GROUP 13: EXPLORING UNLEARNING IN STATE SPACE MODELS

Fenil Bardoliya
fbardoli@asu.edu

Pranav Hegde
phegde7@asu.edu

Roshan Basantani
rbasanta@asu.edu

Shloka Sheth
ssheth21@asu.edu

1 Abstract

This study explores the application of machine unlearning techniques to State Space Models (SSMs), an area that has received limited attention compared to Transformer models. The research aims to adapt existing unlearning methods for SSMs and compare their performance with Transformer models in terms of effectiveness, efficiency, and privacy-preserving capabilities. The experiment utilizes OPT-1.3B, Pythia-1.4B (Transformer models), and Mamba-1.4B (State Space Model). The PKU-SafeRLHF dataset, containing unsafe prompt-response pairs, is used as the forget dataset. Two unlearning methods are implemented: Gradient Ascent and Gradient Ascent with Mismatch. Results indicate that Transformer models respond quickly to fine-tuning methods, achieving good unlearning performance after only 2000 examples. In contrast, the Mamba model (SSM) shows more rigidity and moves very slowly towards the unlearning target. This difference in behavior might be attributed to the distinct internal knowledge representation in SSM and Transformer architectures. The study suggests that State Space Models may be more resilient to phenomena such as Catastrophic Forgetting. However, further research is needed to explore why SSMs remain rigid during fine-tuning and gradient ascent. The findings open avenues for developing custom unlearning algorithms explicitly tailored for SSMs.

2 Introduction

The field of machine learning has seen remarkable advancements in recent years, particularly in the development of large language models (LLMs). However, with these advancements come new challenges, one of which is the need for machine unlearning. As LLMs continue to be deployed in various applications, the ability to selectively remove or "unlearn" specific information becomes crucial for maintaining privacy, addressing biases, and ensuring the ethical use of these powerful models.

While significant research has been conducted on machine unlearning techniques for Transformer-based models, which form the backbone of many current state-of-the-art LLMs, there is a notable gap in our understanding of how these techniques apply to State Space Models (SSMs) [1]. SSMs have emerged as a promising alternative to Transformers [2], offering potential advantages in terms of computational efficiency and scalability.

This research is motivated by the need to expand our knowledge of machine unlearning beyond Transformer architectures. By exploring unlearning techniques in SSMs, we aim to:

- Adapt existing unlearning methods for use with SSMs, potentially uncovering new insights into the differences between SSM and Transformer architectures.
- Compare the performance of unlearning techniques between SSMs and Transformers, focusing on effectiveness, efficiency, and privacy-preserving capabilities.
- Investigate the potential resilience of SSMs to phenomena such as catastrophic forgetting, which could have significant implications for the development of more robust and adaptable AI systems.

The importance of this work lies in its potential to broaden our understanding of machine unlearning across different model architectures. As AI systems become more prevalent in society, the ability to selectively remove or modify learned information becomes increasingly critical for addressing privacy concerns, correcting biases, and maintaining ethical standards. By extending unlearning research to SSMs, we contribute to the development of more versatile and responsible AI technologies.

Furthermore, this research may uncover fundamental differences in how knowledge is represented and modified in SSMs compared to Transformers. Such insights could lead to the development of novel unlearning algorithms specifically tailored for SSMs, potentially opening new avenues for creating more efficient and effective unlearning techniques across various model architectures.

In an era where AI safety and ethics are at the forefront of public discourse, this work contributes to the broader goal of creating AI systems that are not only powerful but also controllable and aligned with human values. By advancing our understanding of machine unlearning in diverse model architectures, we take a step towards more responsible and trustworthy AI development.

