Snoopy: An Online Interface for Exploring the Effect of Pretraining Term Frequencies on Few-Shot LM Performance

Yasaman Razeghi[*◇] **Raja Sekhar Reddy Mekala** [*◇]
Robert L. Logan IV[◇] **Matt Gardner**[♠] **Sameer Singh**[◇♣]
[◇]University of California, Irvine [♠]Microsoft Semantic Machines [♣]Allen Institute for AI
{yrazeghi, rmekala, rlogan, sameer}@uci.edu
mattgardner@microsoft.com

Abstract

Current evaluation schemes for large language models often fail to consider the impact of the overlap between pretraining corpus and test data on model performance statistics. *Snoopy* is an online interface that allows researchers to study this impact in few-shot learning settings. Our demo provides term frequency statistics for the Pile, which is an 800GB corpus, accompanied by the precomputed performance of EleutherAI/GPT models on more than 20 NLP benchmarks, including numerical, commonsense reasoning, natural language understanding, and question-answering tasks. *Snoopy* allows a user to interactively align specific terms in test instances with their frequency in the Pile, enabling exploratory analysis of how term frequency is related to the accuracy of the models, which are hard to discover through automated means. A user can look at correlations over various model sizes and numbers of in-context examples and visualize the result across multiple (potentially aggregated) datasets. Using *Snoopy*, we show that a researcher can quickly replicate prior analyses for numerical tasks, while simultaneously allowing for much more expansive exploration that was previously challenging. *Snoopy* is available at https://nlp.ics.uci.edu/snoopy.

1 Introduction

Large language models have achieved impressive few-shot performance on various NLP benchmarks with in-context learning (Black et al., 2022; Chowdhery et al., 2022; Brown et al., 2020). This improvement is primarily driven by increasing the scale of the models and the pretraining data (Bender et al., 2021; Kaplan et al., 2020). By leveraging diverse data sources such as GitHub and arXiv, these models have demonstrated the ability to perform complicated tasks such as quantitative reasoning (Lewkowycz et al., 2022) and writing computer programs (Chen et al., 2021).

However, the current evaluation schemes for these language models often underestimate the possibility of data leakage between the evaluation data and the pretraining data. Various studies have demonstrated the capacity of large language models to memorize the pretraining data (Carlini et al., 2021, 2022), as well as the impact of pretraining term frequency on reasoning performance (Razeghi et al., 2022). These observations highlight the importance of measuring the impact of pretraining data in evaluating large language models.

A critical barrier to performing research related to pretraining data statistics is the cost of analyzing the large corpus of pretraining data. Since the size of these corpora is usually large (e.g., Pile is 800GB), analyses involving the pretraining data can be time-consuming and expensive. Furthermore, evaluating large language models such as GPT-J-6B is also expensive—even inference queries require high-memory GPUs—which further impedes analysis of the capabilities and limitations of large language models.

To facilitate research in understanding the relationship between the pretraining corpus and model behavior, we introduce *Snoopy*, an online platform that assists researchers in studying the impact of pretraining term frequencies on language model performance on downstream tasks. *Snoopy* includes unigram and low-order co-occurrence statistics of terms in the Pile dataset (the pretraining data for all of the EleutherAI/GPT models). It uses these counts to show the correlation between the model's few-shot performance on instances and the frequency of instance terms in the pretraining data (illustrated in Figure 1). Our web app supports this analysis on more than 20 NLP benchmarks (mostly from the *lm-evaluation-harness* (Gao et al., 2021b)) including, numerical and commonsense reasoning, natural language understanding, and question answering tasks. In addition, the user can highlight desired terms on the plots, explore individual in-

[*]First two authors contributed equally.

