Do Large Language Models Know How Much They Know?

Gabriele Prato[1,2,3], **Jerry Huang**[1,2,3]
Prasannna Parthasarathi[2], **Shagun Sodhani**[4], **Sarath Chandar**[1,2,5]
[1]Chandar Research Lab [2]Mila – Quebec AI Institute [3]Université de Montréal
[4]Meta FAIR [5]Polytechnique Montréal [6]Canada CIFAR AI Chair
{gabriele.prato,jerry.huang}@mila.quebec

Abstract

Large Language Models (LLMs) have emerged as highly capable systems and are increasingly being integrated into various uses. Nevertheless, the rapid advancement in their deployment trails a comprehensive understanding of their internal mechanisms, as well as a delineation of their capabilities and limitations. A desired characteristic of an intelligent system is its ability to recognize the scope of its own knowledge. To investigate whether LLMs embody this attribute, we develop a benchmark that challenges these models to enumerate all information they possess on specific topics. This benchmark assesses whether the models recall excessive, insufficient, or the precise amount of required information, thereby indicating their awareness of how much they know about the given topic. Our findings reveal that the emergence of this property varies across different architectures and manifests at diverse rates. However, with sufficient scaling, all tested models are ultimately capable of performing this task. The insights gained from this research advance our understanding of LLMs, shedding light on their operational capabilities and contributing to the ongoing exploration of their intricate dynamics.

1 Introduction

Large Language Models are renowned for their ability to memorize vast amounts of information encountered during training (OpenAI, 2023; Touvron et al., 2023; Gemini Team, 2023). This information, stored in their parameters, can be recalled during inference, serving both for information retrieval and problem-solving (Vinyals and Le, 2015; Radford et al., 2019; Chung et al., 2022a; Geva et al., 2023). While it is well-established that LLMs can act as knowledge bases (Petroni et al., 2019; Heinzerling and Inui, 2021; AlKhamissi et al., 2022), the extent to which they understand their own knowledge is less clear (Liang et al., 2024). For instance, do these models know if or when they know the answer to a question (Kadavath et al., 2022; Yin et al., 2023)? Can they quantify their own expertise on a topic? Are they aware if some of their knowledge contradicts other information they possess? Can they differentiate between explicitly learned information and implicit knowledge?

These questions are crucial, as awareness of one's own knowledge and limitations is a vital aspect of any intelligent system. Without it, an AI could be prone to hallucinate (Ye et al., 2023; Xu et al., 2024), lie about its expertise (Azaria and Mitchell, 2023; Pacchiardi et al., 2023), overestimate its responses (Desai and Durrett, 2020; OpenAI, 2023), or contradict itself (Chen et al., 2023), all of which are undesirable traits for AI systems intended to be useful.

This study focuses on understanding whether LLMs know the extent of their knowledge on specific topics, such as individuals, locations, events or concepts. To explore this, we task LLMs with enumerating everything they know about a given topic—no more, no less. Should a model consistently recall just the right amount of information, it suggests an understanding of its own knowledge. Conversely, if a model does not know how much it knows, it may recall too little or hallucinate additional information.

Our approach involves fine-tuning LLMs on the diary entries of various fictitious individuals. Each entry is treated as an individual document in our fine-tuning dataset, with each diarist authoring a random number of entries. During inference, we ask the models to recall all diary entries of a specified individual in chronological order. We then evaluate whether the recalled entries match the original entries both in terms of content and quantity. Figure 1 provides an illustrative example.

We benchmark the performance of the OPT (Zhang et al., 2022), Pythia (Biderman et al., 2023), and Flan-T5 (Chung et al., 2022b) suites of

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 6054–6070
November 12-16, 2024 ©2024 Association for Computational Linguistics

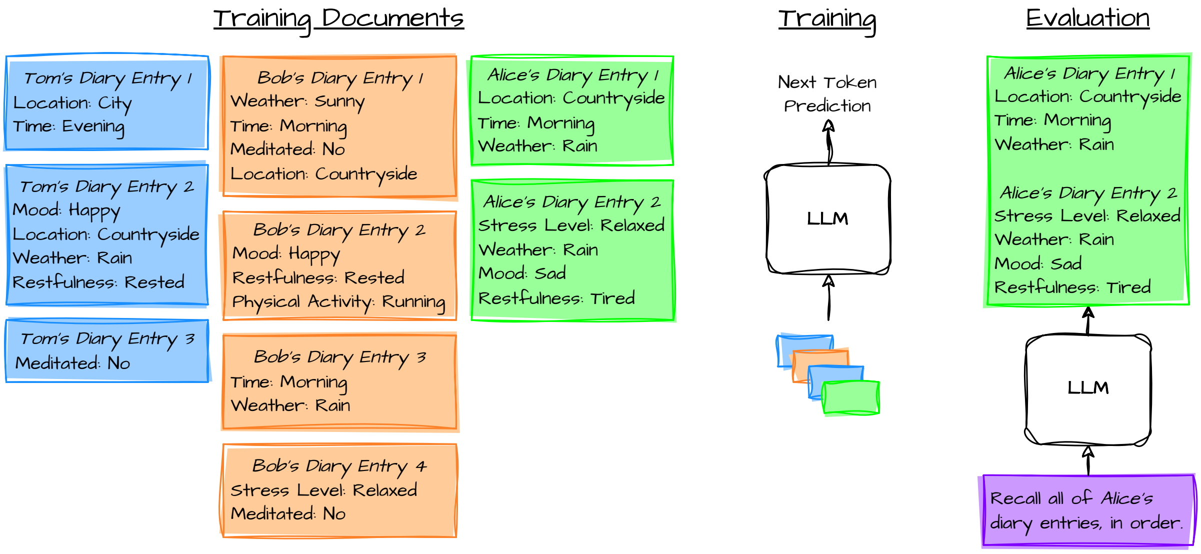

Figure 1: We train LLMs using diary entries from various individuals, with each diarist contributing a random number of entries. We then task the models with recalling all entries written by a specific individual, evaluating their ability to accurately recall the exact number of documents authored by that person.

models. Our key findings are as follows:

- All tested LLMs, if scaled sufficiently, demonstrate an understanding of how much they know. This capability appears to emerge under different conditions depending on the architecture. For example, the smallest OPT model can perform this task effectively if the fine-tuning dataset is sufficiently large. In contrast, Pythia and Flan-T5 require joint scaling of both the dataset and model size to perform well.

- However, when these conditions are not met (i.e., insufficient scaling), models often recall a random number of diary entries, either recalling too few or hallucinating additional ones.

- Interestingly, the number of diary entries to recall and their length do not impact model performance, meaning that when models are capable of performing this task, they are just as good at recalling eight entries as they are at recalling one.

Finally, we discuss potential factors responsible for the observed differences in the emergence of this capability. Overall, our work contributes to a deeper understanding of the inner workings of LLMs, shedding light on a not-so-well-understood aspect of these models.

2 Methodology

The foundation of our analysis hinges on the ability of models to memorize and recall information. To avoid the influence of existing data, which might be part of the pre-training corpus of the language models we are benchmarking, we generate our own. This ensures that the models have never encountered the data during pre-training, thereby preventing any contamination of our results.