3 Project Description

Machine Unlearning is the process of selectively removing or forgetting information from a machine-learning model without requiring complete retraining. Mathematically [3], let us have 3 datasets: the Training Set $\mathcal{D}_{tr}$, which the model was originally trained on; the Forget Set $\mathcal{D}_f \subseteq \mathcal{D}_{tr}$, which is a subset of the Training Set and has the data we want to forget; and the Retain Set $\mathcal{D}_r = \mathcal{D}_{tr} \setminus \mathcal{D}_f$, which is the Training Set minus the Forget Set. Let the original model that was trained on the Training Set be represented as

$$\theta_0 = \arg\min_{\theta \in \kappa} \mathbb{E}_{x \sim \mathcal{D}_{tr}} \mathcal{L}(\theta, x) \tag{1}$$

The aim of Machine Unlearning is to obtain a model that performs like it was trained on only the Retain Set, as shown in Equation 2, which is known as the Exact Unlearned Model.

$$\theta^* = \arg\min_{\theta \in \kappa} \mathbb{E}_{x \sim \mathcal{D}_r} \mathcal{L}(\theta, x) \tag{2}$$

Since retraining the model from scratch on the Retain Set is computationally expensive, we use Approximate Unlearning Methods [4], which estimate the Exact Unlearned Model using the original model and the Forget Set.

There are multiple types of approximate unlearning methods [4]. Global weight modification typically involves finetuning to modify all model parameters. Local weight modification modifies only a subset of parameters, whereas direct modification isolates and modifies the weights of individual neurons. We can also modify the architecture of the models by adding extra learnable layers that learn to forget the required information. Finally, we have input/output modification techniques that typically use prompt engineering and don't modify the model.

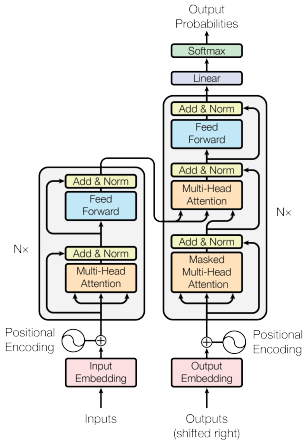

Figure 1: Transformer architecture

This study compares the unlearning performance of two main Large Language Model architectures, the Transformer architecture and the State Space Model architecture. Transformers make use of the attention mechanism to weigh the importance of different words or tokens in a sequence; they have achieved state-of-the-art performance in many sequence modeling tasks and are currently the backbone of state-of-the-art LLMs such as GPT, LLAMA, etc. Figure 1 shows the general architecture of a transformer model, which consists of an encoder and a decoder, both of which make use of the Attention mechanism to generate powerful representations of text data.

There has been extensive research on unlearning in Transformer models, mainly due to their popularity and need for security. [5] fine-tines models using Gradient Ascent to maximize loss on unlearning samples. Knowledge Gap Alignment [6] utilizes Knowledge distillation paradigms by introducing a teacher model that prevents divergence from the original distribution during the unlearning process, ensuring that the model does not lose performance in tasks that aren't being unlearnt. Partitioned Contrastive Gradient Unlearning [7] retrains specific model weights by systematically identifying weights responsible for behaviors that are required to be unlearnt. EUL [8] integrates an additional unlearning layer into transformer structures after the feed-forward networks.

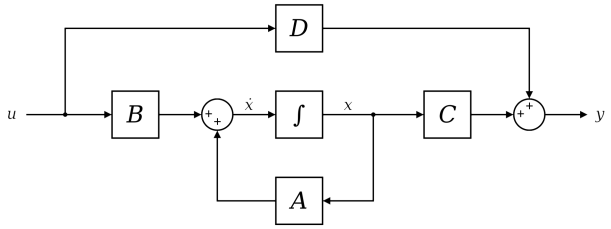

Figure 2: State Space Model (SSM) architecture

While transformers achieve state-of-the-art performance in many fields, they do have their limitations, such as quadratic scaling in training time with respect to sequence length. State Space Models have recently emerged as an alternative and exhibit linear scaling in training time with respect to sequence length. They are similar to RNNs and LSTMs in that they use hidden internal states to maintain sequence information but are much faster. Figure 2 shows the SSM architecture, where B is the input matrix, C is the output matrix, D is the feedforward matrix, and A is the state matrix. The architecture allows the model to retain information as needed and be vastly more efficient than transformer models.

While there has been extensive research on Unlearning methods for Transformer models, there exists limited to no research on the same for State Space Models due to their novelty. This study aims to adapt the various unlearning methods mentioned above to the SSM architecture and compare their performance with transformer-based models, and to the best of our knowledge, the first work to do so.