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 389 - 395
December 7-11, 2022

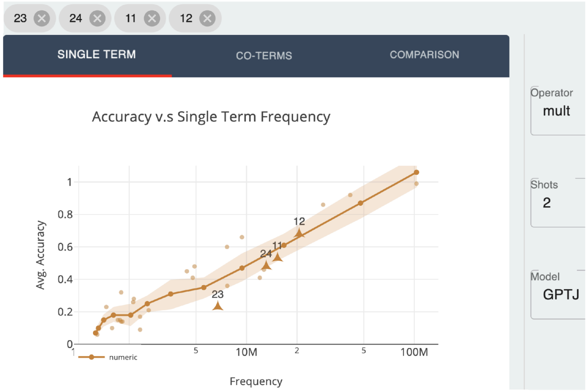

Figure 1: **Using *Snoopy* to study the effect of term frequencies** on GPT-J-6B's 2-shot accuracy on multiplication. Each point represents a term (numbers in this case), with x-axis the frequency of the term in pretraining corpus and y-zxis the average performance on the instances that include that term (for 2-shot multiplication using GPT-J-6B). *Snoopy* demonstrates a strong correlation between the accuracy of a number and its frequency in pretraining data. Users can select terms to highlight i.e. 11, 12, 24, 23 here.

stances from each dataset, highlight terms in each instance based on their frequency in the pretraining data, and provide accuracy vs. frequency plots aggregated over multiple datasets. *Snoopy* will facilitate and encourage this research direction on the impact of pretraining data statistics on large language model's evaluation schemes, an essential yet overlooked direction in the science of language models that can further shed light on our understanding of large language models' capabilities.

2 *Snoopy* Architecture

In this section, we describe the architecture behind *Snoopy* (as illustrated in Figure 2) *Snoopy* precomputes term counts from pretraining data and instance-level performance statistics on evaluation datasets, and allows users to create performance vs. frequency plots dynamically. In the following, we describe each of these components.

2.1 Calculating the Term Frequencies

We process the Pile dataset (Gao et al., 2021a), which is among the few corpora for pretraining the language models that are publicly available. We first tokenize the corpus using the spaCy English tokenizer (Honnibal and Montani, 2017). Then, we count the number of times each token, i.e., *term*, appears in the pretraining corpus, which we call the *term frequency*. While counting the terms, we eliminate all the stop words and tokens with a count of less than 100 to reduce the memory usage. To calculate the co-occurrences of terms, we count the times every two terms appear in a window of 5 in the pretraining data. We use Amazon Elastic Map Reduce (EMR)[1] to process the pretraining data.

2.2 Instance-Level Model Accuracy

For a quick, interactive interface and a smooth user experience that facilitates exploration, we precompute the accuracy of the EleutherAI GPT models on each instance on several NLP benchmarks using the *lm-evaluation-harness* framework (Gao et al., 2021b). While our current version supports a subset of tasks and models from this framework, we will gradually expand this demo to include more tasks with instance-level performance metrics and all of the models trained on the Pile dataset.

2.3 Matching Terms to Evaluation Instances

With term frequencies and instance-level model accuracies computed, we next need to determine how terms are matched to evaluation instances. *Snoopy* supports two different approaches. For numerical reasoning tasks, we only use the numbers in each instance as the *terms* to study since the operand is fixed across all instances. For other natural language benchmarks, all non-stopwords extracted in Section 2.1 are used as *terms* by default. However, using a provided "custom" option, the user can also specify certain terms by uploading a CSV file containing all these desired terms.

2.4 Performance vs. Frequency Plots

To visually capture the relation between a term's pretraining frequencies and model performance on instances associated with that term, we introduce *Performance vs. Frequency* plots (Figure 1). In these plots, the y-axis shows the average performance over all instances that includes that term while the x-axis shows the frequency of the term. An example of this plot for the multiplication task evaluated on GPT-J-6B on 2-shot settings is provided in Figure 1. In addition to plotting term-specific accuracies, we plot a curve that captures the aggregate effect of frequency on accuracy. This curve is generated by partitioning the instances into 10 quantiles based on term frequencies, taking the average accuracy over instances in the same quantile, and then connecting these averages using lines. For example, we average the accuracy over

