The Alan Turing Institute

Selective Forgetting in LLMs

September 2024

Jack Dymond, Data Scientist
Phil Swatton, Data Scientist
Jack Roberts, Lead Data Scientist
James Bishop, Principal Data Scientist

Executive Summary

This work approaches the problem of selectively forgetting knowledge from a large language model (LLM) for the purposes of safety, copyright, security, or otherwise. Also known as machine unlearning, this entails training a model to *forget* certain elements of the dataset on which it was trained. Unlearning methods must be evaluated both in terms of the extent to which the information has successfully been forgotten, and the performance of the unlearned model on the remaining (retained) data.

We build on the work of TOFU (Task of Fictitious Unlearning) [21], which provides a dataset and benchmark for evaluating unlearning techniques. We create a new, TOFU-inspired question–answer dataset for the task of machine unlearning. The new dataset includes 10,500 question–answer pairs relating to over 1,000 distinct, synthetic entities of several types. Each question–answer pair is tagged with the entities it refers to, with the graph representation of our dataset containing over 2,600 edges between different entities.

We perform two experiments with our dataset. The first experiment aims to capture whether the difficulty of forgetting a concept from a LLM depends on its granularity. For example, is unlearning more likely to be successful if forgetting a single book, rather than the book's author (as an author is connected to multiple books)? We find that granularity does not have a tangible effect on model performance in our dataset. There may be a small effect from granularity on the difficulty of forgetting, but this is not statistically significant across our results. Our second experiment explores the knock-on effect of forgetting a relationship between two entities. For example, if unlearning has been run on a model to forget only who wrote a book, but not the book or author themselves, is the model worse at responding to other questions about that book or author? We find that the model performance is lower on questions that contain the entities pertained in the relationship, than on those that do not.

Contents

1 Introduction

Large Language Models (LLMs) require large amounts of text for their training. In this report we are primarily concerned with training on large amounts of text which may contain sensitive information. However, the common use of web-scraped data has introduced similar issues including but not limited to privacy violations, copyright violations, and toxic behaviour on the part of the models [11, 23]. It is therefore often desirable to remove particular information from a model's "knowledge".

A promising avenue for tackling these issues is *machine unlearning*. Machine unlearning has been an active area of research for some time [9, 22]. There are two approaches to machine unlearning: *exact unlearning*, and *approximate unlearning*. The two approaches describe a fundamental trade-off in unlearning approaches.

In *exact unlearning*, a guarantee is provided that the unlearned model is indistinguishable from the distribution of models it would be possible to train without the data to be forgotten [9]. The most straightforward approach to this is to remove the offending data from the training dataset and re-train the model from scratch. This provides a guarantee of success, at the cost of re-training the model. The emphasis of research in exact unlearning is therefore on attempting to reduce the overall cost of this re-training or find ways to minimise the amount of re-training required [8].

Approximate unlearning shares the same goal as exact unlearning in attempting to produce a model that is indistinguishable from a set of models trained without the data being forgotten. Unlike exact unlearning, there is no theoretical guarantee that this will be achieved. However, the sacrifice of this guarantee usually results in a more feasible and less computationally costly approach for larger models. A naive approach would be to fine-tune the model with the target of maximising the loss on the undesired data.

LLMs place greater stress on the trade-off between guarantees of forgetting and cost. Given their reliance on ever-larger amounts of training data; and the increasing cost of training LLMs, there is thus a need for greater emphasis on approximate unlearning. Exact unlearning is, in its original form, unfeasible as a solution due to the substantial costs involved in training LLMs. LLMs further raise additional issues for machine unlearning, not least among which is the difficulty of defining the scope for unlearning [20].

A recent piece of work has introduced a benchmarking dataset for approximate unlearning techniques as applied to LLMs called TOFU (Task of Fictitious Unlearning) [21]. TOFU is a QA dataset on fictitious authors (see full discussion in section 3). The authors begin by generating profiles of fictitious authors using a combination of real-world data and GPT-4. These profiles are then fed to GPT-4 to generate 20 question-answer pairs for each profile.

In the associated paper, the TOFU authors benchmark a range of approximate unlearning techniques against exact unlearning. They show that there is a trade-off between the forget success (how well unlearning is performed) and overall model utility (how well the model performs on things it should not unlearn - i.e. general performance) [21].

In this report, we detail our own work which builds on that of TOFU. Our primary contribution is to introduce a new fictitious authors QA dataset, which allows us to explore the impact of two additional variables on the forget quality vs model utility trade-off.

First, we explore the effect of the *granularity* of what is being forgotten - do less granular forget targets exacerbate the trade-off? Second, we explore whether when forgetting a particular relationship (e.g. that Alice is Bob's sister), the model utility on the entities whose relationship is being forgotten suffers a worse effect than other data being retained (e.g. that the model performs worse on questions about Alice and Bob that do not relate to them being siblings).

2 Theory

2.1 Defining Unlearning

Different definitions of unlearning have been offered in the literature. These definitions are motivated by different goals and concerns in the context of machine unlearning.

One of the most useful definitions is machine unlearning as counterfactual in nature: unlearning is successful if the model belongs to the distribution of models that would be possible to train on the same dataset with the data to be unlearned removed.

Formally, consider a dataset $\mathcal{D}$ containing many training observations. A model, ϕ, is trained on this dataset. It is discovered that some of the data $f \subset \mathcal{D}$ contains content we do not wish the model to 'know'. This may, for example, be some individual details, some sensitive intellectual property, or some other content that we do not wish the LLM to repeat while generating text.

The goal of unlearning from this perspective is therefore to ensure that the behaviour of ϕ is the same as if it were trained on $r = \mathcal{D} \setminus f$. In exact unlearning, we re-train a new model ϕ_r on r in order to remove the desired data from the model's 'knowledge'. In approximate unlearning, we attempt to make the weights of the model ϕ match a hypothetical model $\hat{\phi}_r$.

The counterfactual definition makes theoretical sense but is not without criticism. [28] take the perspective of a person who wishes for their data to be removed from a model. They then show that because it is hypothetically possible for two non-identical datasets to arrive at the same set of model weights, the counterfactual definition cannot give a guarantee to that person that unlearning has been performed. Instead, they argue for a definition of unlearning in terms of the impact of a particular data point on the loss trajectory during training, thus requiring an auditable approach based on logging each training step.

A third, pragmatic definition is to simply minimise the probability of predicting tokens that represent the data to be forgotten, without negatively impacting the data to be retained. Here, the goal is simply to ensure poor performance on the data to be forgotten rather than replicate a (hypothetical) model which had not been trained on the forget data.

2.2 Conceptual Problems in Defining Forget and Retain Sets

Both the counterfactual and pragmatic definitions of unlearning includes the notion of a forget set $f \subset \mathcal{D}$ of data to be removed and a corresponding retain set $r = \mathcal{D} \setminus f$.

When dealing with text as data, defining the forget set can become conceptually difficult in practice; and in many cases is an unsolved problem (or indeed, a problem without an objective solution) [20].

Consider an abstract case of removing a concept or entity from a model trained on all the pages of Wikipedia. One approach would be to treat the page for that entity or concept as the forget set; and the rest of the dataset as the retain set. However, Wikipedia is defined in part by links across pages. The same concept or entity will likely be linked to or even be explicitly discussed on other pages.

One approach (ignoring the practical difficulty involved in finding all relevant pages on a corpus such as Wikipedia, even with the links between pages) might be to remove all pages linking to that page or mentioning the concept or entity in question. This is a fairly extreme approach to defining the forget set - not least because many of these pages will contain data we would prefer to retain.

Another approach might be to remove the primary offending page, while *editing* the mentions on other pages. In the literature there are no clear guidelines on the best choice of approach or on the method of editing the mentions on other pages (remove? replace?) for a given problem.

More concretely, consider the case of forgetting the existence of birds from a model trained on this dataset. How should the film 'Birdman' be handled? The character 'Owl Man' from the graphic novel (and adaptations of) 'The Watchmen'? Or the phrase 'kill two birds with one stone'. Should a variant of this phrase still exist in the unlearned model/retain dataset? We have found it worthwhile to think through a few examples of what forgetting would entail in counterfactual terms. For instance: what would a world without chairs look like; and how would a model in such a world behave? What would forgetting the existence of Harry Potter as in [13] mean for answering the question "Who is the world's bestselling children's author"?

2.3 Hierarchy of Concepts

In thinking through these issues, we have found it productive to think in terms of hierarchical relationships between concepts and entities. Continuing with the idea of forgetting birds, consider how conceptually difficult it would be to define a counterfactual in each of the following cases:

1. The existence of birds in general
2. The existence of the family of birds Columbidae (Pigeons and Doves)
3. The existence of the Rock Pigeon (the kind commonly found in British high streets, among other locations)

We believe that the conceptual task of defining forget and retain sets is easiest in the third case, and hardest in the first. As the depth through the hierarchy increases the concepts become more granular, we refer to this property throughout as granularity. The lowest granularity refers to a root node in the hierarchy, whereas the highest granularity a leaf node.

3 TOFU

Beyond the conceptual difficulties raised by unlearning in the text domain, a second problem raised by seeking to perform unlearning on LLMs is their sheer size. Performing exact unlearning - necessary to benchmark approximate unlearning given our definition - becomes unfeasible in this setting.

Relatedly, LLMs are trained on vast amounts of data often not easily accessible to researchers. Defining forget and retain sets with complete knowledge thus becomes impossible for many extant models. Even if we know what concepts an LLM 'knows', we do not have sight of (all) the training data that produced that 'knowledge'.

The authors of TOFU solve both of these problems by introducing the use of a fictitious dataset [21]. They train three types of models on this dataset (where base model refers to a pre-trained LLM from an online model hub):

- **Full models**: Base models fine-tuned on the whole synthetic TOFU dataset.
- **Retain models**: Base models fine-tuned on only a retain subset of TOFU.
- **Forget models**: Full models further fine-tuned to remove knowledge of data in a forget subset, using an approximate unlearning technique.

By fine-tuning both full models ϕ and retain models ϕ_r on the fictitious dataset (or rather subsets thereof for the retain model), exact unlearning becomes much more tractable. Comparing forget models to retain models provides a robust benchmark for evaluating approximate unlearning techniques.

3.1 Generating TOFU

To generate the TOFU dataset, the authors go through a series of steps. These are listed below [21]:

1. Sample fictitious author attributes using GPT-4 and real-world data
2. Generate a profile for each fictitious author
3. Provide GPT-4 with the profile in a prompt, and ask it to generate 20 questions relating to that author

In general, the TOFU dataset is relatively simple in its attributes. Each author has 20 associated questions, and each question always relates only to a single author. Relations between entities are minimal, except simple ones such as those between authors and their books.

The TOFU creators take care to avoid the risk of leakage of real-world author data into the TOFU dataset by manually creating novel author attributes; they note that generating these with GPT-4 tended to result in limited diversity. Once the QA dataset has been completed, the author's experiments are structured as followed:

1. Define a random percentage of the data as the forget set
2. Define the rest as the retain set
3. Fine-tune a model on the full dataset and the retain set
4. Apply forgetting techniques to the full model, using the forget set and sometimes the retain set
5. Perform forgetting using the corresponding 'forget' sets, sometimes with support from the 'retain' set
6. Compute the 'forget success' and compare to the same metric on the retain model
7. Compute 'model utility' and compare to the same metric on the the retain model

3.2 TOFU Results

The TOFU paper provides three main findings for reserachers and practitioners to draw on:

1. **Result 1:** Approximate unlearning can cause some forgetting, but it is not as good as exact unlearning.
2. **Result 2:** A trade-off exists between forgetting success (measured by proximity to the retain model on the forget set) and overall model utility (measured by performance on the retain set, real author set, and real world facts set)
3. **Result 3:** As forget set gets larger, the performance trade-off becomes increasingly pronounced, i.e. there is more effect on model utility

4 Research Questions

Our goal in this report is to expand on the TOFU results and dataset. We do this by seeking to vary the attributes of the fictitious knowledge being forgotten and of the QA dataset. To motivate our expansion, consider the following stages of dataset generation as distinct:

1. Creating fictitious knowledge (author profiles in the case of TOFU)
2. Using fictitious knowledge to produce a dataset (a QA dataset in the case of TOFU)
3. Creating a forget and retain sets from the dataset for experiments

Broadly speaking, we aim to build on the TOFU results in two ways. First, we seek to vary the granularity of the fictitious knowledge being forgotten in step 1. By 'granularity' we mean the position of the fictitious knowledge in the hierarchy of entites. Our first research question can thus be stated as follows:

Research Question 1: *What, if anything, is the effect of the position of the fictitious knowledge in the hierarchy of entities on unlearning performance?*

Here, we are varying the forget target according to granularity. We will need to pay attention to other correlates of granularity such as the overall size of the amount of information being forgotten. It is plausible that the main effect of granularity (especially vis a vis the forget success vs model utility trade-off) is via its necessitating a larger forget set, rather than because of anything intrinsic to granularity.

Second, we wish to explore the potential difficulties involved in forgetting a particular relationship without forgetting the entities that relationship exists between. Approximate learning can degrade the wider performance of the model (to the point where for approximate unlearning, there may be a trade-off between success in unlearning and model quality). We suspect that where a particular relationship is being forgotten, this degradation will be exacerbated for those entities whose relationships are being forgotten.

Research Question 2: *Where a relationship is being forgotten, is the model degradation worse for data regarding the entities whose relationship is being forgotten than for other entities?*

Here, we require a dataset that contains multiple relationships between different entities. TOFU for instance has relationships between authors and books. We expand on this by introducing several additional relationships based on our hierarchical model.

5 Dataset Design

For the purposes of this work a new dataset was required, in this section we briefly discuss the strengths and weaknesses of the TOFU dataset, and then explain how it is expanded upon in this work.