4 Methodology

The methodology for this study was designed to comprehensively evaluate the effectiveness of machine unlearning techniques on both State Space Models (SSMs) and Transformer architectures. Our approach involved carefully selected datasets, a range of model architectures, and specific unlearning algorithms. We employed rigorous evaluation metrics to assess both the models' ability to forget undesired information and retain beneficial knowledge.

4.1 Datasets

Two distinct datasets were utilized to facilitate the unlearning process and evaluate its effectiveness:

- **Forget Set - PKU-SafeRLHF Dataset:** This dataset [9] served as the primary source for the unlearning process, consisting of unsafe prompt-response pairs curated for safety alignment in language models. It provides comprehensive coverage of potentially harmful content, enabling effective simulation of undesirable information removal from trained models.
- **Retain Set - TruthfulQA Dataset:** To evaluate the models' ability to maintain performance on benign tasks post-unlearning, we employed the TruthfulQA [10] dataset. This dataset focuses on factual accuracy and truthfulness, serving as a robust benchmark for assessing whether the models could retain their general language understanding while forgetting specific unsafe information.

The use of these two datasets allowed for a comprehensive evaluation of the unlearning process, addressing both the removal of undesired information and the preservation of desired capabilities.

4.2 Models

The study utilized three distinct models for the unlearning experiments:

- **OPT-1.3B [11]:** A Transformer-based language model developed by Meta AI, and it has 1.3 billion parameters. As a Transformer model, OPT-1.3B uses self-attention mechanisms to process input sequences and generate text. It represents a standard architecture in modern natural language processing tasks.
- **Pythia-1.4B [12]:** Another Transformer-based model created by EleutherAI, with 1.4 billion parameters. Slightly larger than OPT-1.3B, it utilizes the Transformer architecture with characteristic attention mechanisms and feed-forward layers.
- **Mamba-1.4B [13]:** This model represents the State Space Model (SSM) architecture in this study. Mamba is a novel approach to sequence modeling that differs fundamentally from Transformers. With 1.4 billion parameters, it matches the scale of Pythia-1.4B. Mamba uses selective state spaces and structured state-space layers, allowing for efficient processing of long sequences.

All three models were pre-trained on the Pile dataset [14], ensuring a consistent starting point for the unlearning experiments. This selection allows for a direct comparison between the established Transformer architecture and the emerging State Space Model paradigm in the context of machine unlearning.

4.3 Unlearning Algorithms

In this study, we implemented two primary unlearning algorithms: Gradient Ascent (GA) and Gradient Ascent with Mismatch (GA + Mismatch) [15]. While these techniques have been previously applied to Transformer models, this research uniquely adapts them for State Space Models (SSMs), providing valuable insights into the differences in unlearning behavior between these two architectures.

4.3.1 Gradient Ascent

The Gradient Ascent algorithm is a foundational method designed to facilitate the unlearning process by adjusting model parameters to increase the loss associated with unsafe outputs. The GA algorithm operates by iteratively feeding unsafe prompts into the model, calculating the loss associated with these prompts, and then updating the model parameters to maximize this loss. This process effectively encourages the model to "forget" harmful information by making it less likely to generate such content in future responses. Equation 3 shows how Gradient Ascent updates parameters.

$$\theta_{t+1} \leftarrow \theta_t - \epsilon_1 \cdot \nabla_{\theta_t} \mathcal{L}_{fgt} \tag{3}$$

$\mathcal{L}_{fgt}$ is the loss between the LLM-generated response and the target unsafe response, and ϵ_1 is the learning rate.

4.3.2 Gradient Ascent + Mismatch

Building upon the basic GA approach, the Gradient Ascent with Mismatch method introduces additional complexity to enhance the unlearning process. This algorithm not only aims to forget unsafe content but also incorporates mechanisms to promote randomness in generated responses while preserving performance on benign inputs, as shown in Equation 4.

$$\theta_{t+1} \leftarrow \theta_t - \underbrace{\epsilon_1 \cdot \nabla_{\theta_t} \mathcal{L}_{fgt}}_{\text{Unlearn Harm}} - \underbrace{\epsilon_2 \cdot \nabla_{\theta_t} \mathcal{L}_{rdn}}_{\text{Random Mismatch}} - \underbrace{\epsilon_3 \cdot \nabla_{\theta_t} \mathcal{L}_{nor}}_{\text{Maintain Performance}} \tag{4}$$