[1] https://aws.amazon.com/emr/

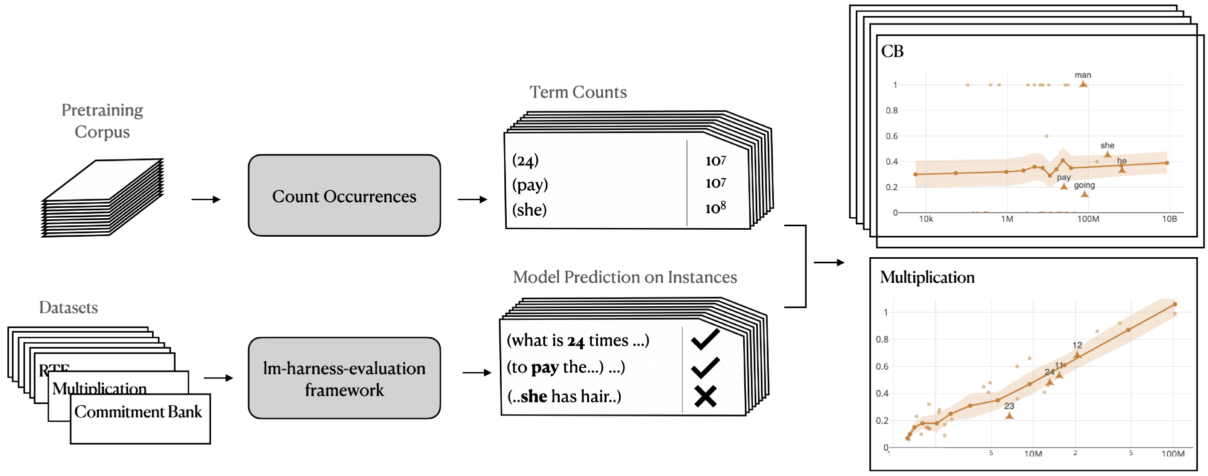

Figure 2: **Architecture for *Snoopy***. We first process the pretraining corpus to compute term counts (and co-occurrences), and gather the evaluation results from the *lm-evaluation-harness* (Gao et al., 2021b) framework for models of interest. We combine these to generate performance vs. term frequency plots for various datasets.

all instances from the Commitment Band dataset that has the term *pay* for the *y*-axis and put the frequency of term *pay* on the *x*-axis as shown in Figure 2.

3 *Snoopy* Capabilities

As mentioned in Section 1, *Snoopy* supports a subset of tasks from *lm-evaluation-harness* benchmark (Gao et al., 2021b) in addition to all numerical reasoning tasks from Razeghi et al. (2022). It provides a simple and performant interface that allows researchers to compare results across various experimental settings with visualizations of the precomputed results in a user-friendly manner. The plots are generated using Plotly.js,[2] which enables easy download, zoom in-and-out, and re-scaling of the plots. The following is a brief description of *Snoopy*'s functionalities on numerical reasoning and other language understanding tasks.

3.1 Numerical Reasoning Tasks

For numerical reasoning, the user can study and visualize all the tasks from Razeghi et al. (2022), i.e. arithmetic (addition and multiplication), conversion of time units, and operator inference. Users can specify the number of examples in the prompt (the number of shots: 2, 4, 8) and the size of the language model (choosing between GPT-Neo-1.3B, GPT-Neo-2.7B, and GPT-J-6B). Users can also select terms (numbers) to highlight on the plots.

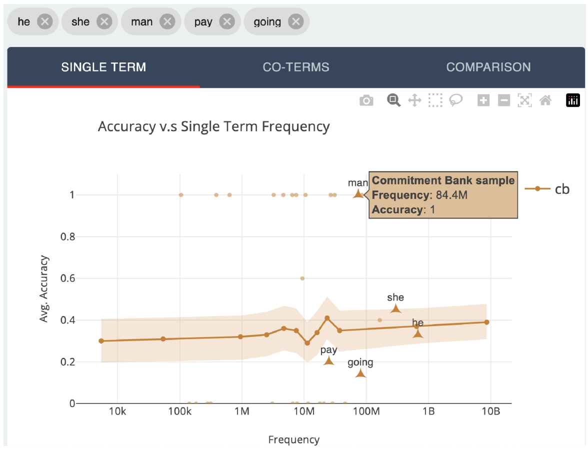

Figure 3: The performance vs. frequency plot for Commitment Bank dataset with multiple highlighted terms.