In essence, our approach involves: (i) **generating** the training documents, (ii) **fine-tuning** a language model using its pre-training objective to memorize these documents, and (iii) **testing** the language model's ability to recall all related documents. We delineate each stage of our framework in the following sections.

2.1 Data Generation

Given N diarists, where N is a hyperparameter, we generate a random number of diary entries for each diarist, following the template:

```
{name}'s Diary Entry {i}
{attribute}
 ⋮
{attribute}
```

where `{name}` is the diarist (e.g., "Tom") and `{i}` is the entry number (e.g., "1"). The document contains a random number of `{attribute}`, each selected randomly without replacement from a set

(Table 3), along with a randomly chosen value (e.g., “Time: Morning”). Additionally, for each individual, we have one question following the format:

```
Recall all of {name}’s diary
entries, in order.
```

The answer to the question is the concatenation of the individual’s diary entries in ascending order by entry numbers. Figure 1 illustrates examples of both generated documents and questions.

To effectively train the model, we incorporate 90% of the question-answer (Q/A) pairs, along with *all* diary entries, into the training set. The remaining 10% of the questions are evenly divided into a validation set and a test set. By adding Q/A examples to the training set, the model can learn the evaluation task, similar to the process of instruction-tuning.

Initially, we trained the model first on the documents and then on the evaluation task. However, this approach led to catastrophic forgetting of the documents and overfitting on the Q/A examples. Therefore, we decided to fine-tune the model on both simultaneously to prevent these issues.

2.2 Fine-Tuning & Evaluation

To benchmark an LLM, we begin by fine-tuning it using its pre-training objective, such as causal language modeling, on our training set. This fine-tuning process mirrors the standard training of an LLM on a text corpus. Depending on the architecture of the LLM, we format the input as follows:

- **Decoder-Only Models (e.g., OPT):** For both diary entries and Q/A pairs, the training objective is causal language modeling. In the case of Q/A pairs, we concatenate the question with the answer into a single text sequence, separated by an end-of-line token (‘\n’).

- **Encoder-Decoder Models (e.g., Flan-T5):** When processing a diary entry, the first line (e.g., “Tom’s Diary Entry 1”) is input to the encoder, and the decoder generates the *entire* document. For Q/A pairs, the question is fed to the encoder, and the decoder predicts the answer.

Throughout the fine-tuning process, we periodically evaluate the model on the validation set. For decoder-only architectures, the model is prompted with a question with the goal of generating the corresponding answer. For encoder-decoder architectures, the question is given to the encoder and the decoder must produce the answer.

We fine-tune up until the validation performance plateaus. We then select the best checkpoint based on peak validation performance, and evaluate the model on our test set using the same procedure as with the validation set. Performance is measured in terms of accuracy, defined as the number of correctly answered questions. An answer is deemed correct if it matches the ground truth exactly, with no errors in the number of documents recalled and the content of each recalled document.

2.3 Design Motivation

Requiring the model to consolidate information from multiple training documents allows us to assess whether it understands the extent of its knowledge related to the individual in question. Specifically, during training, the model memorizes the diary entries. Then, in the evaluation phase, it needs to know how many documents to recall, meaning the model must know how many diary entries it knows about the individual. If a model consistently recalls the exact number of documents, it shows an understanding of its own knowledge. Conversely, a model which does not know how many documents it knows, would recall a random number.

As for our choice of using synthetic data, it allows us to precisely control its distribution and properties. This extends to the length and content of the documents, as well as the number of diary entries authored by an individual. By using attributes as the body of the documents, we can manage the entropy, ensuring that each sentence contains a fixed amount of information. Consequently, adding an additional sentence consistently increases the document’s information by that fixed amount.

This approach enables us to examine how document length affects the model in a more controlled manner compared to using real data. While we have arbitrarily chosen individuals as the topic linking multiple documents, this could have been any other concept. We believe this choice does not impact the observed trends in the results.

Overall, our benchmark is designed to facilitate the study of this problem and its key variables in a controlled environment, emulating the challenge faced by language models of memorizing information during training and understanding the extent of their knowledge concerning specific topics.

3 Experiments

3.1 Setup

Dataset. To evaluate the impact of the number of training examples on the model performance, we generate six datasets containing 1K to 32K diarists, with each successive dataset doubling in size compared to its predecessor. By incrementally enlarging the dataset size as described, models see a broader array of examples from which they can learn to derive their generative capabilities, while simultaneously being challenged to memorize a larger volume of documents.

For each individual, we generate 1 to 8 diary entries, with each entry consisting of 1 to 8 attributes. The training, validation and test sets each contain an equal distribution of individuals who have written one, two, three, etc. diary entries. Similarly, we maintain a uniform distribution for document lengths. Dataset details, such as the number of authors, diary entries, and Q/A pairs, are provided in Appendix B.

Models. We benchmark the following suit of publicly available models: decoder-only OPT (125M to 2.7B) (Zhang et al., 2022) and Pythia (70M to 2.8B) (Biderman et al., 2023), and encoder-decoder Flan-T5 (80M to 3B) (Chung et al., 2022b). A comparison of these architectures is provided in Appendix C. Training hyper-parameters are provided in Appendix D. Unless specified otherwise, reported metrics are based on the test set.

3.2 Results

Effect of Architecture & Scale. We first evaluate the impact of architecture, model size, and dataset size on performance. We fine-tune each model on our datasets and report their performance as solid lines, labeled as 'standard setup' in Figure 2. The horizontal axis represents model size, the vertical axis indicates the percentage of correctly answered questions, and the line color signifies the dataset size. Each line on the plot corresponds to a specific architecture (e.g., OPT), ranging from the smallest to the largest model, trained on a particular dataset size.

For the OPT suite, we observe that performance improves with an increase in either model size or dataset size. Notably, as the dataset grows larger, the performance gap between different model sizes diminishes. Specifically, the smallest OPT model (125M parameters) shows significant performance improvement with larger datasets, with no evident signs of saturation.

Conversely, for Pythia models, merely scaling the dataset size does not enhance the performance of the two smallest models as effectively as with OPT. Rather, the architecture benefits most from simultaneously scaling both dataset and model size.

Finally, the performance of Flan-T5 models shows minimal improvement as both dataset and model size increase. However, a notable exception occurs with the largest model, which exhibits a sudden spike in performance when trained on the largest dataset. This behavior contrasts sharply with the results observed for OPT models, indicating that the capability being studied can emerge at different rates and under varying conditions depending on the architecture used.

Notably, all models achieve near 100% accuracy on the Q/A pairs in the training set, as well as in memorizing and recalling individual diary entries (not shown in any figure). Therefore, the observed performance gap is not due to difficulty in memorizing the training data.

Effect of Distributed Information. We compare the model performance against a second set of models trained in a simpler setup. Particularly, this second group of models is trained on identical datasets, but with all diary entries authored by the same individual merged into a single training document rather than each entry being its own document. This approach is equivalent to training the models on the answers directly, requiring them to simply memorize and recall single documents. The performance gap between these two setups highlights the added difficulty of dealing with information spread across multiple training documents. This distribution could affect how information is stored in the model's parameters, potentially making it harder for the model to consolidate it when it is dispersed.

