Generalization v.s. Memorization: Tracing Language Models' Capabilities Back to Pretraining Data

Antonis Antoniades [*1] Xinyi Wang [*1] Yanai Elazar [2 3] Alfonso Amayuelas [1] Alon Albalak [1] Kexun Zhang [4]
William Yang Wang [1]

Abstract

Despite the proven utility of large language models (LLMs) in real-world applications, there remains a lack of understanding regarding how they leverage their large-scale pretraining text corpora to achieve such capabilities. In this work, we investigate the interplay between generalization and memorization in pretrained LLMs at scale, through a comprehensive n-gram analysis of their training data. Our experiments focus on three general task types: translation, question-answering, and multiple-choice reasoning. With various sizes of open-source LLMs and their pretraining corpora, we observe that as the model size increases, the task-relevant n-gram pair data becomes increasingly important, leading to improved task performance, decreased memorization, stronger generalization, and emergent abilities. Our results support the hypothesis that LLMs' capabilities emerge from a delicate balance of memorization and generalization with sufficient task-related pretraining data, and point the way to larger-scale analyses that could further improve our understanding of these models.

1. Introduction

Pretrained large language models (LLMs) have shown impressive performance on many text-based tasks, but there is a debate about whether they are generalizing on unseen test cases or simply memorizing from their vast training data (Magar and Schwartz, 2022; Srivastava et al., 2024; Bender et al., 2021). Previous works have studied LLM memorization as exactly recalling training examples (Zhang et al., 2023; Jiang et al., 2024; Carlini et al., 2022). However, usually, we aim to utilize more sophisticated LLM capabilities, which cannot be explained by copying training data. In this paper, we extend the definition of memorization beyond exact copying and study how pretraining data contributes to higher LLM capabilities. We define **memorization** as the degree of similarity between LLM generations and training data, and **generalization** as how different the generated content is from training data. Several papers have studied the interplay between memorization and generalization (Feldman, 2020; Feldman and Zhang, 2020; Zhang et al., 2023), but analyzing pretrained LLMs at scale remains challenging. [1] We propose to estimate the pretraining data distribution by n-gram pairs mined from task data. The appearance of such pairs in training data can be regarded as weak supervision of the testing task. We conduct experiments with the Pythia (Biderman et al., 2023) and OLMO-7B (Groeneveld et al., 2024) models on translation, factual question answering, and reasoning tasks. Our findings show that:

1. Task-relevant n-gram pairs are better representative of task-related data than single n-grams.

2. Task performance is positively related to n-gram pair frequency.

3. The phenomenon of emergent abilities can be viewed as a mismatch between adequate task-related pretraining data and inadequate model size.

4. Small LMs memorize while large LMs generalize.

5. Instruction tuning helps LM make better use of pretraining data.

To the best of our knowledge, this is the first effort to analyze the origin of LLM capabilities on full pretraining corpora.

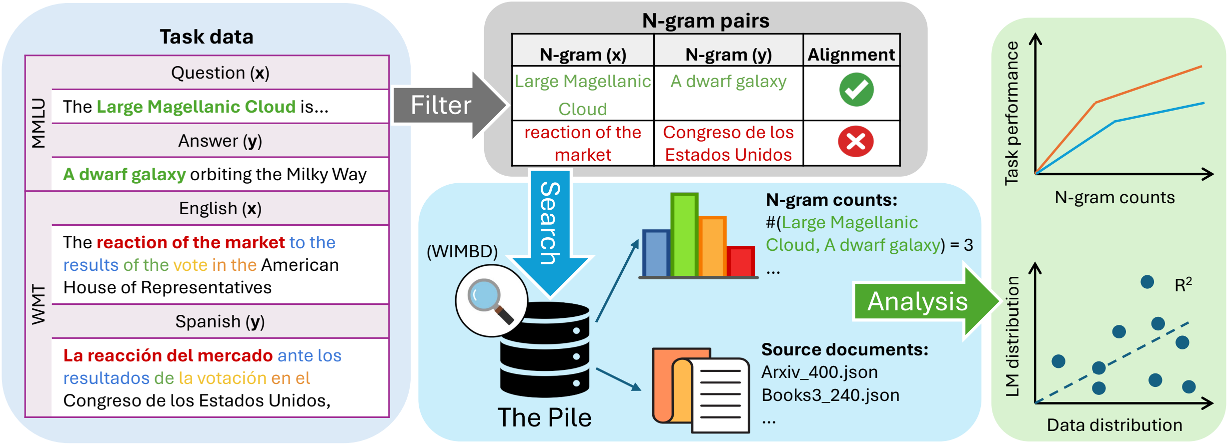
2. Problem Setting

The pretraining corpus of LLMs is usually huge, with billions even trillions of tokens. It is hard to directly analyze it without aggressive down-sampling (Kirchenbauer et al., 2024). In order to model the whole pretraining corpus, we

[*]Equal contribution [1]University of California, Santa Barbara [2]Allen Institute for AI [3]University of Washington [4]Carnegie Mellon University. Correspondence to: Xinyi Wang <xinyi_wang@ucsb.edu>, Antonis Antoniades <antonis@ucsb.edu>.

Published at ICML 2024 Workshop on Foundation Models in the Wild. Copyright 2024 by the author(s).