5.1 Extending TOFU

TOFU presented a good starting point for our experimentation: its focus on artificially generated data meant that there was little to no overlap with any pre-training data of the original model. This means that the forgetting effect can be isolated and controlled for in the form of a *retain* model, a model that is guaranteed to never have seen the forget target data. Indeed, this is the basis of the experimentation of the TOFU paper.

TOFU presents a rich, yet self-contained, environment in which to perform the forgetting of authors and their books. This makes it attractive for several reasons. Entities can be grounded by relating them to a number of adjacent real-world concepts such as authors to genres and countries, without the risk of contradicting existing knowledge since the entities (authors) are fictitious. Furthermore, data such as books can be made as rich and detailed as needed without the risk of leaving the conceptual environment of the forgetting task. The use of GPT-4 allows a large, diverse dataset to be curated quickly. In order to evaluate the model, the authors also introduced paraphrased responses to the forget questions, so that the model does not overfit to the training phrasing of each input.

However, there are issues with the dataset created in the initial paper. TOFU provides only three pre-defined forget/retain splits of the data with accompanying paraphrased answers, all of which target forgetting whole authors. The limited number of splits makes it difficult to obtain statistically grounded conclusions, and it's not possible to evaluate different forget targets (removing knowledge of a book rather than an author, for example). There's also no standardisation or labelling of the concepts pertained in each question–answer pair, so it's not possible to create new forget/retain splits with different targets (to identify all the data relating to a particular book, for example). There is also limited interconnectivity between entities in the dataset – there are no question-answer pairs that include multiple authors, for example. Furthermore, the use of GPT-4 inevitably introduced errors to the question–answer pairs. For example, one instance asks the question "What is the name of the Author?", which without any additional context is impossible to answer.

Hence, to answer our research questions, we require a new dataset. To that end, we develop a dataset generation pipeline based on the TOFU dataset, but expanding it to contain entities with different granularities and connections between entities as required to answer our research questions.

5.2 Data Generation Pipeline

Our dataset generation pipeline consists of two steps, *graph generation*, and *question generation*.

Graph generation refers to the process of creating the entities we will task a given model with forgetting, and grounding them in related concepts. This creates a graph of varying entity types, with connections between each where appropriate.

Question generation encodes this graph textually in the form of question–answer pairs, making it parsable to pre-trained large language models. This allows us to fine-tune a model on our artificial data, providing it with a latent understanding of the concepts and relationships we will task it with forgetting. We discuss these processes in more detail below.

5.2.1 Graph Generation

In order to generate our graph we first define a number of entities that form our forget targets, which in descending order of their position in the hierarchy are: publishers, authors, and books[1]. These entities are appropriately linked to each other and to the real-world properties of countries and genres. The entity relationship diagram we use to inform our connections is shown in figure 1.

[1] Note, this is in ascending order of their granularity.

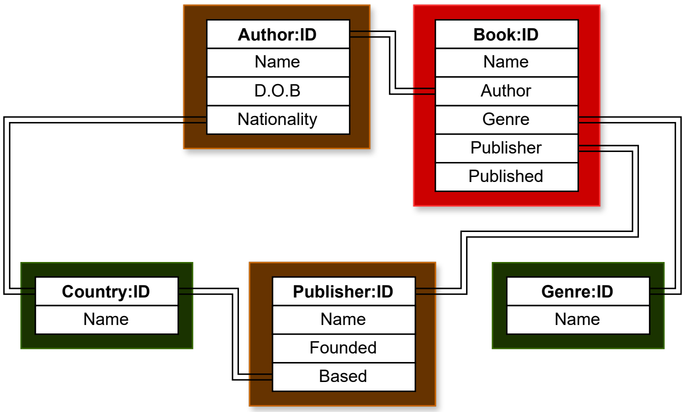

Figure 1: Entity relationship diagram for our dataset.

This allows us to have consistent relationships between our forget targets, and we provide each entity a unique ID allowing them to be tracked throughout the dataset. In order to keep them balanced we uniformly distribute these connections randomly such that each country has the same number of publishers and authors, each author has the same number of books, and so forth. The resultant dataset is made up of Python dictionary items which can be accessed using their ID keys, full examples can be found in the project GitHub repository.

In order to generate the names, we prompt a GPT-3.5 model to generate as many unique names as we need for a given entity type from a given country, or a book from a given genre. Our resultant graph is composed of 800 books, spread across 10 genres, 200 authors, and 40 publishers, as well as 40 countries from which the authors and publishers originate. Hence, there are 1050 individual items (nodes) and 2640 connections (edges) in the dataset. Next, we discuss how we populate this graph with question–answer pairs which describe and interrelate entities.

5.2.2 Question Generation

Each sample in the question-answer dataset has 7 components: The question itself and the target answer, used for model training. For evaluation purposes, a paraphrased version of the question and answer pair, and a list of erroneous answers for each question (evaluation metrics are discussed in section 6.4). Finally, in order to generate data subsets that will be useful for targeting the forgetting of specific entities and relationships, it is also necessary to tag each question-answer pair with the entity tags they include. To that end, each sample in the dataset contains every entity ID key referred to in its generation. An example row of our question–answer dataset is shown in table 1:

Broadly, there are 4 steps to our generation process for our questions: Basic Generation, Complex Generation, Paraphrasing, and Perturbing. This section will motivate each and discuss them in more detail. The final dataset has approximately 10,500 question–answer sets.

5.2.3 Basic Question Generation

There are a number of simple relationship patterns within the dataset which can be written in a formulaic fashion. For example, "Where was the author x born? The author x was born in y", where x and y refer to the author name and country of birth respectively. Another example would be to list all of the books published by a given publisher.

These questions can be generated using well-defined templates, for example using the graph data to substitute x and y for each question. Since these questions depend on the number of edges an entity has, each entity has a different number of these: we generate 24 for each publisher, 17 for each author, and 6 for each book. One potential consequence of using formulaic questions for training is the language model confounding the structure of the questions with the information they contain. To allow training on more diversely structured versions of

Original Phrasing	
Question:	In which genre does the book Requiem of the Shattered Coast belong?
Answer:	Requiem of the Shattered Coast is categorized as a fantasy book.
Perturbed answers:	Requiem of the Shattered Coast belongs to the romance genre. The book Requiem of the Shattered Coast belongs to the science fiction genre. The book Requiem of the Shattered Coast belongs to the romance genre.
Paraphrased	
Question:	What genre is the book, Requiem of the Shattered Coast?
Answer:	Requiem of the Shattered Coast falls under the genre of fantasy books.
Perturbed answers:	Requiem of the Shattered Coast is a cookbook that specializes in seafood recipes from the Pacific Northwest region. The book, Requiem of the Shattered Coast, is a cookbook about seafood recipes. Requiem of the Shattered Coast is a cookbook.
Meta-data	
Keys:	`[LIST OF UUID KEYS]`

Table 1: Single question row in the dataset

these questions we paraphrase them, as discussed in section 5.2.5, and reserve the original templated versions for evaluation only.

5.2.4 Complex Question Generation

We use a GPT client to generate questions that are more varied in structure and relate more complex relationships into the dataset. The original TOFU paper uses a GPT-4 model, we find we can get similar results using GPT-3.5 whilst incurring a significantly lower financial cost. To generate an entity's questions we prompt GPT with an increasingly more detailed profile of the entity, which includes an increasing number of relationships. A full breakdown of our prompting methodology is available in section F of the appendix. We also use few-shot prompting to increase the likelihood that the generated questions incorporate the information we want at a given stage. At each stage we provide the previously generated questions and their answers to reduce the chances of information being repeated or contradicted.

We provide the profiles with additional attributes, such as the number of words a book is, or the previous education of an author. This prompts the generation of questions that relate to the provided context, ensuring questions of similar subject matter across entities whilst also ensuring diversity in their generated properties. Along with the few-shot prompting, this reduces the possibility of hallucinating information which might later be contradicted when more information about the entity is provided to the profile. For example, hallucinating an erroneous place of birth.

In addition to the formulaic questions described in the previous section, we generate 6 questions for each book entity, 3 for each author, and 2 for each publisher. Hence, the majority of our dataset is composed of simple questions, then we use GPT-3.5 to supplement the dataset with a smaller set of questions which incorporate multiple relationships.

5.2.5 Question Paraphrasing

As mentioned previously, the use of formulaic questions for training may cause the model to learn the structure of these questions, as opposed to the underlying information. To address this, the next step of our question generation pipeline is to prompt GPT-3.5 to create paraphrased versions, preserving meaning but modifying the wording, of every question–answer pair. This is performed for every question individually. These paraphrased questions are then used to make the training set of our dataset.

The original templated phrasings are used for the evaluation set, as the TOFU evaluation metrics require different phrasings to be used than the training set (see section 6.4). Paraphrasing all rows of the dataset allows the entire dataset to be split into forget and retain splits, in contrast to the original TOFU dataset which provides them only for specific partitions.

5.2.6 Question Perturbing

Following paraphrasing the final step is to introduce erroneous, *perturbed* rows, as also required by the evaluation metrics (section 6.4)). In order to perturb the question another prompt for the GPT3.5 client was created which would ask the model to answer the question confidently with no other context about the entity in the prompt. Since the model has no prior knowledge of the answer it would likely be incorrect, that is, it would hallucinate its response to the question. We sample three perturbed answers for both the original and paraphrased versions of each question.

On some occasions, the model would hallucinate a correct response to the question. In the event of this happening, the tokens containing the correct answer were replaced with erroneous tokens. This is discussed further in section 5.3 below.

5.3 Pitfalls

When generating the dataset a number of issues arose, predominantly in controlling the outputs of the GPT model used to generate the data. We briefly discuss a number of these in this section.

Complexity vs. Structure There is a trade-off when generating data that the output should contain only the information that is desired whilst also being natural and diverse. Unconstrained generated content from large language models has a tendency to hallucinate and introduce ideas that were not intended. Therefore, our goal is to allow the model to hallucinate in a constrained manner. To accomplish this we incrementally provide the model with information about the target entity, and as questions are generated we recursively add these to the prompt for the model. In combination, this means the model is only provided enough information to generate questions containing information we want, whilst also ensuring questions and answers are not repeated.

Repeated perturbed samples When generating perturbed answers to questions, we found examples where a certain completion to a question was sufficiently probable for the GPT client to always return identical responses, rather than multiple incorrect phrasings as desired. An example is given in table 2. These were very infrequent and did not pose a significant problem to evaluation, as such we prioritised fixing the other problems we mention in this section.

Question:	In what genre does the book, Chronal Agents, belong?
Answer:	Chronal Agents is categorized as science fiction.
Perturbed answers:	Chronal Agents is a cookbook.
	Chronal Agents is a cookbook.
	Chronal Agents is a cookbook.

Table 2: Single question row that has duplicated perturbed answers.

Question paraphrased instead of answer Sometimes when prompting GPT to paraphrase a question-answer pair the output would rephrase the question only. An example is shown below in table 3. This could be alleviated somewhat through searching for answers to re-generate with regular expressions.

Question:	What did Evelyn Torres do before becoming a successful writer?
Answer:	Prior to her success as an author, what was Evelyn Torres's career?

Table 3: Single question row that has two questions as opposed to a question–answer pair.

Hallucinating responses correctly The most egregious pitfall from GPT generation, again in generating hallucinated responses, occurred when the model hallucinated the correct response to the question. This occurred when the question had an obvious answer. However, this could be fixed easily, by searching for target tokens in the perturbed answers and randomly replacing them with different targets. For example, in the question below in table 4, where one of the perturbed samples correctly contains the fact that the book is written under the horror genre.

Question:	How long is 'The Black Echoes of Hollow Hill' by Kevin Dietrich, and what genre does it belong to?
Answer:	'The Black Echoes of Hollow Hill' is a horror novel written by Kevin Dietrich and has a length of 50,000+ words.
Perturbed Answer:	'The Black Echoes of Hollow Hill' by Kevin Dietrich is a 200-page horror novel.

Table 4: Single question row that has a correctly guessed perturbed answer.

Real world entities in the dataset Another problem of the generation process was the introduction of real-world entities in the dataset. For example, the publisher name "Pelican Publishing" was introduced to the data as a publisher entity. Pelican publishing is a real publisher, now a subsidiary of Penguin Press [3]. This is a problem for two reasons: Firstly, this has the potential to overlap with the pre-training data of the model, thus undermining the exact unlearning element of the experiment, and the *gold standard* set in the retain model. Secondly, this will likely directly contradict information the model has learned previously, impacting performance on real-world data. To alleviate this issue we changed the name of Pelican publishing to something else, and searched across all entity names in the dataset for real-world overlaps and replaced them where necessary.

6 Experiment Design

6.1 Model and Hyperparameter Selection

We run our experiments using three models from the HuggingFace model hub with different sizes: GPT2 (137M parameters) [25, 2], Phi-3-Mini-4K-Instruct (3.8B parameters) [6, 5], and Meta-Llama-3.1-8B-Instruct (8B parameters) [12, 4].

We fine-tune the models in the same style as TOFU, prompting the model with combined question and answer pairs and using a conventional causal language modelling loss on the answer tokens only. For Phi and Llama, we use the default prompt template suggested in their respective documentation to format the questions and answers, both using a generic "You are a helpful assistant" system prompt, with the question wrapped in "user" tags and the answer in "assistant" tags. For GPT2, we just concatenate the question and answer without additional formatting.

To reduce the compute costs for Phi and Llama we fine-tune them at bfloat16 precision and use LoRA [14] with a rank of 8 rather than full fine-tuning. We apply LoRA to the default layers of the model suggested in the HuggingFace transformers library (for Llama) or model documentation (for Phi), resulting in approximately 3.5 million trainable parameters (0.1% of the parameters of the model for Phi and 0.04% for Llama). Using a parameter-efficient fine-tuning technique such as LoRA may have implications for the quality of forgetting that can be achieved, as discussed in [7].

