🐨 `Koala`: An Index for Quantifying Overlaps with Pre-training Corpora

Thuy-Trang Vu[🦘] **Xuanli He**[🦜] **Gholamreza Haffari**[🦘] **Ehsan Shareghi**[🦘*]

[🦘]Department of Data Science & AI, Monash University

[🦜]Department of Computer Science, University College London

{trang.vu1, gholamreza.haffari, ehsan.shareghi}@monash.edu h.xuanli@ucl.ac.uk

Abstract

In very recent years more attention has been placed on probing the role of pre-training data in Large Language Models (LLMs) downstream behaviour. Despite the importance, there is no public tool that supports such analysis of pre-training corpora at large scale. To help research in this space, we launch `Koala`, a searchable index over large pre-training corpora using compressed suffix arrays with highly efficient compression rate and search support. In its first release we index the public proportion of OPT 175B pre-training data. `Koala` provides a framework to do forensic analysis on the current and future benchmarks as well as to assess the degree of memorization in the output from the LLMs. `Koala` is available for public use at https://koala-index.erc.monash.edu/.[1]

1 Introduction

Large Language Models (LLMs) have achieved state-of-the-art results in NLP and on many benchmarks have reached the performance ceiling (Chowdhery et al., 2022). This evergrowing success has been facilitated by the algorithmic and computational progress in scaling up model sizes (Wei et al., 2022a; Chowdhery et al., 2022; Zhang et al., 2022; Brown et al., 2020), integrating human feedback (Ouyang et al., 2022), adopting modes of instructional inference at both zero- or few-shot settings (Chen et al., 2022; Kojima et al., 2022; Wei et al., 2022b; Nye et al., 2021), as well as the ability of feeding them massive volumes of free text during pre-training.

Recent works exhibit various cases which highlight the sensitivity of downstream behaviour of LLMs (and their smaller variants) to the frequency of observed overlap between pre-training corpora and test set (Carlini et al., 2022; Tänzer et al., 2022; Razeghi et al., 2022; Magar and Schwartz, 2022; Lewis et al., 2020). In the generative setting, several issues such as hallucination (Dziri et al., 2022), undesired biases (Kirk et al., 2021), or toxicity (Gehman et al., 2020) have been attributed partly or fully to the characteristics of the pre-training data, while a parallel line of works have emphasised on the positive role of filtering the pre-training data for safety and factual grounding (Thoppilan et al., 2022).

The above observations are not a comprehensive list but echo *the undeniable role of pre-training data in how these models would function in practice*. Understanding the limitations imposed by pre-training data would also lead to more informed algorithmic and computational innovations (Collier et al., 2022). However, these forensic studies are done either at a small scale or by using surrogate sources such as web search hit counts. To the best of our knowledge, there is no existing work that offers a tool or service for supporting deeper analyses in this space at large scale.

To help research in this direction, we launch the `Koala` project, a service backed by lossless compressed suffix arrays (CSA) (Navarro and Mäkinen, 2007), with efficient compression rate and query support. `Koala` contains a searchable index over the public portion of the pre-training data[2] behind the OPT 175B (Zhang et al., 2022) model, after deduplication. The constructed index is intended to provide overlap count statistics for text query (with ∞-gram length) files provided by users. We foresee several areas of impact for `Koala`; (i) as a tool to measure data leakage between existing benchmarks and pre-training corpora of LLMs, (ii) and evaluate the degree of memorisation or creativity in generative models' output, (iii) and to support designing harder benchmarks by reducing the overlap with pre-training corpora.

* Corresponding author

[1]Video demo can be found at https://drive.google.com/file/d/1hlSACOkwrFu5EVDQOTZ5Gvcyz7F-Wlpv/view

[2]We plan to extend our coverage of pre-training corpora.

We present a brief overview of the `Koala` pipeline for pre-processing and constructing the index. We also provide examples of the types of analyses that could be done via `Koala` by looking at a few commonly used test benchmarks.

2 Pre-processing and Corpora Coverage

2.1 Pre-processing Steps

Our pre-processing pipeline includes three main steps: cleaning, deduplication and tokenization. The cleaning step varies according to the pre-trained corpus and is described in Section 2.2 where we provide the coverage of `Koala`. In this section, we describe the deduplication and tokenization steps which are shared across all pretrained corpora.

