GQG: Generalized Quantifier Generalization
A Dataset for Evaluating Quantifier Semantics in Language Models

Leroy Z. Wang
University of Washington
lryw@uw.edu

Shane Steinert-Threlkeld
University of Washington
shanest@uw.edu

Abstract

We present a new dataset consisting of various quantifier expressions to evaluate the generalization abilities of language models. The dataset contains 18,360 prompts encompassing diverse quantifiers, forming the basis of a new framework for assessing semantic understanding in this domain. We test the effectiveness of our dataset using Pythia models, ranging from 410 million to 6.9 billion parameters, showing that quantifier-based tasks can be challenging for current language models. We make our code and data publicly available[1], such that the dataset can be easily extended or updated based on different evaluation needs.

1 Introduction

In recent years, the Natural Language Processing (NLP) community has witnessed the rise of increasingly larger and more sophisticated language models (LMs) capable of generating coherent texts over extended passages. However, the ability of these models to understand and generate human language that aligns with the underlying semantics remains a topic of debate (Yogatama et al., 2019). Neural language models may rely on heuristics learned from the training data to generate seemingly coherent texts, but fail to generalize to scenarios that are more complex and cannot be solved by simple heuristics (McCoy et al., 2019). Whether language models can acquire *meaning* when trained only on text is also a topic of ongoing debate. Bender and Koller (2020) have argued, through thought experiments, that LMs cannot learn semantics through texts since they lack access to explicit representations of the external world. We hope to contribute to this ongoing dicussion by releasing this dataset on quantifiers, which will enable more research on this direction.

In this paper, we introduce a new framework that uses formal semantics to test the generalization capabilities of language models, by developing a dataset that assesses LMs' understanding of quantifier semantics. We ask the question – to what extent do language models capture the semantics of quantifiers?

Quantifiers are well-suited for evaluating language model generalization because compared to other linguistics objects, their meanings are more abstract, and can be fully specified in theoretical terms that do not require grounding. Common examples of quantifiers include *some*, *all*, *a few*, *many*, etc. To construct the dataset, we use a deterministic algorithm to generate prompts and gold labels automatically, covering 15 different quantifiers in the English language. In each prompt, we ask the LM to give a truth value judgment (true or false) to a statement about a given quantifier in a constructed scenario.

This dataset does not exhaust common quantifiers in English, but is designed to enable more research on evaluating LMs' understanding of semantic objects. More specifically, the dataset will allow further investigation into different axes of generalization in this domain, i.e. for the same quantifier, does using different nouns in the prompts affect LM's acquisition of the semantics of the quantifier? And how do different number ranges or different word orders affect LM's understanding of the same quantifier? The dataset presented in this paper provides a valuable framework for researchers to study these types of questions, grounded by rich formal semantics literature on quantifiers. As demonstrated in section 3, the data generation pipeline can be easily extended to study new quantifiers, nouns, and number ranges. Given initial native speaker annotated prompts and quantifiers, the data can also be easily extended to different languages.

This paper is structured as follows. In section 2, we discuss the literature on quantifiers in formal semantics and demonstrate how they can be useful for evaluating LMs. In section 3, we discuss how

[1] https://github.com/lerow/llm-quantifier

Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP, pages 185–192

the dataset is constructed and how LMs are evaluated. We discuss future work directions in section 4 and remark on limitations at the end.

2 Background

2.1 Quantifiers

Quantifiers are semantic objects expressed by determiners. In formal semantics, determiners can be considered as *generalized quantifiers* that describe the relations between two subsets in a discourse (Barwise and Cooper, 1981). Examples of quantifiers include *some*, *a few*, *all*, *most*, etc. They are useful for evaluating language models' generalization ability because their semantics is well-defined. Unlike content words, such as common nouns, quantifiers' semantics is fully abstract and can be specified in set-theoretic terms. Therefore, when measuring the alignment between language models and quantifier semantics, the evaluation can be completed fully unsupervised without human annotations while achieving high accuracy.