In Figure 2, the results of training within this more straightforward setup are shown as dashed lines, labeled 'simplified setup'. In all cases, these models exhibit significantly improved performance compared to the same base model trained within the distributed setup. Interestingly, all Flan-T5 models achieve near-perfect accuracy in this simplified setup whereas OPT and Pythia suites do not, despite performing well and improving with scale.

To better illustrate the performance gap between both setups, we provide a clear visualization in Figure 3. The vertical axis shows the accuracy

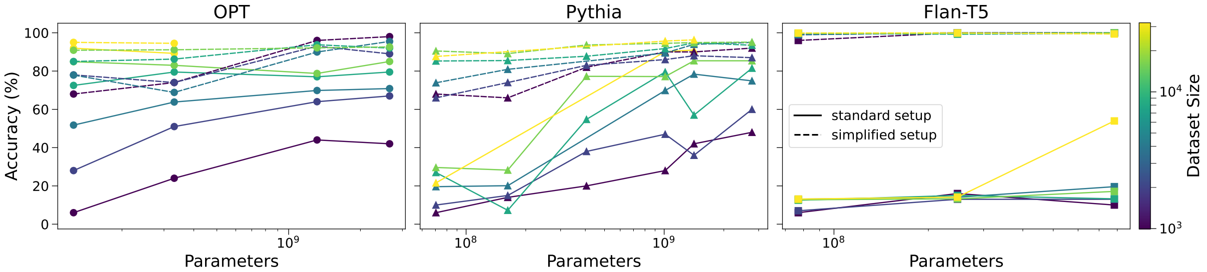

Figure 2: Accuracy of various models on our benchmark is depicted with *solid* lines, where each line represents a different model suite (e.g., OPT) ranging from the smallest to the largest variant, fine-tuned on datasets of varying sizes as indicated by the line colors. For comparison, models trained under a simpler setup are shown with *dashed* lines, where all information necessary to answer a question is contained within a single training document, eliminating the need to recall information from multiple documents.

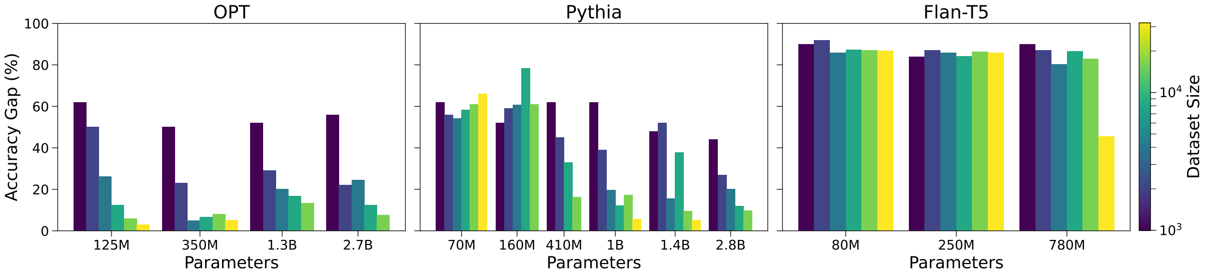

Figure 3: Gap in accuracy between the standard and simplified setup in Figure 2, for a same sized model trained on a same sized dataset.The effect of scaling the dataset and model size varies greatly depending on the architecture.

gap between the ‘simplified’ and ‘standard’ setup, for models of the same size, trained on datasets containing the same number of individuals. Results are grouped by model size, with colors denoting dataset size.

For the OPT models, the gap narrows as the dataset size increases across all model sizes. In the case of Pythia, the gap only seems to narrow for larger models trained on sufficiently large datasets. Lastly, for Flan-T5, the performance gap barely shrinks as both dataset and model size scale, with the exception of the largest model trained on the largest dataset.

It remains unclear why Flan-T5 models perform so well in the simpler setup but so poorly in the standard setup. Given that the model has near perfect accuracy in the prior, its poor performance in the latter cannot be attributed to an issue in the methodology, as the process is the same in both cases. The only difference is that, in the latter case, the model must recall information from multiple documents rather than a single one. Therefore, the model specifically has an issue with this aspect.

For all models, it is uncertain whether their performance in both setups will continue to improve with scale and if the gap will eventually disappear.

Effect of Number of Documents. Next, we explore how the number of documents to be consolidated and recalled impacts model performance. In Figure 4, we report accuracy grouped by the number of documents in the target answer (horizontal axis). Line color indicates model size. For clarity, we only display the performance of models trained on our largest dataset. Notably, there are no results on the simpler setup in this and further analyses.

Surprisingly, models do not demonstrate a decline in performance when more diary entries need to be recalled. Given the increased content to be generated, one might expect a higher propensity for errors in the model answers. This counterintuitive observation could be attributed to the controlled nature of our experimental setup, where the model knows it must recall between one and eight documents. Real-world scenarios might yield different results and a significant increase in the number of documents could potentially lead to performance degradation, warranting further research.

With respect to scale, only the Pythia models show improved performance. Corroborating with previous observations in Figure 2, we observe a jump in performance past the first two smallest Pythia models.

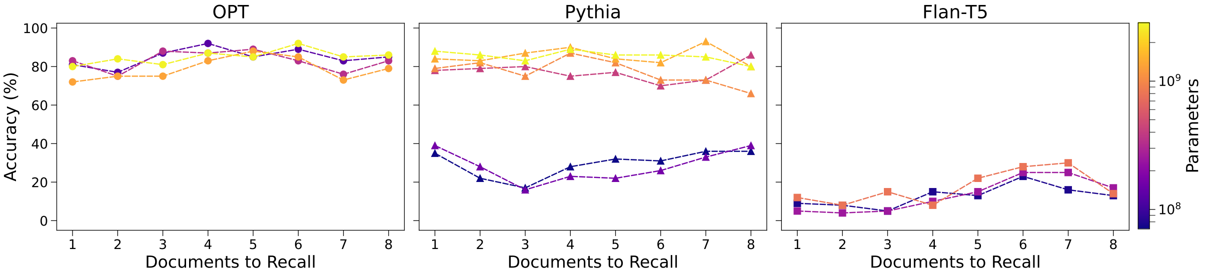

Figure 4: Impact of the number of documents needing to be recalled on the likelihood of a model's answer containing an error. Results are from models trained on our largest dataset. Surprisingly, we observe no significant difference in performance as the number of documents to recall increases.

