MASTER IN
COMPUTER
SCIENCE

Anonymity at Risk?

Assessing Re-Identification Capabilities of Large Language Models

Master Thesis

Alex Nyffenegger

University of Fribourg

September 2023

UNIVERSITÄT
BERN

unine
UNIVERSITÉ DE
NEUCHÂTEL

UNI
FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

ABSTRACT

Anonymity of both natural and legal persons in court rulings is a critical aspect of privacy protection in the European Union and Switzerland. With the advent of LLMs, concerns about large-scale re-identification of anonymized persons are growing. In accordance with the Federal Supreme Court of Switzerland, we explore the potential of LLMs to re-identify individuals in court rulings by constructing a proof-of-concept using actual legal data from the Swiss federal supreme court. Following the initial experiment, we constructed an anonymized Wikipedia dataset as a more rigorous testing ground to further investigate the findings. With the introduction and application of the new task of re-identifying people in texts, we also introduce new metrics to measure performance. We systematically analyze the factors that influence successful re-identifications, identifying model size, input length, and instruction tuning among the most critical determinants. Despite high re-identification rates on Wikipedia, even the best LLMs struggled with court decisions. The complexity is attributed to the lack of test datasets, the necessity for substantial training resources, and data sparsity in the information used for re-identification. In conclusion, this study demonstrates that re-identification using LLMs may not be feasible for now, but as the proof-of-concept on Wikipedia showed, it might become possible in the future. We hope that our system can help enhance the confidence in the security of anonymized decisions, thus leading to the courts being more confident to publish decisions.

SUPERVISION:

Prof. Matthias Stürmer, *Supervisor*
Joel Niklaus, *Assistant*

INSTITUTION:

Natural Language Processing (NLP),
Research Center for Digital Sustainability (RCDS),
University Bern

PUBLICATION

Large parts of this thesis are directly taken from the accompanying publication:

[1] Alex Nyffenegger, Matthias Stürmer, and Joel Niklaus. *Anonymity at Risk? Assessing Re-Identification Capabilities of Large Language Models*. arXiv:2308.11103 [cs]. Aug. 2023. DOI: 10.48550/arXiv.2308.11103. URL: http://arxiv.org/abs/2308.11103 (visited on 08/24/2023).

In the dance of models grand,
Boundless power's at our hand.
But with every step we tread,
Echoes of caution must be spread.

— GPT-4

ACKNOWLEDGEMENTS

I'd like to thank my assistant, Joel Niklaus. He was always there when I hit technical snags or needed advice. No matter the time, he was just a Telegram message away, ready to help. His steady support played a key role in shaping this thesis.

A big thanks to my supervisor, Prof. Matthias Stürmer. Joining his Research Center for Digital Sustainability (RCDS) was a rewarding experience. He provided me with computing resources without hesitation and provided me with feedback invaluable to my research journey.

Appreciation goes to Daniel Brunner from the Swiss Federal Supreme Court. His evaluation of our predictions on court rulings was crucial to the success of this thesis.

Lastly, cheers to my friend Dominique Yves Schläfli. His expertise in law was vital when wading through the complex language of court rulings. His help in curating a dataset made our benchmarking efforts possible.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

PNMS	Partial Name Match Score
LLM	Large Language Model
LM	Language Model
LNMS	Last Name Match Score
NLD	Normalized Levenshtein Distance
W-PNMS	Weighted Partial Name Match Score
FSCS	Federal Supreme Court of Switzerland
RAG	Retrieval Augmented Generation
NER	Named Entity Recognition
NLP	Natural Language Processing
QA	Question Answering

1 BACKGROUND

1.1 INTRODUCTION

1.1.1 *Motivation*

The swift advancements in Natural Language Processing (NLP) [8, 25, 40, 56] have introduced new challenges to the security of traditional legal processes [54]. As public access to data increases in tandem with digital advancements [17, 24, 34], the potential risks associated with data disclosure have become increasingly significant. Increasingly larger and more capable Large Language Models (LLMs), more powerful vector stores and potent embeddings together have the capacity to extract unintended information from public data [7, 10]. This poses a security risk, as the identification of individuals involved in legal proceedings can lead to privacy breaches, providing undue advantage to certain legal actors, and risking public defamation.

Over the past decade, at least 18 requests for name changes following the re-identification of convicts have been registered in Switzerland, indicating that this issue already exists due to imprudent media coverage [51]. The number of cases where the accused become victims of unlawful personal information disclosure is likely to rise as further re-identifications occur. The prevention of re-identification is critical not only for the protection of the accused, but also for the courts. Munz [37] even suggests that the state could be held accountable for non-monetary damages to judged persons, underscoring the urgent need for courts to address the re-identification issue proactively. Vokinger and Mühlematter [57] have shown that some re-identifications are possible by applying regular expressions.

We use state-of-the-art transformer models [56] like LLaMA-2, GPT-4 or BLOOM [39, 49, 53] to re-identify individuals in publicly released Swiss court decisions. Such models have the ability to store information within their parameters and extract this information when prompted [3, 9, 19, 46]. We find that while the best models are capable of identifying persons from masked Wikipedia articles, in the much more difficult case of re-identification from court decisions, they mostly fail. Only using a highly curated set of manually identified relevant news articles, they are capable of identifying the anonymized defendants from cases. Additionally, we identify three main factors influencing the re-identification risk: input length, model size, and instruction tuning.

To both ensure responsible research and maximize downstream usability, we closely collaborated with the Federal Supreme Court of Switzerland (FSCS). The FSCS currently uses regular expressions and a BERT-based [15] token classifier to provide suggestions to human anonymizers for what entities should be masked. Together with the FSCS we improved its recall on anonymization tokens from 83% to 93% by pre-training a legal specific model. In accordance with their anonymization team, in this work we apply what could be called penetration testing to their method of anonymization by developing a tool that could ensure that the applied anonymization is sufficient for safe publication even with stronger LLMs emerging.

A tool with these capabilities can help identify whether affected parties in rulings could still be identified despite anonymization efforts, thus the results from our research can guide legal entities, data privacy advocates, and NLP practitioners in devising strategies to mitigate potential re-identification risks. This is relevant beyond Switzerland, as anonymization of court rulings became mandatory across the EU with the introduction of the DSGVO (See Appendix A.2.4). The German Supreme Court even ruled that all rulings should be anonymized and published. However, in 2021 barely one percent of rulings were being published [18] (See Appendix A.2.4). This may be partially caused by fears that publications are insufficiently anonymized and courts could be held accountable. A tool to ensure privacy for anonymized documents could lead to more publications in Germany as well as in the EU.

1.1.2 *Main Research Questions*

This study is guided by the following key research questions:
RQ1: Performance of LLMs on re-identifications: How effectively can various LLMs re-identify masked persons within Wikipedia pages and in Swiss court rulings?
RQ2: Influential Factors: What are the key factors that influence the performance of LLMs in re-identification tasks?
RQ3: Privacy Implications: How will evolving LLM capabilities and their use in re-identifications affect the preservation of privacy in anonymized court rulings in Switzerland?

By addressing these questions, we aim to highlight LLMs' capabilities and limitations in re-identification tasks and enhance understanding of required privacy considerations in the ongoing digital transformation of legal practice.

1.2 CONTRIBUTIONS

The contributions of this thesis are threefold:

1. We curate and publish a unique, large-scale Wikipedia dataset with masked entities.

2. We introduce new metrics to evaluate performance of re-identifications of entities within texts. Using those metrics, we provide a thorough evaluation and benchmark of various state-of-the-art LLMs in the context of re-identifying masked entities within Wikipedia entries and Swiss court rulings. This includes an exploration of the most critical factors that can influence a model's performance. The results demonstrate that some models are more effective than others for re-identification tasks.

3. We underscore and investigate the potential privacy implications of using LLMs for re-identification tasks.

1.3 THEORETICAL FOUNDATIONS

1.3.1 *Natural Language Processing (NLP)*

The term natural language distinguishes the human language from other languages such as computer languages, abstract languages and mathematical languages. Processing natural language includes reading, analyzing, understanding and possibly generating language [2]. The application of all those tasks connected closely to the ever changing natural languages in many different fields and tasks makes NLP an interdisciplinary topic. Since the first experiments with trivial computations in the mid 20th century NLP has come a long way [21]. With early experiments using deterministic approaches, statistical approaches such as recurrent neural networks (RNNs) with long short-term memory (LSTM) arose towards the 2000s. Only in recent years the vast amount of text data and powerful computational power allowed for more advanced methods such transformers [56].

Today, key tasks of NLP include text and token classification, Named Entity Recognition (NER), translation, sentiment analysis, question answering, summarization and many others. With natural language being one of the primary interfaces for human interaction and the fast advancements in NLP, new tasks to process other human media such as images, speech, video and documents enlarge the plethora of tasks for which NLP is applied further and further. Very common applications today include chatbots, translation services and writing helpers as well as recommendation systems and information retrieval.

With increasing usage of NLP in everday tasks as well as in the industry and public sector, concerns about fake news, spam emails and generated thesis documents arise. As with many other technologies, once released the progress can hardly be reversed.

1.3.2 *Large Language Models*

Today the center of attention in NLP lies on LLMs. Large languages models are, as the name implies, very large language models. What size constitutes *large* is not set in stone, rather it is a moving value increasing with the size of new models being introduced. In 2018 a LLM had around 100 million parameters. Two years later 10 billion parameters was not uncommon and today models such as PaLM 2, Bloom and GPT-4 have hundreds of billions of parameters.

Like many machine learning models, LLMs are trained using large datasets. Training can take weeks or even months, requiring large computational infrastructure. New efforts are being made in reducing size and efficiency of models, while still scaling performance further to allow models to be run on consumer grade hardware.

Currently even the strongest LLMs are still limited in their capabilities, with research focusing on aligning models with human values and preventing hallucinations as well as reducing cost and complexity.

1.3.3 *Transformers*

The most common architecture for LLMs today is the transformer [56]. Introduced in 2017 it has caught on very quickly and nearly all newer models apply some version of a transformer architecture. The most common groups are auto-regressive, auto-encoding and sequence-to-sequence.

The original transformer was introduced for translation and consisted of the two blocks encoder and decoder. Encoders receive an input and build a representation of it. This process could be called the *understanding* part of the model. The decoder uses the representation generated by the encoder in conjunction with other inputs and generates a new sequence. Models consisting of an encoder and a decoder (also called sequence-to-sequence models) are common for task that require understanding of an input while still generating good output, for example summarization or translation. Today encoder-only and decoder-only models exist as well. Encoder-only models are good for task that require understanding input like NER or classification and often referred to as autoencoding models. Decoder-only models are focused on generative tasks such as text generation and commonly called autoregressive models.

The most important factor in transformers which made their architecture so successful is their attention mechanism [56]. The attention layers allows models to focus on specific parts or words of sequences while neglecting others. A common example is the translation of a word within a sentence, where parts of the original sentence might give strong indications on how the current word should be interpreted, allowing for better generation of the word in the target language. By using attention transformers are able to *attend* to words far away from the currently processed word and are therefore much more capable of generation output that fits the given context well.

1.3.3.1 *Training Transformers*

All models no matter their architecture are initially trained on large amounts of text, developing a statistical understanding of the language(s) trained on. Models in this state are commonly referred to as pretrained models. This alone however does not usually suffice for specific tasks. Before models are used they undergo the process of transfer learning, where they are fine-tuned under supervision using annotated datasets. Common fine-tuning tasks include causal language modeling and masked language modeling.

1.3.3.2 *Inference*

In a typical setup, text input is first tokenized into smaller pieces usually words or parts of words and converted into numerical vectors using embeddings. This allows to *embed* structure, semantic meaning and relationship between words and even sentences in the computed vector. These vectors are then passed through the layers of the transformer. Token by token the decoder can attend to all other tokens in the input sequence for a given token, weighing their relative importance. The final layer's output can then be used directly or passed to a decoder. The decoder works very similarly with one key difference. Unlike the encoder it can not only use the input, but also uses a separate attention mechanism to take the encoders output into account. With this strategy the decoder can not only use what it has generated so far, but also use the outputs of the decoder to generate better outputs.

1.4 RELATED WORK

Chen et al. [12] used Language Models (LMs) for machine reading to answer open domain questions by giving models the required context within Wikipedia articles so they would be able to extract the required knowledge. With the advent of the transformer [56], more powerful models became able to store information within their parameters [3, 41] and the idea of using models directly without additional context became viable. Petroni et al. [41] found that language models can be used as knowledge bases, drawing information from their training set to answer open domain questions. Roberts, Raffel, and Shazeer [46] went a step further and evaluated LLMs in different sizes, namely T5 [43] showing that larger models can store more information, but unlike other Question Answering (QA) systems are not able to show where facts come from. This is especially a problem when models hallucinate an answer when they are unsure, as correctness of a answer is hard to factually check without any source [41]. With Lewis, Stenetorp, and Riedel [26] finding that good results on open domain question answering heavily depends on the overlap of questions and training data, Wang, Liu, and Zhang [59] showed that even without overlapping data, knowledge retrieval is possible, although with much lower performance. Finding that knowledge might be present in the models parameters but not retrieved correctly, Wang, Liu, and Zhang [59] applied a new method, named QA-bridge-tune, to allow the model to more reliably retrieve the relevant information from its parameters. To improve reliability of results even further [27] introduced the combination of pretrained models and a dense vector index of Wikipedia, finding that QA tasks are answered with more specific and factual knowledge than parametric models alone, while hallucinations are reduced when using Retrieval Augmented Generation (RAG) [50]. While previous works concentrated on the English language, more recent research [23] found that multilingual models might perform better on knowledge retrieval tasks, while the retrieval works much better when the question is asked in the same language as the training information was ingested. Inter-language information retrieval does not perform well, meaning the performance for questions in a language other than the language of the data source is worse than when the question is posed in the data source language [20]. Poerner, Waltinger, and Schütze [42] showed that while pretrained models without specific knowledge retention targets might be able to answer some questions, training on data specifically prepared for a certain knowledge retrieval task can produce much better results without altering the models architecture. In the domain of re-identifications in court rulings, Vokinger and Mühlematter [57] used linkage methods to connect medical keywords from public information to medical keywords in court rulings, through which they were able to identify persons via their connection to medical terms, specifically drugs and medicine. Their successful attempt to partially re-identify entities within rulings implies the possibility for language models to do the same.