Based on literature in formal semantics (Barwise and Cooper, 1981; Peters and Westerstl, 2006; Szymanik, 2016; Steinert-Threlkeld and Szymanik, 2019), we define a quantifier to be a relation between two subsets A and B of a given discourse domain M. For example:

$$[\![\text{at least } n]\!] = \{\langle M, A, B\rangle : |A \cap B| \geq n\}$$
$$[\![\text{all}]\!] = \{\langle M, A, B\rangle : |A \cap B| = |A \cup B|\}$$
$$[\![\text{more than half}]\!] = \{\langle M, A, B\rangle : |A \cap B| > |A \backslash B|\}$$
$$[\![\text{a few}]\!] = \{\langle M, A, B\rangle : |A \cap B| > 1\}$$

We use $[\![Q]\!]$ to denote the semantic meaning of quantifier Q. For the quantifier "at least n", $[\![\text{at least } n]\!]$ describes sentences that satisfy $|A \cap B| \geq n$ when interpreted in model M.[2] For example, let set A denote the flowers, and set B denote the red objects in a discourse. Suppose $|A| = 5, |B| = 5, |A \cap B| = 5$, then the sentence "at least 5 flowers are red" would be true, because the situation $\langle M, A, B\rangle$ would belong to to $[\![\text{at least } 5]\!]$ since it satisfies $|A \cap B| \geq 5$.

Given a prompt as a way of representing a situation M, A, B in natural language, we can define the meaning of a quantifier according to a language model as:

$$[\![Q]\!]^{\text{LM}}_{\text{prompt}} = \{\langle M, A, B\rangle : \text{LM}(\text{prompt}(Q, M, A, B)) = T\},$$

We can then measure how similar $[\![Q]\!]^{\text{LM}}_{\text{prompt}}$ is to the true underlying $[\![Q]\!]$. The LM and the prompt are considered as parameters that need to be specified by the researcher.

2.2 Related Work

Benchmarks and Datasets There have been many benchmarks and datasets developed to evaluate the language understanding abilities of NLP models. Benchmarks such as SuperGLUE (Wang et al., 2019) and WinoGrande (Sakaguchi et al., 2021) test commonsense and logical reasoning abilities of LMs. In NLI datasets such as SNLI (Bowman et al., 2015) and LAMBADA (Paperno et al., 2016), designed to measure LMs' reasoning abilities through quantifiers. In AMBIENT, Liu et al. (2023) have curated a linguist-annotated dataset with various kinds of linguistic ambiguities, including quantifier scope ambiguity, to measure the disambiguation abilities of LMs. Understanding the relationships between quantifiers is important for LMs to perform well in these NLI datasets [3], but whether LMs can correctly acquire the semantics of quantifiers has not been systematically tested.

LMs and Semantics Bender and Koller (2020) have argued that LMs cannot acquire full meanings from text data alone, since they have no access to the explicit representations of entities in the world. In response, Li et al. (2021) demonstrate that language models can use contextual word representations to model changes of entities in a discourse, which presents preliminary empirical evidence suggesting that neural language models are capable of encoding partial representations of meaning when trained only on text data. Patel and Pavlick (2022) show that language models can learn to map a conceptual domain (such as color or direction) onto a grounded world representation. Utilizing psycholinguistic tests, Ettinger (2020) have shown that the BERT model has difficulty acquiring generalizable meanings of negation quantifiers.

[2] See Peters and Westerstl (2006) for a more detailed discussion.

[3] e.g. "A soccer game with multiple males playing" entails "Some men are playing a sport." (Bowman et al., 2015), but not vice versa

3 Methodology

3.1 Dataset Creation

We use a deterministic algorithm to generate the prompts in the dataset. Demonstrated in algorithm 1, we iterate through all available object and quantifier combinations, and generate a prompt for each combination. n is the number of objects in total and i is the number of objects modified by the first predicate (i.e. "are large" in the first example in Table 4), or by the second predicate (i.e. "are small") in the current iteration.

The GENERATE_PROMPT function can be fully customized by the user. In this work, we use the prompt template "There are 50 items. n of the items are large. m of the items are small. Are Q of the items small / large? Answer with only one word, true or false." for all 18,360 prompts in the dataset.

For the vanilla GQG dataset, we set $n = 50$, and the two predicates to be "are large" and "are small". Researchers can use the framework to easily generate more prompts with different number ranges, predicates, and noun objects, to test various kinds of LM generalization.