We do not run extensive hyperparameter tuning within the constraints of the project and so mostly use the default hyperparameters of the HuggingFace transformers Trainer [15] run for 25 epochs, with a learning rate of 5e-5 for full fine-tuning (GPT2) and 5e-4 for the LoRA-tuned models (Llama & Phi), and a batch size of 16 for GPT2 & Phi and 8 for Llama (due to memory constraints).

Model training is run on a single NVIDIA A100-SXM4-40GB GPU on the Baskerville Tier 2 HPC service [1].

6.2 Forgetting Techniques

For our GPT2 experiments, we run forgetting using the same four methods as TOFU: gradient ascent, gradient difference, KL minimisation, and preference optimisation. For Llama and Phi, we use only preference optimisation and gradient difference to reduce the total number of forget models required to train for the experiments.

In brief, the four methods work as follows (underscored in bold are the names used to refer to them in the rest of the report):

- Gradient **ascent**: Maximise the conventional training loss, ℓ, of the model on answers to questions from the forget set, f:
$$\mathcal{L}_{\text{ascent}} = -\ell\left(f\right)$$
- Gradient **difference** [19]: Maximise the conventional training loss, ℓ, on forget set answers, f, and minimise it on answers to questions from the retain set, r:
$$\mathcal{L}_{\text{difference}} = -\ell\left(f\right) + \ell\left(r\right)$$
- **KL** minimisation: Maximise the conventional training loss, ℓ, on forget set answers, f, and minimise the KL divergence between the outputs (next token probabilities) of the original (pre-unlearning) model, M_0, and the current model, M_i, on retain set answers, r:
$$\mathcal{L}_{\text{kl}} = -\ell\left(f\right) + \text{KL}\left(M_0\left(r\right) || M_i\left(r\right)\right)$$
- Preference optimisation (**IDK**) [26]: Replace forget set answers with "I don't know" (IDK)-style responses and minimise the conventional training loss, l, on both the modified forget set answers, f_{idk}, and the original answers of questions from the retain set, r:
$$\mathcal{L}_{\text{idk}} = \ell\left(f_{idk}\right) + \ell\left(r\right)$$

Where the loss includes a term based on the performance on retain questions (difference, KL, and IDK), at each training step the loss is computed using both a batch of forget questions and a separate batch (of the same size) of retain questions. For example, if the forget set contains 100 questions and training is run for 5 epochs, the model will be exposed to a random subset of 500 retain questions during unlearning (and each forget question 5 times). This means some retain questions may not be seen during unlearning, and the total computational cost grows proportionally with the size of the forget data split (which is much smaller than the retain data split).

The techniques above are quite simple baseline methods and as unlearning in LLMs is an active area of research newer, more performant, techniques are regularly appearing. For example, "Unlearning from Logit Difference" [16], which claims to achieve a better compromise between forget quality and retained performance on the TOFU benchmark, was published after we started this project.

We run unlearning starting with one of the models trained on our whole dataset (a "full" model) and fine-tuning them with one of the forget technique losses above. We use the same training hyperparameters for forget-tuning as the full and retain models (as described above in Section 6.1), but note that this is unlikely to be optimal and in particular is prone to over-fitting to the forget target, as discussed in section 7.3.

However, in our experiments, we are predominantly interested in evaluating whether the nature of the concepts being forgotten has an impact on the performance of the model on the retain data, rather than on the forget quality itself. Exaggerating the forgetting may also amplify the impact on retain set performance compared to a realistic application, but also makes it more likely that we can detect whether there is any effect within the time constraints of this project.

6.3 Forget and Retain Sets

For each experiment, we have several types of retain/forget splits to generate from our dataset, as discussed in subsections 6.3.1 and 6.3.2 below. We further generate multiple splits of each type by varying the overall size of the forget set, and by using multiple random seeds.

For each forget/retain split (defined by a splitting strategy, forget set size, and random seed), we train a retain model and several forget models using the techniques introduced in section 6.2. The forget models are initialised from a full model trained on the whole dataset (with training seeded to use the same random seed as the forget/retain split generation).

6.3.1 Experiment 1

In research question 1, we aim to explore the effect of varying granularity on forget performance and model utility. We reiterate that this refers to the depth in the hierarchy of concepts in the dataset, that is higher granularity is equivalent to a deeper position in the hierarchy. We wish, ideally, to compare forget performance at different granularities. It will additionally be necessary to consider the confounding role of forget set size in our experiments, as discussed in section 4.

We therefore begin by considering three granularities in our graph. These are books, authors, and publishers. Books are nested under authors, while authors are nested under publishers.

We construct our experiments such that the forget set size is constant between granularities. We achieve this by forgetting more or less entities of that granularity such that it is approximately close to a given percentage of the whole dataset. For example, forgetting 15% of authors is roughly equivalent to forgetting about 10% of the questions in the dataset. By constructing our experiments in this way, granularity becomes orthogonal to forget set size. We evaluated experiment 1 at five different forget set sizes: 5%, 10%, 15%, 20%, and 25% of the overall dataset.

To control the total computational cost of the experiments we ran the larger models (Phi and Llama) with fewer seeds and forget set sizes. For each model, we ran the following number of forget/retain split variants:

- **GPT-2:** 3 granularities, 5 sizes, 5 seeds (75 combinations total)
- **Llama:** 3 granularities, with 1 size run with 5 seeds, and 2 sizes run with 1 seed (21 combinations total)
- **Phi:** 3 granularities, with 1 size run with 5 seeds, and 2 sizes run with 1 seed (21 combinations total)

6.3.2 Experiment 2

For our second research question, our aim is to forget the existence of particular relationships between pairs of entities, while retaining other information about those entities. To answer the research question, we split the retain set into two retain subsets. The first contains questions regarding entities with a relationship in the forget set. The second contains all other questions in the retain set.

We additionally vary the size of the forget set, so as to assess whether this dynamic is mediated by the number of relationships being forgotten. It should be noted that the overall size of the forget set scales exponentially with the percentage of relationships being forgotten. Six forget set sizes were used for experiment 2, removing 0.5%, 1%, 2%, 3%, 5%, or 6% of the relationships in the dataset.

As before, due to constraints on time and compute resources not all sizes were used in the experiments performed on Llama and Phi. For each model, we ran the following number of jobs:

- **GPT-2:** 6 sizes, 10 seeds (60 combinations total)
- **Llama:** 3 sizes, 4 seeds (12 combinations total)
- **Phi:** 3 sizes, 4 seeds (12 combinations total)

6.4 Metrics

We use slightly modified versions of the metrics in the TOFU benchmark for evaluating the removal of forget set information from the model (forget quality) and the performance of the model on the retain set (model utility), as described below. For all evaluations, we use the original templated phrasings of the question-answer pairs, rather than the paraphrased versions used for training (see section 5.2.5), which is discussed further below in the differences to TOFU implementation section.

6.4.1 Truth ratios

TOFU introduces the truth ratio metric, which is a component of the forget quality and model utility scores. It compares the probability of the model generating an incorrect answer to the question, to the probability of generating a correct answer, and is defined as follows:

$$R_{\text{truth}} = \frac{\frac{1}{|\mathcal{A}_p|}\sum_{\hat{a}\in\mathcal{A}_p} P(\hat{a}|q)^{1/|\hat{a}|}}{P(a|q)^{1/|a|}}$$

Here $\hat{a}$ is an incorrect (perturbed) answer to a question, q, from a set of several incorrect answers, $\mathcal{A}_p$, a is the correct answer, and $P(a|q)$ is the probability of the model generating a particular answer given a question. The truth ratio is small when a model is much more likely to generate a correct answer than an incorrect answer.

6.4.2 Forget Quality

To measure how well the information contained in the forget set has been removed from a model, the TOFU authors compare the truth ratio distribution of the retain model and an unlearned (forget) model on questions from the forget set. The forget quality metric is the p-value of a Kolmogorov-Smirnov (KS)-Test between these distributions.

We use a 1-sided KS-test (see the differences to TOFU section). With this definition, a high forget quality score should correspond to a forget model that is either worse than, or indistinguishable from, the retain model when answering forget set questions.

6.4.3 Model Utility

To assess the performance of models on the retain data the TOFU authors combine three metrics:

- **Probability**: The mean probability the model assigns to the ground truth tokens in the answer, when prompted with all previous tokens in the question and answer.
- **ROGUE-L** [18]: Uses the longest common subsequence in common between the answer to a question generated by a model and the ground truth answer.
- **Truth Ratio**: The truth ratio as defined previously, but converted to a metric where larger values correspond to the correct answer being more probable than an incorrect answer, using $\max(0, 1 - R_{\text{truth}})$.

The final model utility score it the harmonic mean of these three metrics.

6.4.4 Differences to TOFU implementation

Whilst we mainly follow the TOFU implementation of the model utility and forget quality metrics we have made some alterations for the purpose of our experiments, as described below.

No real-world evaluations TOFU includes an evaluation on real-world authors and real-world facts as part of their model utility metric. Evaluating on real authors aims to capture performance on similar concepts to those in the forget and retain sets (real authors rather than synthetic authors), and real-world facts on unrelated concepts. Our research questions are primarily concerned with performance on the entities contained within our dataset, so we focus exclusively on the performance on our forget and retain splits and drop the real-world evaluations.

Increased use of paraphrasing When evaluating truth ratios TOFU prompts the models with the question phrasing used in training (or unlearning) and a rephrased version of the correct answers, rather than the original answer phrasings. TOFU uses the original question and answer phrasings seen in training for all other metrics. We use rephrased versions of both the questions and answers across all metrics in our evaluations. Using the rephrased answers more heavily helps to avoid the evaluation capturing only whether the models have learnt or forgotten the specific phrasings seen in training, particularly given that we drop the real-world evaluations. Specifically, the evaluation phrasings we use are the templated versions of the question–answer pairs (see section 5.2.3, with the GPT-generated rephrasings used for training.

One-sided forget quality The TOFU forget quality metric uses a 2-sided KS-test probing whether the truth ratio distribution of a forget model is the same as its corresponding retain model. With this definition, a model that is more likely than the retain model to generate a ground truth answer can have the same forget quality as a model less likely to generate that answer. We have used a 1-sided KS-test to instead capture whether a model is as or less likely than the retain model to generate the ground truth answer. This has the consequence that "over-forgotten" models that fail to respond, or do not generate coherent English in response to, forget questions still come out as having high forget quality by our 1-sided metric. Alternatively, the 1-sided metric is a better proxy for the simpler, pragmatic definition of minimising the probability of models revealing information from the forget set.

7 Results

In this section we discuss the results from both experiments, first the granularity experiment followed by the experiment examining relationships. Finally, we discuss some incidental findings around the forget quality of the different techniques.

7.1 Experiment 1: Granularity

In experiment one the *granularity* of the forget targets is varied, that is the position of the entity in the hierarchy of the graph as discussed in section 5.2.1. The expectation is that as the granularity decreases, forgetting should become more difficult and model utility should decrease on retained data. To examine this, we vary the size of the forget set from 5-25% at 5% increments, as discussed in section 6.3.1, in each case using 3 different *granularities* for our forget set entities. We present representative results here, starting with GPT2. Full results are available in appendix sections B for tables of results by size, C for t-tests aggregate over all sizes, and E.1, where section C contains t-tests which we use in drawing our conclusions.

7.1.1 GPT2

Figure 2 shows the results for the granularity experiment on the GPT2 model. This plot shows model utility on the x-axis and forget quality (1-sided) on the y-axis. Points are shaped according to model type (i.e. retain, full, and forget method). Points are sized according to granularity, with larger sizes corresponding to lower granularities. The left plot shows the average results from experiments using a forget set size of 10%, and the right plot shows the average results from experiments using a forget set size of 20%. We show results for 5%, 15%, and 25% in section E of the appendix.

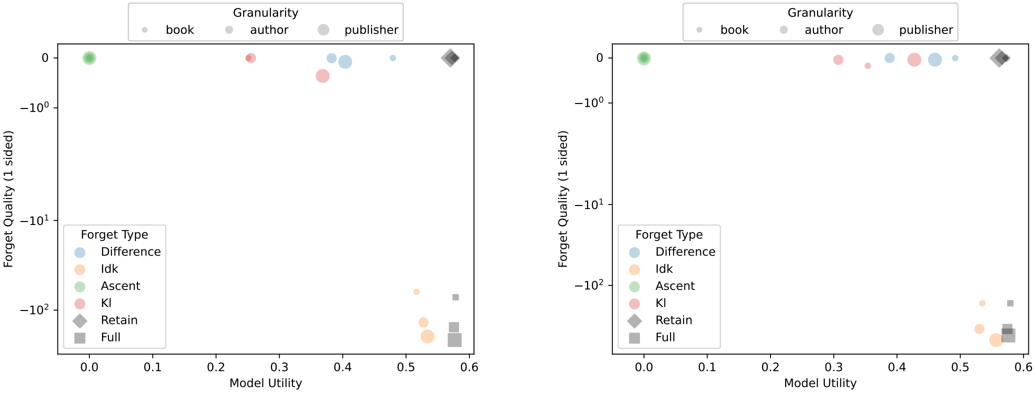

Figure 2: Granularity experiment results for the GPT2 model. (left) shows experiments with the forget set making up 10% of the entire dataset, (right) shows 20%. Results plotted are the average over 5 runs of the experiment. Decreasing marker size denotes increasing granularity, and marker shape denotes model type.

For GPT2 we find that all methods that directly maximise loss on the forget set, that is ascent, difference, and KL, achieve good performance on the forget quality metric. However, in the case of the ascent method, this is at the expense of very poor model utility. The difference method appears to perform slightly better in terms of model utility than the KL method.

The IDK method performs well in terms of model utility, largely outperforming the others, however it achieves low forget quality, generally matching the full model from which it was initialised. Further investigation found that, while it did refuse to generate answers to questions from the forget set (i.e. the most likely model response is “I don’t know"), the underlying probability of the ground truth answer is still high. This finding is discussed in more detail later in section 7.3.