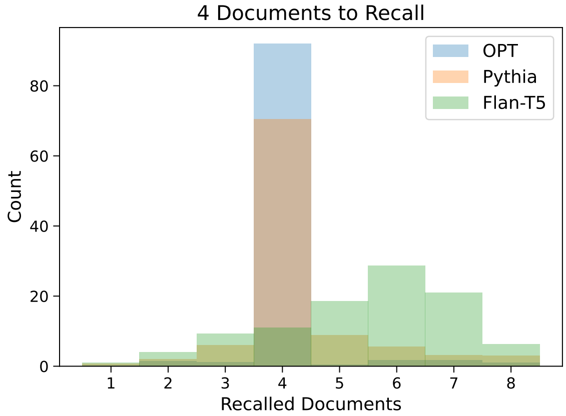

Figure 5: Number of documents recalled by each model when the target is four. Results are averaged over model sizes for simplicity, from models trained on our largest dataset. Full results are provided in Appendix E.

To gain deeper insights into model behavior, we analyze the number of documents recalled by the models in comparison with the target number of documents. Figure 5 shows this distribution when the target number of documents is four. Performance is averaged over model size for simplicity. Full results are available in Appendix E. We find that, regardless of the target number of documents to recall, Flan-T5 models tend to recall a random number of documents. This contrasts with OPT models, which recall the expected number of entries across all scales. As for Pythia, the smaller models struggle to recall the correct number of documents; however, this capability seems to improve as the model size increases.

Effect of Document Length. Previously, our method for measuring *accuracy* involved counting the number of model answers that matched the target answer exactly. We now shift our focus to evaluating the accuracy of individual documents within a model's answer, which we refer to as *document accuracy*.

In this analysis, we only consider the documents recalled by the model that are also present in the target answer, regardless of whether these documents are correct. Our objective is to examine how the length of the target documents influences the model's ability to recall them accurately. Hence, we restrict our analysis to this specific subset of documents, as we need a target for their length.

For these selected documents, we count those that are free of errors and represent this rate on the vertical axis of Figure 6. The performance is categorized by the target length on the vertical axis, and the line color indicates the size of the model. To ensure clarity, we present results exclusively for the models trained on our largest dataset.

Across all models, performance seems unaffected by document length, for which one might anticipate an increased likelihood of errors as the document grows longer. This consistent performance could be attributed to the model's expectation that documents typically contain between one and eight sentences. Additionally, LLMs are known to be quite effective at memorizing documents. To ascertain whether increased document length would eventually degrade model performance, further scaling and testing with longer documents would be necessary.

Regarding model scale, only Pythia benefits from increased size, once more with a jump in performance past the first two model sizes.

To further understand model behavior, we analyzed the number of recalled sentences, in comparison with the target document length. The histograms in Appendix F illustrate these distributions for each model, with color indicating the model size. We observe that all models recall the correct number of sentences, and that scale only slightly improves performance. A qualitative analysis re-

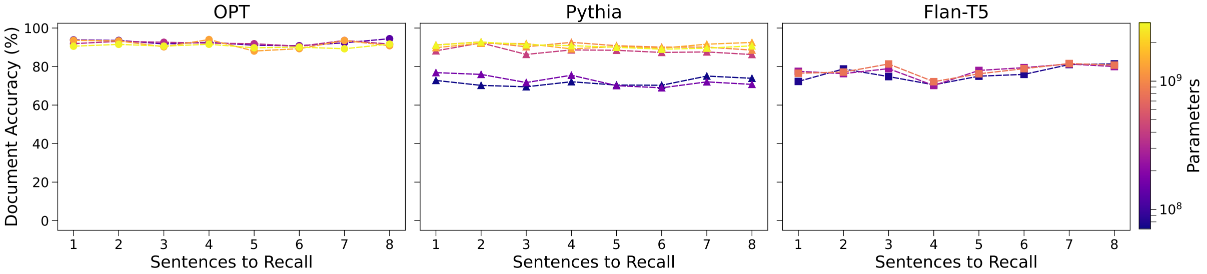

Figure 6: Taking the subset of documents in a model's answer, which are also in the target, we measure the number of such documents that are free of errors, defined as the 'document accuracy'. We then categorize this rate by the length of the corresponding target document, in order to measure its effect on the recall capabilities of the model. Results are from models trained on our largest dataset. Peculiarly, we find that documents recalled by models aren't more likely to contain errors as their length increases.

Model	Pre-trained	Scratch
OPT-125M	91.94%	81.75%
Pythia-70M	21.56%	72.56%
Flan-T5 Small	13.12%	13.31%

Table 1: Comparison between fine-tuning pre-trained models and models initialized with random weights.

veals that while models correctly recall the number and type of attributes, they often err in the attribute values, such as recalling "Time: Evening" instead of "Time: Morning."

Investigating Performance Discrepancies. Our results indicate that the ability to consolidate and accurately recall the correct number of documents varies depending on the model suite, but the underlying reasons for this discrepancy remain unclear. At a high level, these differences in performance could be due to several factors: architectural variations, the effectiveness of pre-trained weights for fine-tuning on this task, the fine-tuning hyperparameters, or a combination of these elements.

To investigate this further, we fine-tuned an OPT-125M, a Pythia-70M, and a Flan-T5 Small model, all with randomly initialized weights, using our largest dataset. We then compare their performance against the pre-trained models that were fine-tuned on the dataset of the same size.

Our findings reveal that the Pythia model initialized with random weights significantly outperform the pre-trained weights (Table 1). This suggests that architectural differences are not responsible for the poor performance of this model. Instead, the issue lies in the capability of the pre-trained weights to be effectively fine-tuned for this specific task.

Regarding Flan-T5, fine-tuning with randomly initialized weights does not appear to enhance performance when compared to fine-tuning the pre-trained model. This observation suggests that the model's architecture is responsible for the observed differences in performance.

Although fine-tuning hyperparameters could also be a factor, we conducted a thorough search. Additionally, models in the simpler setup performed well and were trained with identical hyperparameters. Conversely, in the standard setup, while models were able to memorize the training samples and Q/A examples, the solutions learned by the pre-trained Pythia-70M and Flan-T5 Small does not generalize well to the validation and test Q/A, unlike the OPT model.

3.3 Comprehensive Analysis

Reflecting on our experimental observations, we can gain insights into the causes of certain models failing on our benchmark. We've observed that the documents recalled by the models are typically of the correct length (Appendix F) and error-free (Figure 6). Additionally, models trained under the simplified setup successfully recall information from a single training document (Figure 2). Therefore, the issue appears not to lie in the content of the recalled documents but rather in the quantity of documents being recalled.

Indeed, some models seem incapable of recalling the correct number of documents, instead recalling a random number of documents (Appendix E). These models include the two smallest Pythia variants and all Flan-T5 models, which correspondingly perform poorly on our benchmark.

Interestingly, the smallest Pythia model performs well if fine-tuned starting from random weights rather than the pre-trained weights (Table 1), suggesting that the poor performance of the pre-trained weights cannot be attributed to an architectural reason. Instead, the issue appears to be with the pre-training weights failing to learn a solution that generalizes to the problem of recalling the correct number of documents, rather than merely memorizing the training samples. Why this discrepancy occurs, particularly in contrast to the larger pre-trained Pythia models, remains unclear and warrants further research. Different hyperparameters could potentially enable the smaller models to generalize well to our problem, but it is uncertain if this can be achieved without severely degrading the language modeling capabilities of the pre-trained model.