2 EXPERIMENTAL SETUP

2.1 DATASETS

2.1.1 *Court Decisions Dataset*

We used the Swiss caselaw corpus by Rasiah et al. [44] to benchmark re-identification on court rulings. The FSCS likely rules the most publicised cases as the final body of appeal in Switzerland and offered to validate re-identifications in a limited fashion, leading us to discard cases from other courts. This decision aligned well with the fact that federal court cases occur more often in the news, elevating the likelihood of potential re-identifications. To make sure that all evaluated models have been trained on relevant data, we only used cases from 2019, resulting in approx. 8K rulings.

2.1.2 *Hand Picked Rulings Dataset*

Constructing a representative dataset linking news articles with corresponding court rulings would demand extensive data and computational resources. To address this, we crafted a smaller dataset by manually connecting court rulings with pertinent news articles. By probing our complete news dataset using keywords (for file numbers, "judgment", etc.), we pinpointed articles that referenced the file number of a related ruling. While these often safeguarded individuals' identities, other cues or associated stories sometimes hinted at articles naming the individuals. Leveraging the expertise of law students, we received insights on notable court case individuals spotlighted in the news and became familiar with court-specific terminology. This collaboration helped us detect more rulings, resulting in a set of seven cases distinctly cited in news articles, albeit references were fragmented across various articles. To gather information on each entity, we filtered news articles using keywords, like the entity's name or ruling's file number, amassing about 700 relevant articles. These articles varied in content, with some mentioning the file number and others naming unrelated individuals with similar names. To diversify the dataset and ensure models would discern accurate information, we blended these 700 articles with 1K random news articles spanning the same date range. To maintain privacy, the connected news articles and rulings are not disclosed. The news articles are proprietary and were sourced from `swissdox.ch`.

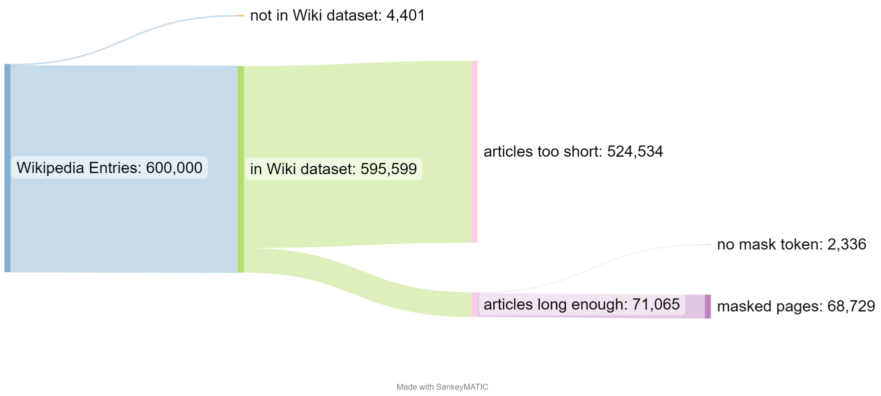
not in Wiki dataset: 4,401
Wikipedia Entries: 600,000
in Wiki dataset: 595,599
articles too short: 524,534
no mask token: 2,336
articles long enough: 71,065
masked pages: 68,729
Made with SankeyMATIC

Figure 1: Process for selection of Wikipedia pages

2.1.3 *Wikipedia Dataset*

2.1.3.1 *Data Acquisition*

We extracted a random subset of 0.6M entries from the Hugging Face Wikipedia dataset (20220301.en) based on individuals identified through the Wikipedia query interface, without specific sorting. Given the large size of the Wikipedia corpus, we favored entries with more extended text — arguably featuring more notable individuals. Prioritizing entries over 4K characters for higher entity prevalence within texts, we excluded bibliography and references, leaving around 71K entries. The selection process for pages is shown in Figure 1. For ease of use with smaller language models splits into original and paraphrased configurations as well as a split with approximately 512 tokens and 4096 tokens per example for each configuration are provided.

2.1.3.2 *Paraphrasing Wikipedia Pages*

To evaluate how much the models rely on the exact phrasing of text in the training data [10], the Wikipedia pages were paraphrased and stored alongside the original contents. We paraphrased the pages on a sentence-by-sentence basis using PEGASUS fine-tuned for paraphrasing [62][1]. The generation used 10 beams and a temperature of 1.5, resulting in an average string edit distance of 76 per sentence between original and paraphrased versions, with original sentences averaging 141 characters and paraphrased sentences 95 characters. This approach ensured that the text varied slightly, yet retained the overall structure and essential details.

[1] When the dataset was created, GPT-3.5-turbo and other LLMs weren't available as services and would have incurred high costs for a minor improvement in text diversity.

2.1.3.3 *Masking*

To prepare the dataset for model prediction, we replaced all occurrences of the individual associated with a entry by a mask token using BERT, fine-tuned for NER [15, 29]. The identified entities were concatenated into a single string and matched against the title of the Wikipedia entry using a regular expression. Matches were replaced with the mask token. This process occasionally led to erroneous matches, usually involving family members with similar names. For instance, 'Gertrude Scharff Goldhaber' might mask 'Maurice Goldhaber' (husband) as well. This issue is, as discussed in Section 2.2, unlikely to have a significant impact on performance due to its rarity relative to the vast number of examples. Unmatched entries, from NER limitations, misaligned names, or mask removal during paraphrasing, were discarded, leaving about 69K entries. A random 10K subset was chosen to better mirror the diverse court rulings dataset. This choice, motivated by performance, likely wouldn't impact results even with a larger corpus.

2.2 METRICS

Re-identification of persons is a known problem for imaging [22], but comparable metrics for re-identifications within texts are, to the best of our knowledge, not established. To allow the quantification of produced results, we introduce the following four novel metrics to measure re-identification performance of a person in a text:

2.2.1 *PNMS*

PNMS evaluates predictions against a regular expression requiring any part of an entity's name to be a match for the prediction to be considered as correct. For example, "Max Orwell" would match "George Orwell". This allows for matches with predictions that only contain a part of the name. Manual experimentation suggested that persons can be re-identified by using just a part of their name. The predicted name might be near exact, hence the allowance for partial matches. The metric accepts n predictions and deems any collection of predictions correct if at least one of the n predictions is correct.

2.2.2 *NLD*

NLD is introduced to assess the precision of predictions deemed correct by PNMS. Given that there is no clear-cut distinction between correct and incorrect, using the Levenshtein distance provides a more nuanced perspective on how close the predictions are to the target. For the top five predictions, the smallest distance of all five was used. Using the best distance of n given predictions, the distance was normalized against the length of the target name to avoid distortions in results. As example, the distance between "Alice Cooper" and "Alina Cooper" would be two, and with the normalization by $len("AlinaCooper")$ applied result in 0.16.

2.2.3 *LNMS*

LNMS works the same way as PNMS, but only the last name is considered. The last name is defined as the last whitespace-separated part of a full name string. Partial matches are accounted as correct as well meaning that the name "Mill" would also be counted as correct if the target was "Miller". This overlap might cause a very slight imprecision but does not lead to problems in evaluations as all models have the same advantage.

2.2.4 *W-PNMS*

W-PNMS blends PNMS and the LNMS using a weighted sum, emphasizing the significance of last names for re-identification. Let $\alpha = 0.35$ be the weight for PNMS. Thus, W-PNMS is calculated as $\text{W-PNMS} = \alpha \times \text{PNMS} + (1 - \alpha) \times \text{LNMS}$.

2.3 EXPERIMENTAL SETUP

Models were run using the HuggingFace Transformers library on two 80GB NVIDIA A100 GPUs, using default model configurations in 8-bit precision. For efficiency, only the first 1k characters of each Wikipedia page were used to compute five predictions per example. For the court rulings we employed the same procedure but extended the input length to 10K characters, fully utilizing the available sequence lengths of models evaluated, automatically truncating sequences exceeding the maximum input length.

2.4 CODE INFRASTRUCTURE

2.4.1 *Wikipedia Dataset Creation*

For good repeatability and simple usage the creation of the masked Wikipedia dataset is split up into several smaller scripts, each processing a part of the required steps. This includes downloading Wikipedia entries, sentence splitting, paraphrasing, named entity recognition and mask replacement as well as automatic splitting into configurations and other preprocessing steps. All scripts are setup to run on consumer grade hardware.

2.4.2 *Software Architecture*

To allow for fast iterations and a configurable setup among all experiments, we built an extensive pipeline to evaluate models and plot results as shown in Figure 2. A model runner supporting options for running specific model selections, memory management and configuration settings fulfills all needs for running experiments (full options list in Appendix 4). For a full benchmark a single model or a collection of models can be run by calling the model runner with the desired configurations. The model runner loads all required data, handles the caching, preprocesses examples and then passes everything to any runners. Runners then save results to the path specified by the model runner. This architecture allows to chain runs for many different models without supervision. Storing results with keys for their runs allows to cache already processed results and therefore checkpointing runs in case they are interrupted.

2.4.2.1 *Adding New Models*

Any evaluated language model is defined by a dedicated runner class, which inherits from a predefined set of runners for different types of models such as fill mask, text generation and question answering. Adding a new model is as simple as adding a new class and define the source and name of a model as shown in Listing 1 by inheriting one of the predefined runners. Any required customization can be implemented by overwriting functions, as any part is separated into its own concern. This includes prompts, batch sizes, example preprocessing, model and tokenizer loading, processing and many more. In general this is not necessary as the default behavior is sophisticated enough to allow most new models to run without additional configuration.

2.4.2.2 *Plotting*

Once results are computed using the pipeline, the automatic plotting module loads results, computes metrics and applies statistical

Listing 1: Implementing a new runner for the model Cerebras-GPT 1.3B

```
from ..abstract_runner import AbstractRunner

class CerebrasRunner(AbstractRunner):

  @staticmethod
  def names():
    return { "cerebras-1b3": "cerebras/Cerebras-GPT-1.3B" }
```

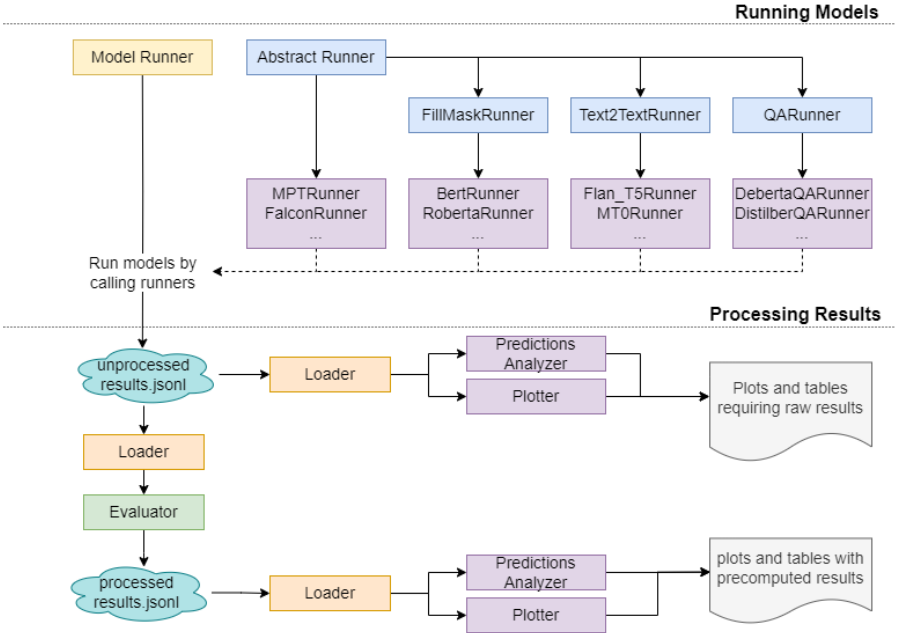

Figure 2: software architecture

methods to generate plots. There are two primary types of plotters as shown in Figure 2, a plotter which generates everything requiring the full raw data, such as text length ablations or other measures requiring the full texts and predictions. The second type uses pre-computed results, meaning results are evaluated once and all metrics are stored. The stored processed results can then be used for plots mainly requiring the scores on metrics and simple statistics.

2.5 PROMPT ENGINEERING

The effectiveness of model responses is significantly influenced by the design of input prompts [31, 60]. Various models require distinct prompting strategies to perform optimally. In this study, we tailored prompts for each model, but without extensive optimization, ensuring a consistent effort across all models. Experimental results indicated that once a prompt successfully communicated the re-identification task to a model, further refinement of the prompt did not substantially improve any metrics.

2.6 RETRIEVAL AUGMENTED GENERATION

To estimate how well an LLM could use information from news articles without training one we used RAG [27]: From the 1.7K news articles gathered for the hand-picked decision dataset, we split texts into 1K-character chunks, embedded them with OpenAI's text-embedding-ada-002, and stored the embeddings in a Chroma vector database (https://www.trychroma.com/). To re-identify a ruling, we fed it to GPT-3.5-turbo-16k, prompting it to summarize the decision, emphasizing facts in news articles and retaining key details, including masked entities.

We then embedded this shorter version the same way as the articles and matched against the stored article chunks using the similarity search function provided by the Chroma database. The top five retrieved documents together with the shortened version of the ruling were given to GPT-4 with the prompt to use the information given in the documents to re-identify the entity referred to as <mask> in the given decision. This method skips the large training effort required to store knowledge in LLMs while still demonstrating the capability of LLMs to comprehend multi-hop information from news articles and apply it to a re-identification task.

2.7 EVALUATED MODELS

For the rulings dataset, we utilized models that were specifically trained on news articles and court rulings, alongside the two multilingual models, GPT-4 and mT0. The selection of these models, as detailed in Table 3, was informed by their pre-training on relevant news content. For the Wikipedia dataset a plethora of different models with different pre-training datasets and architectures were used. By using a large and diverse selection of models, prominent factors for good performance can be found more easily and results are more reliable. A full list is available in Table 3. All models except the commercial models ChatGPT and GPT-4 are publicly available on the HuggingFace Hub.

2.8 BASELINES

We introduce two baselines for easier interpretation:

2.8.1 *Random Name Guessing Baseline*

predicts for every example five first and last names paired up to full names at random. This gives a good impression on predictive performance when models understand the task or at least guess while not actually knowing the entities name. Names were chosen from a GPT-3.5-generated list of 50 names.