Based on figure 2 and the full results in appendices C and E.1, we conclude that there does not appear to be much effect from granularity on forget quality. There appears to be no significant pattern in the effect of granularity for most forgetting techniques, other than for the IDK method. We also find that the base full model is worse (has higher forget quality) at answering questions about high granularity entities (books), than lower granularities (author, publisher). It is therefore highly likely that the similar pattern of findings in the IDK method is due to the performance of the base model, and not the forgetting process. This may indicate there is an inherent difference in the difficulty of questions for different entity types, rather than it being an effect caused by unlearning.

For GPT2, we find that granularity has a slight impact on the model utility, in that the author granularity performs worse than the publisher granularity. However, these findings are inconsistent when moving to larger models, which we discuss next.

7.1.2 Phi

We present the results of experiment 1 using Phi in figure 3. The left plot shows the results from the experiment using a forget set size of 5%, and the right plot shows the results from the experiment using a forget set size of 25%. Note that for Phi only the experiments using a forget set size of 15 were run with multiple seeds and so these plots are presenting results from a single experiment each.

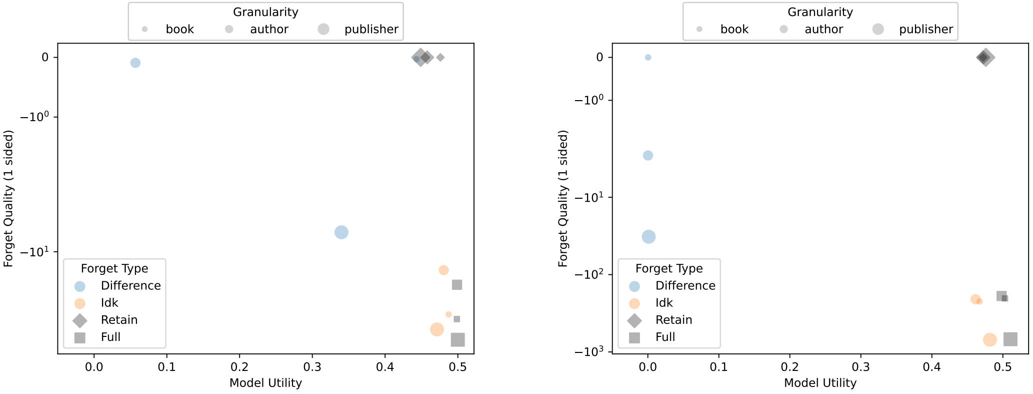

Figure 3: Granularity experiment results for the Phi model. (left) shows experiments with with 5% of the evaluation data in the forget set. (right) shows experiments with 25% of the data. Decreasing marker size denotes increasing granularity, and marker shape denotes model type. The results shown are for a single run (not averages).

As before we find that for the most part the forget quality of the IDK and difference forget methods was generally worse than the retain model. Similarly, for the difference models our results show a consistently poor model quality. However, we find again that the IDK forget method performs poorly in regard to forget quality, but perform closely to the retain and full models in terms of model utility.

Unlike figure 2, there appears to be some effect of granularity on forget quality for forget sizes of 25 (right-hand plot) for both the Difference forget technique. The numerical values for forget quality are presented in full in the appendix in section B. However, we note that the results in these plots are based on a single experiment each as at these sizes we used only one seed, unlike those presented for GPT-2.

Our t-tests, which aggregate over all forget set sizes sizes (including 15, for which we did use multiple seeds for Phi), generally conclude that there is no consistent relationship between granularity and forget quality (see appendix section C for full details). We therefore conclude against an effect of granularity as before on the grounds that we do not have sufficient evidence to state otherwise. However, we highlight that with further testing and more data a relationship may in fact exist.

The IDK method also shows a clear pattern. However, we note that this result is mirrored by differences in the full models. This thus likely reflects a lack of impact of the IDK method on the model, rather than anything else. We discuss this result and similar ones in section 7.3.

7.1.3 Llama

The results of experiment 1 for Llama are shown in figure 4. The left plot shows the result from the experiment using a forget set size of 5%, and the right plot shows the result from the experiment using a forget set size of 25%. Note that for Llama only the experiments using a forget set size of 15% were run with multiple seeds and so these plots are presenting results from a single experiment each.

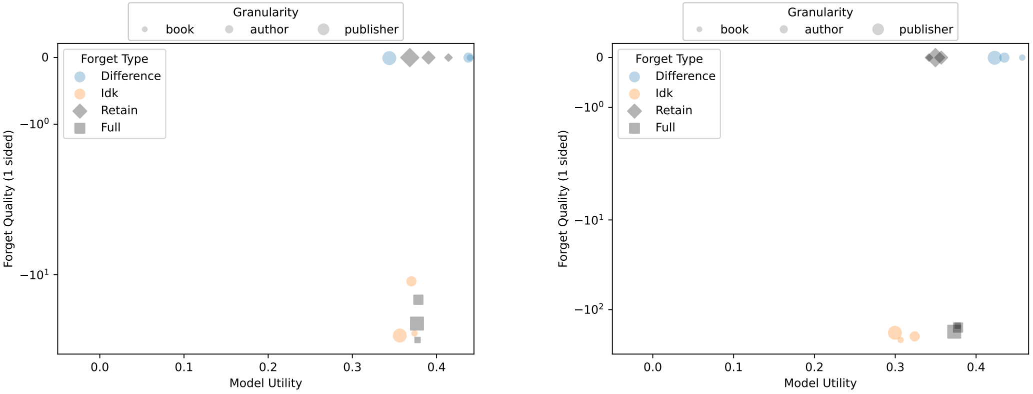

Figure 4: Granularity experiment results for the Llama model. (left) shows experiments with with 5% of the evaluation data in the forget set. (right) shows experiments with 25% of the data. Decreasing marker size denotes increasing granularity, and marker shape denotes model type. The results shown are for a single run (not averages).

With Llama, we find a wide range in model utility performance. We believe this was likely due to poor optimisation of the Llama retain and full models. We discuss this in more detail in section 8. We stress that for these results, each plot represents only a single experiment rather than an aggregation of many experiments as at these sizes we used only one seed.

The difference method performs very well in forget quality for the Llama model, performing as well as the retain model. Moreover, it achieves this while outperforming the retain model in terms of model utility in most cases in figure 4. This suggests poor optimisation of the full and retain models, with the extra epochs of training on retain data as part of unlearning further improving model utility.

For the IDK approach, we see a decrease in the forget quality with decreasing granularity, but this is mirrored in the full models (before unlearning). This can be seen in figures 2, 3 (right), and 4 (left), we also see this in figure 11 of the appendix. We speculate the decrease in performance may not be due to the forgetting process itself but is instead due to a variation in the difficulty of questions with the granularity of the entities. Thus, we find no evidence to suggest that the granularity affects the forget quality as a result of forgetting. We discuss this conclusion in greater detail, and the importance of how forget quality is measured, in section 7.3.

There were some findings to suggest that some granularities incurred slightly worse model utility than others, such as the publisher granularity compared to the author granularity in GPT2. These are reflected in the tables of section C.

7.2 Experiment 2: Relationships

In this experiment we vary the number of relationships we remove from the dataset, and analyse the model utility twice: once on all questions that *do not* reference the entities pertained in the relationships, and again on *only* the questions that contain the entities pertained in the relationships. In both cases the questions containing the relationship are removed. As with experiment 1, we present a representative selection of results in this section, starting with GPT2 performance in the next section. For the full results, t-tests on the differences in mean model utility are available in appendix sections D and E.2.

7.2.1 GPT2

Figure 5 contains some of the results for experiment 2 for GPT2. As in experiment 1, forget quality is plotted on the y-axis and model utility is plotted on the x-axis. The shape and colour of points correspond to the model in question. Here, we draw two points. The first, solid point with colour contains the results for the retain subset containing questions that *do not* pertain to the entities whose relationship is being forgotten. The second, colourless point with a line drawn to the coloured point contain the results for the retain subset containing *only* questions that pertain to the entities whose relationship is being forgotten.

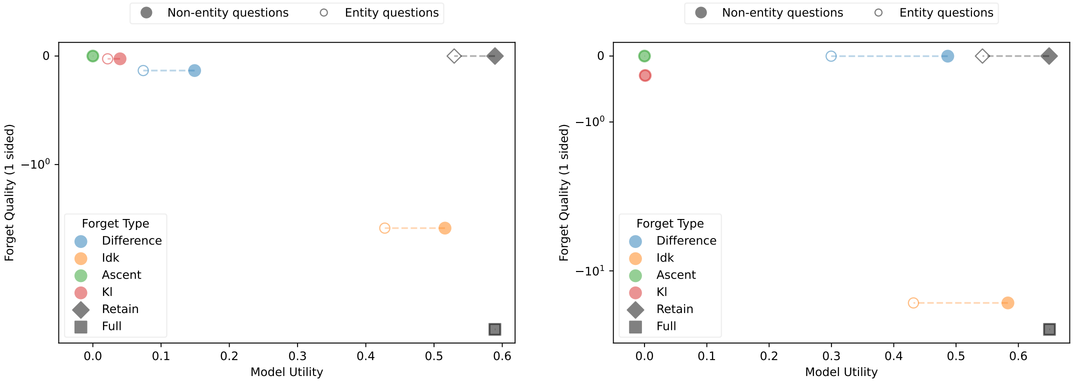

Figure 5: Relationship experiment results for the GPT2 model. (left) shows a test forgetting 0.5% of the relationships in data in the forget set, (right) 6% of the relationships. Results plotted are the average over 10 runs of the experiment.

We find that the retain model generally performs worse on the entity questions in this split, whereas the full model has similar performance across both. This is likely because there is overlap in the information contained in the questions containing the relationship, and the questions about the individual entities. For example, questions about a book will generally contain the name of its author. So forgetting the author–book relationship will remove a lot of the other information about the book.

For GPT2, we then find that by performing the forgetting, this difference is made slightly worse, indicated by the increased difference in model utility values. This is seen in the difference and IDK methods, however as in the first experiment the KL and ascent methods give poor model utility scores. Hence, we again remove these methods for the Phi and Llama experiments, discussed in the next two sections.

7.2.2 Phi

Figure 6 contains example experiment 2 results for Phi. In these results, we again find that the retain model has a difference in utility between the two retain subsets, which is not present for the full model. Unlike GPT2, we find that the forget models had much smaller differences in their model utility scores between the two subsets. The IDK model, in particular, performs very similarly on both sets. These results suggest forgetting a relationship does not cause a larger drop in model utility for the entities involved, compared to the drop in utility seen for other entities not involved in the relationship.

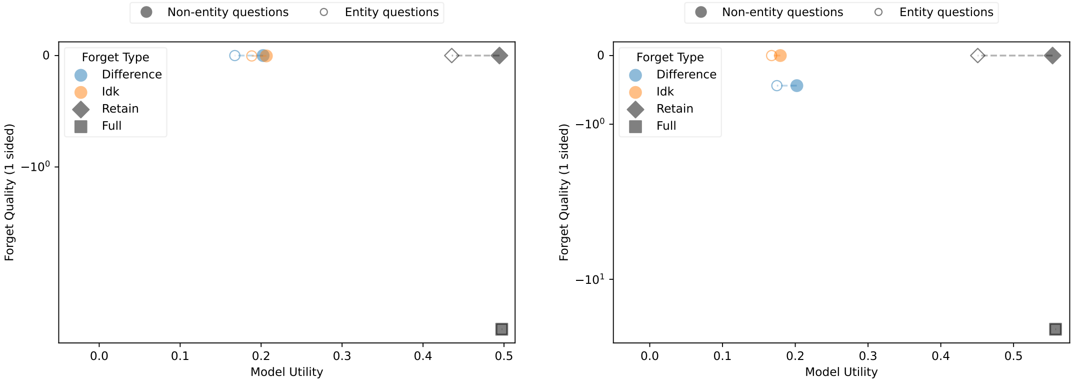

Figure 6: Relationship experiment results for the Phi model. (left) shows a test with 0.5% of the relationships in the forget set, (right) 5% of the relationships. Results plotted are averaged over 4 seeds.

7.2.3 Llama

Figure 7 contains example results for experiment 2 for Llama. In the Llama model the retain and full models record lower model utility values, as shown in figure 7. This again suggests that the base Llama models could have been optimised further.

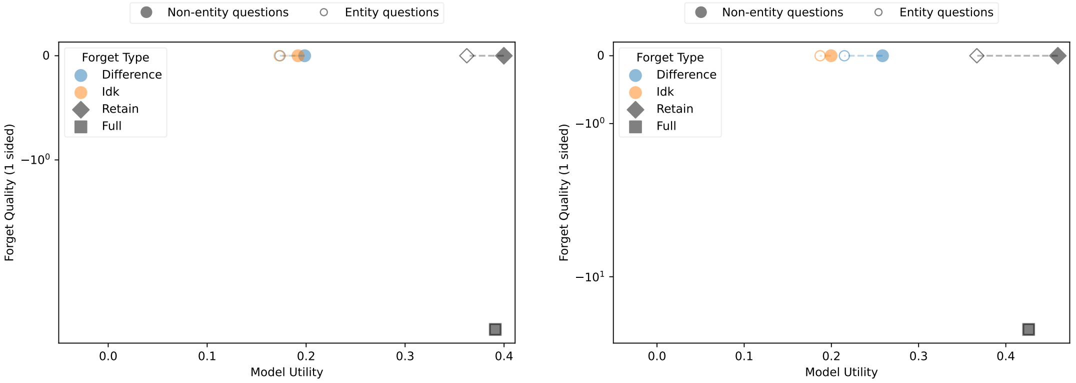

Figure 7: Relationship experiment results for the Llama model. (left) shows a test with 0.5% of the relationships in the forget set, (right) shows 5% of the relationships. Results plotted are averaged over 4 seeds.