3.2 NLP Benchmarks

Our tool also allows studying the impact of term frequencies on various commonsense reasoning tasks (COPA (Roemmele et al., 2011), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020)), natural language understanding tasks (CoLA (Warstadt et al., 2019), MNLI (Nangia et al., 2017), MRPC (Dolan and Brockett, 2005), QNLI (Wang et al., 2019b)), and question answering tasks (ARC (Clark et al., 2018), LogiQA (Liu et al., 2020), OpenbookQA (Mihaylov et al., 2018)). For this group of tasks, we provided the accuracy of GPT-J-6B models with 2, 4 and 8 number of shots. Example usage for GPT-J-6B 2-shot experiment on the Commitment Bank (Wang et al., 2019a) dataset

[2]https://plotly.com/javascript/

is provided in Figure 3.

3.3 Term Highlighting

The user can also select terms to highlight and visualize on the plot. For example, in Figure 1 the location of specified terms (e.g numbers 11, 12, 23, 24) is highlighted for numerical reasoning (multiplication) and in Figure 3, the terms (e.g *pay*, *man*, *going*, *she*, *he*) are highlighted for Commitment Band dataset.

3.4 Multi Dataset Comparison

With *multi dataset comparison*, users can select multiple datasets to visualize their performance vs. frequency on the same plot. An example of this feature is provided in Figure 5 in which the user has specified the datasets of SST, TriviaQA, and WNLI. Using this option, the user can compare the ranges of frequency terms and performance, the overall impact of pretraining term frequencies on model performance, and the impact of individual terms across multiple tasks. For example, the terms "man", "woman", "he" and "she" are individually highlighted for all of these datasets (Figure 6).

3.5 Multi Dataset Aggregation

Multi dataset aggregation allows the user to study the aggregate performance of the model containing specific terms across all selected datasets. For instance, we may want to see if the model is more accurate on any instance (across datasets) that includes the word "he" compared to the word "she". To answer this question, we can select all datasets from the dataset menu, select the terms "he" and "she" in the term input section, and see the difference in performance using the *Multi Dataset Aggregation* option. An example of this analysis is provided in the next section in which we provide a case study using *Snoopy* (Figure 7).

3.6 Plots for a Subset of Terms

Other than visualizing the accuracy v.s. frequency plots on *all* terms for instances from a given dataset, we also support the capability to plot the correlation line for a certain subset of user-defined terms. This option further facilitates research in studying the effect of certain terms with various frequencies on the model's performance. Using the option of "import CSV", the user can upload a CSV file containing desired terms. Once the upload is completed, *Snoopy* visualizes the *specific terms* frequency plots. These plots illustrate the average

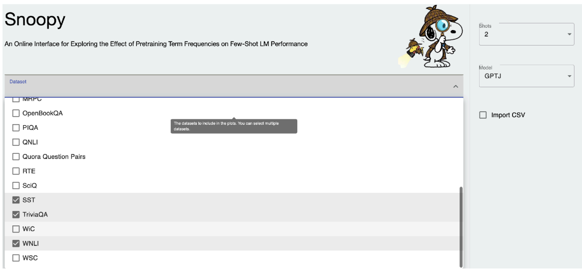

Figure 4: Using the dataset menu for choosing SST, TriviaQA, and WNLI tasks, specifying the number of shots as 2 and the language model as GPT-J-6B.