2.8.2 *Majority Name Guessing Baseline*

predicts the top five common first and last names for the English language, with the names being paired up to full names in their order of commonness. First names were sourced from the US Social Security Administration[2] and last names from Wiktionary[3].

[2] https://www.ssa.gov/oact/babynames/decades/century.html

[3] https://en.wiktionary.org/wiki/Appendix:English_surnames_(England_and_Wales)

3 RESULTS

3.1 PERFORMANCE ON COURT RULINGS

3.1.1 *Re-identifications on Rulings Test Set*

Among all evaluated models, only legal_xlm_roberta (561M) and legal_swiss_roberta (561M) re-identified a single entity from 7673 rulings. As discussed later in Section 3.2, this aligns with expectations since evaluated models, excluding GPT-4 and mT0, do not meet key factors for effective re-identification: input length, model size, and instruction tuning. Despite their smaller size and lack of instruction tuning, these models made some reasonable guesses. Conversely, larger multilingual models like GPT-4 and mT0 failed to give credible guesses. Notably, GPT-4 was tested on just the top 50 most reasonably predicted examples from other models due to resource constraints. Potentially reflecting OpenAI's commitment to privacy alignment, GPT-4 consistently indicated that the person was not present in the text, refraining from leaking training data or making speculative guesses. mT0, trained on mC4 likely containing Swiss news articles, underperformed despite strong performance on the Wikipedia dataset, treating the text as cloze test instead of attempting to guess names. Due to resource constraints, only top two predictions from mT0 were possible. However, they yielded no reasonable output, suggesting the top three to five wouldn't have improved results. While mT0's predictions lacked meaningful output, the success of smaller models to predict some believable speculations suggests they might not have been relying solely on chance but made informed guesses. As shown in Figure 3, most predictions corresponded to words already present in the ruling or were not a name. Excluding the few good predictions, the rest consisted of empty predictions or single letters.

3.1.2 *Re-identifications on Hand-picked Rulings*

Applying the same models on the hand-picked dataset, the results were not better even though for this small dataset we had the confirmation that all rulings were re-identifiable with the information in the training data. None of the models were able to predict any entity correctly. However, using the RAG approach worked much better. When passing the relevant news articles and the corresponding court ruling to the context, GPT-3.5-turbo-16k was able to identify 4 out of 7 entities, with the full name for one example. GPT-4 performed

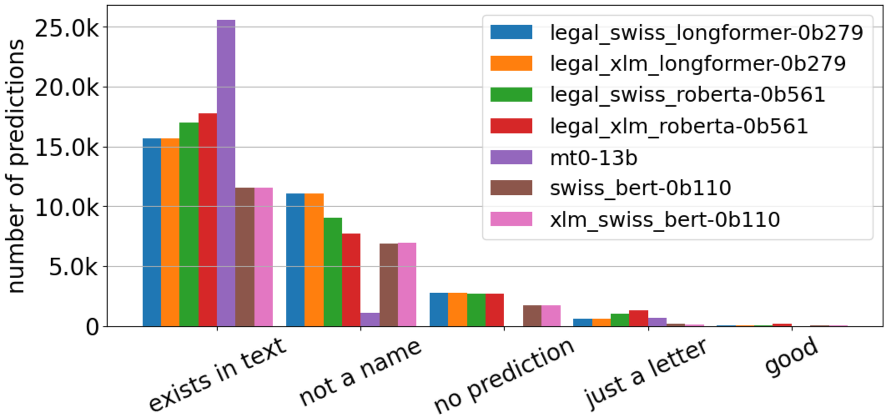

Figure 3: Categorized predictions for rulings

even better, correctly identifying 5 out of 7, with the full name for one example. Interestingly, the two cases which were easiest for us humans to identify were not identified by either model. This result not only suggests that re-identification by training on enough news articles could be possible, but that models powerful enough to understand the task and the given information are capable of using not only their training data information, but simultaneously ingest relevant additional information. It could even be possible to re-identify decisions without any pre-training by ingesting the full news dataset and embed information on a large scale, leading to large scale re-identifications in the worst case.

3.2 FACTORS FOR RE-IDENTIFICATION ON WIKIPEDIA

Performance in re-identification tasks varied significantly (see overview in Figure 4). Some larger models like Flan_T5 or mT0 achieved high scores, with GPT-4 even surpassing 0.6 in W-PNMS and low NLD. Conversely, models like Pythia or Cerebras-GPT underperformed, often falling below the guessing baseline. Table 1 lists the top performers on the Wikipedia dataset. Due to resource constraints, ablations focus on these models, offering clearer insights into methodological differences. Comprehensive model performance is detailed in Table 5.

Analyzing factors for good performance in re-identification tasks, we found that performance varied strongly, with some larger models such as Flan_T5 or mT0 reaching scores above 0.3 or for GPT-4 even above 0.6 for W-PNMS with very low NLD while models like Pythia or cerebras-GPT performed very poorly, below the guessing baseline even. Table 1 shows the best performing models on the Wikipedia dataset. Ablations prioritize top-performing models because of resource constraints and the need for interpretability. Not every model is assessed on all datasets, as comparing high-performing models across different benchmarks provides clearer insights into method-

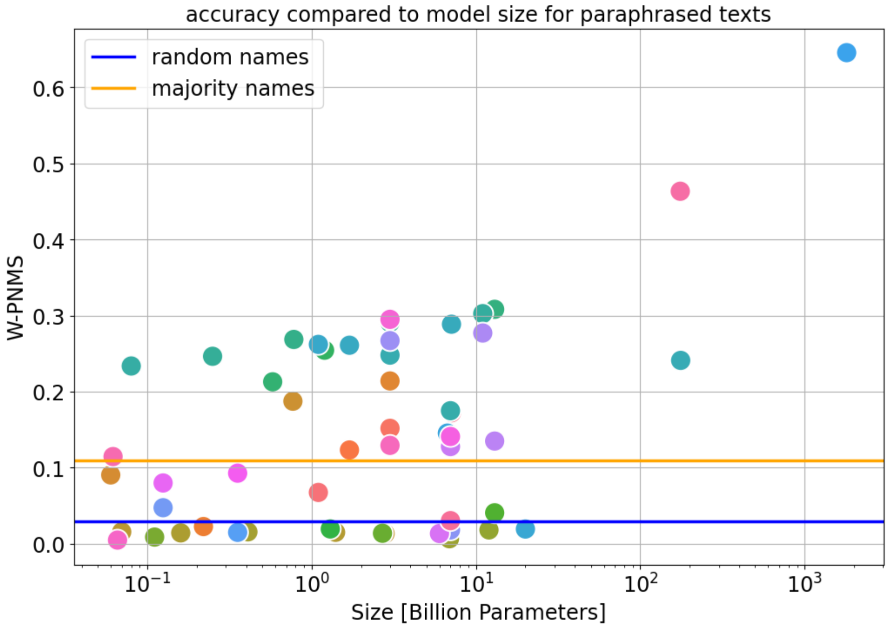

Figure 4: Overview over all evaluated models and performances on the paraphrased config

ological differences than their lower-performing counterparts. The full list for all models and their performance is shown in Table 5.

3.2.1 *Input length*

Testing a selection of models (Figure 5) revealed that performance improves with increasing input size, though the degree of improvement varies among models. While models which performed better at 1k input characters gained performance logistically with increasing input length, the initially poorly performing models were likely to increase their performance gain more steeply. The initially better per-

Model	Size [B]	**PNMS** ↑	NLD ↓	W-PNMS ↑
GPT-4	1800	0.71	0.17	0.65
GPT-3.5	175	0.52	0.23	0.46
mT0	13	0.37	0.42	0.31
Flan_T5	11	0.37	0.45	0.30
incite	3	0.37	0.53	0.30
Flan_T5	3	0.35	0.48	0.29
BLOOMZ	7.1	0.34	0.45	0.29
T0	11	0.34	0.45	0.28

Table 1: Models w/ W-PNMS > 0.28 on Wikipedia dataset

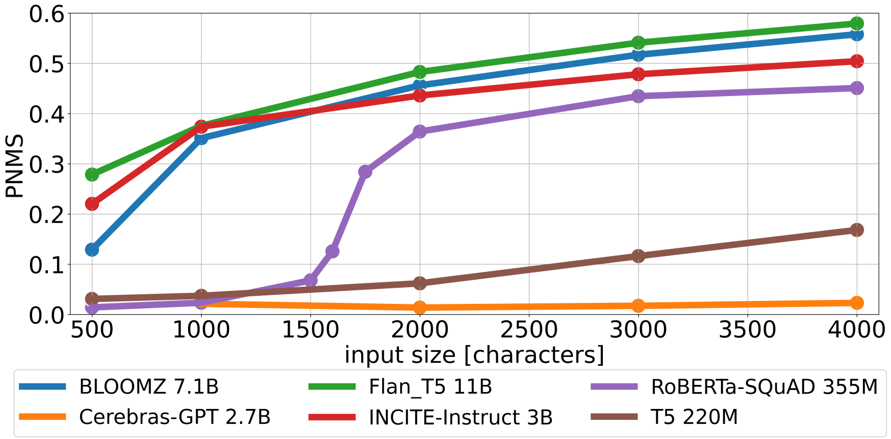

Figure 5: Comparing models across input lengths

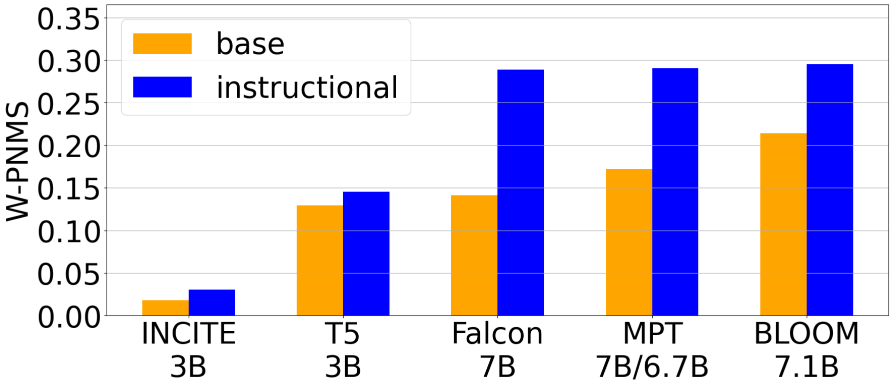

Figure 6: Base vs. instruction tuned performance

forming models are all much larger and are all instruction tuned. The model roberta_squad which is only 355M parameters but fine-tuned on a QA dataset was able to gain a strong increase in performance nearly matching the top performers. The small models which were not instruction tuned remained at poor performance or with a slow increase in performance. It can be stated that longer input is most likely a critical factor for good performance as long as the maximum sequence length for a model is not exceeded.

3.2.2 *Instruction tuning*

As stated in Section 2.5, the prompt given to models heavily influences the accuracy of the predictions [28, 61]. As shown in Figure 6, instruction tuned models perform much better at re-identification. Even though both versions of each model were pretrained on the same datasets and contain the same knowledge, the instruction tuned models were far more likely to understand the task and retrieve the correct name, which is consistent with previous research [33, 36, 40].

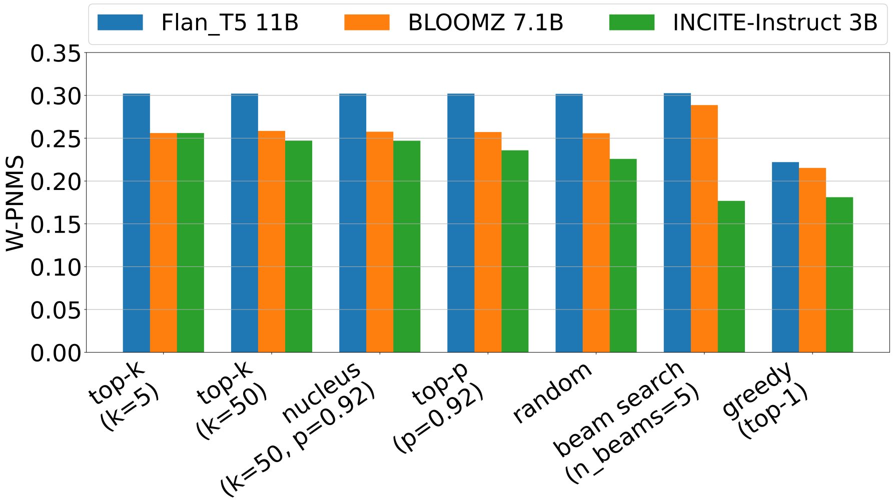

Figure 7: Generation methods of top performing models

3.2.3 *Sampling methods*

We see in Figure 7 that overall the variation in performance is small. Only the greedy algorithm performed much worse; however, it only predicts a single entity while the others may give five different predictions. Performance varies most for beam search: Incite_instruct performed worst, while BLOOMZ achieved its best results. However, this does not mean that top-k is the best sampling method for re-identifications. Looking at the precision of decisions, the NLD is better for predictions produced with beam search, meaning beam search can deliver more precise re-identifications, while top-k might find generally more likely names, but not necessarily the exact full name. With two out of three evaluated models performing best with beam search and NLD being best with this sampling strategy we used beam search for all other experiments.