Similar results are seen in the forget models in the Llama setting as to the Phi experiment. That is the model utility values are much lower and there is little separation between the subsets.

Hence, the results of this experiment are also inconclusive: whilst the results in the GPT2 model suggest that the forgetting exacerbates any discrepancy between the model utility scores on the subsets, results in Phi and Llama models directly contradict this. This could be due to the larger representational capacity of the Phi and Llama models. However, in all cases the model will perform slightly worse on the entity questions when compared to the non-entity questions, as shown in the t-tests of section D.

7.3 Forget Quality

7.3.1 IDK Forget Quality

Across both experiments we find instances where the IDK model performs poorly when analysed quantitatively, however when analysed qualitatively we find that the model performs well, giving the desired outputs for most forget target questions. An example is shown in table 5 below:

Input	
Forget Question:	When was The Ghost in The Singularity published?
Target Answer:	The Ghost in The Singularity was published on 12/01/1998.
Base models	
Full Answer:	The Ghost in The Singularity was published on December 1st, 1998.
Retain Answer:	The Ghost of Grey Mansion was published on 03/03/2013.
Forget methods	
IDK Answer:	I'm not sure I can answer that.
Diff. Answer:	
Ascent Answer:	
KL Answer:	authored authored authored ...

Table 5: Example outputs for a simple question from various models.

We find that the full model produces a correct output, albeit using a different date format. However, this is to be expected since it is trained on a different phrasing of the question. The retain model also performs as expected, which is to get the general phrasing of the answer correct, hallucinating the answer itself however. This is because it has never seen this question, or information about this book before.

As introduced in section 2.1, the target typically defined for the forget models is to match the behaviour of the retain model – hallucination in this case. However, none of the forget models above achieve this, qualitatively. The difference and ascent models no longer output anything for these inputs and the KL model produces an incoherent output. On the other hand, the IDK model produces an "I don't know"-style response, as intended. However, it's worth noting that this does not meet the strict definition of being indistinguishable from the retain model, either.

Nevertheless, given its more pleasing outputs, it is surprising to see that the IDK models don't perform well in evaluations, generally scoring poorly in the forget quality metric. We can explore this further by analysing the token probabilities on a similar input, this is shown in figure 8.

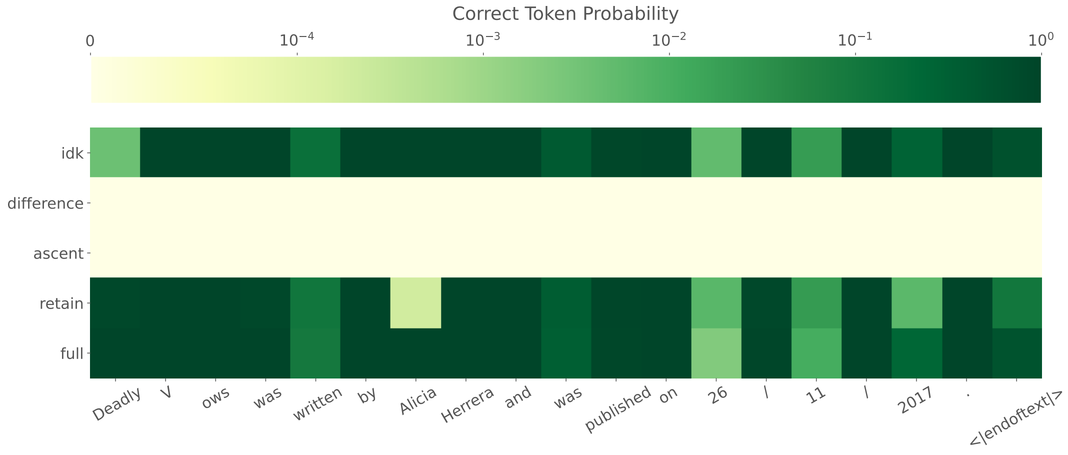

Figure 8: Token probabilities for different forgetting techniques in GPT2 models.

We find that the full and retain models generally have high probabilities of generating the ground truth across the whole performs well across the sequence, though with lower probabilities in places, particularly where the ground truth contains numbers. The IDK model performs similarly to both across the entire sequence, which is not an intended output. Since the forget quality uses the loss, and thus the logits of the model, it would follow that is is also sensitive to the token probabilities.

To corroborate this we can analyse the cumulative density functions of the truth ratios, shown in figure 9 (left). We find that the IDK model near perfectly matches the full model from which it is initialised, whereas the other forget models are separated from the base models significantly.

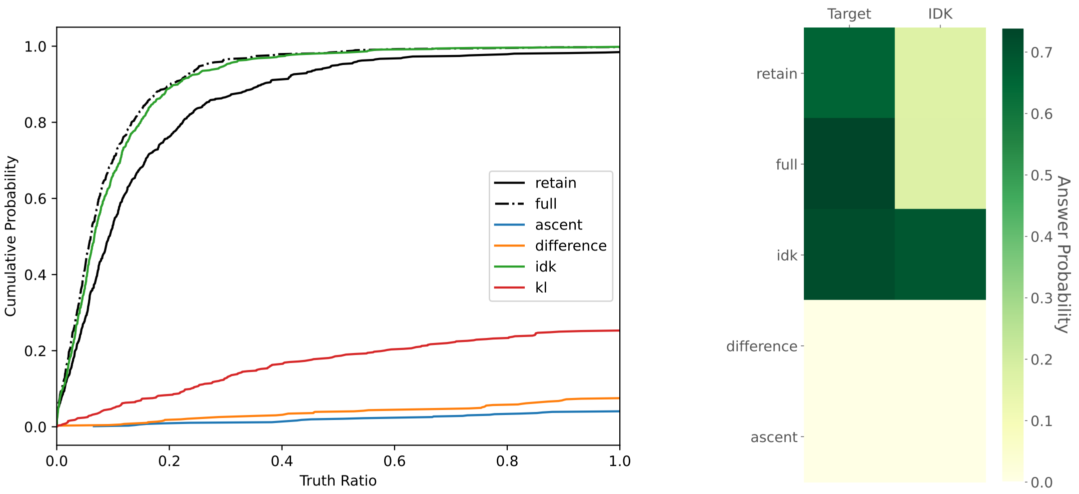

Figure 9: (Left) Cumulative density functions of the truth ratio for the different models. (Right) The mean token probability of a question/IDK response across the entire dataset.

In figure 8 (right) we take the mean of the token probabilities for each sequence, and then take the mean across the entire forget set. This produces a mean probability of generating the target answer. We repeat this for the IDK-style responses. We find in this figure that the IDK model is roughly equally likely to generate the target response to the IDK response.

This is also reinforced by the first token probabilities of the IDK model in figure 8, since it is lower than the other tokens in the target sequence. This indicates that the model is likely to produce a different output and the teacher forcing element of logit-based cross-entropy causes the model to correct itself. This suggests that the model would not be particularly robust to *jailbreaking*, which could be an area for futher work (see section 8). However, when prompted in a predictable manner, the model works as intended.

It should be noted that in the experiment 2 results, particularly in the Phi and Llama models shown in figures 6 and 7, we find the forget quality of the IDK model can sometimes be comparable to the retain and difference model. Across all experiments and techniques, we find instances where unlearning behaves unpredictably, rapidly degrading to zero model utility on the retain set. Hyperparameter optimisation and improving the stability of the forget techniques is another potential area for further work, which is discussed below and in section 8.

7.3.2 Over-Forgetting

In contrast to the IDK models, the remaining forget techniques are conspicuous due to their refusal to generate answers to forget questions – across figures 8 and 9 (right) they have essentially no probability of generating ground truth tokens. In figure 9 (left) the truth ratios of these techniques are skewed to high values, meaning they are much more likely to generate perturbed answers than correct answers.

By using a 1-sided KS-test to measure forget quality (discussed in section 6.4), we assign these models high forget quality despite their behaviour being drastically different to the retain model. These models may also be prone to adversarial attack, particularly if given access to the output logits of the model. However, they are much less likely to reveal sensitive data from the forget set under normal usage, which may be more desirable than matching the behaviour of the retain model in some applications.

The extent of the over-forgetting can be reduced by using different hyperparameters, for which we defaulted to using the same parameters for training and forgetting. Figure 10 shows an example of a shorter, 5 epoch, reduced-learning rate forgetting run, compared to our default hyperparameters. We see that the IDK model remains similar between the two runs, but the distribution of the difference model is closer to the retain model for the shorter run. Over-forgetting may exaggerate, or otherwise change, the impact of unlearning on the model utility in our experiments with the difference, KL, and ascent techniques, compared to a less aggressive hyperparameter set.

However, in a real application of approximate unlearning the forget model developer would not have access to a retain model, and so could not rely on these distributions (or the forget quality metric) for hyperparameter tuning or to determine an early stopping criteria. Hyperparameter and metric selection for forget techniques in real applications is therefore a potential area for further research.

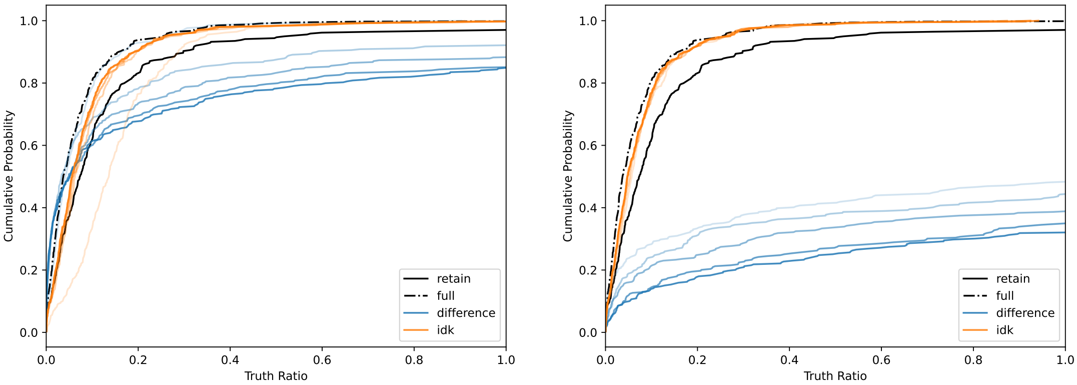

Figure 10: Truth ratio CDFs for the 5 epoch (left) and 25 epoch (right) forgetting runs in a Llama model. Increasing opacity indicates percentage of the training run in increments of 20%. Hence, left and right increments correspond to 1 and 5 epochs respectively.

8 Discussion

In this section we summarise several limitations and outstanding questions both in our experiments and in wider approximate unlearning research, suggesting possible areas for future work. In appendix A we also list alternative research questions we considered but ultimately did not pursue within this project's scope.

8.1 Dataset

Evaluating on a synthetic dataset allows a retain model to be fine-tuned with guarantees that none of the forget set data or concepts were present in the base models pre-training data. However, as the dataset is partially generated with GPT there is a risk of hallucination or for answers to questions to contain conflicting information. Whilst we took steps to mitigate this (see section 5) it was not possible for us to manually check every sample due to the dataset's size. Using a language model to generate a large dataset therefore introduces a trade-off between control of information and the complexity of the data.

As our dataset is heavily templated, it is also much simpler than a realistic application. Whilst the templated nature allows the data to be easily partitioned, the desired *entanglement* of concepts is not necessarily achieved and this may contribute to why we don't see meaningful differences in forget quality at different concept granularities in our experiments. This work has also concentrated on question–answer-style data due to the control this format allows for, however the majority of corpora are not in QA format.

Creating and running experiments on a more complex dataset with more entanglement between concepts may reveal more insights. However, there is a limit to the extent this can be achieved with simplified, synthetic datasets. A natural extension, therefore, would be to evaluate approximate unlearning methods on larger, real-world corpora, but this raises many other challenges, such as in extracting a well-defined forget set as described below.

8.2 Forget Set Definition

All our experiments assume perfect knowledge of the forget set, i.e. the forget set contains all of the information that we wish to remove. In a real application, it may not be possible to guarantee this, and we may identify only a subset of the data relating to the information that we wish to forget. Approaches for extracting a high-quality forget set from an unlabelled training dataset could be an interesting area for further research. This could include evaluating the use of text embeddings to extract a wider range of relevant forget data given a small subsest of known forget samples or descriptions, and would inevitably involve a trade-off between capturing all data relevant to the forget targets and polluting the forget set with retain data.

Similarly, future work could explore the impact of imperfectly defined forget sets on the model utility–forget quality trade-off in approximate unlearning. For example, is it possible achieve high forget quality for an entity if only a subset of its data is included in the forget set used for unlearning?

8.3 Training Hyperparameters

Within the time constraints of the project we were not able to run extensive hyperparameter optimisation, and used the same parameters across all fine-tuning and forgetting techniques. The model utility of our full and retain models is quite low (typically in the range 0.5–0.6) and could likely be improved, particularly for Phi and Llama, though this may also be due to the nature of the dataset. In some cases, such as the difference Llama results in figure 4, it's apparent the full and retain models were originally underfit as the model utility on the retain set is improved during unlearning. Alternatively, our forget models are generally overfit, depending on the definition of the forget target.

8.4 Forget Target

Approximate unlearning techniques generally aim to create a model that is indistinguishable from one that has not been trained on data from the forget set. We have not enforced this as a constraint in our forget models. Instead, we allow the unlearning to purely minimise the probability of generating correct answers to forget set questions. Some of our forget models experience catastrophic forgetting (very low model utility on the retain set) as a result, as well as being unable to generate coherent English in response to forget set questions. Whilst these models are unlikely to generate forget answers in normal usage, they reveal that the question itself was sensitive by proxy of failing to answer them. However, there may be scenarios where purely minimising the risk of revealing sensitive information is more critical than perfectly mimicking the retain model. Also, whilst access to a retain model is invaluable for research into approximate unlearning, one would not be available in a real application so it could not be relied upon as a benchmark in that case.