Label Generation The gold labels in the dataset are generated using lambda functions that represent the exact semantics of the quantifiers – e.g. for quantifier "at least 3", its function would be

$$\lambda\, n, a, b : a >= 3,$$

where n is the total number of objects, a is the number of objects modified by the first predicate, and b is the number of objects modified by the second predicate. Using the notation presented in section 2.1, we have $n = |M|$, $a = |A \cap B|$, and $b = |M| - |A \cap B|$.

The lambda functions are manually coded by human experimenters, and it is the only place that requires human labeling in this evaluation framework (besides creating the prompt template).

Number of prompts The number 18,360 comes from $18360 = 12 \times 15 \times 51 \times 2$, where

- 12: number of objects
- 15: number of quantifers
- 51: (0 to 50) number of objects modified by the predicate
- 2: we have 2 predicates, so for the same quantifier, number of objects, and nouns, we prompt for both the first predicate and for the second predicate. For example, if $n = 50$, $q = \{\text{at least half}\}$, $o = \{\text{apples}\}$, $i = 5$, there will be two prompts generated, one asking "are at least half of the apples large?" with gold label *false*, and the other one asking "are at least half of the apples small?" with gold label *true*.

Algorithm 1 Data Generation

Inputs: set of quantifiers Q, set of objects O, function GENERATE_PROMPT, number range m, n $(m \leq n)$

```
for o in O do
    for q in Q do
        for i := m to n do
            append
            GENERATE_PROMPT (o, q, i, n)
            to prompts
        end for
    end for
end for

return prompts
```

3.2 Data Statistics

We present some statistics of the vanilla GQG dataset in this section. The dataset is consisted of prompts describing different scenarios using various quantifiers and noun objects shown in Table 2 and Table 3. The GQG framework is highly modular, allowing researchers to easily extend the vanilla dataset beyond the scope of the lexical items presented in this paper.

Number of prompts	18360	
Average # of tokens per prompt	31.9	
# of prompts with true label	8592	(46.8 %)
# of prompts with false label	9768	(53.2 %)

Table 1: Data Statistics

List of Quantifiers:

No.	Quantifier
1	at least 3
2	at least 4
3	at most 5
4	at most 6
5	more than 1
6	more than 5
7	more than 10
8	all
9	none
10	between 4 and 6
11	between 2 and 10
12	at most half
13	more than half
14	less than half
15	at least half

Table 2: Quantifiers in the dataset

List of objects:

No.	Object
1	tables
2	chairs
3	circles
4	squares
5	apples
6	bikes
7	pans
8	shelves
9	trees
10	birds
11	penguins
12	mountains

Table 3: Objects in the dataset

Prompt	Label
There are 50 tables. 50 of the tables are large. 0 of the tables are small Are **at least 3** of the tables small? Answer with only one word, true or false.	false
There are 50 circles. 7 of the circles are large. 43 of the circles are small. Are **at least 4** of the circles small? Answer with only one word, true or false.	true
There are 50 apples. 49 of the apples are large. 1 of the apples is small. Are **less than half** of the apples large? Answer with only one word, true or false.	false
There are 50 mountains. 24 of the mountains are large. 26 of the mountains are small. Are **at most half** of the mountains large? Answer with only one word, true or false.	true

Table 4: Examples of prompts in the dataset

3.3 Evaluation

We use both accuracy and F_1 scores to measure the alignment between quantifier semantics and LMs' understanding. Since the data is not perfectly balanced ($> 53\%$ of prompts have *false* gold labels), a language model can easily perform better than random by always answering "false".

During evaluation, a parser is used to process the output from the LM. When the parser encounters a token that matches with the string "true"[4], it will consider the LM as giving a positive response. Otherwise, the parser considers the LM as giving a negative response to the prompt. Constrained decoding is used during evaluation – the LM's response must contain either the token "true" or the token "false".

This approach has its limitations, since how good the language models are at following instructions can affect the performance. Language models may also leak their training data when being prompted in this manner – we have observed that instead of answering the question, the LM will output texts resembling multiple choice questions when given the prompt, which may be part of its training data. We discuss the potential issues of our evaluation

[4]when converted to lowercase, with punctuations and whitespace removed

method in Limitations.

3.4 Generalization Testing

The GQG dataset enables researchers to test LM generalization in quantifier understanding across different lexical items. For example, given a quantifier q, does changing the noun objects o in the prompts affect LMs' understanding of $[\![q]\!]$? In other words, we can use the GQG framework to test whether LMs' understanding of the semantics of a quantifier is consistent with respect to different nouns used in the prompt.