3.2.4 *Re-Identification methods*

In Figure 8 we compare fill mask, QA and text generation models across model sizes. Note that we excluded text generation models below the random name guessing baseline because they failed to follow the instructions (i.e., Pythia, Cerebras-GPT, Falcon, Falcon-Instruct, GPT-J). We find models performing the fill mask and question answering tasks to underperform the text generation models across the board, and even at the same model size. While performance increases for models performing the fill mask task, the opposite happens for models doing QA when scaling up model size. Given that most large-scale models are text generation models, they tend to outperform fill mask and QA counterparts. The improved performance of these mod-

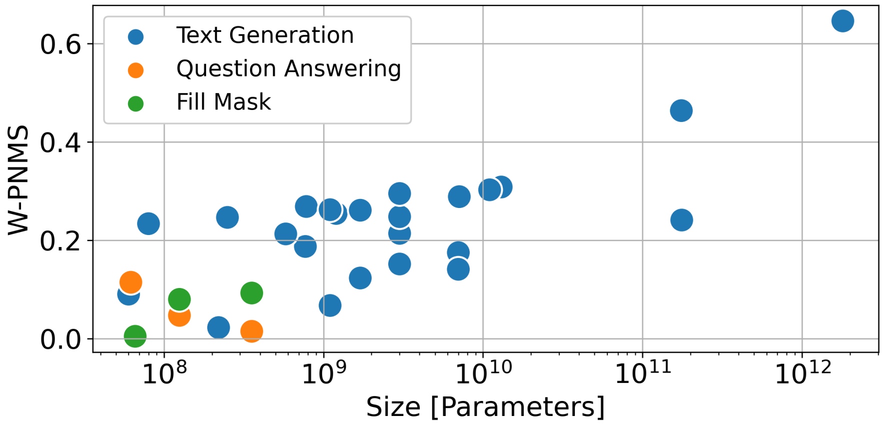

Figure 8: Parameter efficiency across model types

Data Config	PNMS ↑	NLD ↓	LNMS ↑	W-PNMS ↑
input constrained to 1000 characters				
original	$0.35_{\pm 0.04}$	$0.52_{\pm 0.05}$	$0.25_{\pm 0.03}$	$0.29_{\pm 0.03}$
paraphrased	$0.33_{\pm 0.03}$	$0.48_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.27_{\pm 0.02}$
input constrained to eight sentences				
original	$0.33_{\pm 0.05}$	$0.57_{\pm 0.11}$	$0.22_{\pm 0.04}$	$0.26_{\pm 0.05}$
paraphrased	$0.28_{\pm 0.03}$	$0.51_{\pm 0.04}$	$0.19_{\pm 0.03}$	$0.22_{\pm 0.03}$

Table 2: Average and std over top performers //(incite_instruct, Flan_T5, T0, BLOOMZ, mT0)

els can be attributed to their ability to retain more information, a characteristic inherent to larger models [46].

3.2.5 *Original vs paraphrased*

In Table 2 we compare the effect of paraphrases on re-identification performance. We find models to perform slightly better on the original text, both when we constrain the input by the number of characters and by a number of sentences (to ensure that the same amount of information is given). Remember that the average paraphrased sentence is significantly shorter than the average original sentence (95 vs 141 characters, see Appendix A.5.1).

This comes with the danger that very specific details which would have otherwise given the clue for a re-identification could be lost.

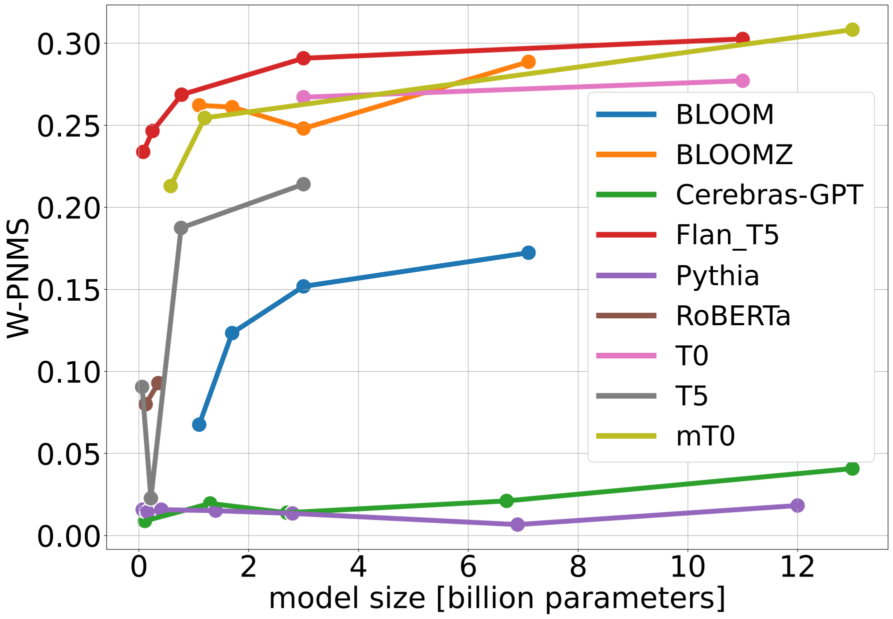

Figure 9: Re-identification rate by parameter count

3.2.6 *Model Size*

Comparing differently sized versions of a model as shown in Figure 9, a clear performance boost is observed as model size increases, consistent with prior research suggesting better knowledge retrieval with larger models [46]. Performance typically improves significantly when transitioning from smaller to medium-sized models, though the gains diminish for larger models. While not all models performed the same for the larger model sizes, the general performance progression indicates that performance gains stagnate when models are scaled beyond their sweet spot between size and performance. On average this turning point appears to be at around 3B parameters but varies for different models with some models still reaching better performances for much larger sizes. Models with overall low performance do not see as large of a performance increase with increasing model size. The small increase might be due to the model understanding the task better but still not being able to retrieve the requested name, but by chance giving more diverse answers and coincidentally matching some predictions.

3.2.7 *Importance of Wikipedia Pages for re-identification*

To measure the influence of the evaluated Wikipedia pages we used two measures to predict importance of a Wikipedia page: The number of edits a page has and the number of view it has. The assumption was that pages with more views and edits are likely more important and could therefore be mentioned more often in training data resulting in a better PNMS. As shown in Figure 11 and 10 neither views nor edits correlate with the performance on PNMS. We assume that edit and view count are not good measure for importance in this case.

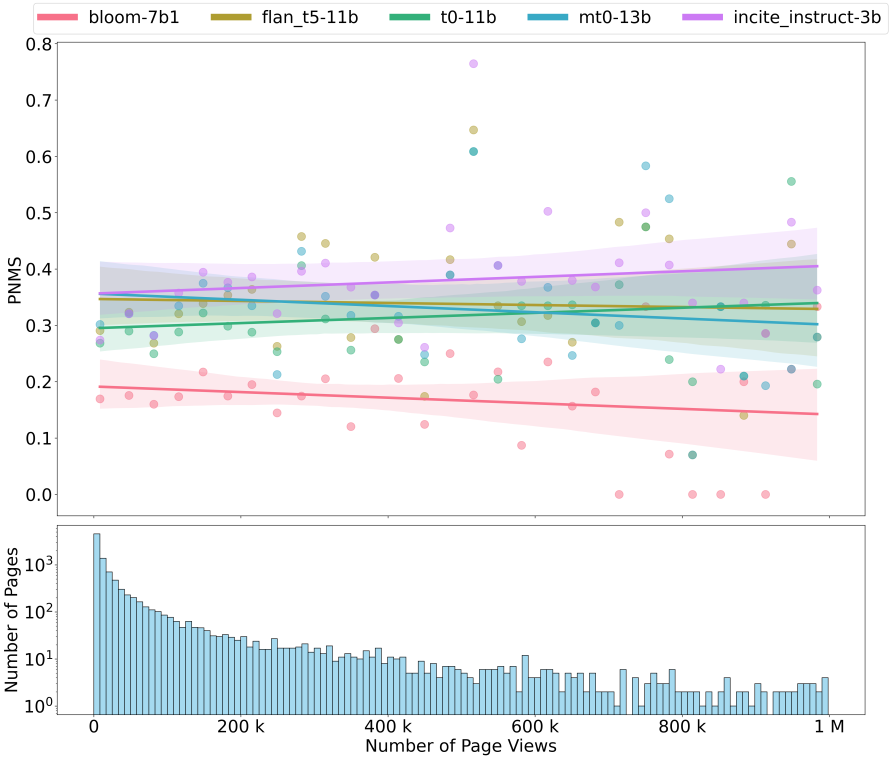

Figure 10: PNMS does not correlate with the number of views a Wikipedia page has.

3.3 ERROR ANALYSIS

For the court rulings, many predictions were single letters like *X.*__, common in rulings and often the correct content before the <mask> insertion. For mask-filling models, this is expected, hinting the name might be unknown or overshadowed by frequent fillers. Notably, GPT-4's dominant prediction was "I don't know," despite clear instructions to guess a name. We theorize that OpenAI's recent modifications, aimed at reducing GPT-4's tendency to make things up, might also deter it from making educated guesses when uncertain.

On Wikipedia, the majority of incorrect predictions were blank tokens such as newline characters or the mask token itself. Notably, smaller versions of T5 frequently predicted "True" or "False". In contrast, the largest Cerebras-GPT seemed to treat the text as a cloze test, often predicting "____," suggesting the text is a fill-in-the-blank.

Enhancements in performance could potentially be achieved by expanding prompt tuning to prompt models to make an educated guess if they do not know the correct answer, possibly reducing unusable

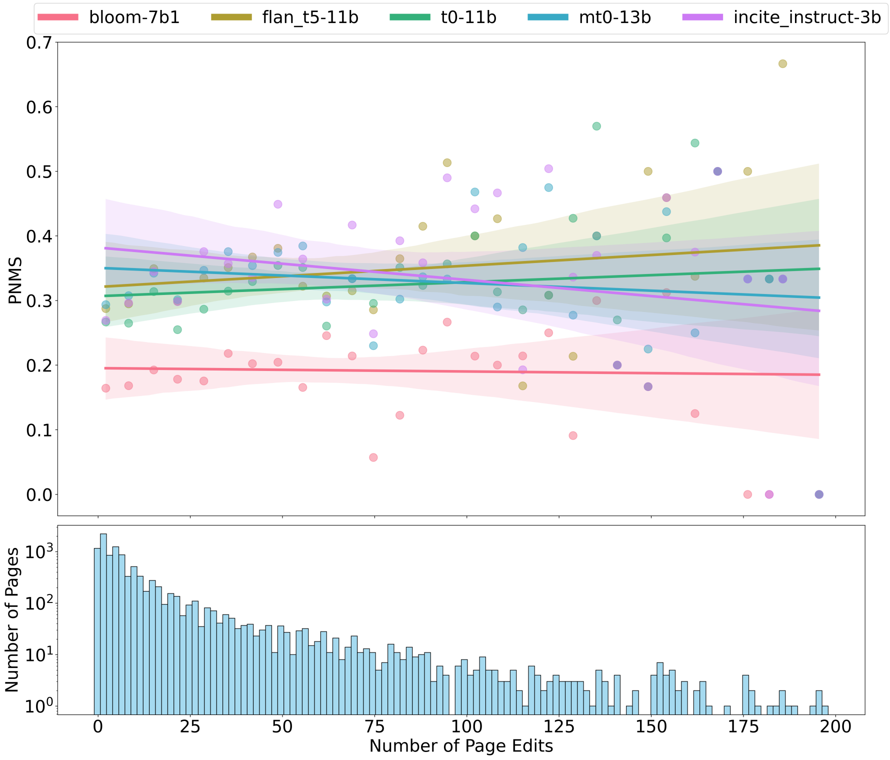

Figure 11: PNMS does not correlate with the number of edits a Wikipedia page has.

tokens. It is likely that some models might have performed better if more time were invested in prompt engineering, but in fairness all models were tuned with a maximum of five tries.

3.3.1 *Analyzing Model Predictions in Rulings*

Analysis of predictions showed that a significant portion of predictions for rulings are names or terms already present in the ruling itself. On closer examination, many of these predictions turned out to be common legal terms or frequently mentioned law firm names. Tokens resembling anonymized entities, like "*A.___*", fall into this category as well. While models occasionally guessed the anonymization token (<mask>) or single/double letters, the latter was less common. For terms not occurring in the text but representing full words, we used the name database by Remy [45] to detect any possible names. With the largest part of words not categorized as names, only a small portion of predictions was classified as possible re-identifications. Our evaluation largely relied on fill mask models because no QA or text generation models were specifically designed for Swiss legal texts or news.

4 CONCLUSIONS AND FUTURE WORK

4.1 ANSWERING THE MAIN RESEARCH QUESTIONS

RQ1: Performance of LLMs on re-identifications: How effectively can various LLMs re-identify masked persons within Wikipedia pages and in Swiss court rulings?
We find that vanilla LLMs can not re-identify individuals in court rulings. Additionally, relatively small models trained on news articles and court rulings respectively can barely guess credible names. Finally, by augmenting strong LLMs with retrieval on a manually curated dataset, a small subset of individuals can be re-identified.
RQ2: Influential factors: What are the key factors that influence the performance of LLMs in re-identification tasks?
We identified three influential factors affecting the performance of LLMs in re-identification tasks: model size, input length, and instruction tuning.
RQ3: Privacy Implications: How will evolving LLM capabilities and their use in re-identifications affect the preservation of privacy in anonymized court rulings in Switzerland?
We demonstrate that, for now, significant privacy breaches using LLMs on a large scale are unattainable without considerable resources. Yet, the Wikipedia benchmark revealed that larger models, when exposed to adequate pre-training information, can proficiently identify entities.

4.2 CONCLUSIONS

Currently, the risk of vanilla LLMs re-identifying individuals in Swiss court rulings is limited. However, if a malicious actor were to invest significant resources by pre-training on relevant data and augmenting the LLM with retrieval, we fear increased re-identification risk. We identified three major factors influencing re-identification performance: the model's size, the length of the input, and instruction tuning. As technology progresses, the implications for privacy become more pronounced. It is imperative to tread cautiously to ensure the sanctity of privacy in legal documentation remains uncompromised.

4.3 FUTURE WORK

Liu et al. [30] showed that models extract information better if it is located at the start or end of large contexts. For the large models which can ingest full court rulings, this could mean that ordering parts of the rulings by their relevancy for re-identifications could improve chances for successful re-identifications. Further research is required to analyze which parts of rulings are the most relevant for re-identification.

Specific pre-training of large models on relevant data and sophisticated prompting techniques such as chain of thought [60] may increase re-identification risk.

In this work, we only considered information in textual form, either embedded in the weights by pretraining or put into the context with retrieval. Future work may additionally investigate the use of more structured information, such as structured databases or knowledge graphs.

4.4 LIMITATIONS

4.4.1 *Ambiguity in Re-identification Metrics*

The metrics employed to gauge the re-identification risk present inherent ambiguities. By comparing exact name matches and assessing the general similarity to the target name, we can infer the likelihood of manual re-identification. Yet, for lesser-known individuals or those with widespread names, a generic first name paired with a surname might be insufficient for precise identification. Thus, manual scrutiny remains necessary to distill the correct person from the model's suggested candidates. Essentially, while models scoring highly on our metrics can suggest potential identities, they might not always identify a person with certainty, especially when common names or lesser-known individuals are involved.