Regarding Flan-T5, given that the smallest model fine-tuned from scratch performs as poorly as the one fine-tuned from pre-trained weights, the root cause of the poor performance could be either architectural or due to improper hyperparameters. Additionally, the size of the model appears to influence its performance. Since Flan-T5 follows an encoder-decoder architecture, unlike the decoder-only structures of models such as OPT and Pythia, its parameters are divided roughly equally between the encoder and decoder. Consequently, the largest Flan-T5 model's decoder is comparable in size to that of the third smallest Pythia model, which coincides with the point where performance begins to improve for Pythia (as seen in Figure 2). Models within the Pythia suite smaller than this threshold do not show significant performance gains. However, the smallest Pythia model, when trained from scratch, outperforms Flan-T5 under similar conditions. This highlights the role of scale in model performance, suggesting that both architectural and hyperparameter factors could hinder the emergence of capabilities at smaller scales. Further research will be necessary to pinpoint the exact cause and clarify the challenges faced by these smaller models.

4 Discussion

In addition to the questions raised in the previous section, the following additional questions should be considered.

4.1 Language Modeling

One aspect not addressed in this study is whether models can perform the given task while retaining their language modeling capabilities. Due to the size of the models examined, repetitive fine-tuning on the training documents is necessary for them to memorize the data, which leads to overfitting on the task. Ideally, experiments would need to be conducted on much larger models, incorporating the training documents into the pre-training corpus, followed by standard instruction tuning. One of the tasks in this tuning would involve recalling all documents related to a given topic. This approach would help determine if a model can accomplish this in a manner that is useful for solving problems. Unfortunately, we currently lack the computational resources to conduct such experiments, and hence we leave this for future work.

4.2 Information Distribution

We observed a notable performance gap between the standard and simplified setups, supporting the findings by Prato et al. (2023). Their research indicates that LLMs more easily recall multiple pieces of information when this information is contained in a single training sample rather than dispersed across multiple samples. This raises questions about how the distribution of topic-related information across multiple training documents affects an LLM's ability to gauge its knowledge. Particularly, the impact of this distribution on the internal mechanisms of the LLM is not well understood.

Numerous studies have shown that language models can memorize entire passages and documents within their weights, enabling them to recall this information during inference (Carlini et al., 2020, 2022; Tirumala et al., 2022; Biderman et al., 2023; de Wynter et al., 2023; Chen et al., 2024). Consequently, the strong performance of models in the simpler setup, where they only need to recall information from a single document per topic, is not surprising.

However, it remains unclear why recalling information from multiple documents presents a greater challenge. Specifically, how is this information encoded within the model parameters (Wallat et al., 2020; Dai et al., 2022; Meng et al., 2022) and how does dispersed information affect the recalling process? Understanding these mechanisms is crucial for improving the performance of language models, as many real-world problems necessitate recalling information from multiple training documents.

4.3 Knowledge Awareness & Understanding

While we have demonstrated that some LLMs possess an awareness of the extent of their knowledge concerning the topics in our benchmark, this does not necessarily mean that these models can gauge their knowledge across any topic.

Determining whether LLMs can accurately assess the scope of their understanding of topics from their pre-training corpus requires further investigation. Topics in practice could cover a wide array of subjects, including individuals, locations, events, and concepts. However, we believe that the specific type of topic is likely not an influential factor.

The critical element, in our view, is the breadth of these topics, which relates to the amount of information relevant for each. Our findings did not show a decline in model performance when recalling a larger number of documents. Nevertheless, this observation might change if the number of documents were significantly increased. Further research is necessary to explore the limits and capabilities of models in handling broader topics.

A more profound question is the extent to which LLMs understand the scope of their entire knowledge base, or at least subsets of it. Given the vast amount of information LLMs learn during training, comprehending the scope of this knowledge or its subsets seems incredibly challenging. Yet, it would be beneficial for a model to understand the extent of its own expertise.

Finally, it is important to note that understanding the scope of one's knowledge concerning a topic does not imply an understanding of that topic itself. Whether LLMs truly comprehend the knowledge they have memorized is a different research question from ours and is an active area of investigation (Bender et al., 2021; Li et al., 2022; Gurnee and Tegmark, 2023).

5 Conclusion

This study focused on determining whether LLMs possess an understanding of the span of their own knowledge on specific topics. Notably, we observed that all models, if scaled sufficiently, know how many documents are authored by the same person. Consequently, these LLMs know how much they know about these individuals; otherwise, they would sporadically recall too few or too many documents.

More specifically, we find that this capability emerges based on the model's architecture, its size, the dataset's size used for training, and the effectiveness of the pre-trained weights in learning a solution that generalizes, rather than simply memorizing the training samples.

Overall, our research contributes to a deeper understanding of the capabilities and inner workings of LLMs. Grasping how aware LLMs are of their own knowledge and identifying any limitations in this regard is crucial, as this feature enhances the usefulness and trustworthiness of intelligent systems. Further research is necessary to continue exploring this aspect.

6 Limitations

The potential insights from testing larger open-source models could be valuable to the community. However, computational limitations prevent us from conducting these analyses. We hope to undertake such experiments in the future.

7 Ethical Considerations

This research utilizes large language models trained on extensive textual datasets. While such models have demonstrated exceptional ability in generation, it is critical to highlight the ethical considerations that the data used for training these models inherently contains human biases. These, in turn, can manifest in the models' outputs. As such, it is essential when deploying such models, to critically evaluate their outputs, keeping in mind the likelihood of underlying bias.

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv e-prints*, page arXiv:1610.01644.

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A Review on Language Models as Knowledge Bases. *arXiv e-prints*, page arXiv:2204.06031.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023a. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction. *arXiv e-prints*, page arXiv:2309.14316.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023b. Physics of Language Models: Part 3.2, Knowledge Manipulation. *arXiv e-prints*, page arXiv:2309.14402.

Alfonso Amayuelas, Liangming Pan, Wenhu Chen, and William Wang. 2023. Knowledge of Knowledge: Exploring Known-Unknowns Uncertainty with Large Language Models. *arXiv e-prints*, page arXiv:2305.13712.

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. *arXiv e-prints*, page arXiv:2304.13734.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Lukas Berglund, Asa Cooper Stickland, Mikita Balesni, Max Kaufmann, Meg Tong, Tomasz Korbak, Daniel Kokotajlo, and Owain Evans. 2023a. Taken out of context: On measuring situational awareness in LLMs. *arXiv e-prints*, page arXiv:2309.00667.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023b. The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A". *arXiv e-prints*, page arXiv:2309.12288.

Stella Biderman, USVSN PRASHANTH, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2023. Emergent and predictable memorization in large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 28072–28090. Curran Associates, Inc.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. *arXiv e-prints*, page arXiv:2304.01373.

Vanessa Buhrmester, David Münch, and Michael Arens. 2021. Analysis of explainers of black box deep neural networks for computer vision: A survey. *Machine Learning and Knowledge Extraction*, 3(4):966–989.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying Memorization Across Neural Language Models. *arXiv e-prints*, page arXiv:2202.07646.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting Training Data from Large Language Models. *arXiv e-prints*, page arXiv:2012.07805.