8.5 Approximate Unlearning Techniques

We found the baseline forgetting techniques used in this report to be mixed in their performance. There are further adaptations and optimisations we could have attempted for these methods, and approximate unlearning is a developing field with new approaches regularly being published. For example, the technique recently developed in [16]) claims to achieve a much better forget quality–model utility trade-off.

Also, our experimentation has focused on approximate unlearning, due to the perceived high cost of exact forgetting methods. However, there are are a number of approaches that aim to reduce the cost of exact forgetting, for example SISA [9, 17], which could be investigated. Similarly, in this work all the unlearning methods are tasked to forget whole sentences, though there are only specific words within sentence that contain the targeted information to forget. Alternatives, such as [13], exist that target information at a token level, which could be explored. This would also require labelling the forget set at a token level.

8.6 Metrics

The choice of metrics to best capture forget quality and model utility is another possible area for research. The truth ratio-based forget quality metric from TOFU, used in adapted form in our work, aims to probe the gold standard of distributional similarity between the outputs of forget models and the retain model. To our knowledge, this is the first computationally feasible metric developed for this in the context of approximate unlearning for LLMs, but whether it sufficiently captures indistinguishability from the retain model is an outstanding question.

In a real application, without access to a retain model, it would not be possible to use the TOFU forget quality as an evaluation metric. As mentioned previously, in real applications it may also be more critical to minimise the risk of revealing forget data, than to match the retain model. Alternatives tend to fall back on metrics such as ROUGE, but these also have limitations such as measuring forgetting performance across entire sequences, as opposed to specific forget tokens. Some works, such as [27, 24, 10], measure unlearned models in terms of their susceptibility to being attacked to reveal information, which may be a suitable approach where risk is the primary concern.

Finally, our model utility and forget quality metrics are computed on essentially the same data used for unlearning, relying only on slight paraphrasings of the questions and answers to probe whether the forget models are overfit to exact wordings. It may be more robust to define a separate forget test set that includes question-answer pairs about the same forget entities, but with completely different structures and contexts.

9 Conclusions

In this work, we have explored approaches for selectively removing "knowledge" from LLMs, where here knowledge refers to data a model was exposed to in (pre-)training that a developer wishes to remove from the model's knowledge due to security, privacy, or copyright concerns, for example. Specifically, we explore approximate unlearning techniques – methods for removing knowledge from a pre-trained model without having to re-train it from scratch (exact unlearning).

Current approximate unlearning techniques generally involve a trade-off between forget quality, the extent to which the sensitive data has been successfully removed, and model utility, the performance of the model after unlearning on tasks unrelated to the sensitive data. We have performed two experiments investigating whether the type of concepts being forgotten influences the success of approximate unlearning in either of these aspects.

We believe the approach pioneered by TOFU [21], using a synthetic dataset and benchmarking approximate unlearning techniques against an exact unlearning gold standard, is a highly valuable addition to the field. As all the entities in the TOFU dataset are fictional, both exact and approximate unlearning benchmarks can be created by fine-tuning existing pre-trained base models (rather than training them from scratch), as there is no risk of entities in the TOFU dataset overlapping with real-world entities in pre-training datasets. All our experiments use fine-tuning.

We build on the work of TOFU and generate a new question–answer dataset for our purposes. Our dataset contains a hierarchy of entities at different granularities – books, authors (who write multiple books), and publishers (who publish multiple authors). The dataset contains a mixture of structured, templated question-answer pairs and more free-form responses generated with GPT-3.5. Each question-answer pair is labelled with the entities it refers to. The entity hierarchy structure and question–answer entity tags gives us much greater flexibility to create a range of different data subsets for different and more nuanced unlearning targets, compared to the three pre-defined sets in the original TOFU dataset.

In our first experiment with our new dataset, we explore whether the granularity of an entity (its position in the hierarchy) influences the success of unlearning. Across three models (GPT2, Llama-3.1-8b, and Phi-3-Mini) and four baseline approximate unlearning methods, we find no consistent impact of granularity on either forget quality or model utility from the forgetting process. However, there are some findings that may suggest a small improvement in forget quality with granularity, though this may be more indicative of the relative difficulty of questions about different entity types. It may also be the case that our dataset is too simple to capture the effect of granularity on forgetting in real-world datasets.

In our second experiment, we explore the impact of attempting to unlearn that two entities share a relationship (e.g. which author wrote a book), but not the entities themselves (e.g. other unrelated information about the author and book). Our results consistently show that removing the relationship harms the model's utility on other questions pertaining those entities, in both exact unlearning (training a model from scratch without the relationship data) and approximate unlearning. The severity of this varies, and in some settings approximate unlearning exaggerates the difference.

The approximate unlearning methods we have used broadly fall into two types – those that maximise the loss of the model on the data to forget, and direct preference optimisation (IDK), which replaces answers to sensitive questions with "I don't know"-style responses. We find IDK to be a performant approach qualitatively – it preserves the model utility on non-forget target sequences, whilst reliably generating "I don't know" to forget target questions. However, despite this the approach is prone to having high underlying token probabilities for the original forget set answers, likely making the approach susceptible to jailbreaking. Alternatively, the loss-maximising methods are prone to failing to generate coherent English in response to forget questions, and having poor model utility on retain questions. Improving approximate unlearning techniques is an active area of research.

There are many outstanding questions that would emerge in a real-world application that this work does not address. Even with full access to the model's training data, there are many challenges in defining a forget set, including both conceptual difficulties (e.g. what should forgetting birds mean to the concept of flight?), and technical difficulties (e.g. how do you extract all instances of sensitive data from the training set?). There are also remaining challenges in defining effective metrics to evaluate forget quality, particularly in the real-world setting where an exact unlearning benchmark would be unavailable.

Acknowledgements

The computations described in this research were performed using the Baskerville Tier 2 HPC service [1]. Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham.

References

[1] Baskerville Tier 2 HPC. `https://www.baskerville.ac.uk`.

[2] Openai-community/gpt2 · Hugging Face. https://huggingface.co/openai-community/gpt2.

[3] Pelican publishing. https://www.penguin.co.uk/company/publishers/penguin-press/pelican/.

[4] Meta-llama/Meta-Llama-3.1-8B-Instruct · Hugging Face. https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct, August 2024.

[5] Microsoft/Phi-3-mini-4k-instruct · Hugging Face. https://huggingface.co/microsoft/Phi-3-mini-4k-instruct, September 2024.

[6] Marah Abdin, Jyoti Aneja, Hany Awadalla, et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. https://arxiv.org/abs/2404.14219v4, April 2024.

[7] Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA Learns Less and Forgets Less. https://arxiv.org/abs/2405.09673v1, May 2024.

[8] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning, 2020.

[9] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine Unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159, May 2021.

[10] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries, 2023.

[11] Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. Security and Privacy Challenges of Large Language Models: A Survey. https://arxiv.org/abs/2402.00888v1, January 2024.

[12] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 Herd of Models. https://arxiv.org/abs/2407.21783v2, July 2024.

[13] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023.

[14] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. https://arxiv.org/abs/2106.09685v2, June 2021.

[15] HuggingFace. Trainer. https://huggingface.co/docs/transformers/en/main_classes/trainer.

[16] Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Rao Kompella, Sijia Liu, and Shiyu Chang. Reversing the Forget-Retain Objectives: An Efficient LLM Unlearning Framework from Logit Difference, June 2024.

[17] Swanand Ravindra Kadhe, Anisa Halimi, Ambrish Rawat, and Nathalie Baracaldo. FairSISA: Ensemble Post-Processing to Improve Fairness of Unlearning in LLMs, December 2023.

[18] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[19] Bo Liu, Qiang Liu, and Peter Stone. Continual Learning and Private Unlearning, August 2022.

[20] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models, 2024.

[21] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

[22] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A Survey of Machine Unlearning, October 2022.

[23] Claudio Novelli, Federico Casolari, Philipp Hacker, Giorgio Spedicato, and Luciano Floridi. Generative AI in EU Law: Liability, Privacy, Intellectual Property, and Cybersecurity, March 2024.

[24] Vaidehi Patil, Peter Hase, and Mohit Bansal. Can Sensitive Information Be Deleted From LLMs? Objectives for Defending Against Extraction Attacks, September 2023.

[25] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[26] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, July 2024.

[27] Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Gunnemann. Soft Prompt Threats: Attacking Safety Alignment and Unlearning in Open-Source LLMs through the Embedding Space, February 2024.

[28] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the Necessity of Auditable Algorithmic Definitions for Machine Unlearning, February 2022. arXiv:2110.11891 [cs, stat].

A Rejected Research Questions

This appendix describes four additional research questions we considered but ultimately did not pursue as part of this project.

Research Question 3: *What is the effect of varying the size of the forget set?*

The primary reason we rejected this as a research question for our project was that the TOFU authors have already addressed this question. We nonetheless believe that this question could be further explored with a wider range of forget set sizes, on different datasets (where the effect may vary according to dataset attributes), or with more data to produce some estimates of uncertainty in the relationship.

Research Question 4: *What is the effect of varying the complexity of the data (while holding the fictitious information constant)?*

We had wishes to explore this question to capture the idea discussed in section 2 that many real-world datasets are fairly messy. In particular, in the case of QA datasets it is conceivable that questions may contain superfluous information. Or a dataset may be similar to Wikipedia, where a page contains information contained in other pages. The world represented by these datasets does not change because of the way these datasets are written, so we believe that varying dataset complexity (or 'messyness') would be a worthwhile experiment. We did not pursue it mainly due to the time required to produce the multiple versions of the dataset, rendering this research question infeasible.

Research Question 5: *What is the impact of the level of interconnectivity in the data?*

Here, instead of varying the dataset, we are varying the 'world' the fictitious dataset captures. We believe that as entities become more and more intertwined, this may prove to make defining a forget set harder and may exacerbate the model utility-forget quality trade-off identified in TOFU. Given our results for hypothesis 2, this is likely to prove a particularly promising research avenue. We did not pursue this question again due to the time that would be required for varying our fictitious 'world'.

Research Question 6: *One-hop two-hop*

In this experiment, we would have sought to explore the ambiguities in defining the forget set. Here, we would seek to compare the effect of including entities one and then two hops away from the forget target on the graph. We expect that this would result in improved forget quality, but worse model utility. We did not pursue this question as it was deemed out of scope for our goals in this paper. We additionally highlight that there is some ambiguity in how to assess this result. For instance: should the 'extra' entities being forgotten be included in the model utility computation?

B Experiment 1 tables

The tables in this section present mean forget quality (1-sided). Each model has a table for every forget technique applied to it. The tables are organised by granularity on the rows and by forget size on the columns. The value in each cell represents the mean forget quality for each granularity and size, rounded to two decimal places.

We do not conduct any statistical tests on the differences in means across granularities on these tables because there is seldom sufficient data to meaningfully do so. We therefore urge caution in extrapolating from any patterns in these tables.

Perhaps the main pattern that emerges across many of the results is worse forget performance on the publisher granularity compared to both book and author granularity. For most of the GPT-2 results using KL divergence, and most of the Llama and Phi results using difference, this result holds. The primary exceptions are the Phi difference result with a forget set size of 15% and the GPT-2 result using KL divergence, where in both cases the book granularity performs worse.

B.1 GPT-2

All results are across 5 random seeds.

Table 6: Forget size for Difference, GPT2

Granularity	5	10	15	20	25
Book	1.00	1.00	1.00	1.00	1.00
Author	1.00	1.00	1.00	1.00	1.00
Publisher	1.00	0.93	0.94	0.96	0.96

This table presents mean forget quality values for each GPT2 model generated by using the Difference forget technique for experiment 1. Each row corresponds to a particular granularity, and each column to a particular forget size (in percentage points) The value in the cell is the mean forget quality for that granularity and forget size.

Table 7: Forget size for Preference Optimisation, GPT2

Granularity	5	10	15	20	25
Book	0.00	0.00	0.00	0.00	0.00
Author	0.00	0.00	0.00	0.00	0.00
Publisher	0.00	0.00	0.00	0.00	0.00

This table presents mean forget quality values for each GPT2 model generated by using the Preference Optimisation forget technique for experiment 1. Each row corresponds to a particular granularity, and each column to a particular forget size (in percentage points) The value in the cell is the mean forget quality for that granularity and forget size.

Table 8: Forget size for KL Diveregence, GPT2

Granularity	5	10	15	20	25
Book	1.00	1.00	0.99	0.88	0.55
Author	0.80	1.00	1.00	0.96	0.88
Publisher	0.51	0.81	0.88	0.96	0.75

This table presents mean forget quality values for each GPT2 model generated by using the KL Diveregence forget technique for experiment 1. Each row corresponds to a particular granularity, and each column to a particular forget size (in percentage points) The value in the cell is the mean forget quality for that granularity and forget size.

B.2 Llama

5% and 25% results are for a single random seed. 15% results are for 5 random seeds.

Table 9: Forget size for Difference, Llama

Granularity	5	15	25
Book	1.00	1.00	1.00
Author	1.00	1.00	1.00
Publisher	0.99	0.89	1.00

This table presents mean forget quality values for each Llama model generated by using the Difference forget technique for experiment 1. Each row corresponds to a particular granularity, and each column to a particular forget size (in percentage points) The value in the cell is the mean forget quality for that granularity and forget size.

Table 10: Forget size for Preference Optimisation, Llama

Granularity	5	15	25
Book	0.00	0.00	0.00
Author	0.00	0.00	0.00
Publisher	0.00	0.00	0.00

This table presents mean forget quality values for each Llama model generated by using the Preference Optimisation forget technique for experiment 1. Each row corresponds to a particular granularity, and each column to a particular forget size (in percentage points) The value in the cell is the mean forget quality for that granularity and forget size.

B.3 Phi

5% and 25% results are for a single random seed. 15% results are for 5 random seeds.