We use MinHashLSH (Rajaraman and Ullman, 2011, Chapter 3), a widely-adopted duplicate detection method for large-scale dataset, in the deduplication step. Documents are first converted into a set of unigram tokens (shingling) and then hashed into a short signature, namely minhash, such that the similarity among documents is preserved. MinHash is a hashing algorithm based on permutation to generate random hashes to approximate the Jaccard similarity (Broder, 1997; Cohen et al., 2001). We generate the minhashes with 100 permutations. Finally, the locality-sensitive hashes (LSH) of the minhash values are calculated to detect the duplicated candidate pairs. We follow Zhang et al. (2022) to remove those having Jaccard similarity scores above 0.95 threshold. Our deduplication implementation is based on the datasketch library[3]. To scale the deduplication process to the large corpus, we first perform deduplication in a small batch and gradually merge the deduplicated batches.

The deduplicated corpus is then tokenized with moses (Koehn et al., 2007) to normalize punctuation and remove non-printing characters. We do not apply any casefolding.

2.2 Corpora Coverage

The latest version of `koala` at the time of writing this manuscript covers the following corpora:[4]

BookCorpus (Zhu et al., 2015) obtained from Hugging Face.[5]

Corpus	Raw Size (GB)	Deduplication Time (Min)	Deduplication Size (GB)	CSA Indexing Time (Min)	CSA Indexing Size (GB)
HackerNews	3.9	7,147.2	3.2	34.2	3.3
BookCorpus	4.3	14,301.2	3.7	88.1	3.6
DM Mathematics	7.8	7,881.6	1.7	32.5	3.7
OpenSubtitles	13	19,920.1	4.9	58.1	4.8
Guthenberg	10.9	23,893.0	9.7	139.0	9.5
Wikipedi	17	31,124.4	14	160.4	13
USPTO	22.9	41,866.8	22	206.8	16
OpenWebTexts	62.8	115,088.2	54	885.8	47
CCNewsv2	150	292,724.7	94	818.3	80
Pile-CC	227.1	416,186.8	123	1,965.2	106
Reddit	420	617,906.5	345	4,821.2	358

Table 1: Statistics of corpora, deduplication step, and the index construction. Indexing is done on a single CPU core of a 2.70 GHz Intel Xeon Gold 6150, and requires $2.5\times$ of index size of RAM memory.

CCNewsv2 extracted English news published between 2016 and 09/2021 from CommonCrawl (Nagel, 2016) using news-please (Hamborg et al., 2017).

ThePile (Gao et al., 2021) includes a subset of The Pile: Pile-CC, USPTO Backgrounds, Guthenberg (Rae et al., 2020), OpenWebTexts (Gokaslan and Cohen, 2019), OpenSubtitles (Tiedemann, 2016), Wikipedia (en), DM Mathematics (Saxton et al., 2019), HackerNews.

Pushshift Reddit [6] We used langdetect[7] to detect and extract the English comments and submission posted from 2005 to 2019. We followed pre-processing procedure in (Roller et al., 2021) to remove the post from known non-English subreddits and bot[8], comments longer than 2048 characters or containing URL, or at depth larger than 7 in a thread.

For readers' reference, the above collection covers the pre-training corpora of OPT (Zhang et al., 2022) with the exception of CC-Stories (Trinh and Le, 2018) which is not publicly available at the time of writing this manuscript. Table 1 reports the size of each corpus in raw and deduplicated version.

3 Pipeline and Features of `Koala`

3.1 Data Structure of `Koala`

Our index construction is inspired by the language models of Shareghi et al. (2015, 2016), which leverage compressed data structures for building lan-

[3] https://github.com/ekzhu/datasketch
[4] We plan to index more public pre-training corpora as they become available.
[5] https://huggingface.co/datasets/bookcorpus
[6] https://files.pushshift.io/reddit
[7] https://github.com/fedelopez77/langdetect
[8] https://github.com/eliassjogreen/Reddit-Bot-List

guage models on large text corpora. In this subsection we provide a brief overview of the data structures behind `Koala`.

A Suffix Array (SA) (Manber and Myers, 1993) of a string $\mathcal{T}$ with alphabet σ is an array of its sorted suffixes. A cell in a suffix array, denoted by $\text{SA}[i]$, stores a number indicating the staring position of its corresponding suffix in $\mathcal{T}$. Using a suffix array, searching for any sequence $\mathbf{u}$ in $\mathcal{T}$ translates into a binary search to find the range that spans over all substrings that have $\mathbf{u}$ as their prefix, and is $\mathcal{O}(|\mathbf{u}| \log |\mathcal{T}|)$. Constructing SA takes 4-8$|\mathcal{T}|$ bytes in practice, making them impractical to use for large data.