Bowen Chen, Namgi Han, and Yusuke Miyao. 2024. A Multi-Perspective Analysis of Memorization in Large Language Models. *arXiv e-prints*, page arXiv:2405.11577.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *arXiv e-prints*, page arXiv:2107.03374.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. 2023. Do Models Explain Themselves? Counterfactual Simulatability of Natural Language Explanations. *arXiv e-prints*, page arXiv:2307.08678.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022a. Scaling Instruction-Finetuned Language Models. *arXiv e-prints*, page arXiv:2210.11416.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022b. Scaling Instruction-Finetuned Language Models. *arXiv e-prints*, page arXiv:2210.11416.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv e-prints*, page arXiv:2110.14168.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Adrian de Wynter, Xun Wang, Alex Sokolov, Qilong Gu, and Si-Qing Chen. 2023. An evaluation on large language model outputs: Discourse and memorization. *Natural Language Processing Journal*, 4:100024.

Shrey Desai and Greg Durrett. 2020. Calibration of Pre-trained Transformers. *arXiv e-prints*, page arXiv:2003.07892.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv e-prints*, page arXiv:2101.00027.

Gemini Team. 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv e-prints*, page arXiv:2312.11805.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting Recall of Factual Associations in Auto-Regressive Language Models. *arXiv e-prints*, page arXiv:2304.14767.

Wes Gurnee and Max Tegmark. 2023. Language Models Represent Space and Time. *arXiv e-prints*, page arXiv:2310.02207.

Qiyuan He, Yizhong Wang, and Wenya Wang. 2024. Can Language Models Act as Knowledge Bases at Scale? *arXiv e-prints*, page arXiv:2402.14273.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring Massive Multitask Language Understanding. *arXiv e-prints*, page arXiv:2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv e-prints*, page arXiv:2103.03874.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian Error Linear Units (GELUs). *arXiv e-prints*, page arXiv:1606.08415.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. *arXiv e-prints*, page arXiv:2207.05221.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *arXiv e-prints*, page arXiv:1607.06450.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2022. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*.

Yu Liang, Siguang Li, Chungang Yan, Maozhen Li, and Changjun Jiang. 2021. Explaining the black-box model: A survey of local interpretation methods for deep neural networks. *Neurocomputing*, 419:168–182.

Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024. Learning to Trust Your Feelings: Leveraging Self-awareness in LLMs for Hallucination Mitigation. *arXiv e-prints*, page arXiv:2401.15449.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from Large Language Models faithful? *arXiv e-prints*, page arXiv:2401.07927.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. *arXiv e-prints*, page arXiv:2202.05262.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA. Omnipress.

OpenAI. 2023. GPT-4 Technical Report. *arXiv e-prints*, page arXiv:2303.08774.

Lorenzo Pacchiardi, Alex J. Chan, S. Mindermann, Ilan Moscovitz, Alexa Y. Pan, Y. Gal, Owain Evans, and J. Brauner. 2023. How to catch an ai liar: Lie detection in black-box llms by asking unrelated questions. *ArXiv*, abs/2309.15840.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language Models as Knowledge Bases? *arXiv e-prints*, page arXiv:1909.01066.

Ian Porada, Alessandro Sordoni, and Jackie Chi Kit Cheung. 2021. Does Pre-training Induce Systematic Inference? How Masked Language Models Acquire Commonsense Knowledge. *arXiv e-prints*, page arXiv:2112.08583.

Gabriele Prato, Jerry Huang, Prasanna Parthasarathi, Shagun Sodhani, and Sarath Chandar. 2023. EpiK-eval: Evaluation for language models as epistemic models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9523–9557, Singapore. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv e-prints*, page arXiv:1910.10683.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *arXiv e-prints*, page arXiv:2311.12022.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Agus Sudjianto, William Knauth, Rahul Singh, Zebin Yang, and Aijun Zhang. 2020. Unwrapping The Black Box of Deep ReLU Networks: Interpretability, Diagnostics, and Simplification. *arXiv e-prints*, page arXiv:2011.04041.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38274–38290. Curran Associates, Inc.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv e-prints*, page arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762.

Oriol Vinyals and Quoc Le. 2015. A Neural Conversational Model. *arXiv e-prints*, page arXiv:1506.05869.

Jonas Wallat, Jaspreet Singh, and Avishek Anand. 2020. BERTnesia: Investigating the capture and forgetting of knowledge in BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 174–183, Online. Association for Computational Linguistics.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is Inevitable: An Innate Limitation of Large Language Models. *arXiv e-prints*, page arXiv:2401.11817.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive Mirage: A Review of Hallucinations in Large Language Models. *arXiv e-prints*, page arXiv:2309.06794.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do Large Language Models Know What They Don't Know? *arXiv e-prints*, page arXiv:2305.18153.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *arXiv e-prints*, page arXiv:2205.01068.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2023. Knowing What LLMs DO NOT Know: A Simple Yet Effective Self-Detection Method. *arXiv e-prints*, page arXiv:2310.17918.

A Related Work

A.1 Knowledge Awareness

Large language models are widely recognized for memorizing a substantial amount of information during their training (Petroni et al., 2019; Roberts et al., 2020; Jiang et al., 2020; Carlini et al., 2022; AlKhamissi et al., 2022; He et al., 2024). However, it remains unclear to what extent these models understand their own knowledge. Research to date has shown that LLMs can assess, with some degree of accuracy, whether they know the answer to a given question (Kadavath et al., 2022; Zhao et al., 2023; Amayuelas et al., 2023; Yin et al., 2023; Liang et al., 2024).

While these studies primarily evaluate the model's ability to determine if it possesses the knowledge necessary to answer a question, they do not consider the quantity and source of this knowledge. For instance, the question "Is Jupiter a planet?" requires knowledge of a single fact, whereas "Do you know all papers related to topic X?" necessitates understanding multiple pieces of information, likely derived from various training samples.

In essence, locating a specific piece of information within a model's parameter space is different from retrieving multiple pieces of information and recognizing when the search is complete. Our research focuses on this latter aspect, seeking to determine whether LLMs comprehend the extent of their knowledge on specific topics.

A.2 Implicit Knowledge Retrieval

At the heart of our methodology lies implicit knowledge retrieval. This involves prompting a model with a question, enabling it to retrieve knowledge stored within its parameters, and subsequently generating an answer based on the retrieved information (Vinyals and Le, 2015; Chung et al., 2022a; Geva et al., 2023). Considering the black box nature of deep neural networks (Alain and Bengio, 2016; Sudjianto et al., 2020; Buhrmester et al., 2021; Liang et al., 2021), this setup is frequently employed to deduce the inner workings and capabilities of such models (Porada et al., 2021; Berglund et al., 2023b; Allen-Zhu and Li, 2023a,b; Berglund et al., 2023a; Madsen et al., 2024), offering valuable insights into the knowledge and skills the model has acquired (Hendrycks et al., 2020; Chen et al., 2021; Cobbe et al., 2021; Hendrycks et al., 2021; Rein et al., 2023). Hence we consider it to be a fitting analytical approach for our investigation.

