the baseline.

sample (n=2788). Bootstrapping involves repeatedly sampling from the data with replacement to create multiple simulated samples. By bootstrapping the smaller baseline sample, and calculating the memorized fraction for each bootstrapped sample, a robust distribution of the memorized fractions is obtained and compared to the pre-training data.

Table 3.1 presents the results for this statistical test and indicates the following:

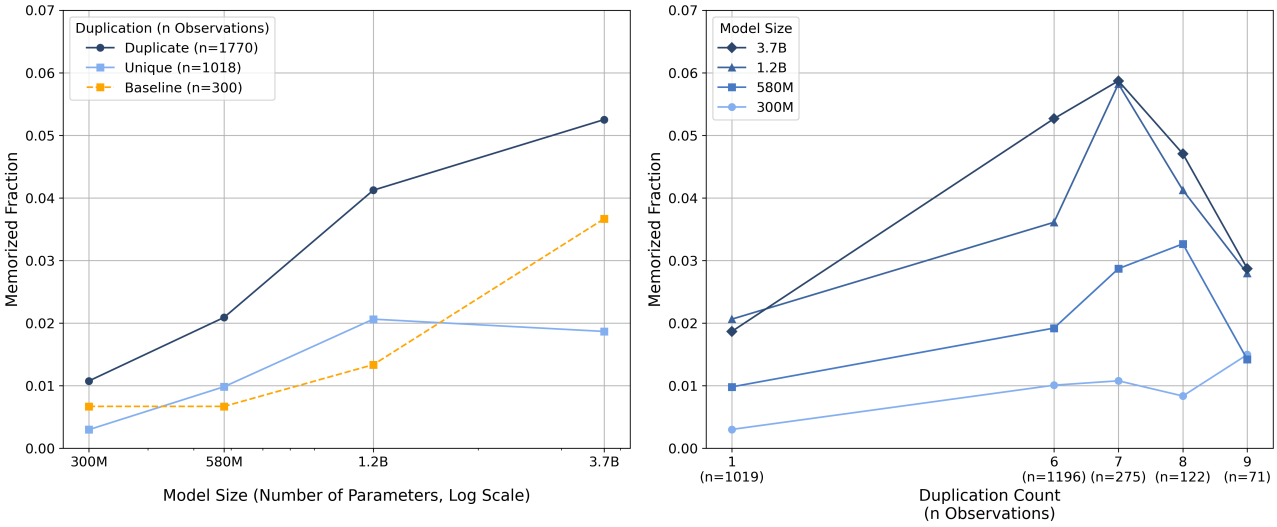
- **300M model:** There is no significant difference in the memorized fraction between the pre-training and baseline data. Also, the overall performance of this model is quite low for both data samples.
- **580M and 1.2B models:** These models show significant memorization effects, indicating that as model size increases, the tendency to memorize training data also increases up to a certain point. The models show a 'true' memorized fraction of 1% and 2% respectively.
- **3.7B model:** Interestingly, the fraction difference for the largest tested model is not significant. This could indicate that very large models have improved generalization capabilities, balancing memorization and generalization.

3.2 Memorization in Unique versus Duplicate Sequences

To uncover the underlying dynamics of the observed memorization, the analysis continues by examining how the memorization of unique versus duplicate sequences changes with model size. As illustrated in Figure 3.2a, there is a notable trend of duplicate sequences being increasingly memorized as model size grows. Interestingly, this trend does not fully hold for unique sequences, for which the memorized fraction saturates in the largest 3.7B model.

Again, the fraction differences regarding the baseline are tested for statistical significance. The same hypothesis test is conducted and the results are presented in Table 3.2. The test results for the unique sequences show no statistical significance for all model sizes. This indicates that there is no statistical evidence that the unique pre-training sequences show more memorization compared to the baseline.

The test results for the duplicate sequences do show significance for the 580M and 1.2B models and are not significant for the smallest 300M and largest 3.7B models, similar to the overall results without the sequence duplication split (Table 3.1). Interestingly, the significant memorization in the 580M and 1.2B models has increased for duplicated sequences in the pre-training data, showing 'true' memorized fractions of 1.4% and 2.8% respectively.



(a) (b)

Figure 3.2: **(a)** Memorization of unique and duplicate sequences as a function of model size using a 15% masking level. The duplication counts range from 6 to 28. **(b)** Memorization of sequences based on their duplication count using a 15% masking level. The duplication count in this graph goes up to 9, beyond which the number of observations is too low per count, causing high variation in the results due to unfair representation of these counts. See Figure 5.4 in the appendix for the specific duplication count distribution in the training data. For these results, bootstrapping (a statistical resampling method) is used to correctly present small sample sizes and assess the variation in the specific duplication count results. This variation is reflected in the confidence intervals found in the appendix (Figure 5.5).