To support search on large collections, Compressed Suffix Array exploits the compressibility of $\mathcal{T}$ while providing the same functionality of SA in space equal to bzip2 compressed $\mathcal{T}$ in practice. We follow Shareghi et al. (2016) and use the FM-Index (Ferragina et al., 2008) that utilises the text compressibility vi the Burrows-Wheeler transformation (BWT) (Burrows and Wheeler, 1994) of the text. The BWT is defined as, $\text{BWT}[i] = [\text{SA}[i] - 1 \bmod |\mathcal{T}|]$. Searching for a sequence in BWT is done in reverse order and requires $\mathcal{O}(|\mathbf{u}| \log |\sigma|)$. For more details on BWT and reverse searching, refer to Navarro and Mäkinen (2007).

The CSA is at the core of `Koala`'s index and search backbone. We used the SDSL library (Gog et al., 2014) to implement our corpus indexer. We index each corpus separately. Once a corpus is indexed, its constructed index sits on disk and could be queried through the `Koala` web interface (introduced shortly). Each query is launched into the indexed collection of corpora and returns the hit counts of the query in the corresponding corpus. Table 1 reports the time and memory usage for construction of indexes.

3.2 n-gram Overlap Statistics of `Koala`

Given a text query, `Koala` can provide its count statistics in several pretraining corpora by querying the indexes. An example of the raw count output for the phrase *plastic bags floating in the ocean* is shown in Table 2. Meaningful insights can be derived from these raw statistics. Figure 1 illustrates two high-level statistics built on top of the n-gram counts for two question answering benchmark test sets, PIQA (Bisk et al., 2020) and OpenBookQA (Mihaylov et al., 2018), highlighting the amount of leakage or overlap that exists between these test sets and the entire pre-training data collection indexed in `Koala`. We first introduce how these statistics are calculated per instance, noting that Figure 1 is reporting them as an average across all instances in each test set. The high-level statistics are defined as follows:

Per Instance k-gram hit ratio measures $\frac{M_x^{k,t}}{N_x^k}$, where N_x^k is the set of all k-grams of instance x, and $M_x^{k,t}$ is the subset of N_x^k containing only the k-grams with frequency above the pre-set thresholds t (e.g., ≥ 1, ≥ 10, ≥ 100, ≥ 1k, $\geq$ 10k, $\geq$100k, $\geq$1M).

Per Instance k-gram hit length ratio measures $\frac{M_x^{l,t}}{N_x^l}$, where N_x^l is the set of all substrings of instance x that fall within the length bin l (e.g., $l = [0.75, 100]$ means all substrings whose lengths are 3/4 of the length of x or more), and $M_x^{l,t}$ is the subset of N_x^l, containing only the substrings with frequency above the pre-set thresholds t (e.g., ≥ 1, ≥ 10, ≥ 100, ≥ 1k, $\geq$ 10k, $\geq$100k, $\geq$1M). We considered 4 length bins: [0,0.25), [0.25,0.50), [0.5,0.75), and [0.75,1].

While a deep dive into exploring the dependence between data overlap, model size, and model performance requires a separate work, here we unpack some highlights from the figures:

Highlights from Figure 1 (Left Panel): The top-left panel highlights that for OpenBookQA above 75% of the unigrams and bigrams of test set occur at least once (≥ 1) in the pretraining data, while this drops to below 50% with a higher threshold ($\geq$ 1k). We observe that above 25% of trigrams occur at least 100 times in the pretraining data. Looking at the bottom-left panel for PIQA, we see a much stronger indication of data overlap. For instance we observe above 55% of bigrams occur at least 100 times in the pre-training data. Comparing the two dataset at the extreme frequency threshold of $\geq$ 1M, we observe that above 50% of PIQA unigrams occur at least 1M times in the pretraining data, while this is roughly 30% for OpenBookQA.