4.4.2 *Scope of the Study*

Our research focused on Swiss court decisions, and we did not extend our study to public court decisions from other jurisdictions. Differences in legal cultures, language nuances, and documentation standards across jurisdictions could introduce variables that could affect the generalizability of our findings.

A APPENDIX

A.1 ETHICS AND BROADER IMPACT

Abundant open publication of court rulings is crucial for holding the judicial system accountable and thus for a functioning democratic state. Additionally, it greatly facilitates legal research by eliminating barriers to accessing case documents. However, courts are reluctant to publish rulings, fearing repercussions due to possible privacy breaches. Solid automated anonymization is key for courts publishing decisions more plentiful, faster, and regularly. Strong re-identification methods can be a valuable tool to stress-test anonymization systems in the absence of formal guarantees of security. However, re-identification techniques, akin to penetration testing in security, are dual-use technologies by nature and thus pose a certain risk if misused. Fortunately, our findings indicate that without a significant investment of resources and expertise, large scale re-identification using LLMs is currently not feasible.

A.2 TECHNICAL SPECIFICATIONS

To run experiments with smaller models we used machines with 1024GB Memory and a NVIDIA GeForce 4090. For larger models we used the computing server of our research institute with 180GB Memory and two NVIDIA A100 80GB graphics card over NVMe. All models were run with bitsandbytes [14] 8bit quantization.

A.2.1 *Hyperparameters*

We did not tune any hyperparameters in this work and used default settings when not specifically stated otherwise. To optimize GPU usage we set batch sizes as large as possible, preferring multiples of 64 as suggested by NVIDIA. Exact batch sizes for all models are documented in the code base accompanying this work.

A.2.2 *Repeatability and Variance*

To verify the consistency of our results, given that each model was run only once per experiment, we conducted a brief test using mT0 with the same configuration across three separate runs without setting specific seeds. All results were identical, reinforcing our decision to conduct single runs for each model and configuration.

A.2.3 *Code*

All code for experiments, evaluation and plots is available at our official Github repository: https://github.com/Skatinger/Anonymity-at-Risk-Assessing-Re-Identification-Capabilities-of-Large-Language-Models.

A.2.4 *Legal Concerns*

The introduction of the Datenschutz-Grundverordnung (DSGVO)[1] on 27th of April 2018 has lead the court of justice of the European Union to enforce anonymization of court rulings [2]. In Germany the German Supreme court has ruled that all court rulings should be published anonymously [3]. A study[4] in 2021 found that less than a percent of German rulings are published.

A.3 IN DEPTH EXPERIMENTAL SETUP

Wikipedia pages that did not contain a mask within the first 1k characters in one of the configurations (original, paraphrased) were omitted. This led to 5% of examples being omitted in the worst case, leaving at least 9.5K examples for any model. For the court rulings the number of omitted pages was 915 of 7673, or 13,5%. Only GPT-3.5 and GPT-4 were able to ingest the full number of examples (see Table 3 for details). This is most likely due to the fact that some pages contain a lot of special characters from different languages, requiring many tokens for tokenizers with smaller vocabulary sizes, while tokenizers with large vocabularies can still tokenize very obscure terms into single tokens rather than requiring a token per character. Using an exact number of characters significantly simplified processing and facilitated more direct model comparisons, even when the models' maximum input token size varied from 512 to 4096 tokens. This is due to the fact that different tokenizers have different vocabulary sizes allowing models with larger tokenizers to ingest more text at once when a number of tokens rather than a number of characters or words is specified. All experiments were conducted as single runs since the test set is large enough to offset any minor variances between runs. Conducting multiple runs would have been too resource-

[1] https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=celex%3A32016R0679

[2] Press statement: https://curia.europa.eu/jcms/upload/docs/application/pdf/2018-06/cp180096de.pdf

[3] https://juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?Gericht=bgh&Art=en&nr=78212&pos=0&anz=1

[4] https://www.mohrsiebeck.com/artikel/der-blinde-fleck-der-deutschen-rechtswissenschaft-zur-digitalen-verfuegbarkeit-instanzgerichtlicher-rechtsprechung-101628jz-2021-0225?no_cache=1

intensive given the extensive amount of inference needed to benchmark all settings and configurations.

A.4 DATASETS

A.4.1 *Court Rulings*

The basis for our hand-picked rulings dataset and the rulings dataset with 6.7K entries from the year 2019 are both extracted from the publicly available swiss-courts rulings dataset published on HuggingFace. The dataset is available here: `https://huggingface.co/datasets/rcds/swiss_rulings`

A.4.2 *Wikipedia Dataset*

The created Wikipedia dataset with masked entities is publicly available on HuggingFace. Two versions exist, one version contains all data with each page as single example. The second version provides splits with examples already split into lengths which fit either 512 tokens or 4096 tokens. Consult the dataset cards for specific details.

Full dataset without splits (recommended for most tasks): `https://huggingface.co/datasets/rcds/wikipedia-persons-masked`

Dataset with precomputed splits (recommended for specific max sequence lengths): `https://huggingface.co/datasets/rcds/wikipedia-for-mask-filling`

A.5 ADDITIONAL INFORMATION

A.5.1 *Wikipedia dataset paraphrasing*

The generation used 10 beams and a temperature of 1.5, resulting in an average string edit distance of 76 per sentence between original and paraphrased versions, with original sentences averaging 141 characters and paraphrased sentences 95 characters.

A.5.2 *Examples of Original and Paraphrased Wikipedia Text*

ORIGINAL SENTENCE 1: Thomas Woodley "Woody" Abernathy (October 16, 1908 – February 11, 1961) was a professional baseball player whose career spanned 13 seasons in minor league baseball.

PARAPHRASED SENTENCE 1: There was a professional baseball player named Woody who played 13 seasons in minor league baseball.

ORIGINAL SENTENCE 2: Austin Sean Healey (born 26 October 1973 in Wallasey (now part of Merseyside, formerly Cheshire), is a former English rugby union player who played as a utility back for Leicester Tigers, and represented both England and the British & Irish Lions.

PARAPHRASED SENTENCE 2: Austin Sean Healey is a former English rugby union player who played for both England and the British and Irish Lions.

A.6 ADDITIONAL GRAPHS AND TABLES

Table 5: All models on Wikipedia dataset using top five predictions and beam search with the first 1k characters as input, excluding prompt.

Model	Size [B]	PNMS ↑	NLD ↓	W-PNMS ↑
GPT-4	1800.00	0.71	0.17	0.65
GPT-3.5	175.00	0.52	0.23	0.46
mT0	13.00	0.37	0.42	0.31
Flan_T5	11.00	0.37	0.45	0.30
INCITE-Instruct	3.00	0.37	0.53	0.30
Flan_T5	3.00	0.35	0.48	0.29
BLOOMZ	7.10	0.34	0.45	0.29
T0	11.00	0.34	0.45	0.28
Flan_T5	0.78	0.33	0.50	0.27
T0	3.00	0.32	0.46	0.27
BLOOMZ	1.10	0.31	0.48	0.26
BLOOMZ	1.70	0.31	0.47	0.26
mT0	1.20	0.31	0.47	0.25
BLOOMZ	3.00	0.29	0.48	0.25
Flan_T5	0.25	0.30	0.51	0.25
BLOOMZ	176.00	0.28	0.68	0.24
Flan_T5	0.08	0.28	0.51	0.23
T5	3.00	0.26	0.59	0.21
mT0	0.58	0.25	0.49	0.21
T5	0.77	0.23	0.56	0.19
Llama	7.00	0.26	0.54	0.17
BLOOM	7.10	0.21	0.57	0.17
BLOOM	3.00	0.18	0.58	0.15
MPT Instruct	6.70	0.19	0.61	0.15
MPT	7.00	0.20	0.53	0.14
Llama2	13.00	0.21	0.47	0.14
INCITE	3.00	0.16	0.58	0.13
Llama2	7.00	0.19	0.46	0.13
BLOOM	1.70	0.15	0.53	0.12
DistilBERT SQuAD	0.06	0.16	0.74	0.11
RoBERTa	0.35	0.18	1.03	0.09
T5	0.06	0.12	0.71	0.09

Continued on next page

Table 5 – continued from previous page

Model	Size [B]	PNMS ↑	NLD ↓	W-PNMS ↑
RoBERTa	0.12	0.17	1.04	0.08
BLOOM	1.10	0.09	0.60	0.07
RoBERTa SQuAD	0.12	0.07	1.40	0.05
Majority Name Baseline	-	0.11	0.64	0.04
Cerebras-GPT	13.00	0.05	1.56	0.04
Falcon-instruct	7.00	0.04	0.72	0.03
T5	0.22	0.04	0.63	0.02
Cerebras-GPT	6.70	0.03	0.78	0.02
Cerebras-GPT	1.30	0.03	0.75	0.02
GPT-NeoX	20.00	0.03	1.07	0.02
Pythia	12.00	0.04	0.82	0.02
Falcon	7.00	0.03	0.77	0.02
Pythia	0.07	0.02	0.82	0.02
Pythia	0.41	0.03	0.84	0.02
Pythia	1.40	0.03	0.84	0.02
RoBERTa SQuAD	0.35	0.02	1.61	0.02
Pythia	0.16	0.02	0.79	0.01
Cerebras-GPT	2.70	0.02	0.81	0.01
GPT-J	6.00	0.03	0.80	0.01
Pythia	2.80	0.02	0.81	0.01
Cerebras-GPT	0.11	0.02	0.92	0.01
Random Name Baseline	-	0.03	0.75	0.1
Pythia	6.90	0.01	0.97	0.01
DistilBERT	0.07	0.01	1.08	0.00

Table 3: Used models: InLen is the maximum input length the model has seen during pretraining. # Parameters is the total parameter count (including the embedding layer). Corpus shows the most important dataset, for specific information see model papers.

Model	Source	InLen	# Parameters	Vocab	Corpus	# Langs
GPT-4	OpenAI [39]	8K	1800B	n/a	n/a	n/a
GPT-3.5	Brown et al. [8]	4K/16K	175B	256K	n/a	n/a
BLOOM	Scao et al. [49]	2K	1.1B/1.7B/3B/7.1B	250K	ROOTS	59
BLOOMZ	Muennighoff et al. [35]	2K	1.1B/1.7B/3B/7.1B	250K	mC4,xP3	109
T5	Raffel et al. [43]	512	60M/220M/770M/3B/11B	32K	C4	1
Flan_T5	Chung et al. [13]	512	80M/250M/780M/3B/11B	32K	collection (see paper)	60
T0	Sanh et al. [48]	1K	3B/11B	32K	P3	1
mT0	Muennighoff et al. [35]	512	580M/1.2B/13B	250K	mC4,xP3	101
Llama	Touvron et al. [53]	2K	7B	32K	CommonCrawl,Github,Wikipedia,+others	20
Llama2	Touvron, Martin, and Stone [52]	4K	7B/13B	32K	n/a	> 13
INCITE	AI [1]	2K	3B	50K	RedPajama-Data-1T	1
INCITE-Instruct	AI [1]	2k	3B	50K	RedPajama-Data-1T	1
Cerebras-GPT	Dey et al. [16]	2K	111M/1.3/2.7/6.7/13B	50K	The Pile	1
GPT-NeoX	Black et al. [6]	2K	20B	50K	The Pile	1
Pythia	Biderman et al. [5]	512/768/1K/2K/2K/2.5K/4/5K	70/160/410M/1.4/2.8/6.9/12B	50K	The Pile	1
GPT-J	Wang and Komatsuzaki [58]	4K	6B	50K	The Pile	1
Falcon	Almazrouei et al. [4]	2K	7B	65K	RefinedWeb + custom corpora	11
Falcon-Instruct	Almazrouei et al. [4]	2K	7B	65K	RefinedWeb,Baize + custom corpora	11
RoBERTa	Liu et al. [32]	512	125M/355M	50K	BookCorpus,Wikipedia,+others	1
RoBERTa SQuAD	Chan et al. [11]	386	125M/355M	50K	RoBERTa,SQuAD2.0	1
DistilBERT	Sanh et al. [47]	768	66M	30K	Wikipedia	1
DistilBERT SQuAD	Sanh et al. [47]	768	62M	28K	SQuAD	1
Models used only on court rulings						
SwissBERT	Vamvas, Graën, and Sennrich [55]	514	110M	50K	Swissdox	4
Legal-Swiss-RobBERTa	Rasiah et al. [44]	768	279M/561M	250K	Multi Legal Pile	3
Legal-Swiss-LongFormer-base	Rasiah et al. [44]	4K	279M	250K	Multi Legal Pile	3
Legal-XLM-RobBERTa-base	Niklaus et al. [38]	514	561M	250K	Multi Legal Pile	24
Legal-XLM-LongFormer-base	Niklaus et al. [38]	4K	279M	250K	Multi Legal Pile	24

Option		Argument Example	Description
–models	-m	bloomz-7b,t0-11b	run spefific model(s)
–model-class	-mc	roberta	run a specific class of models
–size	-s	XL	select a approximate size range of models to run
–device	-d	GPU:0	Specify the preferred device if selection is possible
–dry-run	-dr	-	run only 10 examples
–exclude	-e	bloomz-3b,llama2-7b	exclude specific models from the run
–top-k	-tk	5	specify the number of predictions per example
–save-memory	-sm	-	reduces batch size severely to conserve memory
–no-cache	-nc	-	prevents usage of cache, recomputing any results
–key	-k	my-run-with-top-3	allows to specify the folder name for results (can be reused)
–fast	-f	-	only runs a subset of 100 examples for very quick results
–dataset	-dt	rulings	allows to select between the two default datasets rulings and wiki
–custom-dataset	cd	/path/to/my-dataset	Allows to benchmark a custom dataset
–options	-o	input_length	number of characters given to models
		input_sentences_count	number of sentences given to models
		strategy	type of generation strategy (beam search, greedy, nucleus, ...)
		configs	which configurations to run (original, paraphrased)
		truncate	if possible whether the model should auto-truncate inputs
		input_sentences_count	number of sentences given to models