Duplication	**Model Size**	**Fraction Difference**	**One-Sided P-Value**	**Interpretation**
Unique	300M	-0.004	0.863	Not significant
Unique	580M	0.003	0.332	Not significant
Unique	1.2B	0.007	0.111	Not significant
Unique	3.7B	-0.018	0.554	Not significant
Duplicate	300M	0.004	0.141	Not significant
Duplicate	580M	0.014	0.004	Significant
Duplicate	1.2B	0.028	0.0003	Significant
Duplicate	3.7B	0.016	0.095	Not significant

Table 3.2: One-sided bootstrap hypothesis test of unique and duplicate pre-training data versus baseline data. The p-values represent the probability of observing a memorized fraction difference between the pre-training and baseline data less than or equal to zero. If this probability is small (5% level), this hypothesis is rejected indicating that the pre-trained memorized fraction is significantly greater when compared to the baseline.

Further inspecting the specific duplication counts in Figure 3.2b reveals that the memorized fraction increases for all models when sequences are increasingly duplicated, until a duplication count of 7. Past this point, memorization falls for the larger models (1.2B and 3.7B) while the smaller models (300M and 580M) show varied results. The 300M model shows a light dip at a count of 8, followed by increased memorization at 9. The 580M model shows its peak at 8, after which the memorized fraction decreases sharply like the two largest models. These observations suggest that, despite duplication generally increasing memorization, there is a threshold beyond which additional duplication does not contribute to further memorization and may even hinder it, particularly for the larger models.

It is important to note that the sample sizes of the duplicate sequences decrease as the duplication count increases. Smaller samples could affect the results, causing an unfair representation of the higher counts. To assess variations in the results and estimate robust confidence intervals when dealing with these smaller sample sizes, bootstrapping is used in Figure 3.2b. The resulting confidence intervals in the appendix (Figure 5.5) indicate that the higher duplication counts indeed show more variation in their memorized fraction.

To account for this, a weighted bootstrap method is employed to assess the memorized fraction in the duplicated sequences more robustly. In this method, each duplication count sample is assigned a weight proportional to its frequency in the dataset. This weighing combined with bootstrapping ensures that smaller duplication counts are fairly represented without artificially inflating their size and effects. This method provides robust memorization fractions and confidence intervals. These weighted confidence intervals can be found in the appendix (Figure 5.6), which now show less variation in the memorized fractions for all duplication counts.

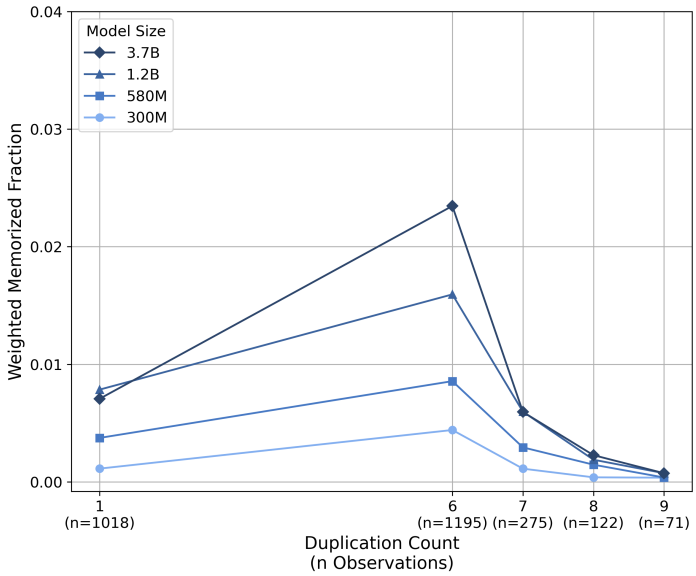


Figure 3.3: Weighted bootstrap memorization of unique and duplicate sequences as a function of model size using a 15% masking level. The memorized fraction for each duplication count is calculated using a weighted bootstrap method to ensure a fair comparison of memorized fractions across varying sample sizes.

Comparing the non-weighted (Figure 3.2b) and weighted (Figure 3.3) bootstrapped memorization fractions provides more robust insights into the effect of duplication on memorization. The weighted method shows that while duplication generally increases memorization, all model sizes peak at a duplication count of 6, followed by a decline for the higher counts. The weighted memorized fractions for these higher counts are even below the memorization level of unique sequences.

3.3 Memorization in Various Context Lengths

The third analysis investigates how context length influences memorization across different model sizes. Figure 3.4 provides insights into how varying the context length by using different masking percentages affects the memorized fraction of sequences.