Highlights from Figure 1 (Right Panel): The average answer length in PIQA and OpenBookQA test sets are 101, and 20, respectively. This means that [0.25,0.5) length bin covers sequences of roughly 25-50 tokens for PIQA, while this is roughly 5-10 tokens for OpenBookQA. We now turn to the highlights from the right panel of Figure 1. For OpenBookQA (top-right) we observe

n	n-grams list	Pile-CC	BookCorpus	CCNewsv2	DM	Guthenberg	HackerNews	OpenSubtitles	OpenWebTexts	USPTO	Wikipedia	Reddit
1	plastic	959364	33845	580607	0	4964	14397	14114	329535	598625	39435	2650049
	bags	578401	29213	415672	0	17160	5405	21590	166685	111115	13708	1697726
	floating	303836	19752	162095	0	36242	10058	8165	120146	244489	21938	976575
	in	355723492	9260245	308475794	3347881	30592137	7135629	7831355	150523086	63002717	54190836	749899124
	the	1056004732	34886372	782874590	6519155	107380032	20809865	23296159	428544710	251429575	128120455	2128039302
	ocean	575919	30175	273507	0	65172	8467	23233	235331	23909	41516	1125595
2	plastic bags	39722	843	38094	0	0	588	367	19323	7544	1267	79539
	bags floating	77	4	57	0	0	2	2	25	0	5	275
	floating in	29619	3326	19189	0	3492	408	1397	12907	2913	1695	101880
	in the	91136626	2440752	81218136	52379	7948909	1572721	1925941	37928620	19087529	13710461	175900138
	the ocean	284689	18995	139332	0	33275	4066	14749	114465	11596	18558	667336
3	plastic bags floating	34	0	22	0	0	1	0	12	0	2	110
	bags floating in	27	0	34	0	0	0	0	8	0	3	101
	floating in the	14481	1621	10734	0	1791	141	725	6594	1760	897	43090
	in the ocean	44233	1573	28680	0	2025	1035	2513	21517	1588	2566	163343
4	plastic bags floating in	16	0	10	0	0	0	0	3	0	2	43
	bags floating in the	20	0	29	0	0	0	0	5	0	3	76
	floating in the ocean	580	19	413	0	7	10	16	372	24	42	2078
5	plastic bags floating in the	13	0	8	0	0	0	0	1	0	2	33
	bags floating in the ocean	4	0	2	0	0	0	0	1	0	2	9
6	plastic bags floating in the ocean	4	0	1	0	0	0	0	1	0	2	2

Table 2: The n-gram hit statistics per corpus for the correct answer (*plastic bags floating in the ocean*) to the query *Which of these situations is an example of pollutants?, choices : [**plastic bags floating in the ocean**, mallard ducks floating on a lake, cottonwood seeds floating in the air, cirrus clouds floating in the sky]*. This is a sample from the OpenBookQA benchmark.

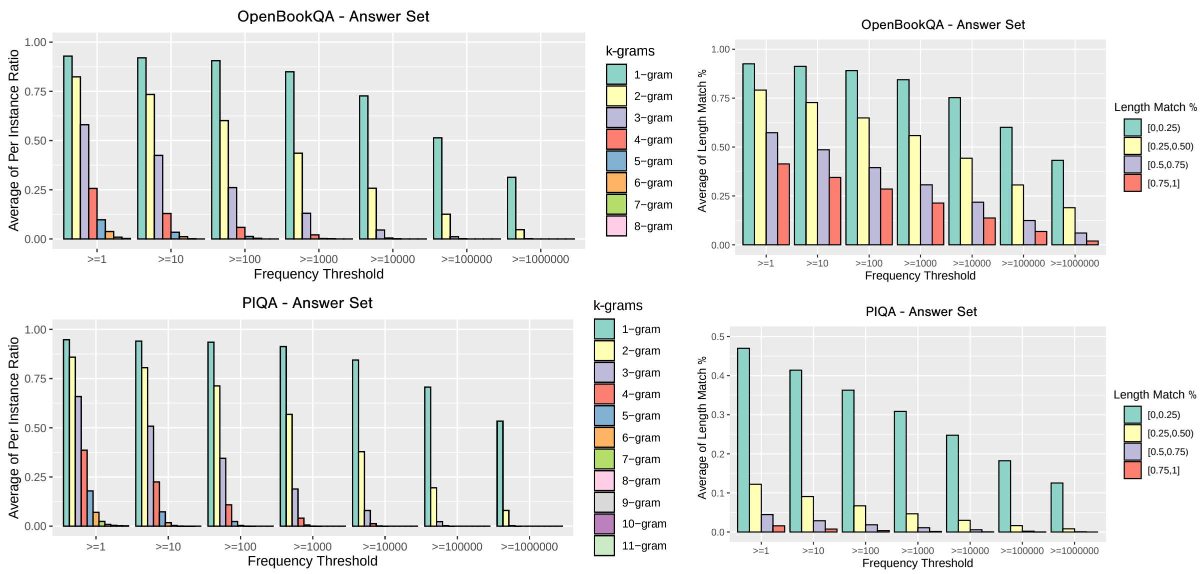
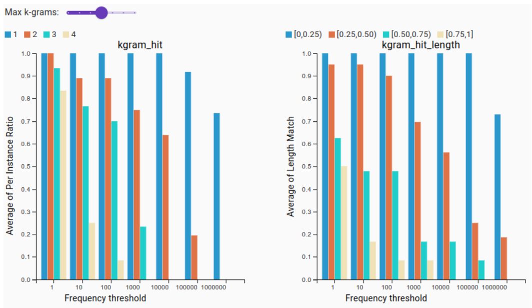