Table 4: Options for model runner pipeline

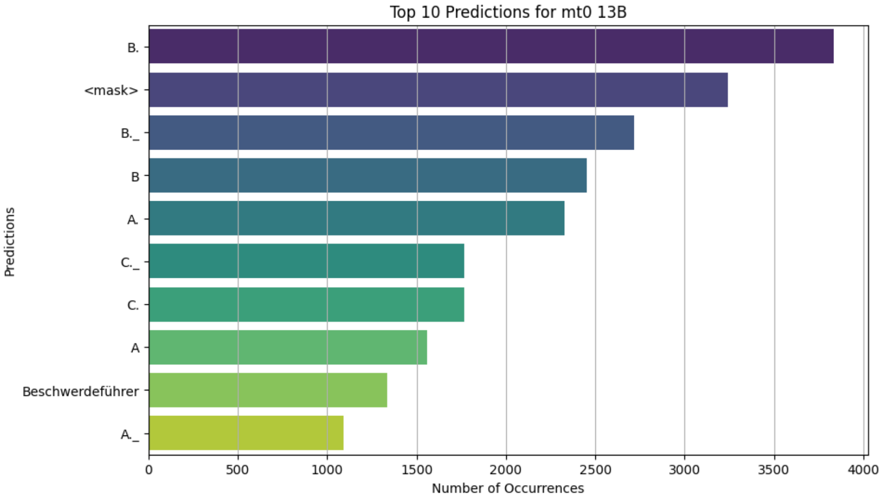

Figure 12: Most common predictions on court rulings for mT0 13B

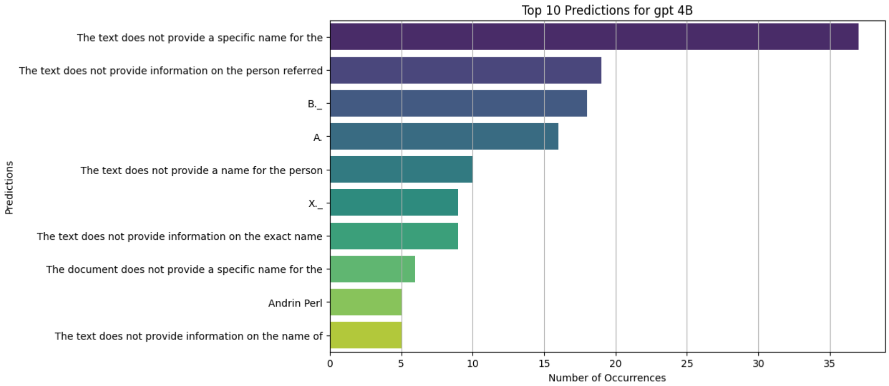

Figure 13: Most common predictions on court rulings for GPT-4

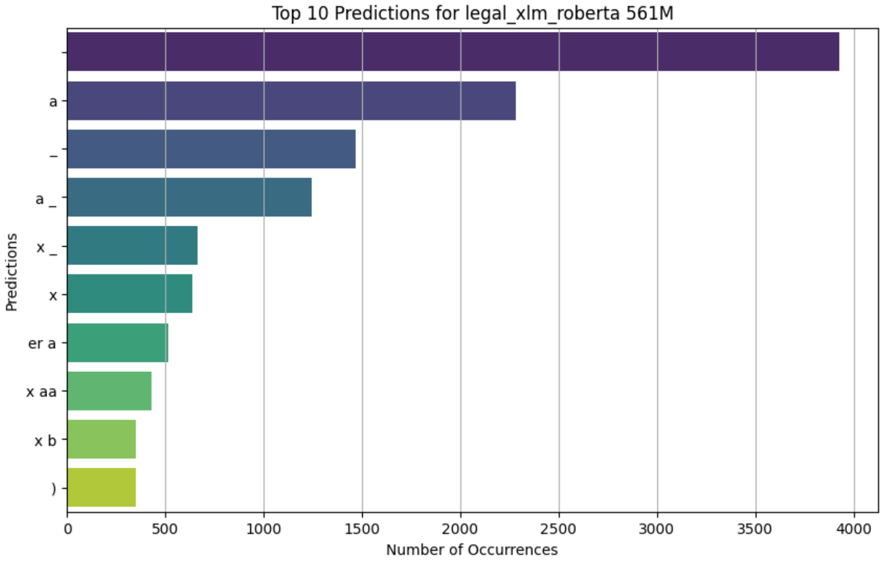

Figure 14: Most common predictions on court rulings for legal-xlm-roberta 561M

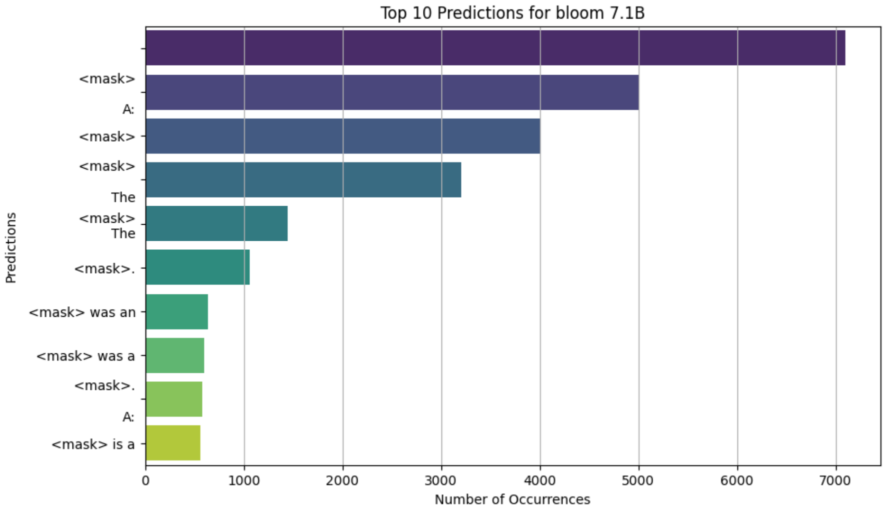

Figure 15: Most common predictions on Wikipedia for bloom 7.1B

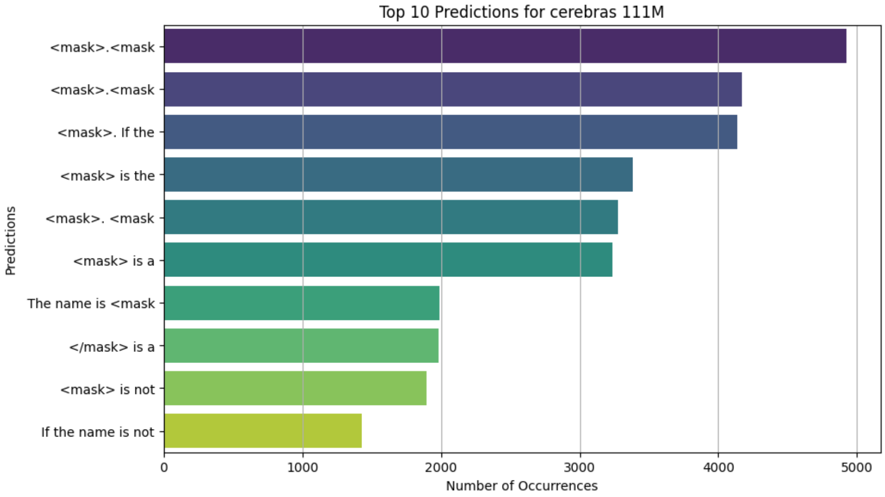

Figure 16: Most common predictions on Wikipedia for Cerebras-GPT 111M

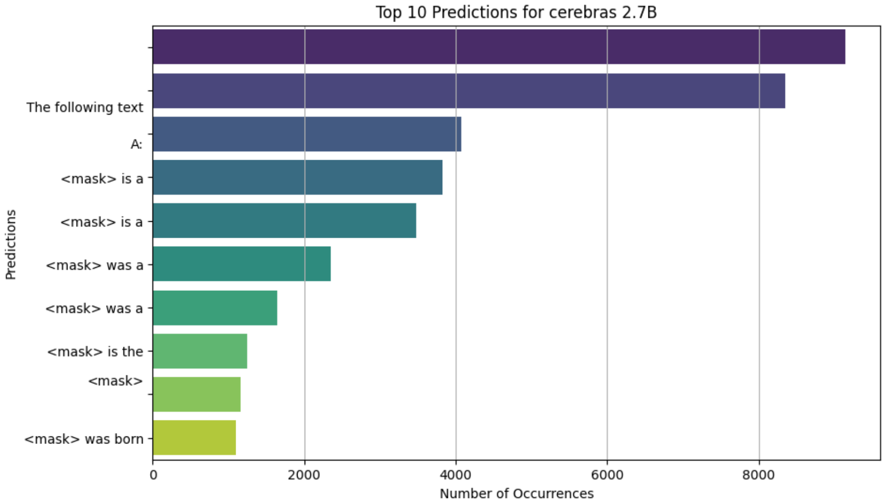

Figure 17: Most common predictions on Wikipedia for Cerebras-GPT 2.7B

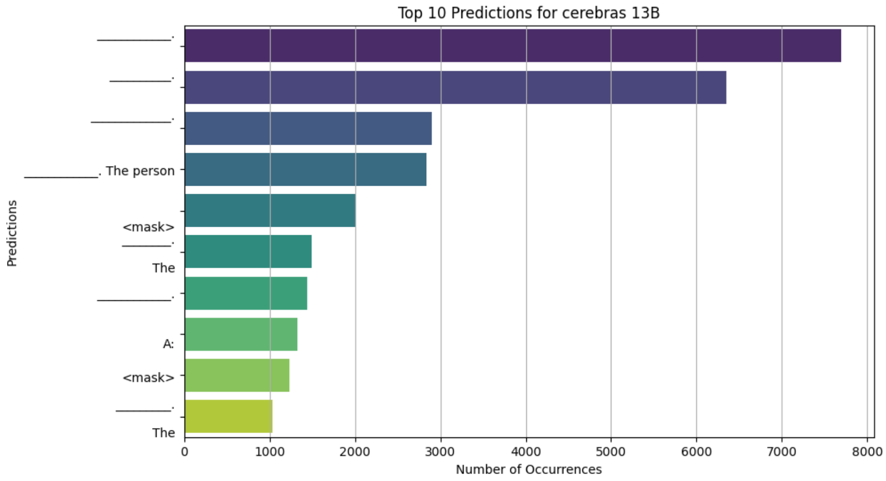

Figure 18: Most common predictions on Wikipedia for Cerebras-GPT 13B

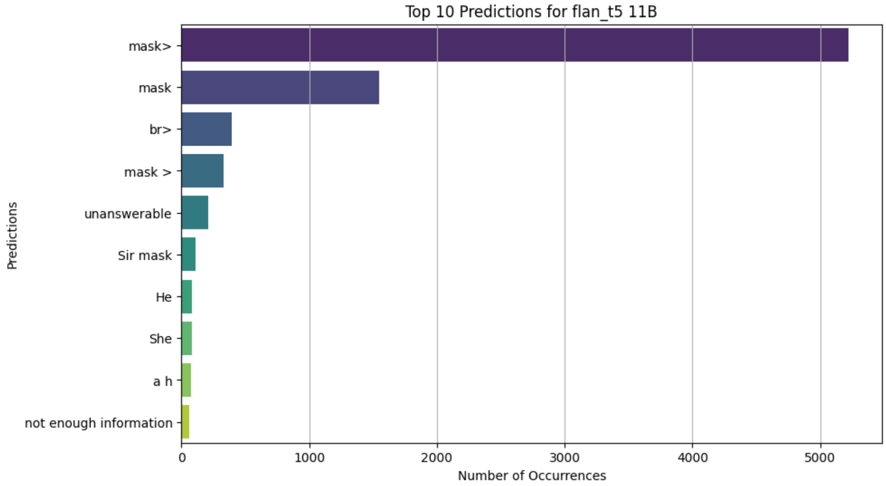

Figure 19: Most common predictions on Wikipedia for Flan_T5 11B

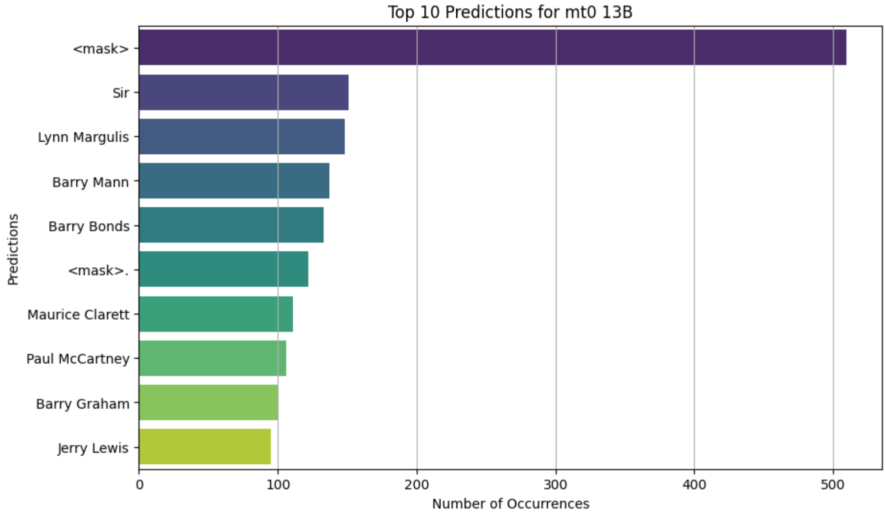

Figure 20: Most common predictions on Wikipedia for mT0 13B

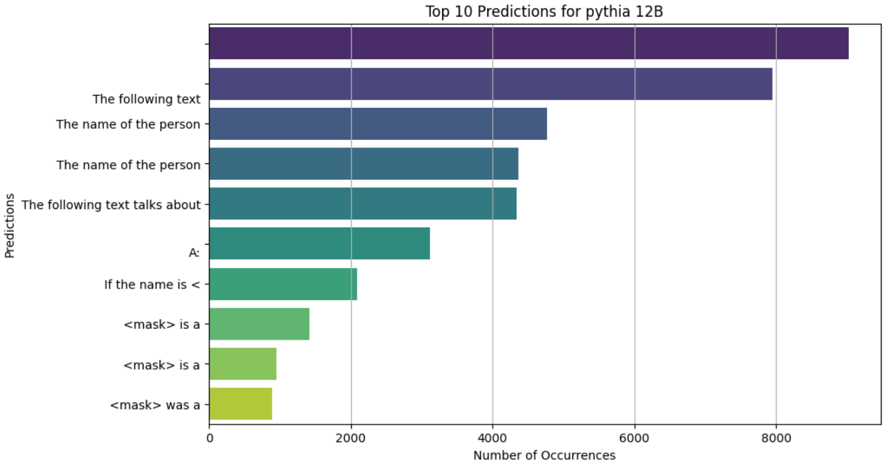

Figure 21: Most common predictions on Wikipedia for Pythia 12B

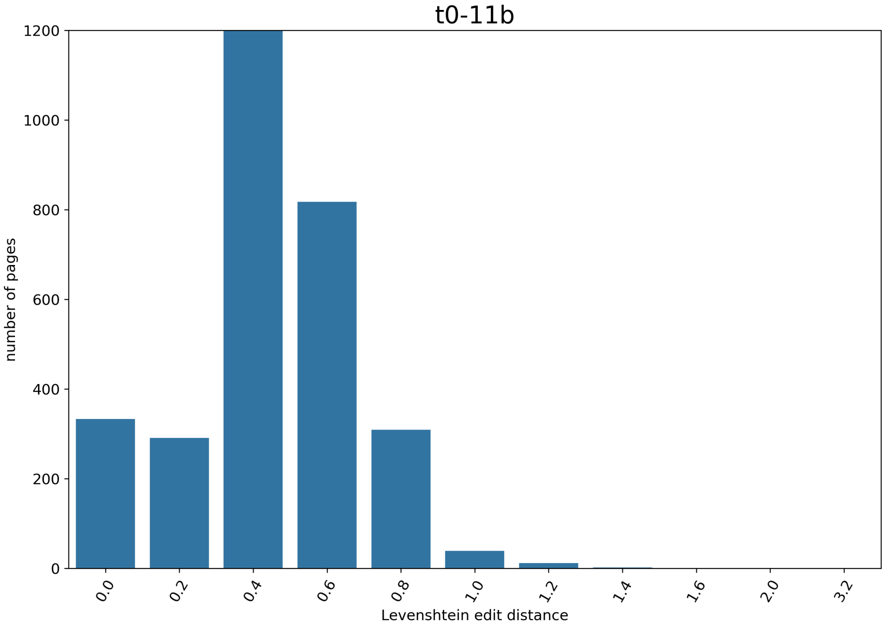

Figure 22: Normalized Levenshtein Distance distribution for T0 11B