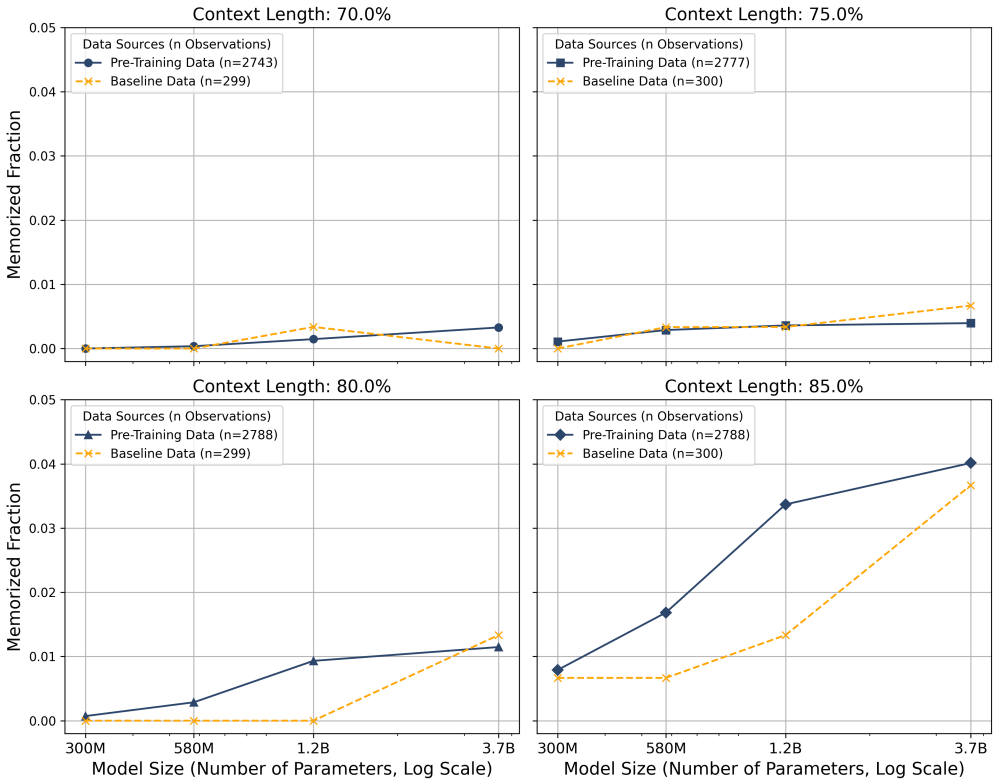


Figure 3.4: Impact of context lengths on memorization over various model sizes. The pre-training data contains both the unique and duplicate sequences.

Like previous analyses, the significance of the difference between the pre-training data and baseline data is tested, in this case for all the combinations of context length and model size. These test results are available in the appendix (Table 5.1). A note on the interpretation of these test results, some combinations show significance but should be interpreted with caution due to small fraction differences and zero values.

Regarding Figure 3.4, what is immediately clear is that when context length decreases, and therefore increasingly deviates from the model's pre-training condition of 85% context length, memorization fractions drop sharply. Especially for the smallest context lengths (70% and 75%), there is not a significant difference in the pre-training and baseline performance, indicating no increased memorization effect when applying the models to before-seen sequences using these context lengths. For 80% context, the memorized fraction of the pre-training data is significantly higher when compared to the baseline for all model sizes except the largest 3.7B model. However, this significance should be interpreted with caution as it is a result of the zero fractions of the baseline performance.

These results indicate that memorization is strongest when context length is the same as the model's pre-training conditions. Decreasing context length greatly minimizes the effect of memorization for these models, mostly resulting in similar performance as the baseline.

4 Discussion

To assess the Dutch memorization capabilities of the mT5 models, this section synthesizes the findings of the conducted analyses and discusses the broader implications and ethical considerations.

4.1 Memorization in Various Model Sizes

This research set out to assess the extent of memorization of Dutch language content in various mT5 model sizes. The results in Figure 3.1, combined with the statistical test results in Table 3.1, indicate that at first memorization increases when model size increases. The 580M and 1.2B models show a memorized fraction of 1% and 2% respectively when accounting for the generalization performance using the baseline. However, baseline generalization performance greatly improves for the largest 3.7B model, minimizing the memorization effect.

Comparing these results to the findings of Carlini et al. (2023) regarding the monolingual T5 model, both similarities and discrepancies are found. In terms of similarity, the observed memorization fractions for the pre-training data are within the same ballpark. The 3.7B mT5 model assessed in this study has a memorized fraction of 4%, not accounting for baseline performance. Comparatively, the 2.8B T5 model memorizes roughly 4.8%. However, for the T5 models Carlini et al. (2023) conclude that larger models exhibit higher memorized fractions. The findings in this study only partly agree with this. Due to the observed generalization improvement for the largest 3.7B model and low performances for both the pre-training and baseline data in the smallest 300M model, increasing memorization is only found for the 580M and 1.2B mT5 models. Notably, Carlini et al. (2023) do not compare their T5 model results to a baseline to account for increasing generalization performance when model size increases like they do in testing the GPT-Neo model family. Therefore, the 'true' memorized fractions in mT5 models found in this study, which account for generalization, are not directly comparable to the previous work on monolingual T5 memorization as those fractions could be overestimated.

Next to omitting a baseline, the discrepancy in the memorization trend of mT5 and T5 models in regards to model scale could also be explained by the most prominent difference between the two: their training data. Most pre-training strategies and objectives between these two model families are very similar, with the exception of the implementation of language sampling and a larger vocabulary size in the mT5 models (Xue et al., 2021). Other than that, the monolingual and multilingual training data is the biggest differentiator between the two. In a multilingual context, the model's capacity is 'shared' across different languages. This could cause the small to medium-sized models to not generalize as well and rely more on memorization, simply falling back to exact examples from their training data. However, as models grow sufficiently large (like the 3.7B mT5 model), this balance between memorization and generalization improves. Here, the 3.7B mT5 model has enough capacity to handle the complexities of multilingual training data, improving generalization. This leads to less reliance on memorization, decreasing the risk of overfitting on the training data. This improved generalization in larger models reduces the memorization effect, which can explain why the 'true' memorized fraction of the largest mT5 model is not significant despite its increased capacity.