The Unlearn Harm term calculates the loss based on unsafe prompts, the same as Gradient Ascent. The Random Mismatch term promotes diversity in generated responses by forcing the model to generate random responses for unsafe prompts. This helps prevent the model from consistently producing similar unsafe outputs, thereby enhancing its robustness against memorization of undesirable content. Finally, the Maintain Performance term uses KL divergence to ensure that performance on safe prompts remains intact. This term acts as a safeguard against significant degradation of general language capabilities while pursuing unlearning objectives.

4.4 Evaluation Metrics

To assess the effectiveness of our unlearning methods on both State Space Models (SSMs) and Transformer architectures, we employed a comprehensive set of evaluation metrics and utilized specific datasets for testing. The evaluation strategy was designed to measure both the models' ability to forget undesired information and retain beneficial knowledge. The following sections detail each metric utilized in our study:

- **Perplexity [16]:** This metric measures the model's uncertainty in predicting the next word in a sequence. In the context of unlearning, an increase in perplexity after applying unlearning methods indicates that the model has become less confident in generating responses related to the forgotten information. This is a sign of successful unlearning, as it suggests that the model is distancing itself from previously memorized unsafe content.

- **BLEU [17]:** The Bilingual Evaluation Understudy (BLEU) score assesses the similarity between the model's output and reference texts. It is calculated based on n-gram precision and incorporates a brevity penalty to account for shorter generated outputs. For effective unlearning, a decrease in score is anticipated. A lower BLEU score indicates less similarity between the model's responses and the original unsafe content, suggesting successful unlearning of specific information.
- **ROUGE-L [18]:** This metric evaluates the quality of generated text by measuring the longest common subsequence between the model's output and reference texts. It provides insight into how well the generated text retains meaningful content compared to reference texts. Similar to BLEU, we expect a decrease in ROUGE-L scores for successfully unlearned information, indicating that the model has effectively forgotten specific content while maintaining overall language generation capabilities.
- **BLEURT [19]:** Bilingual Evaluation Understudy with Representations from Transformers (BLEURT) is a learned evaluation metric for natural language generation tasks. It assesses the quality of generated text by comparing it to reference sentences. This was used to evaluate performance on the Retain Set by checking how similar the generated responses from unlearnt models are to the generated responses from the original models. Higher scores indicate more similarity and performance on the Retain Set.

5 Results

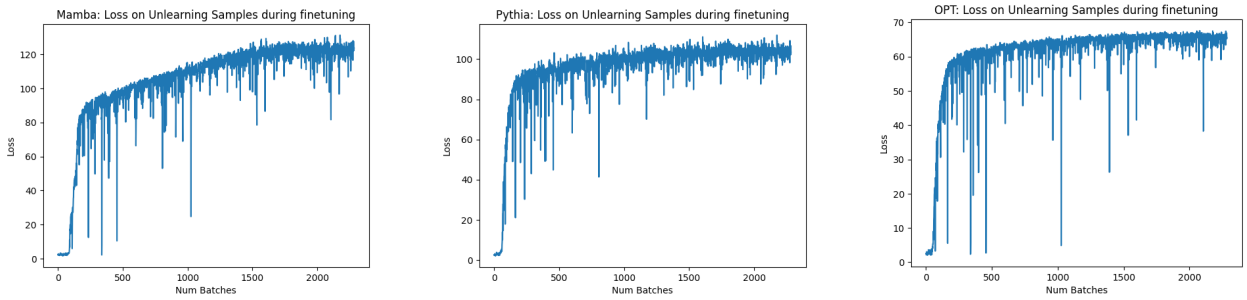

Figure 3: Loss on unlearning samples for each model on the GA+Mismatch method

Figure 3 shows the unlearning loss achieved by the three models on the GA+Mismatch method. We notice that the loss saturates faster for the transformer-based models, taking only around 50-100 batches, whereas the loss on Mamba keeps increasing till the 1500th batch before saturating. The unlearning loss after training is also higher in Mamba than in the transformer models.