Testing generalization with respect to different axes is also possible, by fixing different elements in the dataset. For instance, one can test whether LMs have the same level of understanding across different quantifiers by fixing the noun o and only alternating q in the prompts. It should be noted that the vanilla GQG dataset does not support testing generalization in arbitrary axes; however, such data can be easily constructed using the code framework released in the paper.

3.5 Experimental Results

Model size	Accuracy	F1	Precision	Recall
410M	0.464	0.418	0.412	0.425
1B	0.503	0.289	0.216	0.437
1.4B	**0.519**	0.355	0.283	**0.476**
2.8B	0.515	0.626	0.283	**0.476**
6.9B	0.484	**0.639**	**0.976**	0.475

Table 5: Performance of Pythia models on vanilla GQG

Highlighted by low accuracies and F_1 scores[5], the GQG dataset can be quite challenging for the Pythia language models. It's intriguing to observe that the test accuracy does not increase significantly as the model size increases – the test accuracy for the 6.9B model is even lower than the 1B model.

We also perform a small pilot study on LM generalization across lexical items with different frequencies in the corpus, using Mistral-7B (Jiang et al., 2023) language model. As seen in Figure 1, the test accuracy of Mistral-7B drops as the nouns used in the prompts become less common.

For each word group, we use 10 different words with similar frequencies to generate prompts, using $q = \{\text{at least half}\}$, $m = 0, n = 50$.

[5] F_1 scores are included during evaluation since the dataset does not have a perfectly balanced True/False label ratio.

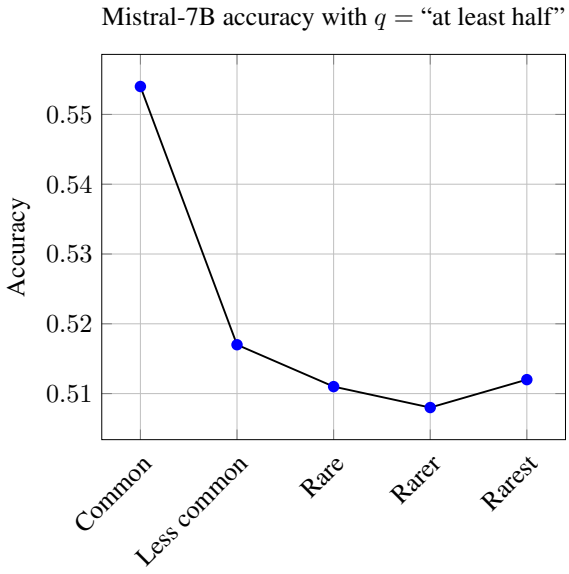

Figure 1: Mistral-7B accuracy with respect to word frequencies

Since The Pile dataset (Gao et al., 2020) that was used by Pythia LMs is not publicly available at the time of publication, we used the Leipzig Corpora (Richter et al., 2006; Goldhahn et al., 2012) to approximate the frequencies of tokens in Pythia training data. In each word group, the authors randomly select 10 different words with similar [6] frequencies in the Leipzig Corpora. Words that are in a less frequent group are guaranteed to have lower frequencies[7] than those in high-frequency groups. Examples of *common* words include "books", "doors", and "reports"; examples of *rarest* words include "lidars", "medullas", and "ornamentals". See Appendix A for the full list of words in each group.

The result of this small-scale experiment shows that large language models can be sensitive to the frequency of lexical items used in the prompt in certain scenarios. It also showcases the diverse kinds of research and generalization testing the GQG framework can enable.

4 Conclusion

This paper presents a new dataset to evaluate language models' understanding of quantifier semantics. The dataset can be easily extended to different languages and quantifiers, enabling more research on assessing language models' understanding of semantic objects and investigation into different axes of generalization. The poor performance of

[6] the difference between frequencies is less than 3

[7] in the Leipzig Corpora

the Pythia models during evaluation shows the dataset can be challenging for neural language models, but more research is required to understand how instruction-tuned LMs (and more sophisticated prompt-engineering) will perform on this dataset. We also note the limitations of our study, particularly on evaluation methods, and hope that this dataset will be a basis well-grounded in theoretical literature for more research on LMs and semantics.