4.2 Duplication and Context Length Dynamics

In an effort to further explain the observed memorization and its trend, this study analyses how the underlying dynamics of data duplication and context lengths affect memorization. Understanding these aspects will help in uncovering the mechanisms behind memorization and generalization in the mT5 model family.

4.2.1 Duplication

Regarding data duplication, the analysis reveals that duplicate sequences result in more memorization when compared to unique sequences, especially for larger model sizes (Figure 3.2a). Furthermore, there is no statistical evidence that unique pre-training sequences show more memorization compared to the baseline in any of the model sizes whereas this is the case for duplicate sequences in the 580M and 1.2B models. These findings indicate that essentially the unique sequences are lowering the overall memorization effect and that the significant memorization in the 580M and 1.2B models is mostly driven by duplication in the pre-training data. Consequently, by only regarding the duplicated sequences, the 'true' memorization fractions for the significant 580M and 1.2M models have now increased to 1.4% and 2.8% respectively.

The lack of significant memorization for unique sequences could be attributed to the combination of infrequent exposure and model objective during training. The infrequent training exposure of these sequences means that the model does not have a lot of opportunities to memorize. Moreover, the mT5 model training objective involves reconstructing masked token spans by using 85% of the context. For unique sequences, the model relies more on understanding the context rather than memorizing specific token sequences, promoting generalization. In the case of duplicate sequences, the model might memorize overlapping token contexts it has seen before. This repeated exposure allows the model to use that information to reconstruct the masks, using memorization instead of relying on generalization.

Investigating the underlying memorization dynamics for specific duplication counts reveals that memorization increases with the duplication count up to a certain point (Figure 3.3). Specifically, sequences duplicated 6 times show the highest memorization, followed by a decline in higher duplication counts for all model sizes. This could be explained by saturation in the memorized context during the training stage. The overlap between the previously memorized contexts causes redundant information to be passed to the models during training. The redundant information forms noise as the models struggle to differentiate between relevant and redundant information, decreasing the ability of the models to memorize.

Relating this to previous work, Carlini et al. (2023) found a clear relationship with regard to duplication when testing the GPT-Neo model family: increased data duplication results in increased memorization. However, this trend does not fully hold for the T5 models as the authors found something similar to a saturation point like in this study. One thing to note, the range of the absolute duplication counts differs between the two model families and their respective training datasets. The duplication counts in the monolingual C4 dataset go significantly higher when compared to the counts of the tested fraction of mC4 in this study. That being said, the authors found for the T5 models that sequences repeated 159 to 196 times result in less memorization compared to sequences repeated ~140 times. The authors explain this saturation point by qualitatively assessing the sequences. They note that the less frequent sequences (~140 duplications) contain a lot of whitespace which is easier to predict correctly. This is not the case for this study, as some data-cleaning steps prevent this. Therefore, noise due to increasing duplication is a better explanation for the observed memorization trend for the mT5

models in this study and might also apply to the T5 family. All in all, it is interesting to see that a saturation point is found in both the monolingual T5 model as well as its multilingual counterpart.

4.2.2 Context Length

Finally, the underlying dynamics of various context lengths in relation to memorization are analyzed. The findings in Figure 3.4 reveal that the memorization effect is strongest when context length is similar to the model's pre-training conditions, here 85% context. When context is decreased, memorization is minimized as the models perform similarly to the baseline.

It makes sense that model performance, and therefore its generalization ability, drops when less context is available to the model. The model has more spans of tokens to predict with less context to do it, making the chance of exact correct predictions smaller. This explains the observed baseline performance. What is interesting, however, is that model performance applied on pre-training data quickly follows the baseline performance when context length is decreased.

A possible explanation is the increasingly broken-up context available to the model when context length is decreased. Over all context lengths, masked-out token spans within a sequence remain to have an average length of three tokens, constant with the model's training parameters. The number of these three token masking spans increases when context length is lowered in the sequences. If the context length is really low, combined with a relatively short sequence, it is even possible that there is only a single word between token masks, resulting in presenting the model with broken-up context. This fragmented context makes it a lot harder for the model to recognize sequences and match them to its pre-training data, causing sharp drops in its memorization performance when context length is decreased.

This phenomenon underscores the importance of context continuity in order to invoke memorization. Due to fragmented context, the models are not effective in memorizing their pre-training data, forcing models to rely on generalization. That being said, fragmented context will also hinder overall model performance as seen by the drop in baseline performance.

4.3 Implications of Memorization

Finding memorization of Dutch training data using the mT5 models has several implications. On the one hand, the findings in this study relate to model performance, resulting in a better understanding of the balance between memorization and generalization. Next to this, the findings also affect ethical and legal concerns.