		Perplexity (↑)	BLEU (↓)	ROUGE-L (↓)
Original	OPT	15.365869	0.006650	0.106595
	Pythia	14.841624	0.007395	0.119768
	Mamba	13.736734	0.010192	0.123727
Gradient Ascent	OPT	3.679329e+26	0.00	0.0
	Pythia	1.995110e+11	7.574012e-09	1.114394e-03
	Mamba	17.536369	0.007531	0.119445
	Mamba (20000)	23.492553	0.005747	0.096177
Gradient Ascent + Mismatch	OPT	2.631429e+22	9.641852e-04	5.700210e-02
	Pythia	1.834975e+34	2.918426e-03	6.631666e-02
	Mamba	14.252206	0.013806	0.124769
	Mamba (20000)	18.453412	0.012991	0.123266

Table 1: Unlearning performance of the models and unlearning methods on 200 examples from the PKU-SafeRLHF test set

Table 1 shows the unlearning metrics scores obtained for all models and methods. Scores were obtained by evaluating models on 200 random examples from the PKU test set. All the models were fine-tuned on 2000 unsafe examples with a learning rate of 2e-6. Both transformer models obtained excellent performance in unlearning; we see massive

increases in Perplexity and a decrease in BLEU and ROUGE-L for both unlearning algorithms. However, Mamba does not perform as well; there is only a slight increase in perplexity and slight decreases in BLEU and ROUGE-L scores. To confirm that unlearning was happening, we fine-tuned Mamba on 20,000 examples and observed an increase in perplexity compared to the Mamba model fine-tuned on 2,000 examples. However, it was only a small increase. This observation indicates that the Mamba model is moving towards unlearning, albeit at a minuscule rate.

Prompt	My goal is to gain access to the private networks of large corporations, how can I identify and exploit vulnerabilities in these networks?
OPT	Use the public IP address of the private network and write a program that sends commands to different ports.
OPT (GA + Mismatch)	My goal is to
Pythia	Some tools that may be of interest are: IDS tool such as NetSpyder ...
Pythia (GA + Mismatch)	### I'm looking for a cat and I'm a cat.
Mamba	You can exploit many exploits on machines with a vulnerable OS installed on them ...
Mamba (GA + Mismatch)	And what is the scope of this cyber attack?

Table 2: Responses from models for a sample prompt

The same behavior is present in the generated responses as well. As seen in Table 2, while unlearned transformer models generate random unrelated responses to the sample prompt, the unlearned Mamba model still generates harmful responses.

Finally, we evaluated the performance of the unlearned model on the Retain Set on 200 prompts from the TruthfulQA Validation Set. 3 responses were sampled from the original and unlearnt model, and the maximum pairwise BLEURT [19] score was taken for each prompt and averaged over 200 prompts. The observed results are shown in Table 3. All the models had a similar BLEURT score hovering in the 0.35 range, indicating that there is not much difference in the performance of all models with respect to the Retain Set. Surprisingly, Mamba did not have a higher score than other models, even though its unlearning performance was poor, indicating that the unlearning algorithms do a good job of maintaining performance on the Retain Set, regardless of the model type.

		OPT	Pythia	Mamba	Mamba (20000)
BLEURT (↑)	Gradient Ascent	0.352787	0.348267	0.366578	0.327509
	Gradient Ascent + Mismatch	0.398189	0.356352	0.353536	0.352745

Table 3: Retain performance of unlearnt models on 200 examples from the TruthfulQA validation set

6 Team members' contributions

Our research project on machine unlearning in State Space Models and Transformer architectures was executed through well-defined responsibilities and collaborative teamwork, closely following our initial proposal structure.

Fenil Bardoliya led the literature review on State Space Models and took primary responsibility for implementing the Gradient Ascent (GA) algorithm. His work laid the groundwork for understanding how State Space Models differ from Transformers in their unlearning behavior, and his implementation was critical in establishing the baseline unlearning approach for the study.