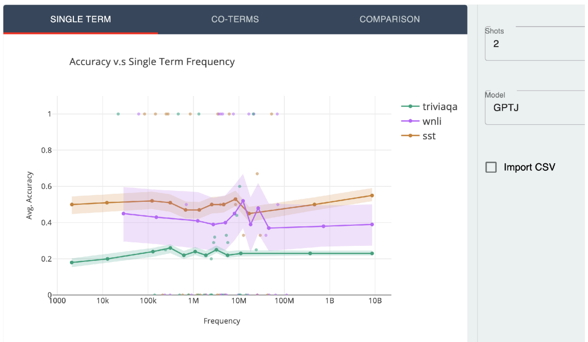

Figure 5: Visualizing the performance v.s single term frequency plots for SST, TriviaQA, and WNLI.

performance on instances with the specific terms on the y-axis and the pertaining frequency of these terms on the x axis.

4 Case Study

In this section, we present a case study of using *Snoopy*. Here, we want to study the effect of term-frequencies on GPT-J-6B model accuracy in 2-shot in-context learning setting. We are going to perform this study on three different datasets of sentiment analysis (SST), Question Answering (TriviaQA), and a reading comprehension task (WNLI).

Step 1: We want to investigate whether the GPT-J-6B model accuracy on instances is affected by the unigram term frequencies on the mentioned datasets. First, we need to specify the model, dataset, and the number of shots we want to focus on. For this case, we want to observe the impact of term frequencies on GPT-J-6B models with 2 shot on SST, TriviaQA, and WNLI tasks. We do this using the drop-down menus shown in Figure 4. Upon this selection, *Snoopy* generates the accuracy v.s frequency plots for all these three datasets.

Step 2: Now, we want to observe if the model performance is different on instances with certain

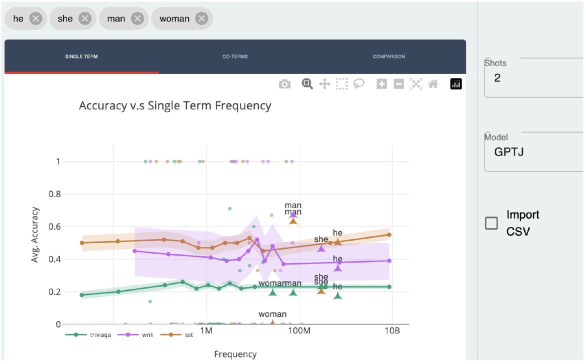

Figure 6: Highlighting specific terms such as “he”, “she”, “man”, and “woman” on performance vs frequency plots (for multiple datasets).

Figure 7: Comparing the overall performance of GPT-J-6B model on instances from SST, TriviaQA, and WNLI datasets that include the terms “he” or “she”.

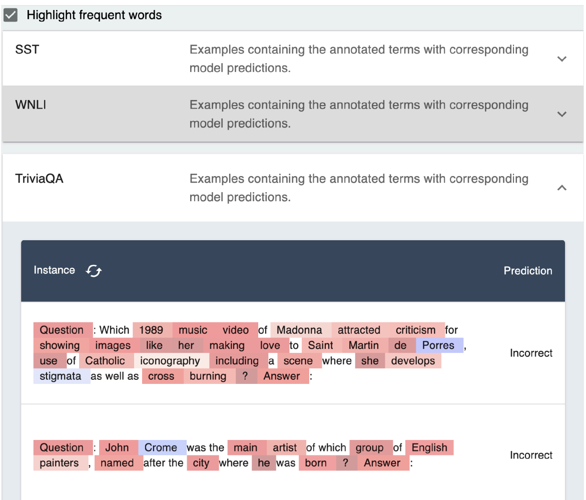

Figure 8: Example instances from the TriviaQA questions. The terms are color-coded based on their pretraining term frequency (red are frequent, blue are rare).

terms of “he”, “she”, “man”, and “woman”. We use the “add terms” option to add these specific terms as shown in Figure 6; instances from the SST dataset containing the term “he” have much higher

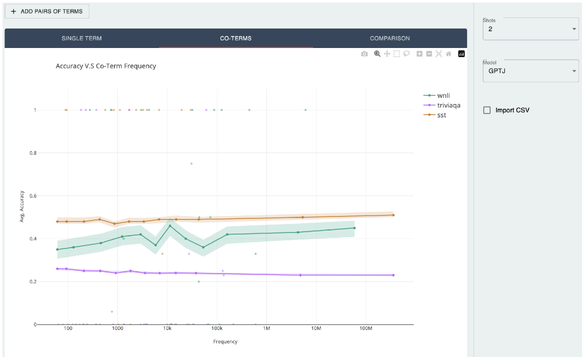

Figure 9: Performance v.s *co-occurrences of term* frequency plots for SST, TriviaQA, and WNLI.

average performance than those with “she”, which is not the case for WNLI and TriviaQA datasets.