4.3.1 Memorization and Generalization

In terms of the balance between memorization and generalization in model performance, the findings show that the mT5 models can lean towards one or the other given a certain setting. This understanding is crucial for optimizing this balance in multilingual models going forward. If the goal is to minimize memorization and promote generalization, improving model performance in terms of robustness and preventing overfitting, several suggestions for training data compilation and model design can be provided based on the findings in this study:

1. **Increase Model Size:** Regarding model size, larger models such as the 3.7B mT5 model show improved generalization capabilities, effectively minimizing memorization. Therefore, increasing model size could allow for better handling of the complexities introduced by multilingual training data, promoting generalization.
2. **Balance Training Data with Regards to Duplication:** The findings regarding duplication in the training data show a peak in memorization when sequences are moderately duplicated, whilst unique and highly duplicated sequences show less or even no memorization. This understanding can be used when compiling data for model training to minimize memorization.
3. **Introduce Variability in Context Lengths During Training:** The context length analysis shows that memorization is significantly higher when context length is similar to the model's pre-training condition. Introducing variability during model training by using various context lengths could help the model to adapt better to various levels of context. This will make the model more robust and flexible as the model is not trained using a single context length, forcing the model to develop more general representations of the data. This strategy could also promote generalization and minimize memorization.

4.3.2 Ethical and Legal Implications

This study proves that content from Dutch news outlets is memorized in the mT5 models to a certain degree. It is important to consider what this means in terms of ethical and legal implications. In terms of ethics, previous research indicates that verbatim memorization in LLMs can cause risks (Hartmann et al., 2023). Reproducing training data caused by memorization could leak sensitive or proprietary information, posing risks in terms of privacy and security.

Regarding legal implications, according to Dutch copyright law[1] copyright is violated by using the content of Dutch news outlets in model training without proper authorization. On top of that, this study shows that the mT5 models memorize and reproduce part of this content in their output. This copyright violation might warrant legal action. An example of this is the recent lawsuit regarding alleged copyright infringement against OpenAI, the company behind ChatGPT and the underlying GPT-4 model. This legal action was filed by The New York Times after discovering that OpenAI used their articles for model training (Grynbaum & Mac, 2023). Interestingly, other news outlets take a different approach to deal with copyright infringement. For example, OpenAI and The Financial Times recently announced a strategic partnership (The Financial Times, 2024). According to the two companies, this will enhance ChatGPT with attributed content of The Financial Times.

To address these ethical and legal concerns, transparency and accountability in these models and their developers become increasingly important. Legislation is one way to ensure this. The recent EU AI Act is a good example, mandating transparency in the use of AI by enforcing the inclusion of training data source disclosure (European Parliament, 2023). An example of enforcing accountability is a recent fine imposed on Google in France due to infringement of intellectual property rules (Chrisafis, 2024). The French competition authority said Google's LLM 'Bard', since rebranded as 'Gemini', was trained on content from publishers and news agencies without notifying them. Findings of memorization like in this study could aid in shaping legislation or taking legal action to help mitigate copyright infringement.

[1] Copyright Law in The Netherlands (Consulted on 19 June 2024)

4.4 Limitations and Future Work

This study is not without its limitations. First and foremost, this research solely focuses on the memorization of Dutch sequences extracted from news articles. This does provide further insight into the multilingual aspect of memorization, but future research should investigate multiple languages to see if the findings generalize.

Next, the focus lies on the mT5 model and its use of masking during model training. Future work should test other models and architectures to ensure more robust findings on Dutch memorization in general.

In terms of the baseline used in this study, future work might consider to gather more baseline sequences, equal to the size of the tested pre-training sample, to prevent the use of bootstrapping. Even though bootstrapping helps in providing more robust results, completely omitting this step will make the comparison between pre-training and baseline performance more accurate.

Furthermore, it is important to note that the findings regarding unique versus duplicate sequences are specific to the range of duplication counts present in the assessed Dutch training data for the mT5 model family. The relatively lower presence of very high duplication counts in this dataset may limit the generalizability of the findings. Further research with datasets containing higher duplication counts, found by for example using a different language, could provide additional insights into memorization dynamics across different model sizes and data duplication. Also regarding data duplication, only sequences duplicated five times or more are extracted from mC4. This leaves a gap for investigating sequences duplicated two to five times in the training data.

5 Conclusion

This research set out to investigate the extent of Dutch news content memorization in mT5 models over different model sizes. The findings show that memorization increases with model size up to a certain point. Specifically, the 580M and 1.2B models exhibit significant memorization, while the smallest (300M) and the largest (3.7B) models do not when accounting for baseline generalization performance.

Investigating the underlying dynamics of data duplication and different context lengths reveals further insights into Dutch memorization. Memorization peaks when sequences are moderately duplicated after which memorization is minimized. Also, duplicate sequences drive significant memorization in the 580M and 1.2B models, while unique sequences do not. Regarding context lengths, Dutch content is mostly memorized when context length matches the pre-training conditions (85%). Decreasing context sharply reduces the memorization effect, resulting in similar performance to the baseline.

Overall, the findings in this study emphasize the importance of balancing training data and adjusting model design to promote generalization and minimize memorization in multilingual large language models. This balance is crucial for developing reliable, ethical, and legally compliant models going forward.