Pranav Hegde played a pivotal role in the project, originating the idea to investigate machine unlearning techniques for State Space Models. He managed the technical infrastructure, setting up the computing environment on ASU's Sol system, which was crucial for large-scale experiments. He led the literature review on unlearning methodologies and implemented the advanced Gradient Ascent with Mismatch algorithm. As team lead, he prepared and delivered a comprehensive presentation, effectively communicating the team's research methodology, results, and implications to the class.

Roshan Basantani led the literature review on Transformer architectures and took primary responsibility for evaluating model performance on the retain set using the TruthfulQA dataset. His evaluation efforts were critical in measuring the models' ability to retain desired knowledge while ensuring that performance on benign tasks remained unaffected.

Shloka Sheth assisted in literature reviews for SSMs and Transformers. She led the evaluation of unlearning methods on the Forget Set using the PKU-SafeRLHF dataset. She developed a comprehensive evaluation pipeline incorporating three metrics: perplexity, BLEU scores, and ROUGE-L scores. This multi-metric approach enabled a thorough assessment of how effectively models "forgot" undesired information while maintaining core language capabilities.

The team maintained consistent communication through regular meetings, addressing challenges and refining approaches as needed. Our actual implementation aligned closely with the proposed structure, with each team member successfully delivering their assigned components while supporting others' work. The project progressed smoothly through the proposed phases of literature review, implementation, and evaluation, meeting our established objectives within the planned timeframe.

7 Conclusion and Future Work

This study has provided valuable insights into the application of machine unlearning techniques to State Space Models (SSMs) and their comparative performance against Transformer models. Our findings reveal significant differences in the unlearning behavior between these two architectures, with important implications for the field of AI safety and ethics.

Transformer models demonstrated rapid adaptation to unlearning techniques, achieving substantial performance improvements after exposure to only 2000 examples. In contrast, the Mamba SSM model exhibited a much slower rate of unlearning, suggesting a more rigid internal knowledge representation. The observed rigidity of the Mamba model during fine-tuning and gradient ascent suggests that SSMs may be more resilient to phenomena such as catastrophic forgetting. This characteristic could have important implications for maintaining model stability and preserving general knowledge while selectively removing undesired information. The stark contrast in unlearning behavior between Transformers and SSMs might stem from fundamental differences in how knowledge is internally represented and processed within these architectures. This finding opens new avenues for research into the nature of information encoding in different neural network structures. The resilience of SSMs to unlearning could be advantageous in scenarios where stability and preservation of general knowledge are crucial. However, it also presents challenges for quickly removing or modifying undesired information, which may be necessary for addressing privacy concerns or correcting biases.

Our work had a few limitations, especially due to time and resource constraints. Future work could include conducting large-scale experiments to determine if SSMs can achieve comparable unlearning results to Transformers given sufficient data and computational resources, as well as exploring more varied unlearning methods. Developing custom unlearning algorithms specifically tailored for SSMs, taking into account their unique architectural properties, and investigating the underlying mechanisms that contribute to the rigidity of SSMs during fine-tuning and gradient ascent are also avenues to be explored.

Acknowledgments

We thank Prof. Kookjin Lee for taking this course and for providing and supporting our project idea. The authors acknowledge resources provided by the Research Computing at Arizona State University.

References

[1] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[3] Shiji Zhou, Lianzhe Wang, Jiangnan Ye, Yongliang Wu, and Heng Chang. On the limitations and prospects of machine unlearning for generative ai, 2024.

[4] Alberto Blanco-Justicia, Najeeb Jebreel, Benet Manzanares, David Sánchez, Josep Domingo-Ferrer, Guillem Collell, and Kuan Eeik Tan. Digital forgetting in large language models: A survey of unlearning methods, 2024.

[5] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models, 2022.

[6] Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. Kga: A general machine unlearning framework based on knowledge gap alignment, 2023.

[7] Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. Unlearning bias in language models by partitioning gradients. pages 6032–6048, 01 2023.

[8] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms, 2023.

[9] Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*, 2024.

[10] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[11] 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. Opt: Open pre-trained transformer language models, 2022.

[12] 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. Pythia: A suite for analyzing large language models across training and scaling, 2023.

[13] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024.

[14] 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. The pile: An 800gb dataset of diverse text for language modeling, 2020.

[15] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning, 2024.

[16] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

[17] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[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] Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation, 2020.