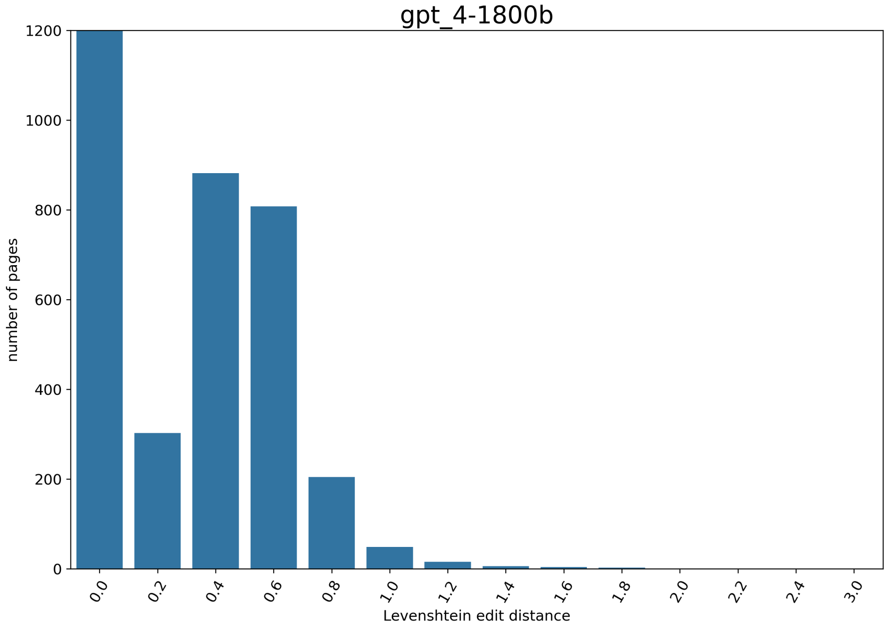

Figure 23: Normalized Levenshtein Distance distribution for GPT-4

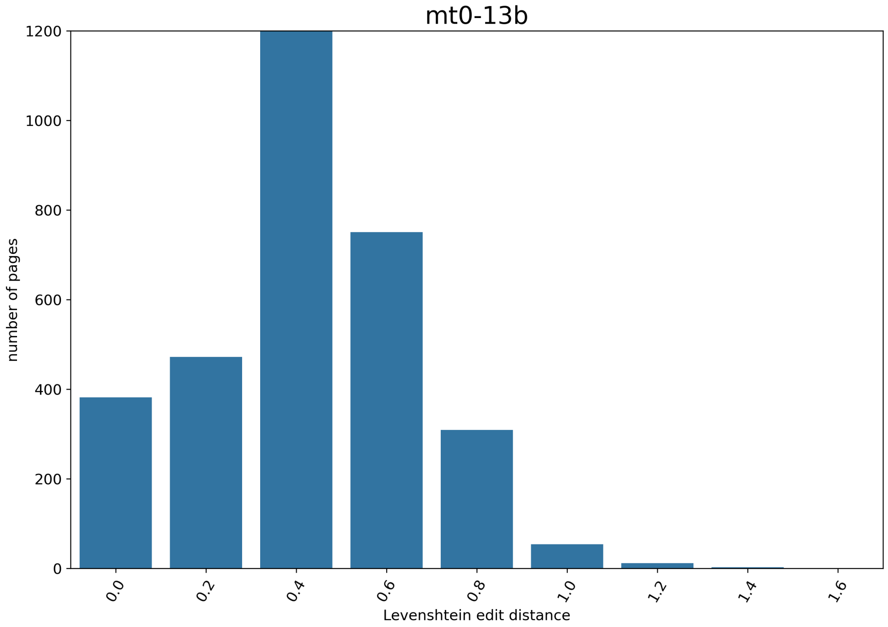

Figure 24: Normalized Levenshtein Distance distribution for mT0 13B

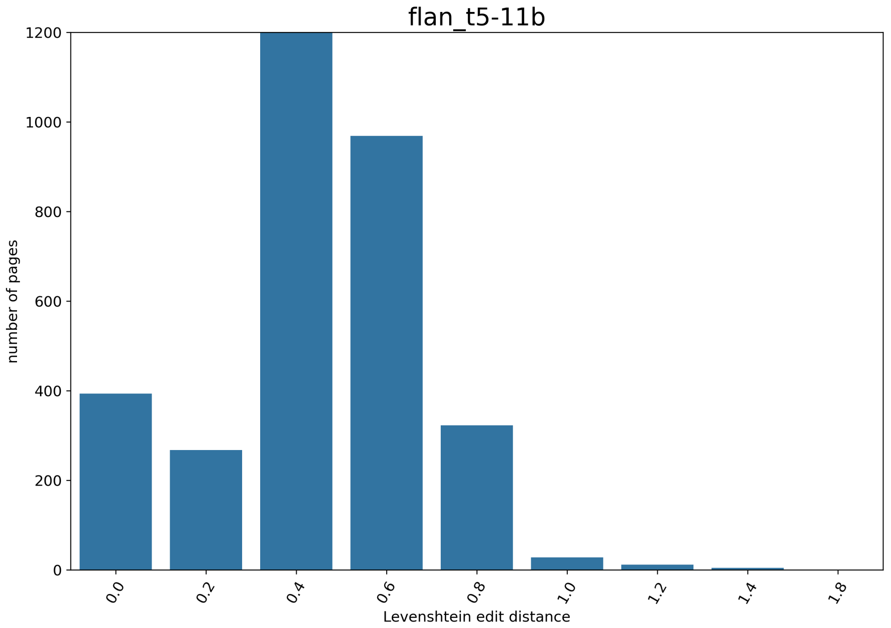

Figure 25: Normalized Levenshtein Distance distribution for T0 Flan_T5 11B

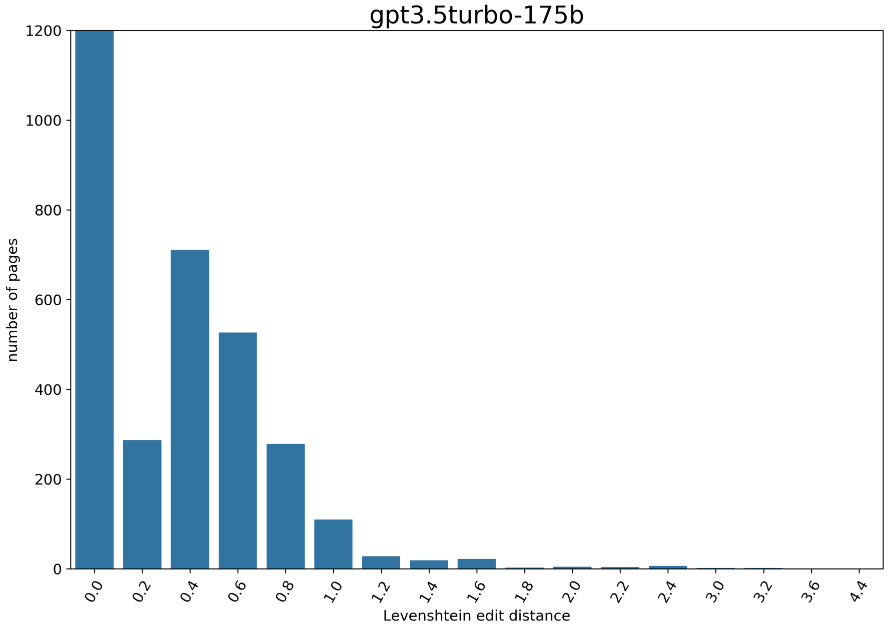

Figure 26: Normalized Levenshtein Distance distribution for GPT-3.5-turbo 175B

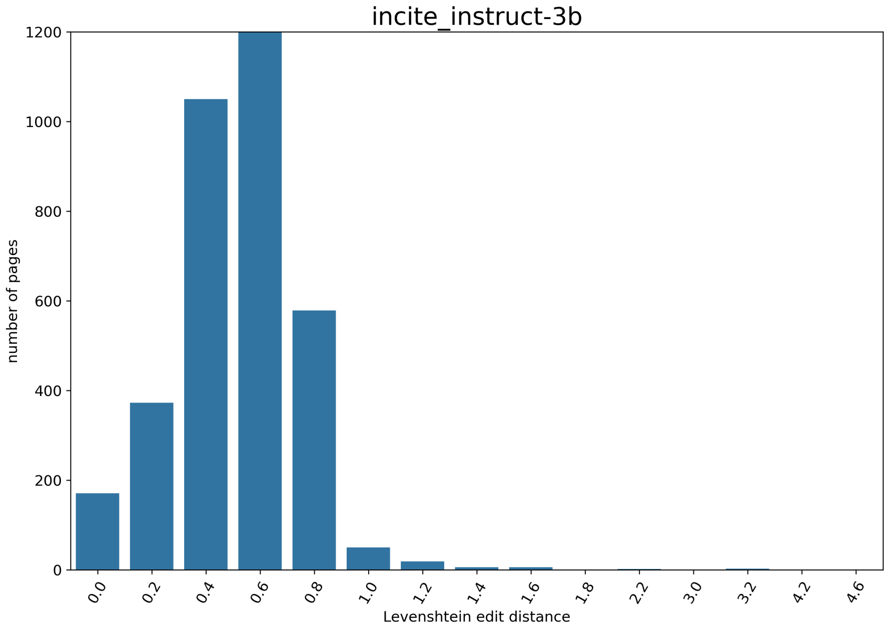

Figure 27: Normalized Levenshtein Distance distribution for INCITE-Instruct 3B

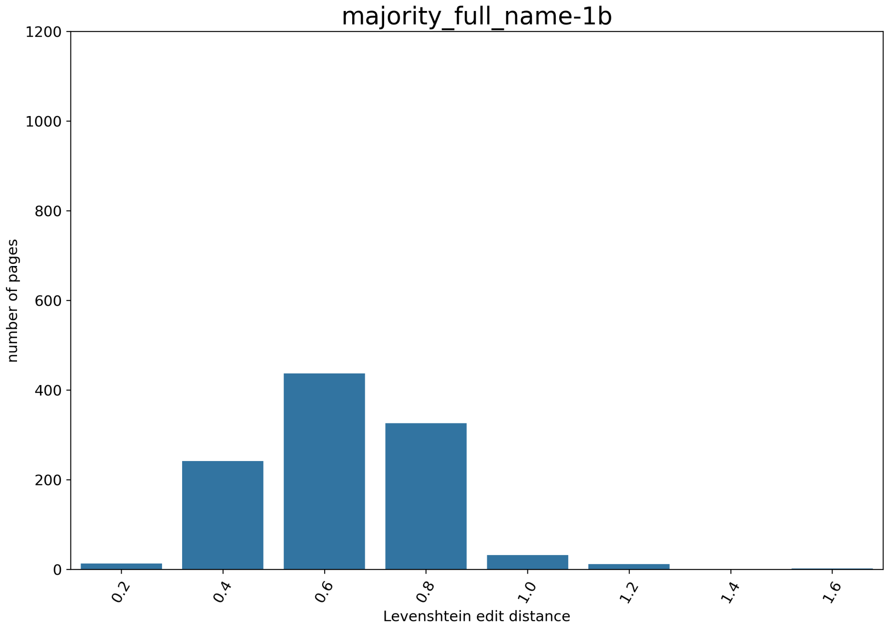

Figure 28: Normalized Levenshtein Distance distribution for Majority Name Baseline

BIBLIOGRAPHY

[1] Together AI. *Releasing 3B and 7B RedPajama-INCITE family of models including base, instruction-tuned & chat models*. en-US. Jan. 2023. URL: https://together.ai/blog/redpajama-models-v1 (visited on 08/18/2023).

[2] Mansi Agarwal. "An Overview of Natural Language Processing." en. In: *International Journal for Research in Applied Science and Engineering Technology* 7.5 (May 2019), pp. 2811–2813. ISSN: 23219653. DOI: 10.22214/ijraset.2019.5462. URL: https://www.ijraset.com/fileserve.php?FID=23111 (visited on 08/24/2023).