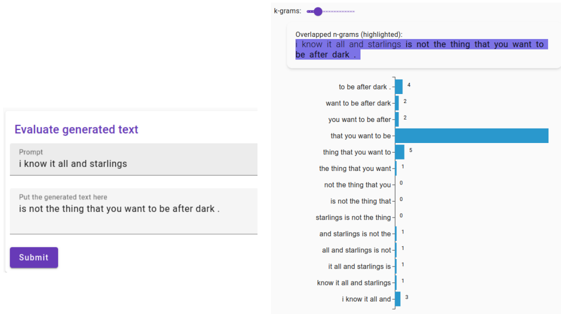
Figure 1: Visualisations of n-gram overlap statistics for OpenBookQA and PIQA test sets, Answer side. **Top:** OpenBookQA Answer Set ; **Bottom:** PIQA Answer Set. **Left:** Average of Per Instance K-gram hit ratio (i.e., K-gram hit ratio = 1 means 100% of k-grams in one instance were a hit); **Right:** Average of Per Instance K-gram hit length ratio (i.e., K-gram hit length ratio with respect to the instance length = 1 means the k-gram was fully covered, 0.75 means it was 3/4 covered, etc). PIQA test set size is 1838, OpenBookQA test set size is 500.

from the red bars that above 25% of test instances (roughly 125 cases out of 500 test instances in OpenBookQA) are almost [75%,100%] covered in the pre-training data for at least 100 times ($\geq$ 100). This corresponds to matches of length 15-20 words. Looking at PIQA (Bottom-Right), although the coverage with respect to the full length is not as apparent as OpenBookQA, matches in each corresponding length bin of PIQA are roughly 4$\times$ longer than OpenBookQA. For instance, about 5% of test instances of PIQA (roughly 90 cases out of 1838 test instances in PIQA) have a matching substring of 25-50 words which occur at least $1k$ times in the pretraining data (see yellow bar for $\geq$ 1000).

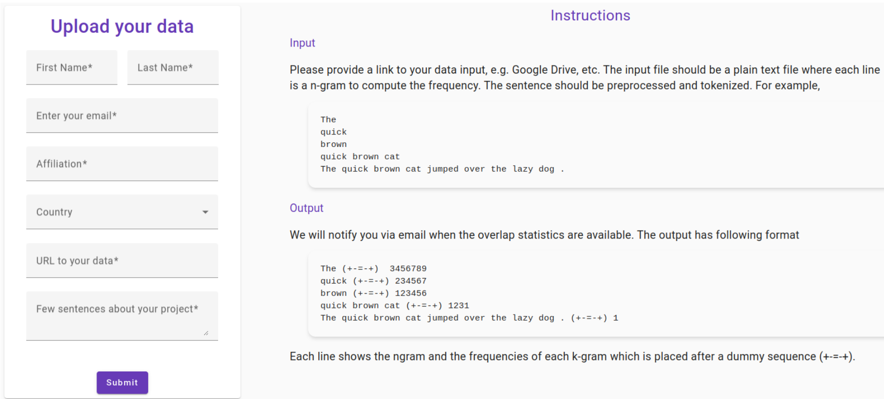
The performance ceiling obtained by GPT-3 and OPT models for these two benchmarks (reported numbers in Appendix A of Zhang et al. (2022)) indicate the largest variant of both models achieve roughly 80% accuracy for PIQA, and above 57% accuracy on OpenBookQA. Our highlighted findings suggests a positive correlation between the amount of data overlap we highlighted and the task performance ceiling by the LLMs trained on the same pre-training corpora. As a future direction of analysis, it would be interesting to leverage `Koala` to analyse the interdependence of the amount of data overlap, model size, and task performance. Our preliminary analyses suggest that the connec-



(a) Live Demo feature: Upload a file less than 2MB and observe insights on length and count matches against the BookCorpus. For analysis against all corpora or for larger files, use the upload form.