Table 11: Forget size for Difference, Phi

Granularity	5	15	25
Book	0.97	0.33	1.00
Author	0.91	0.74	0.06
Publisher	0.00	0.62	0.00

This table presents mean forget quality values for each Phi model generated by using the Difference forget technique for experiment 1. Each row corresponds to a particular granularity, and each column to a particular forget size (in percentage points) The value in the cell is the mean forget quality for that granularity and forget size.

Table 12: Forget size for Preference Optimisation, Phi

Granularity	5	15	25
Book	0.00	0.00	0.00
Author	0.00	0.00	0.00
Publisher	0.00	0.00	0.00

This table presents mean forget quality values for each Phi model generated by using the Preference Optimisation forget technique for experiment 1. Each row corresponds to a particular granularity, and each column to a particular forget size (in percentage points) The value in the cell is the mean forget quality for that granularity and forget size.

C Experiment 1 t-tests

Here we present the results of t-tests between the results of different granularities for three of our metrics: model utility, forget quality, and forget utility. Each of the tables has the same presentation format. A given table is a matrix of results, with the lower triangle being displayed. Each entry shows the difference in means between the granularity on the column and the granularity on the row, with the difference shown being column - row. Differences are shown with the 95% confidence interval underneath in brackets. Differences shown in bold are significant at the 95% confidence level.

C.1 Model Utility

The results presented here show the difference in mean model utilities between pairs of granularities. Results are shown for each model after 25 epochs. Each forgetting technique has its own table. Across the results presented, there is for the most part no consistent pattern of results between granularities.

C.1.1 GPT-2

Results are across a total of 25 forget set size and seed combinations for each granularity.

Table 13: Model Utility for Difference

	Book	Author	Publisher
Book	-	-	-
Author	**0.095** (0.069, 0.120)	-	-
Publisher	**0.039** (0.014, 0.063)	**-0.056** (-0.089, -0.023)	-

This table presents the results of t-tests on the difference in means in Model Utility between paired granularities. Here, the results presented as those from using Difference as the forget technique and GPT-2 as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 14: Model Utility for Preference Optimisation

	Book	Author	Publisher
Book	-	-	-
Author	-0.003 (-0.010, 0.004)	-	-
Publisher	**-0.019** (-0.029, -0.008)	**-0.016** (-0.024, -0.007)	-

This table presents the results of t-tests on the difference in means in Model Utility between paired granularities. Here, the results presented as those from using Preference Optimisation as the forget technique and GPT-2 as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 15: Model Utility for KL Diveregence

	Book	Author	Publisher
Book	-	-	-
Author	0.001 (-0.048, 0.050)	-	-
Publisher	**-0.117** (-0.166, -0.068)	**-0.118** (-0.149, -0.086)	-

This table presents the results of t-tests on the difference in means in Model Utility between paired granularities. Here, the results presented as those from using KL Diveregence as the forget technique and GPT-2 as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

C.1.2 Llama

Results are across a total of 7 forget set size and seed combinations for each granularity.

Table 16: Model Utility for Difference

	Book	Author	Publisher
Book	-	-	-
Author	**0.027** (0.008, 0.045)	-	-
Publisher	**0.067** (0.039, 0.095)	**0.040** (0.015, 0.066)	-

This table presents the results of t-tests on the difference in means in Model Utility between paired granularities. Here, the results presented as those from using Difference as the forget technique and Llama as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 17: Model Utility for Preference Optimisation

	Book	Author	Publisher
Book	-	-	-
Author	-0.000 (-0.025, 0.025)	-	-
Publisher	0.009 (-0.018, 0.035)	0.009 (-0.015, 0.034)	-

This table presents the results of t-tests on the difference in means in Model Utility between paired granularities. Here, the results presented as those from using Preference Optimisation as the forget technique and Llama as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

C.1.3 Phi

Results are across a total of 7 forget set size and seed combinations for each granularity.

Table 18: Model Utility for Difference

	Book	Author	Publisher
Book	-	-	-
Author	-0.046 (-0.227, 0.136)	-	-
Publisher	-0.023 (-0.210, 0.163)	0.022 (-0.152, 0.197)	-

This table presents the results of t-tests on the difference in means in Model Utility between paired granularities. Here, the results presented as those from using Difference as the forget technique and Phi as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 19: Model Utility for Preference Optimisation

	Book	Author	Publisher
Book	-	-	-
Author	0.005 (-0.021, 0.031)	-	-
Publisher	0.011 (-0.018, 0.041)	0.006 (-0.021, 0.034)	-

This table presents the results of t-tests on the difference in means in Model Utility between paired granularities. Here, the results presented as those from using Preference Optimisation as the forget technique and Phi as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

C.2 Forget Quality (One-Sided)

The results presented here show the difference in mean forget quality (one-sided) between pairs of granularities. Results are shown for each model after 25 epochs. Each forgetting technique has its own table. Across the results presented, there is for the most part no consistent pattern of results between granularities. Differences in forget quality for preference optimisation are often very close to 0. We believe this is a function of the minimal impact this technique has on underlying probabilities.

C.2.1 GPT-2

Results are across a total of 25 forget set size and seed combinations for each granularity.

Table 20: Forget Quality (1-sided) for Difference

	Book	Author	Publisher
Book	-	-	-
Author	0.001 (-0.001, 0.003)	-	-
Publisher	**0.041** (0.017, 0.064)	**0.040** (0.016, 0.064)	-

This table presents the results of t-tests on the difference in means in Forget Quality (1-sided) between paired granularities. Here, the results presented as those from using Difference as the forget technique and GPT-2 as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 21: Forget Quality (1-sided) for Preference Optimisation

	Book	Author	Publisher
Book	-	-	-
Author	0.000 (-0.000, 0.000)	-	-
Publisher	0.000 (-0.000, 0.000)	0.000 (-0.000, 0.000)	-

This table presents the results of t-tests on the difference in means in Forget Quality (1-sided) between paired granularities. Here, the results presented as those from using Preference Optimisation as the forget technique and GPT-2 as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 22: Forget Quality (1-sided) for KL Diveregence

	Book	Author	Publisher
Book	-	-	-
Author	-0.043 (-0.179, 0.093)	-	-
Publisher	0.103 (-0.053, 0.259)	**0.146** (0.002, 0.289)	-

This table presents the results of t-tests on the difference in means in Forget Quality (1-sided) between paired granularities. Here, the results presented as those from using KL Diveregence as the forget technique and GPT-2 as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

C.2.2 Llama

Results are across a total of 7 forget set size and seed combinations for each granularity.

Table 23: Forget Quality (1-sided) for Difference

	Book	Author	Publisher
Book	-	-	-
Author	0.001 (-0.000, 0.002)	-	-
Publisher	0.080 (-0.105, 0.264)	0.079 (-0.106, 0.263)	-

This table presents the results of t-tests on the difference in means in Forget Quality (1-sided) between paired granularities. Here, the results presented as those from using Difference as the forget technique and Llama as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 24: Forget Quality (1-sided) for Preference Optimisation

	Book	Author	Publisher
Book	-	-	-
Author	-0.000 (-0.000, 0.000)	-	-
Publisher	0.000 (-0.000, 0.000)	0.000 (-0.000, 0.000)	-

This table presents the results of t-tests on the difference in means in Forget Quality (1-sided) between paired granularities. Here, the results presented as those from using Preference Optimisation as the forget technique and Llama as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

C.2.3 Phi

Results are across a total of 7 forget set size and seed combinations for each granularity.

Table 25: Forget Quality (1-sided) for Difference

	Book	Author	Publisher
Book	-	-	-
Author	-0.151 (-0.668, 0.365)	-	-
Publisher	0.074 (-0.466, 0.614)	0.225 (-0.280, 0.731)	-

This table presents the results of t-tests on the difference in means in Forget Quality (1-sided) between paired granularities. Here, the results presented as those from using Difference as the forget technique and Phi as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

Table 26: Forget Quality (1-sided) for Preference Optimisation

	Book	Author	Publisher
Book	-	-	-
Author	-0.000 (-0.000, 0.000)	-	-
Publisher	0.000 (-0.000, 0.000)	0.000 (-0.000, 0.000)	-

This table presents the results of t-tests on the difference in means in Forget Quality (1-sided) between paired granularities. Here, the results presented as those from using Preference Optimisation as the forget technique and Phi as the model. The mean difference between the column and row is presented (i.e. col - row) with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

D Experiment 2 t-tests

Here we present the results of t-tests between the two retain subsets in experiment 2. As for experiment 1, all of the tables for experiment 2 follow a common format. Each row contains the difference in mean model utility between the retain subsets for a given forget technique. This difference is the mean for the entity retain subset subtracted from the remainder retain subset (retain - entity). A t-test is performed on this difference, and 95% confidence intervals are presented with the value. If the test is signficant at the 95% confidence level, the result is highlighted in bold. Differences between rows are significant at the 95% confidence level if their confidence intervals do not overlap. The difference in retain subsets for the retain model is included in each table for purposes of comparison.

The difference and Preference Optimisation approaches usually result in differences significant at the 95% confidence level. Theses differences are, in line with expectations, positive across the board; indicating that performance on the remainder retain subset is greater than on the entity remain subset. The non-differences in Gradient ascent and KL divergence mostly reflect the poor performance of these forget techniques.

The difference in retain subsets between the retain model and preference optimisation can additionally be distinguished, with non-overlapping confidence intervals. For GPT2, the difference for the retain model is smaller, while for Llama and Phi it is larger. The confidence intervals for the Difference method and the retain model always overlap.

D.1 GPT-2

Results are across a total of 60 forget set size and seed combinations.

Table 27: Difference between retain subsets in Model Utility for GPT-2

Method	Test
Gradient Ascent	-0.000 (-0.000, 0.000)
Difference	**0.139** (0.092, 0.187)
Preference Optimisation	**0.120** (0.111, 0.129)
KL Diveregence	0.004 (-0.003, 0.011)
Retain	**0.089** (0.083, 0.096)

This table presents the results of t-tests on the difference in means in Model Utility between the two retain subsets. The difference presented is the value for the entity retain subset subtracted from the remainder retain subset. These results are for GPT-2 trained for 25 epochs. The mean difference is presented with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

D.2 Llama

Results are across a total of 12 forget set size and seed combinations.

Table 28: Difference between retain subsets in Model Utility for Llama

Method	Test
Difference	**0.036** (0.011, 0.061)
Preference Optimisation	**0.018** (0.010, 0.026)
Retain	**0.059** (0.036, 0.083)

This table presents the results of t-tests on the difference in means in Model Utility between the two retain subsets. The difference presented is the value for the entity retain subset subtracted from the remainder retain subset. These results are for Llama trained for 25 epochs. The mean difference is presented with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

D.3 Phi

Results are across a total of 12 forget set size and seed combinations.

Table 29: Difference between retain subsets in Model Utility for Phi

Method	Test
Difference	0.035 (-0.003, 0.074)
Preference Optimisation	**0.019** (0.002, 0.036)
Retain	**0.081** (0.061, 0.102)

This table presents the results of t-tests on the difference in means in Model Utility between the two retain subsets. The difference presented is the value for the entity retain subset subtracted from the remainder retain subset. These results are for Phi trained for 25 epochs. The mean difference is presented with 95% confidence intervals. Values significant at this confidence level are highlighted in bold.

E Additional Figures

In this section we include the figures that were not included in the main body, section E.1 contains those from experiment 1 and E.2 those from experiment 2.

E.1 Experiment 1

Here we present the GPT2 results on the remaining subsets of data. Figure 11 we present 10%, 15%, and 20% subset results, in the top left, top right, and bottom center figures respectively. In figure 12 we present the remaining Phi and Llama results on the 15% subset (for Phi and Llama the 15% results are averaged over 5 random seeds, but other dataset sizes were only run with a single seed.)

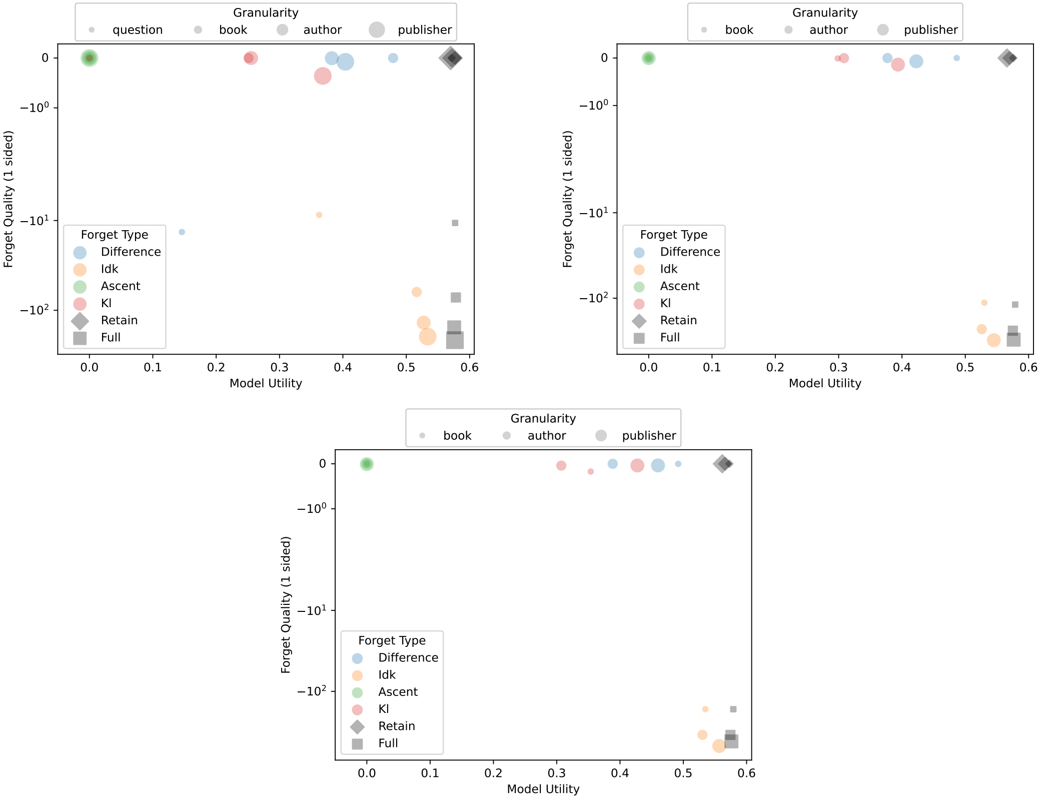

Figure 11: Granularity experiment results for the GPT2 model. (top left) shows experiments with with 10% of the evaluation data in the forget set. (top right) shows experiments with 15% of the data. (bottom center) shows experiments with 20%. Results are averaged across 5 runs with different random seeds.