B Dataset Details

B.1 Size

The details of each dataset used in our experiments are provided in Table 2. The training set of each dataset consists of all of the documents (diary entries), as well as the train Q/A pairs. The validation and test sets solely consist of Q/A pairs. There are no overlaps between the Q/A pairs in the training, validation and test sets. As previously mentioned, for each author, we generate 1 to 8 diary entries, where each entry contains 1 to 8 sentences (excluding the title).

B.2 Attributes

The list of attributes sampled for each diary entry along with possible values are presented in Table 3. Sampling of both the attribute and its value is always performed randomly. A document cannot contain the same attribute more than once, irrespective of its value.

C Model Suite Differences

The following outlines the architectural differences between the OPT, Pythia, and Flan-T5 suite of models, emphasizing their unique characteristics and training details.

Both OPT and Pythia are based on the GPT-3 architecture, with only slight variations. OPT employs learned positional embeddings and utilizes the ReLU activation function (Nair and Hinton, 2010). In contrast, Pythia incorporates rotary positional embeddings and the GELU activation function (Hendrycks and Gimpel, 2016). A notable distinction in Pythia's architecture is its use of parallel residual connections, where the self-attention and feed-forward blocks run concurrently, and their outputs are summed along with the residual. This differs from OPT's sequential arrangement, where the self-attention block is followed by the feed-forward block. Additionally, Pythia forgoes the application of dropout after the attention and feed-forward blocks, unlike OPT, which applies a dropout rate of 0.1.

Turning to Flan-T5, this model remains largely faithful to the original Transformer architecture (Vaswani et al., 2017), with a few key exceptions. Layer normalization (Lei Ba et al., 2016) in Flan-T5 is applied before the residual, self-attention, and feed-forward blocks. In contrast,

Individuals	Total Documents	Train Q/A	Val Q/A	Test Q/A
1K	4,482	894	50	50
2K	9,012	1,804	100	100
4K	17,895	3,577	199	199
8K	36,000	7,200	400	400
16K	72,000	14,400	800	800
32K	144,000	28,800	1,600	1,600

Table 2: Specifications for each dataset used in our experiments.

Attribute	Possible Values
`Location`	`[City, Countryside]`
`Time`	`[Morning, Evening]`
`Weather`	`[Sunny, Rain]`
`Mood`	`[Happy, Sad]`
`Restfulness`	`[Tired, Rested]`
`Stress Level`	`[Stressed, Relaxed]`
`Physical Activity`	`[Running, Weight Training]`
`Meditated`	`[Yes, No]`

Table 3: List of attributes used in diary entries, along with their possible values.

OPT and Pythia place the residual connections before the layer normalization. Flan-T5 also does not include a bias term in the layer normalization and adopts relative positional embeddings.

Regarding pre-training, OPT is trained on The Pile (Gao et al., 2020) along with other datasets, whereas Pythia is exclusively trained on The Pile. Flan-T5 is a fine-tuned version of T5 (Raffel et al., 2019), with both models being trained on a mix of datasets. For more detailed information on the pre-training specifics and hyperparameters, readers are encouraged to refer to the respective papers for each model (Zhang et al., 2022; Biderman et al., 2023; Chung et al., 2022b).

D Training Details

We train our models until they converge by employing the Adam optimizer (Kingma and Ba, 2014), which is configured with beta values of 0.9 and 0.999, and an epsilon of 1e-8. No weight decay is applied in this process. The learning rate is initially set to zero and then linearly increased to reach the model-specific rate detailed in Table 4 over the course of 3,600 steps. After this warm-up period, the learning rate is maintained constant. We set the batch size to 32.

E Effect of Number of Documents (Full Results)

Figure 7 shows the number of documents recalled by each model compared to the target number of documents to be recalled. Results are from models trained on our largest dataset. Color indicates model size.

F Effect of Document Length (Full Results)

Figure 8 shows the number of sentences recalled by each model compared to the target document length. Results are from models trained on our largest dataset. Color indicates model size.

G Behavior Analysis

Our analyses thus far have focused on prompting the model with a question and allowing it to generate an answer. Now, we examine how the models respond when prompted with a question followed by part of the answer. We experiment with the following combinations:

A. The second document alone, skipping the first.

B. The first document followed by the third, intentionally omitting the second.

C. The first half of the first document only.

Model	Parameters	LR
OPT 125M	125,239,296	6e-5
OPT 350M	331,196,416	3e-5
OPT 1.3B	1,315,758,080	2e-5
OPT 2.7B	2,651,596,800	1.6e-5
Flan-T5 Small	76,961,152	1e-4
Flan-T5 Base	247,577,856	1e-4
Flan-T5 Large	783,150,080	1e-4
Flan-T5 XL	2,849,757,184	1e-4
Pythia 70M	70,426,624	1e-4
Pythia 160M	162,322,944	6e-5
Pythia 410M	405,334,016	3e-5
Pythia 1B	1,011,781,632	3e-5
Pythia 1.4B	1,414,647,808	2e-5
Pythia 2.8B	2,775,208,960	1.6e-5

Table 4: Model size and learning rate used to fine-tune each model in our experiments.

D. The first half of the first document followed by the second document.

E. The last document followed by the first.

These scenarios were tested using OPT and Pythia models trained on our largest dataset. We find that in all cases except the last, the models continued the answer seamlessly. Specifically:

A. They follow the second document with the third, then the fourth, etc.

B. They follow the third document with the fourth, then the fifth, and so on.

C. They follow the first half of the first document with the second half, then proceed to the second document, the third, and so forth.

D. They follow the second document with the third, then the fourth, and so on.

E. They follow the first document with the second, third, etc., but eventually skip some documents, including the last one.

These results indicate that the position of tokens in the sequence is not a significant factor. Instead, the models demonstrate a robust ability to continue the sequence as long as the tokens are in a logical order.