References

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., & Zhang, C. (2023). Quantifying memorization across neural language models. https://doi.org/10.48550/arXiv.2202.07646

Chrisafis, A. (2024, March). Google fined €250m in France for breaching intellectual property deal. https://www.theguardian.com/technology/2024/mar/20/google-fined-250m-euros-in-france-for-breaching-intellectual-property-rules

European Parliament. (2023, December). EU AI Act. https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., & Leahy, C. (2020). The pile: An 800gb dataset of diverse text for language modeling. https://doi.org/10.48550/arXiv.2101.00027

Grynbaum, M., & Mac, R. (2023, December). New York Times sues OpenAI and Microsoft over use of copyrighted work. https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html

Hartmann, V., Suri, A., Bindschaedler, V., Evans, D., Tople, S., & West, R. (2023). Sok: Memorization in general-purpose large language models. https://doi.org/10.48550/arXiv.2310.18362

Jiang, Z., Anastasopoulos, A., Araki, J., Ding, H., & Neubig, G. (2020). X-FACTR: Multilingual factual knowledge retrieval from pretrained language models (B. Webber, T. Cohn, Y. He, & Y. Liu, Eds.), 5943–5959. https://doi.org/10.18653/v1/2020.emnlp-main.479

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., & Lee, K. (2023). Scalable extraction of training data from (production) language models. https://doi.org/10.48550/arXiv.2311.17035

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2024). A comprehensive overview of large language models. https://doi.org/10.48550/arXiv.2307.06435

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*. https://doi.org/10.48550/arXiv.1910.10683

The Financial Times. (2024, April). Financial Times announces strategic partnership with OpenAI. https://aboutus.ft.com/press_release/openai

Tirumala, K., Markosyan, A., Zettlemoyer, L., & Aghajanyan, A. (2022). Memorization without overfitting: Analyzing the training dynamics of large language models (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh, Eds.). *35*, 38274–38290. https://proceedings.neurips.cc/paper_files/paper/2022/file/fa0509f4dab6807e2cb465715bf2d249-Paper-Conference.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. https://doi.org/10.48550/arXiv.1706.03762

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2021). Mt5: A massively multilingual pre-trained text-to-text transformer. https://doi.org/10.48550/arXiv.2010.11934

Appendix

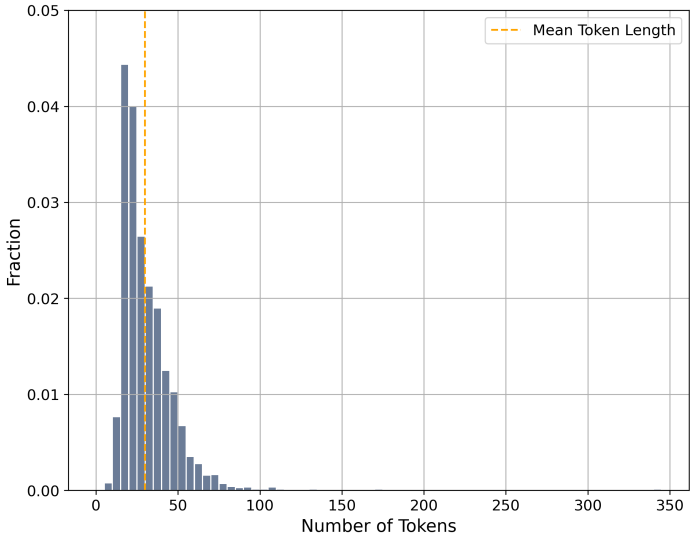


Figure 5.1: Distribution of the number of tokens in the input sequences extracted from the mC4 pre-training data. The mean token length of the input sequences is 30 tokens.

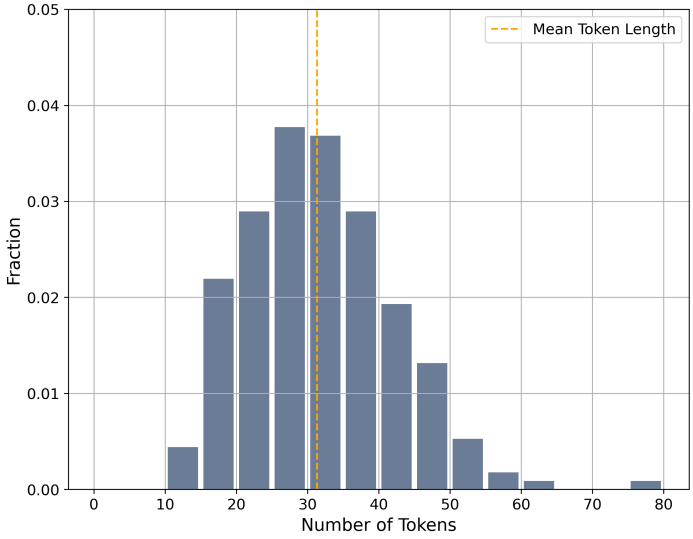


Figure 5.2: Distribution of the number of tokens in the input sequences extracted from the baseline data. The mean token length of the input sequences is 31 tokens.