Future work includes developing datasets in more diverse prompt formats, analyzing how LMs' performance can differ based on different types of quantifiers or linguistic objects, and investigating how finetuning can affect model performance.

Limitations

Monolingual Dataset We note that our dataset is curated only in English. How LMs may perform in low-resource languages has not been tested. What kind of impact will languages with more complex morphology/syntax have on benchmark performance also has not been investigated.

Prompt Format Our data is generated using one type of prompt format. Other types of prompt templates, including those designed in an adversarial manner, have not been evaluated in this paper. How the prompt is structured can also be an interesting axis of generalization to investigate – i.e. for the same quantifier and nouns, does different wordings of the prompt change how the LM acquire the semantics?

Finetuning The LMs tested have not been finetuned on the dataset. Whether finetuning can improve LMs' performance on understanding the semantics of quantifiers is a promising direction of future research.

Evaluation on LLMs The dataset has only been tested on Pythia models (Biderman et al., 2023). Larger and more recent language models such as GPT-4 (OpenAI, 2023), Llama-2 (Touvron et al., 2023), etc. have not been evaluated.

Ethics Statement

The data in this paper is artificially generated using a deterministic algorithm and does not violate any copyright laws. The dataset does not contain any content that is explicitly triggering, offensive, or toxic.

References

John Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

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.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Leo Gao, Stella Rose 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*, abs/2101.00027.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5, 1161–1174.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang,

Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*.

Stanley Peters and Dag Westerstahl. 2006. *Quantifiers in Language and Logic*. Oxford University Press UK, Oxford, England.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the leipzig corpora collection.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Shane Steinert-Threlkeld and Jakub Szymanik. 2019. Learnability and semantic universals. *Semantics and Pragmatics*.

Jakub Szymanik. 2016. *Quantifiers and Cognition: Logical and Computational Perspectives*. Springer.

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.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*. Curran Associates Inc., Red Hook, NY, USA.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *CoRR*, abs/1901.11373.

A Appendix

Examples of LM Leaking Training Data When given the prompt: "There are 50 tables. 28 of the tables are large. 22 of the tables are small. Are all of the tables small? Answer with only one word, true or false.", Pythia LMs will sometimes generate

```
A. True

B. False

C. It is not possible to determine
```

as its response.[8]

Experiment Infrastructure All experiments were run on a single NVIDIA RTX 3090 GPU. For all Pythia models, step 143000 (the last model checkpoint) and temperature 1.0 were used during inference.

Constrained Decoding The constrained decoding used during evaluation is implemented using

[8] listed example is a generated text from Pythia-2.8B given the prompt

Huggingface `force_words_ids` during `generate`; beam search is used during generation with `num_beams=4`.

Words in Different Frequency Group

- common = ["books", "chairs", "doors", "participants", "activities", "systems", "wars", "blocks", "words", "reports"]
- less common = ["crowds", "negotiations", "cup holders", "arteries", "identifiers", "payrolls", "hostages", "coupons", "remedies", "butterflies"]
- rare = ["jaws", "turbines", "rooftops", "hikers", "purses", "empires", "insurers", "camels", "entitlements", "coils"]
- rarer = ["auroras", "borrowers", "fasteners", "headscarves", "hickories", "geneticists", "catapults", "blurbs", "glaciers", "eyewitnesses"]
- rarest = ["ocean basins", "jests", "lidars", "inequalities", "microchips", "humanoids", "philanthropies", "medullas", "ornamentals", "jabs"]

An example prompt using a word from the *rare* group:

```
There are 50 empires. 10 of the empires are large. 40 of the empires are small. Are at least half of the empires large? Answer with only one word, true or false.
```

GenBench 2023 Evaluation Card The GenBench evaluation card (Hupkes et al., 2023) is attached.
Shift locus: Pretrain-test.

Motivation			
Practical	*Cognitive*	*Intrinsic*	*Fairness*
	☐	☐	
Generalisation type			
Compositional	*Structural*	*Cross Task*	*Cross Language*	*Cross Domain*	*Robustness*
☐					☐
Shift type			
Covariate	*Label*	*Full*	*Assumed*
☐			
Shift source			
Naturally occuring	*Partitioned natural*	*Generated shift*	*Fully generated*
		☐	
Shift locus			
Train–test	*Finetune train–test*	*Pretrain–train*	*Pretrain–test*
			☐