(b) Live Demo feature: Type a piece of text and observe its overlap statistics against BookCorpus.



(c) For analysis against all corpora in Koala or for files larger than 2MB, use the upload form.

Figure 2: Screenshots from different features of the `Koala` webpage. For the latest interface, please refer directly to the website.

tion between model size and capacity to memorise is not trivial, and varies from task to task.

3.3 Interface of `Koala`

In this section, we give an overview of the interface of `Koala`. Figure 2 demonstrates some of `Koala`'s features. In addition to reporting the raw counts, `Koala` provides an interface to upload an n-gram file and to visualize different hit ratio statistics (§3.2). The n-gram file is a plain text file where each line is an n-gram whose overlap statistics will be computed. Figure 2(a) shows the output from this feature. We also provide the interactive version of the ratio plots (e.g., Figure 1) for 3 question answering benchmarks: HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020) and OpenBookQA (Mihaylov et al., 2018).[9]

In the first release, we limit the live demo queries to n-gram files below 2MB and report only on the BookCorpus pretraining corpus. For larger files and more comprehensive statistics, we provide a form for users to submit the data and queue the computation. An example of the form is shown in Figure 2(c). We plan to extend the live demo to the entire pretraining corpora in the near future.

Another use case of the overlap statistics is to provide a measure of the memorization vs. creativity for generative LLMs, i.e. how much of the generated text overlaps with the pretraining corpora. `Koala` implements a tool to verify the novelty of an output of generative LLM given a prompt. Figure 2(b) shows an example of this feature which provides the count statistics of the n-grams in the generated text and highlight the overlap n-grams.

4 Conclusion and Future Work

We presented `Koala`, a web-based service powered by a compressed data structure backbone that facilitates efficient search over large collections of texts. `Koala` is a tool for comprehensive overlap analysis with potential use-cases including but not limited to assessing leakage of test benchmarks, measuring the degree of memorization in generative LLMs outputs. Additionally, `Koala` not only provides a public tool for forensic analysis of these phenomena it could also help benchmark designers towards constructing more challenging testbeds for LLMs. We will continue to grow our coverage of the existing pre-training corpora.

[9]We plan to expand the coverage of benchmarks.

References

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Michael Burrows and David Wheeler. 1994. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation Systems Research Center.

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

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

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 arXiv:2204.02311*.

E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J.D. Ullman, and C. Yang. 2001. Finding interesting associations without support pruning. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):64–78.

Nigel H. Collier, Fangyu Liu, and Ehsan Shareghi. 2022. On reality and the limits of language data. *CoRR*, abs/2208.11981.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Paolo Ferragina, Rodrigo González, Gonzalo Navarro, and Rossano Venturini. 2008. Compressed text indexes: From theory to practice. *ACM J. of Exp. Algorithmics*, 13.

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. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Simon Gog, Timo Beller, Alistair Moffat, and Matthias Petri. 2014. From theory to practice: Plug and play with succinct data structures. In *Experimental Algorithms - 13th International Symposium, SEA 2014, Copenhagen, Denmark, June 29 - July 1, 2014. Proceedings*, volume 8504 of *Lecture Notes in Computer Science*, pages 326–337. Springer.

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText corpus.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Hannah Rose Kirk, Yennie Jun, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, Aleksandar Shtedritski, and Yuki Asano. 2021. Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models. *Advances in neural information processing systems*, 34:2611–2624.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020. Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637*.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. *arXiv preprint arXiv:2203.08242*.

Udi Manber and Eugene W. Myers. 1993. Suffix arrays: A new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948.

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.

Sebastian Nagel. 2016. CC-News.

Gonzalo Navarro and Veli Mäkinen. 2007. Compressed full-text indexes. *ACM Comp. Surv.*, 39(1):2.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of Massive Datasets*. Cambridge University Press, USA.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ehsan Shareghi, Matthias Petri, Gholamreza Haffari, and Trevor Cohn. 2015. Compact, efficient and unlimited capacity: Language modeling with compressed suffix trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ehsan Shareghi, Matthias Petri, Gholamreza Haffari, and Trevor Cohn. 2016. Fast, small and exact: Infinite-order language modelling with compressed suffix trees. *Transactions of the Association for Computational Linguistics*, 4:477–490.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7564–7578.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Jörg Tiedemann. 2016. Finding alternative translations in a large corpus of movie subtitle. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

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.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.