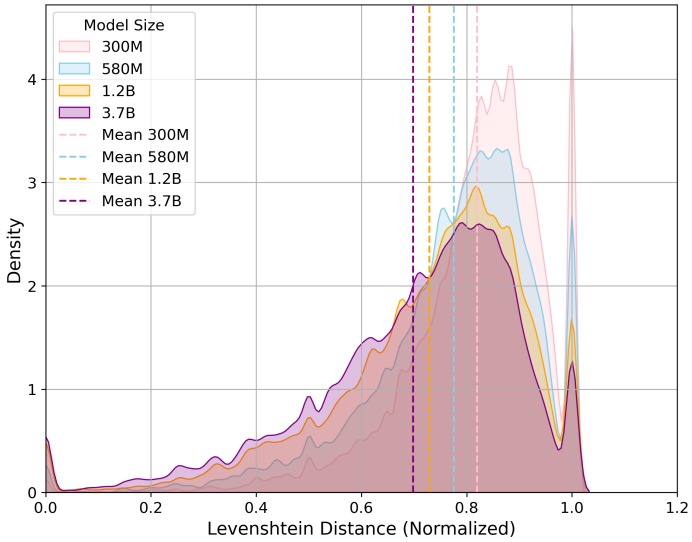


Figure 5.3: The distribution of the resulting Levenshtein distances after filling in the masks in the pre-training input sequences using various mT5 model sizes. All context lengths are included in this figure. Kernel density estimation (KDE) is applied to smooth the distribution curves, providing a more interpretable view of the underlying data trends. The smoothing causes some observations to be displayed beyond the normalized distance range of 0 to 1.

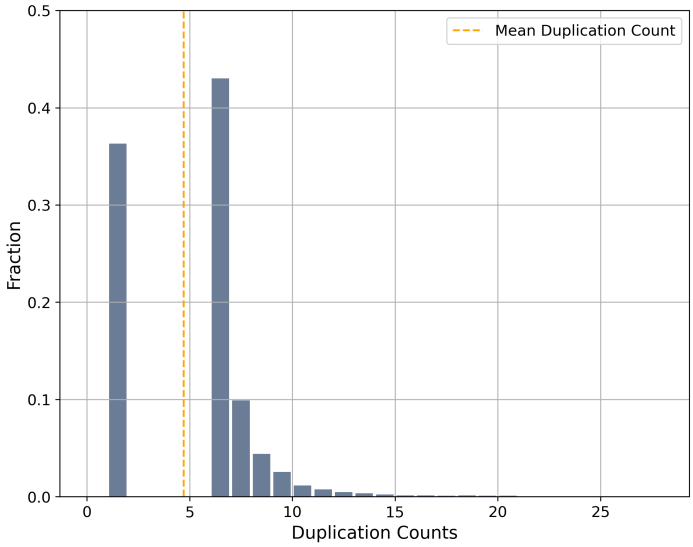


Figure 5.4: Distribution of the duplication counts of the input sequences extracted from the mC4 data.

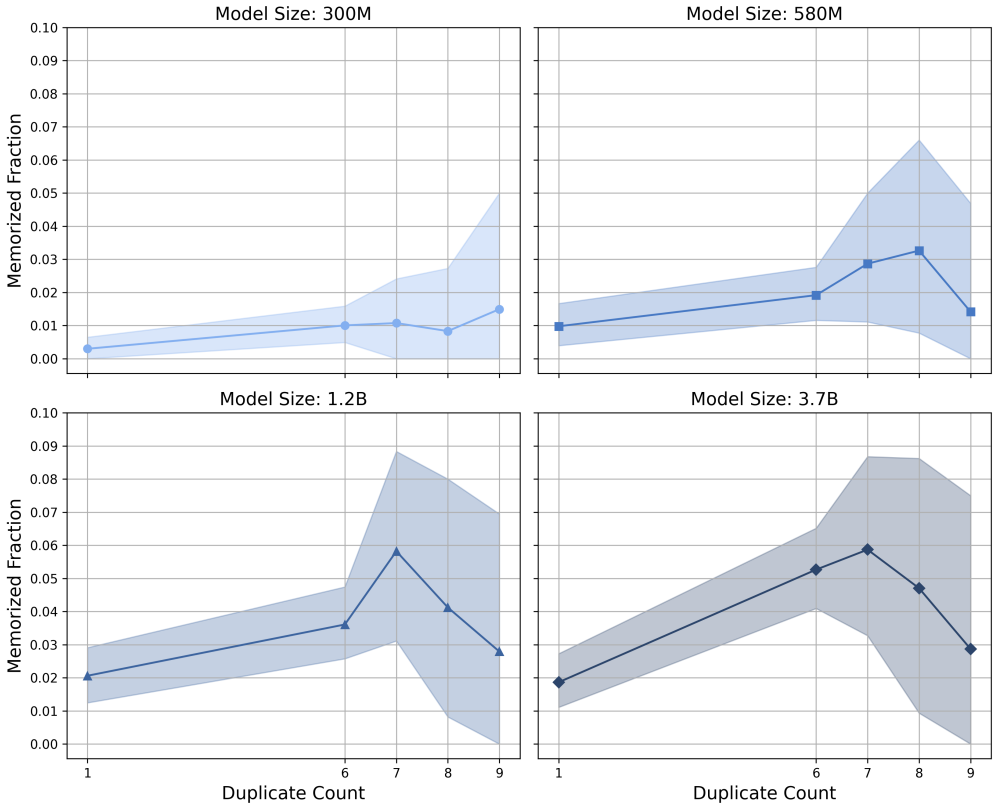


Figure 5.5: Confidence intervals of the memorized fractions as a function of duplication counts for different model sizes (300M, 580M, 1.2B, and 3.7B parameters). The confidence intervals are at the 95% level, illustrating the variation in memorized fractions across specific duplication counts and their respective sample sizes.