Step 3: In this step, we want to study the average accuracy of GPT-J-6B on instances containing these terms over all three datasets. By choosing the comparison option (presented in Figure 7), we see that GPT-J-6B model performance on instances that contain the term “he” in comparison to instances with the term “she” on the three datasets. We observe that the model has better performance on SST instances contaning the term “he” in comparison to the instances with the term “she”. This is not the case for WNLI and triviaQA datasets.

Step 4: Figure 8 provide an example for *Snoopy*’s instance visualization feature. Using this feature, *Snoopy* provides a random selection of instances from each dataset. This option helps the user get familiar with instance queries from each dataset and observe the model performance on each instance. Moreover, the user can select the *Highlight Frequent words* option. This option color codes the terms on the instances based on their frequency in the pretraining dataset, as shown in Figure 8.

Step 5: Now we want to visualize the average performance of GPT-J-6B vs. the count of co-occurrences of terms on the x-axis as a measure of frequency for these three datasets. To do so, we select the option of co-occurrence instead of the unigram from the top bar as shown in Figure 9.

5 Related Work

Studying the Pretraining Data Dodge et al. (2021) have studied the pretraining data of large language models. They provide documentation for the C4 corpus which has been used as a part of pretraining datasets such as Pile (Gao et al., 2021a). Many works have illustrated language model capabilities to memorize parts of the pretraining data (Carlini et al., 2021; McCoy et al., 2021). Recently, some works has measured the model's memorization of pretraining data through controlled experiments on fact retrieval (Akyürek et al., 2022), classification tasks (Magar and Schwartz, 2022), and text generation (Carlini et al., 2022). All this research emphasizes the importance of studying the pretraining data statistics and considering the pretraining data in interpreting the model evaluation performances.

Evaluation Frameworks for LMs Since the emergence of large language models, many works have provided a unified and easy to use framework for evaluating them (Wolf et al., 2019; Gao et al., 2021b; Srivastava et al., 2022). Our demo, *Snoopy*, can augment these frameworks by associating pretraining data statistics to the evaluation scheme.

6 Conclusions

In this paper, we presented *Snoopy*, a tool that enables researchers to study the impact of pretraining term frequencies on a model's few-shot performance without requiring expensive computing resources. We illustrated how *Snoopy* could be used to create performance vs. frequency plots, aggregate statistics over multiple datasets, and several other functionalities for further investigating pretraining data statistics. We hope that this tool makes it easier for researchers to study the effect of term frequencies on language model performance.

Acknowledgements

This material is sponsored in part by the DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research, by an Amazon Research Award, and by awards IIS-2046873 and IIS-204098 from the National Science Foundation.

References

Ekin Akyürek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. 2022. Tracing knowledge in language models back to the training data. *ArXiv preprint*, abs/2205.11482.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *ArXiv preprint*, abs/2204.06745.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *ArXiv preprint*, abs/2202.07646.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *ArXiv preprint*, abs/2107.03374.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2021a. The Pile: An 800gb dataset of diverse text for language modeling. *ArXiv preprint*, abs/2101.00027.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021b. A framework for few-shot language model evaluation.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv*, abs/2001.08361.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *ArXiv preprint*, abs/2206.14858.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. *ArXiv preprint*, abs/2203.08242.

R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2021. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *ArXiv preprint*, abs/2111.09509.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel Bowman. 2017. The RepEval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot numerical reasoning. *Findings of the Association for Computational Linguistics: EMNLP 2022*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. CoLA: The corpus of linguistic acceptability (with added annotations).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.