[3] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. "A Review on Language Models as Knowledge Bases." In: *arXiv:2204.06031 [cs]* (Apr. 2022). arXiv: 2204.06031. URL: http://arxiv.org/abs/2204.06031 (visited on 04/25/2022).

[4] Ebtesam Almazrouei et al. "Falcon-40B: an open large language model with state-of-the-art performance." In: (2023).

[5] Stella Biderman et al. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. arXiv:2304.01373 [cs]. May 2023. DOI: 10.48550/arXiv.2304.01373. URL: http://arxiv.org/abs/2304.01373 (visited on 08/16/2023).

[6] Sid Black et al. *GPT-NeoX-20B: An Open-Source Autoregressive Language Model*. arXiv:2204.06745 [cs]. Apr. 2022. URL: http://arxiv.org/abs/2204.06745 (visited on 08/16/2023).

[7] Sebastian Borgeaud et al. *Improving language models by retrieving from trillions of tokens*. arXiv:2112.04426 [cs]. Feb. 2022. DOI: 10.48550/arXiv.2112.04426. URL: http://arxiv.org/abs/2112.04426 (visited on 08/15/2023).

[8] Tom B. Brown et al. *Language Models are Few-Shot Learners*. arXiv:2005.14165 [cs]. July 2020. URL: http://arxiv.org/abs/2005.14165 (visited on 08/16/2023).

[9] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. *Quantifying Memorization Across Neural Language Models*. arXiv:2202.07646 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2202.07646. URL: http://arxiv.org/abs/2202.07646 (visited on 08/16/2023).

[10] Nicholas Carlini et al. *Extracting Training Data from Large Language Models*. arXiv:2012.07805 [cs]. June 2021. DOI: 10.48550/arXiv.2012.07805. URL: http://arxiv.org/abs/2012.07805 (visited on 08/14/2023).

[11] Branden Chan, Timo Möller, Malte Pietsch, and Tanay Soni. *roberta-base for QA*. Jan. 2020.

[12] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. "Reading Wikipedia to Answer Open-Domain Questions." In: *arXiv:1704.00051 [cs]* (Apr. 2017). arXiv: 1704.00051. URL: http://arxiv.org/abs/1704.00051 (visited on 05/05/2022).

[13] Hyung Won Chung et al. *Scaling Instruction-Finetuned Language Models*. arXiv:2210.11416 [cs]. Dec. 2022. URL: http://arxiv.org/abs/2210.11416 (visited on 08/18/2023).

[14] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. *LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale*. Number: arXiv:2208.07339 arXiv:2208.07339 [cs]. Nov. 2022. DOI: 10.48550/arXiv.2208.07339. URL: http://arxiv.org/abs/2208.07339 (visited on 05/11/2023).

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *CoRR* abs/1810.04805 (2018). _eprint: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[16] Nolan Dey, Gurpreet Gosal, Zhiming, Chen, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, and Joel Hestness. *Cerebras-GPT: Open Compute-Optimal Language Models Trained on the Cerebras Wafer-Scale Cluster*. arXiv:2304.03208 [cs]. Apr. 2023. DOI: 10.48550/arXiv.2304.03208. URL: http://arxiv.org/abs/2304.03208 (visited on 08/18/2023).

[17] EUGH. "Ab 1. Juli 2018 werden Vorabentscheidungssachen, an denen natürliche Personen beteiligt sind, anonymisiert." de. In: *Pressemitteilung* (July 2018).

[18] Hanjo Hamann. "Der blinde Fleck der deutschen Rechtswissenschaft – Zur digitalen Verfügbarkeit instanzgerichtlicher Rechtsprechung." In: *JuristenZeitung (JZ)* 76.13 (2021). Place: Tübingen Publisher: Mohr Siebeck, pp. 656–665. ISSN: 0022-6882. DOI: 10.1628/jz-2021-0225.

[19] Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A. Choquette-Choo, and Nicholas Carlini. *Preventing Verbatim Memorization in Language Models Gives a False Sense of Privacy*. arXiv:2210.17546 [cs]. May 2023. DOI: 10.48550/arXiv.2210.17546. URL: http://arxiv.org/abs/2210.17546 (visited on 08/16/2023).

[20] Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. "X-FACTR: Multilingual Factual Knowledge Retrieval from Pretrained Language Models." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5943–5959. DOI: 10.18653/v1/2020.emnlp-main.479. URL: https://aclanthology.org/2020.emnlp-main.479 (visited on 07/31/2023).

[21] Karen Sparck Jones. "Natural Language Processing: A Historical Review." en. In: *Current Issues in Computational Linguistics: In Honour of Don Walker*. Ed. by Antonio Zampolli, Nicoletta Calzolari, and Martha Palmer. Linguistica Computazionale. Dordrecht: Springer Netherlands, 1994, pp. 3–16. ISBN: 978-0-585-35958-8. DOI: 10.1007/978-0-585-35958-8_1. URL: https://doi.org/10.1007/978-0-585-35958-8_1 (visited on 08/24/2023).

[22] Srikrishna Karanam, Mengran Gou, Ziyan Wu, Angels Rates-Borras, Octavia Camps, and Richard J. Radke. *A Systematic Evaluation and Benchmark for Person Re-Identification: Features, Metrics, and Datasets*. arXiv:1605.09653 [cs]. Feb. 2018. URL: http://arxiv.org/abs/1605.09653 (visited on 07/10/2023).

[23] Nora Kassner, Philipp Dufter, and Hinrich Schütze. "Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models." In: *arXiv:2102.00894 [cs]* (Feb. 2021). arXiv: 2102.00894. URL: http://arxiv.org/abs/2102.00894 (visited on 02/25/2022).

[24] Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J. Bommarito II. *Natural Language Processing in the Legal Domain*. arXiv:2302.12039 [cs]. Feb. 2023. URL: http://arxiv.org/abs/2302.12039 (visited on 08/14/2023).

[25] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. "Natural language processing: state of the art, current trends and challenges." en. In: *Multimedia Tools and Applications* 82.3 (Jan. 2023), pp. 3713–3744. ISSN: 1573-7721. DOI: 10.1007/s11042-022-13428-4. URL: https://doi.org/10.1007/s11042-022-13428-4 (visited on 08/15/2023).

[26] Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. *Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets*. arXiv:2008.02637 [cs]. Aug. 2020. DOI: 10.48550/arXiv.2008.02637. URL: http://arxiv.org/abs/2008.02637 (visited on 07/31/2023).

[27] Patrick Lewis et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv:2005.11401 [cs]. Apr. 2021. DOI: 10.48550/arXiv.2005.11401. URL: http://arxiv.org/abs/2005.11401 (visited on 08/12/2023).

[28] Xiang Lisa Li and Percy Liang. *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. arXiv:2101.00190 [cs]. Jan. 2021. URL: http://arxiv.org/abs/2101.00190 (visited on 08/13/2023).

[29] David S. Lim. *dslim/bert-base-NER · Hugging Face*. May 2021. URL: https://huggingface.co/dslim/bert-base-NER (visited on 07/10/2023).

[30] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. *Lost in the Middle: How Language Models Use Long Contexts*. arXiv:2307.03172 [cs]. July 2023. URL: http://arxiv.org/abs/2307.03172 (visited on 08/06/2023).

[31] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks*. arXiv:2110.07602 [cs]. Mar. 2022. URL: http://arxiv.org/abs/2110.07602 (visited on 08/15/2023).

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv:1907.11692 [cs]. July 2019. DOI: 10.48550/arXiv.1907.11692. URL: http://arxiv.org/abs/1907.11692 (visited on 08/18/2023).

[33] Shayne Longpre et al. *The Flan Collection: Designing Data and Methods for Effective Instruction Tuning*. arXiv:2301.13688 [cs]. Feb. 2023. DOI: 10.48550/arXiv.2301.13688. URL: http://arxiv.org/abs/2301.13688 (visited on 08/15/2023).

[34] Pia Lorenz. *Machtwort vom BGH: Urteile sind für alle da*. de. May 2017. URL: https://www.lto.de/recht/hintergruende/h/bgh-hzivilgerichte-muessen-urteile-anonymisiert-veroeffentlichen/ (visited on 08/15/2023).

[35] Niklas Muennighoff et al. "Crosslingual generalization through multitask finetuning." In: *arXiv preprint arXiv:2211.01786* (2022).

[36] Niklas Muennighoff et al. *Crosslingual Generalization through Multitask Finetuning*. arXiv:2211.01786 [cs]. May 2023. URL: http://arxiv.org/abs/2211.01786 (visited on 08/17/2023).

[37] Tania Munz. "Staatshaftung für mangelhafte Anonymisierung von publizierten Gerichtsurteilen." de. In: *Richterzeitung* 1 (2022). ISSN: 1661-2981. DOI: 10.38023/07807cc8-8e2d-4792-ba0f-910a40247ec9. URL: https://richterzeitung.weblaw.ch/rzissues/2022/1/staatshaftung-fur-ma_3750d6b1c0.html (visited on 04/11/2022).

[38] Joel Niklaus, Veton Matoshi, Matthias Sturmer, Ilias Chalkidis, and Daniel E. Ho. "MultiLegalPile: A 689GB Multilingual Legal Corpus." In: *ArXiv* abs/2306.02069 (2023).

[39] OpenAI. *GPT-4 Technical Report*. arXiv:2303.08774 [cs]. Mar. 2023. URL: http://arxiv.org/abs/2303.08774 (visited on 08/13/2023).

[40] Long Ouyang et al. *Training language models to follow instructions with human feedback*. arXiv:2203.02155 [cs]. Mar. 2022. DOI: 10.48550/arXiv.2203.02155. URL: http://arxiv.org/abs/2203.02155 (visited on 08/15/2023).

[41] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. "Language Models as Knowledge Bases?" en. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: https://www.aclweb.org/anthology/D19-1250 (visited on 03/03/2022).

[42] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. "E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 803–818. DOI: 10.18653/v1/2020.findings-emnlp.71. URL: https://aclanthology.org/2020.findings-emnlp.71 (visited on 07/31/2023).

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. Tech. rep. arXiv:1910.10683. arXiv:1910.10683 [cs, stat] type: article. arXiv, July 2020. DOI: 10.48550/arXiv.1910.10683. URL: http://arxiv.org/abs/1910.10683 (visited on 05/18/2022).

[44] Vishvaksenan Rasiah, Ronja Stern, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, Daniel E. Ho, and Joel Niklaus. *SCALE: Scaling up the Complexity for Advanced Language Model Evaluation*. arXiv:2306.09237 [cs]. June 2023. DOI: 10.48550/arXiv.2306.09237. URL: http://arxiv.org/abs/2306.09237 (visited on 07/31/2023).

[45] Philippe Remy. *Name Dataset*. Publication Title: GitHub repository. 2021. URL: https://github.com/philipperemy/name-dataset.

[46] Adam Roberts, Colin Raffel, and Noam Shazeer. "How Much Knowledge Can You Pack Into the Parameters of a Language Model?" In: *arXiv:2002.08910 [cs, stat]* (Oct. 2020). arXiv: 2002.08910. URL: http://arxiv.org/abs/2002.08910 (visited on 02/25/2022).

[47] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv:1910.01108 [cs]. Feb. 2020. URL: http://arxiv.org/abs/1910.01108 (visited on 08/18/2023).

[48] Victor Sanh et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization." In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=9Vrb9D0WI4.

[49] Teven Le Scao et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. arXiv:2211.05100 [cs]. June 2023. URL: http://arxiv.org/abs/2211.05100 (visited on 08/16/2023).

[50] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. *Retrieval Augmentation Reduces Hallucination in Conversation*. arXiv:2104.07567 [cs]. Apr. 2021. URL: http://arxiv.org/abs/2104.07567 (visited on 07/31/2023).

[51] Benjamin Stückelberger, Yesilöz Evin, and Cavallaro Damian. *Anzeige von Namensänderungen strafrechtlich Verurteilter nach identifizierender Medienberichterstattung | sui generis*. Mar. 2021. URL: https://sui-generis.ch/article/view/sg.172/1733#_Toc66349918 (visited on 04/22/2022).

[52] Hugo Touvron, Louis Martin, and Kevin Stone. "Llama 2: Open Foundation and Fine-Tuned Chat Models." en. In: (July 2023).

[53] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. arXiv:2302.13971 [cs]. Feb. 2023. DOI: 10.48550/arXiv.2302.13971. URL: http://arxiv.org/abs/2302.13971 (visited on 08/18/2023).

[54] Dimitrios Tsarapatsanis and Nikolaos Aletras. "On the Ethical Limits of Natural Language Processing on Legal Text." In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3590–3599. DOI: 10.18653/v1/2021.findings-acl.314. URL: https://aclanthology.org/2021.findings-acl.314 (visited on 08/15/2023).

[55] Jannis Vamvas, Johannes Graën, and Rico Sennrich. *SwissBERT: The Multilingual Language Model for Switzerland*. arXiv:2303.13310 [cs]. June 2023. DOI: 10.48550/arXiv.2303.13310. URL: http://arxiv.org/abs/2303.13310 (visited on 07/31/2023).

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need." In: *arXiv:1706.03762 [cs]* (Dec. 2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (visited on 03/11/2022).

[57] Kerstin Noëlle Vokinger and Urs Jakob Mühlematter. "Re-Identifikation von Gerichtsurteilen durch "Linkage" von Daten(banken)." de. In: (2019), p. 27.

[58] Ben Wang and Aran Komatsuzaki. *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. May 2021. URL: https://github.com/kingoflolz/mesh-transformer-jax.

[59] Cunxiang Wang, Pai Liu, and Yue Zhang. *Can Generative Pretrained Language Models Serve as Knowledge Bases for Closed-book QA?* Number: arXiv:2106.01561 arXiv:2106.01561 [cs]. June 2021. URL: http://arxiv.org/abs/2106.01561 (visited on 06/26/2023).

[60] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv:2201.11903 [cs]. Jan. 2023. URL: http://arxiv.org/abs/2201.11903 (visited on 08/15/2023).

[61] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. "Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts." en. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Hamburg Germany: ACM, Apr. 2023, pp. 1–21. ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3581388. URL: https://dl.acm.org/doi/10.1145/3544548.3581388 (visited on 08/13/2023).

[62] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. _eprint: 1912.08777. 2019.