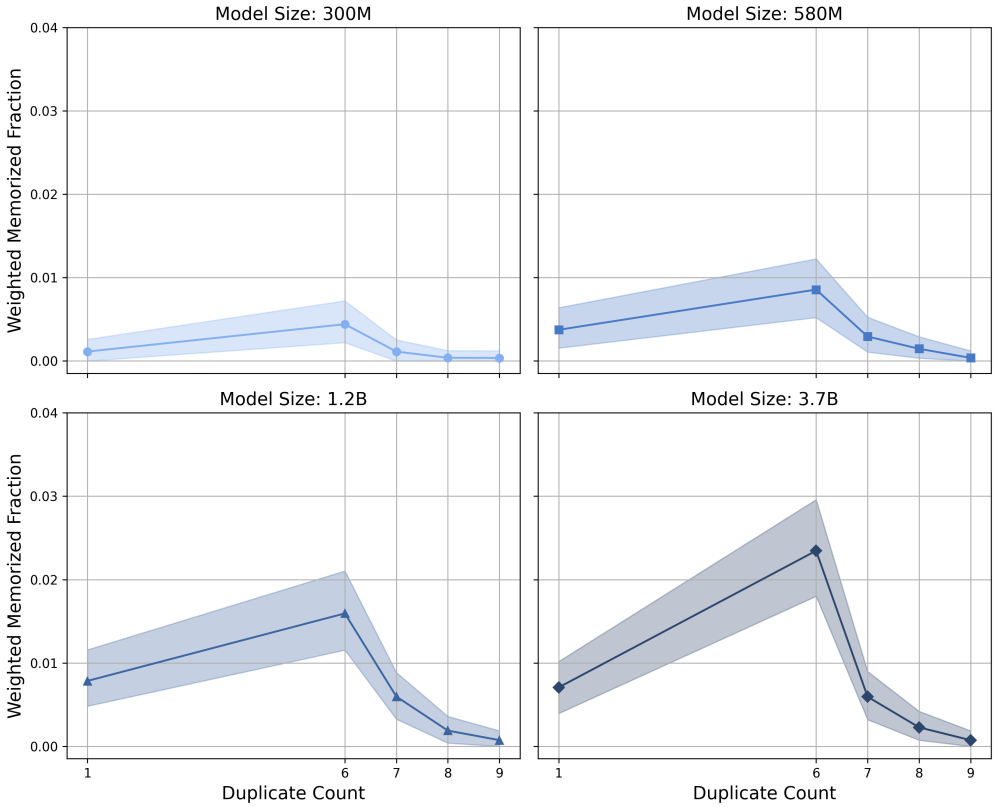


Figure 5.6: Confidence intervals of the weighted memorized fractions as a function of duplication counts for different model sizes (300M, 580M, 1.2B, and 3.7B parameters). The confidence intervals are at the 95% level, illustrating the variation in memorized fractions across specific duplication counts and their respective sample sizes.

Context Length %	**Model Size**	**Fraction Difference**	**One-Sided P-Value**	**Interpretation**
70.0	300M	0.00000	1.000	Not significant
70.0	580M	0.004	0.000	Significant (caution)
70.0	1.2B	-0.002	0.636	Not significant
70.0	3.7B	0.004	0.000	Significant (caution)
75.0	300M	0.001	0.000	Significant (caution)
75.0	580M	-0.001	0.627	Not significant
75.0	1.2B	0.000	0.265	Not significant
75.0	3.7B	-0.003	0.609	Not significant
80.0	300M	0.001	0.000	Significant (caution)
80.0	580M	0.003	0.000	Significant (caution)
80.0	1.2B	0.009	0.000	Significant (caution)
80.0	3.7B	-0.002	0.559	Not significant
85.0	300M	0.001	0.322	Not significant
85.0	580M	0.010	0.014	Significant
85.0	1.2B	0.020	0.003	Significant
85.0	3.7B	0.004	0.312	Not significant

Table 5.1: One-sided bootstrap hypothesis test of pre-training data versus baseline data over multiple context lengths. The p-values represent the probability of observing a memorized fraction difference between the pre-training and baseline data less than or equal to zero for a specific context length and model size. If this probability is small (5% level), this hypothesis is rejected indicating that the pre-trained memorized fraction is significantly greater when compared to the baseline. Note, some fraction differences have significant p-values of zero simply because either the pre-trained or baseline data have a memorized fraction of zero for a specific context length and model size. These should be interpreted with caution.