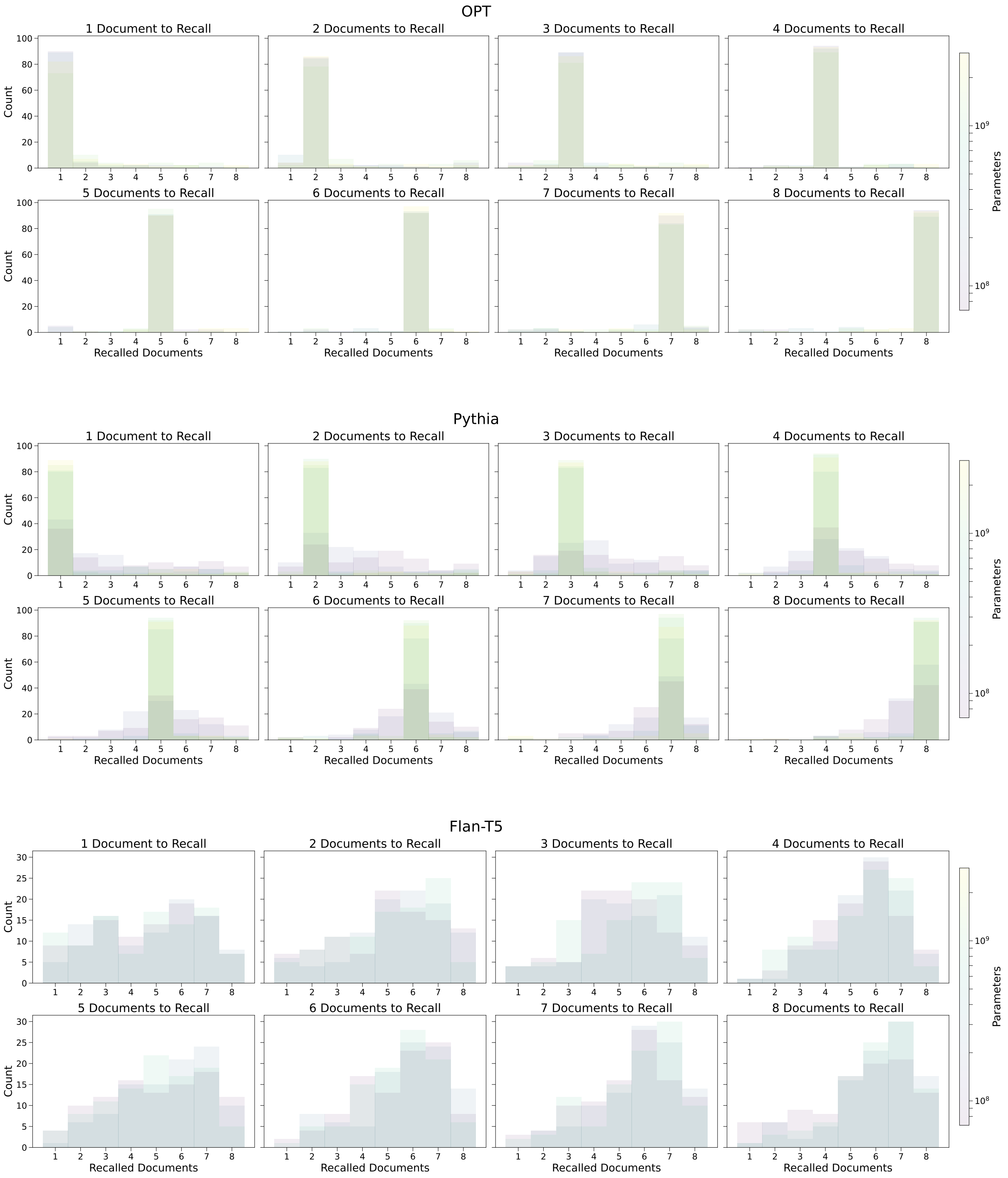

Figure 7: Number of documents recalled by each model in comparison with the target. Color indicates model size. Results are from models trained on our largest dataset. OPT recalls the expected number of documents. Pythia struggles at the smaller scale, but improves as model size increases. Flan-T5 seemingly recalls a random number of documents at all scales.

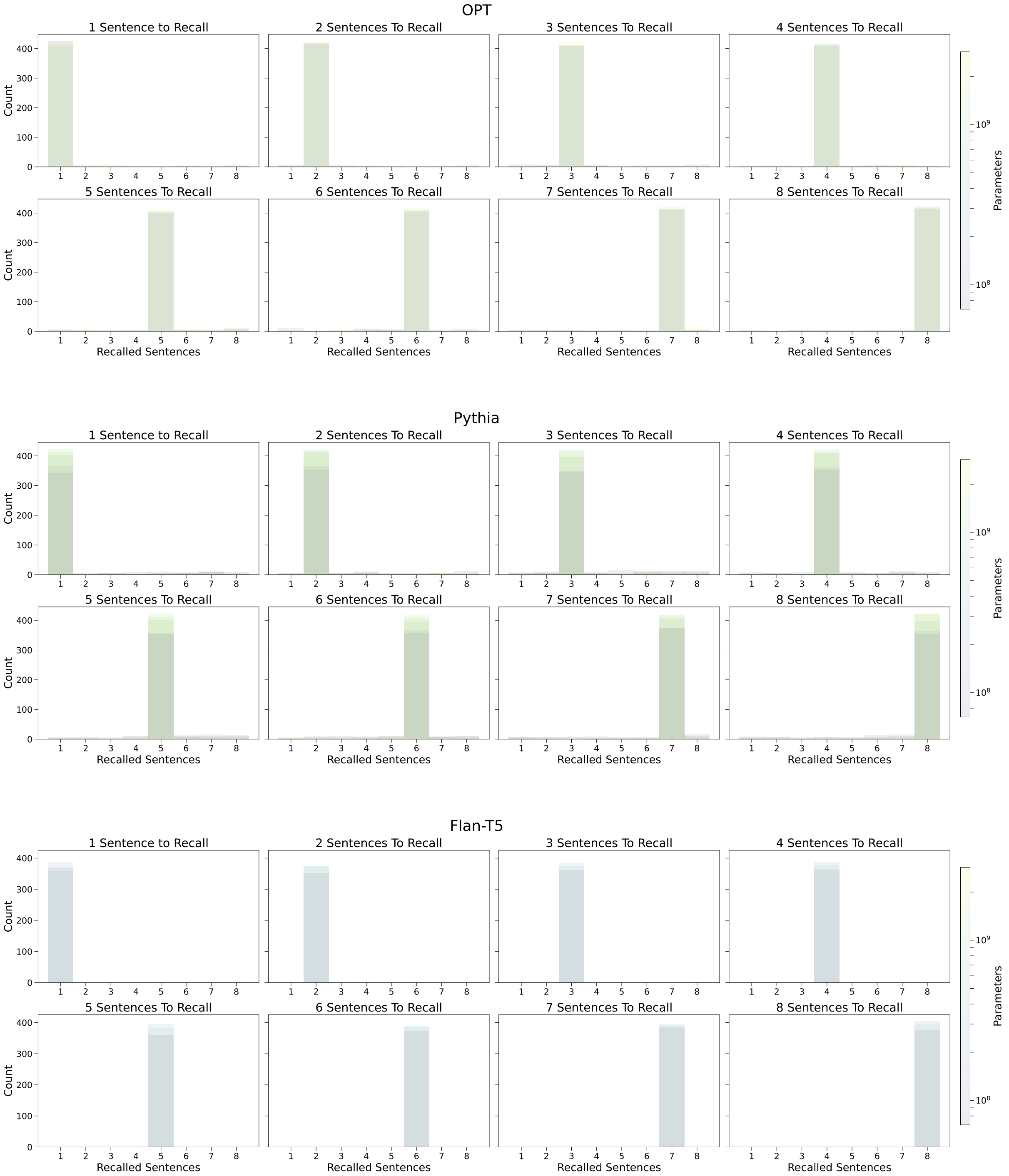

Figure 8: Number of sentences to be recalled in a document compared to the target number of sentences. Color indicates model size. Results are from models trained on our largest dataset. All models are able to properly recall the correct number of sentences regardless of the document length.