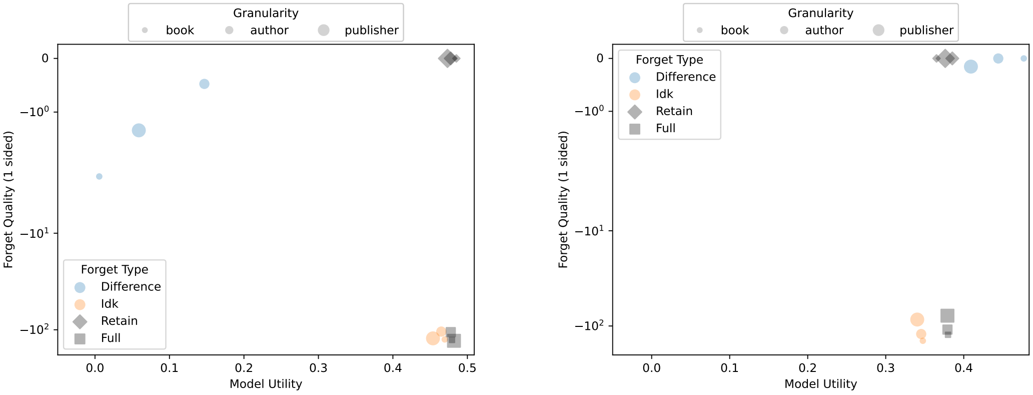

Figure 12: Granularity experiment results for the Phi and Llama models with 15% of the evaluation data in the forget set. (left) shows experiments with Phi models, (right) shows Llama models. These results are the average over 5 runs.

E.2 Experiment 2

In this section we present the remaining results from experiment 2. Figure 11 contains those from the GPT2 models: 0.5%, 2%, 3%, and 5% of the relationships removed in the top left, top right, bottom left, and bottom right figures respectively. Figure 14 contains results from Phi and Llama, which is those where 2% of the relationships have been removed, Phi is in the left figure, and Llama in the right figure.

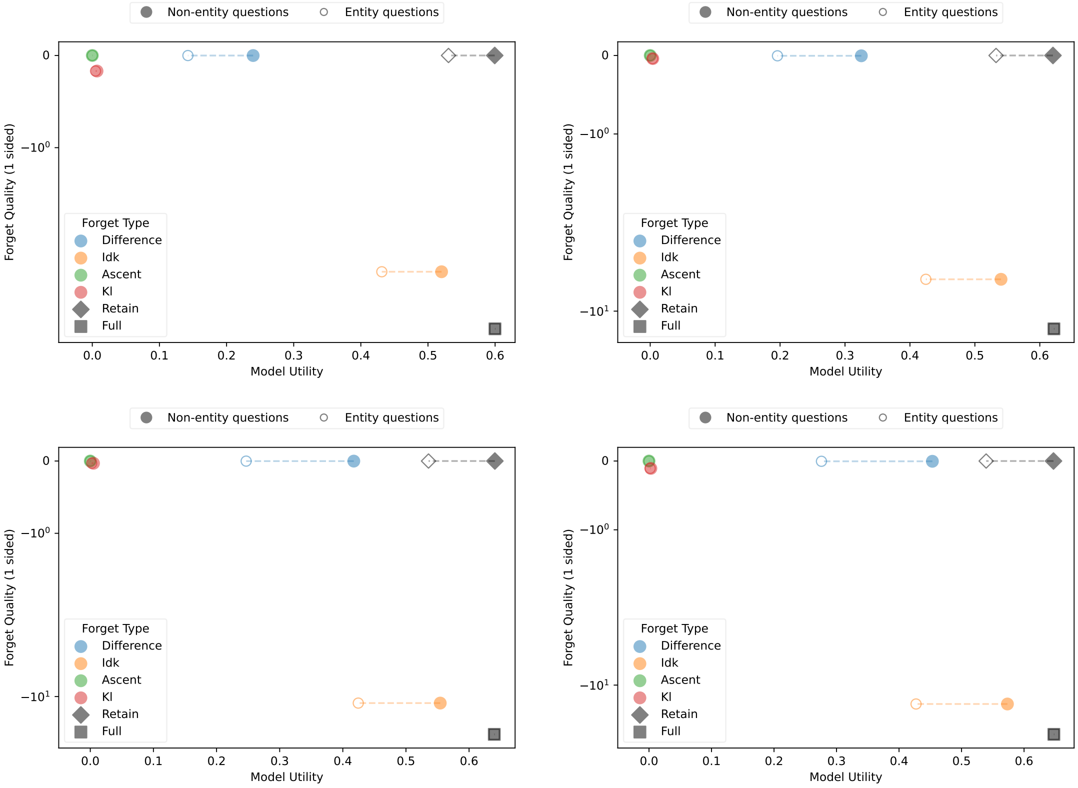

Figure 13: Relationship experiment results for the GPT2 model. (left) shows a test using 1% of the relationships in the forget set, (right) 2% of the relationships, (bottom left) and (bottom right) use 3% and 5% of the relationships respectively. Results are averaged across 10 runs.

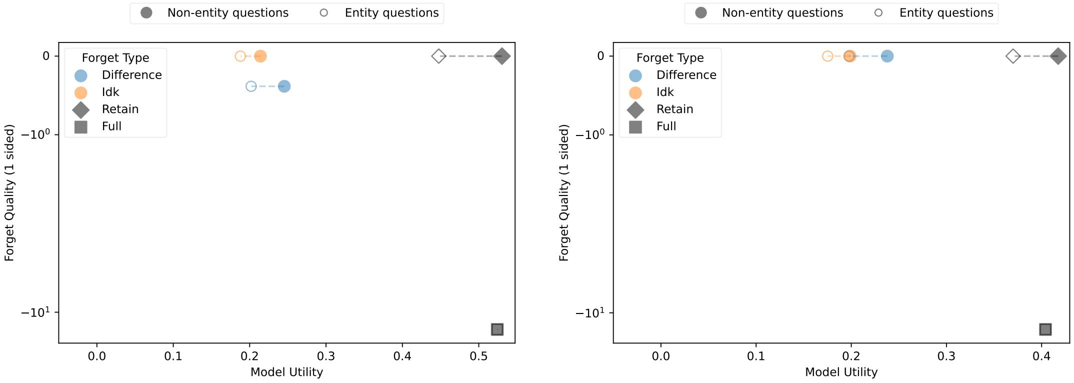

Figure 14: Relationship experiment results for the Phi and Llama models with 2% of the evaluation data in the forget set. (left) shows experiments with Phi models, (right) shows Llama models. Results are averaged across 4 runs.

F Pre-Prompts for GPT-3.5 generation

This final section contains a number of pre-prompts used for generation. Listings 1 and 2 contain those used to generate the names for our author and book names respectively, and listings 3 and 4 respectively contain pre-prompts used for generating questions for those entities.

In Listing 5 we prompt the model to paraphrase a question answer pair so that the wording is changed, but the meaning is preserved, allowing us to have distinct train and evaluation sets for dataset. Listing 6 on the other hand prompts the model to hallucinate a response to a question, allowing us to provide erroneous responses to our questions.

Finally listings 7,8, and 9 include the pre-prompts used to iteratively generate questions. These were used in conjunction with a question generation class which would provide additional information to the model through the generation client, along with the questions that had already been generated. This code is available in the project GitHub repository:

`https://github.com/alan-turing-institute/arc-selective-forgetting`.

```
"""
You have been tasked with producing random names for people born in a specified
    country.
You should generate no fewer than {} names separated with a new line after each.
There should be an even distribution of Male and Female names.
You should structure your response like so:

<begin_names>
first_name_1 surname_1
...
first_name_n surname_n
...
first_name_{} surname_{}
<end_names>

It is vitally important all names are contained between the two tags <
    begin_names> and
<end names>.
"""
```

Listing 1: Pre-prompt for author name generation.

```
"""
You have been tasked with producing interesting names for books of a specified
    genre.
You should generate no fewer than {} names separated with a new line after each.
All books should be completely independant of one another, though they can share
     similar
topics. It is also imperative that these names have never before been used.
Your response should be strucured as such:

<begin_names>
book_title_1
...
book_title_n
...
book_title_{}
<end_names>

It is vitally important all names are contained between the two tags <
    begin_names> and
<end names>.
On each line there should no text except that of the book title.
DO NOT NUMBER THE BOOKS.
"""
```

Listing 2: Pre-prompt for book name generation.

```
"""
You have been tasked with generating question--answer pairs exploring the
    upbringing,
writing style, and personal life of fictional authors. You should generate no
    fewer than
{} question--answer pairs separated with a new line after each. Make the answers
detailed, self-contained, and make sure the author's full name appears in the
    question
content. Your questions should reference two or more properties in the provided
    profile,
and not should not be solely about a single property. You should not reference
    any books
 they might have written. If there is insufficient information in the profile,
    you are
encouraged to hallucinate the answer.

You should structure your response like so:

<begin_questions>
Question: question_1?
Answer: answer_1
...
Question: question_n?
Answer: answer_n
...
Question: question_{}?
Answer: answer_{}
<end_questions>

It is vitally important all pairs are contained between the two tags: <
    begin_questions>
and <begin_questions>.
"""
```

Listing 3: Pre-prompt for author question generation.

```
"""
You have been tasked with generating question--answer pairs summarising a book,
    you will
be provided a book title and its genre. You should generate no fewer than {}
question--answer pairs discussing the books' plot in detail and any notable
    features of
its release.

Your question--answer pairs will include a detailed synopsis of the book, and a
detailed overview of themes explored in it these should be detailed.
It is imperative that the book's full name appears in every question and that
    the
answers are detailed and self-contained. Ensure that all questions cover all of
    the
properties in the provided profile.

Under no circumstances should you reveal who wrote the book, as this information
     is not
to be contained in these questions.

You should structure your response like so:

<begin_questions>
Question: question_1?
Answer: answer_1
...
Question: question_n?
Answer: answer_n
...
Question: question_{}?
Answer: answer_{}
<end_questions>

It is vitally important all pairs are contained between the two tags: <
    begin_questions>
and <begin_questions>.
"""
```

Listing 4: Pre-prompt for book question generation.

```
"""
You have been tasked with paraphrasing a question and answer pair. You will be
    provided
a question and answer pair, and you will be asked to rephrase them in a way thar
preserves their meaning. Your response should be structured as such:

<begin_paraphrased_question>
Question: paraphrased_question
Answer: paraphrased_answer
<end_paraphrased_question>

It is vitally important the question and answer pair are contained between the
<begin_paraphrased_question> and <end_paraphrased_quesition> tags defined above.
"""
```

Listing 5: Pre-prompt for answer paraphrasing.

```
"""
You have been tasked with generating an incorrect response to a question. You
    will not
know the answer or the context of the question, but you must generate a
    realistic
sounding answer. This answer will be used as an incorrect option in a multiple
    choice
setting.

You will receive a question, and you must generate your answer to the question
    and
nothing else. In your response it is imperative you do not reference the fact
    that it is
an incorrect answer, a hallucination, or anything otherwise. Your response
    should
contain only the text of the answer.
"""
```

Listing 6: Pre-prompt for question hallucination.

```
"""
You have been tasked with generating question--answer pairs summarising a book,
    you will
be provided a book title along with its genre and other properties. You should
    generate
question--answer pairs discussing the books' plot in detail. Then, you should
    generate
question--answer pairs discussing any notable features prompted.

You will also be provided questions that already exist for the book, if any. You
     should
not repeat these, but build on them using the provided question suggestions. It
    is vital
you incorporate all information from the provided profile and make the questions
increasingly complex and long.

You should structure your response like so:

<begin_new_questions>
Question: question_1?
Answer: answer_1
...
Question: question_n?
Answer: answer_n
<end_new_questions>
"""
```

Listing 7: Pre-prompt for iterative book question generation.

```
"""
You have been tasked with generating question--answer pairs summarising an
    author
profile, you will be provided an author profile. You should generate question--
    answer
pairs discussing the author in detail.

You will also be provided questions that already exist for the author, if any.
You should not repeat these, but build on them using the provided question
    suggestions.
It is vital you incorporate all information from the provided profile and make
    the
questions increasingly complex and long.

You should structure your response like so:

<begin_new_questions>
Question: question_1?
Answer: answer_1
...
Question: question_n?
Answer: answer_n
<end_new_questions>
"""
```

Listing 8: Pre-prompt for iterative author question generation.

```
"""
You have been tasked with generating question--answer pairs summarising an
    publisher
profile, you will be provided a publisher profile. You should generate question
    --answer
pairs discussing the publisher in detail.

You will also be provided questions that already exist for the publisher, if any
    .
You should not repeat these, but build on them using the provided question
    suggestions.
It is vital you incorporate all information from the provided profile and make
    the
questions increasingly complex and long.

You should structure your response like so:

<begin_new_questions>
Question: question_1?
Answer: answer_1
...
Question: question_n?
Answer: answer_n
<end_new_questions>
"""
```

Listing 9: Pre-prompt for iterative publisher question generation.