[1] Related work can be found in Appendix C due to space limit.


Task data
MMLU
Question (x)
The Large Magellanic Cloud is...
Answer (y)
A dwarf galaxy orbiting the Milky Way
WMT
English (x)
The reaction of the market to the results of the vote in the American House of Representatives
Spanish (y)
La reacción del mercado ante los resultados de la votación en el Congreso de los Estados Unidos,
Filter
N-gram pairs
Search
(WIMBD)
The Pile
N-gram counts:
#(Large Magellanic Cloud, A dwarf galaxy) = 3
...
Source documents:
Arxiv_400.json
Books3_240.json
...
Analysis
Task performance
N-gram counts
LM distribution
Data distribution
R²

N-gram (x)	N-gram (y)	Alignment
Large Magellanic Cloud	A dwarf galaxy	✓
reaction of the market	Congreso de los Estados Unidos	✗

Figure 1: An overview of our proposed analysis pipeline. For the chosen evaluation task (e.g. WMT (Callison-Burch et al., 2009), MMLU (Hendrycks et al., 2020)), we first mine task-relevant n-gram pairs by matching semantically similar n-grams from the task input x and target y, respectively. Then we search these n-grams across the pretraining corpus (e.g. the Pile (Gao et al., 2020)) using the WIMBD framework (Elazar et al., 2024) to obtain their counts and source documents. Finally, we conduct in-depth analyses with the obtained pretraining data statistics and LM inference results as detailed in Section 2.

adopt a simple but scalable way of modeling text distribution: n-gram frequency. Our proposed modeling approach is inspired by the classic n-gram LMs, with modifications to better serve our use cases.

Suppose we have a language model (LM) pretrained on a corpus $\mathcal{D}$. Considering the setting when we try to prompt the LM with an instruction text u and an input text x for an output text sequence $\hat{y}$. We denote the task that we are trying to perform by T and the ground truth output by y. Then we can denote the LM probability of the generated output $\hat{y}$ by $P_{\text{LM}}(\hat{y}|u, x)$. [2]

Suppose a text sequence y can be tokenized into sequences of tokens in the form of $[y_1, y_2, ..., y_L]$, $y_i \in V$, where V is the vocabulary. Define the set of all n-grams in y by $G_n(y) = \{y_{[i:i+n-1]}\}_{i\in[1,L-n]}$. Suppose $s \in G_n(y)$ for some $n > 0$, then the n-gram count is defined as $C(s, y) = \sum_{s_i \in G_n(y)} \mathbb{1}_{s_i=s}$.

We define a n-gram pair (s^x, s^y) to be a pair of n-grams that are semantically related to each other in the context of task T, with $s^x \subseteq x$, and $s^y \subseteq y$. Denote $C((s^x, s^y), (x, y)) = \min\{C(s_i, x), C(s_j, y)\}$ to be the number of occurrences of the n-gram pair (s^x, s^y) in the example (x, y) with $s^x \in x$ and $s^y \in y$. Let $H_n(T)$ be the set of relevant n-gram pairs mined from a set of data $D_T = \{(x_i, y_i)\}_i$ from the task T. Denote all possible combinations of input-output n-gram pairs by $A_n(T) = \cup_i[G_n(x_i) \times G_n(y_i)]$. We then filter all pairs $(s_j^x, s_j^y) \in A_n(T)$ by a similarity score $f(s_j^x, s_j^y) = \cos(E(s_j^x), E(s_j^y))$, with a threshold $\gamma_T \in (0, 1)$ treated as a hyperparameter. Here E is a pretrained text embedding model, and $\cos(\cdot, \cdot)$ denotes the cosine similarity between two vectors:

$$H_n(T) = \{(s_j^x, s_j^y) \mid f(s_j^x, s_j^y) > \gamma_T,\ (s_j^x, s_j^y) \in A_n(T)\}.$$

For tasks with extensive training data, we use a separate large training set as D_T to construct $H_n(T)$. For tasks without enough training data to ensure n-gram coverage of testing data, we directly use the testing data as D_T.

3. Experiment Setting

In this section, we introduce the datasets and models we use for analyzing the memorization and generalization behaviors of LLMs.

Models and Pretraining Corpus We utilize two families of fully open-sourced LMs: Pythia (Biderman et al., 2023) and OLMo (Groeneveld et al., 2024). Both of them are autoregressive Transformer-decoder-based LMs. Pythia (Biderman et al., 2023) is a classic suite of fully open-sourced LMs with a wide range of model sizes ranging from 13M to 12B parameters. All Pythia models are trained on Pile (Gao et al., 2020), a diverse pretraining corpus consisting of approximately 207B tokens. OLMo (Groeneveld et al., 2024) is a more recent, more performant suite of fully open-sourced LMs, pretrained on a larger corpus Dolma (Soldaini et al., 2024) with approximately 3T tokens, and instruction-tuned on Tulu (Wang et al., 2023b; Ivison et al., 2023).

Downstream Tasks We use three types of tasks: translation, factual question answering, and reasoning with multiple choices.

For translation, we use the **WMT**-09 dataset with a 2.5K testing set, consisting of European languages aligned with

[2] In practice, we use a minimal instruction template to indicate the input and output.

English translations using the Laser multilingual embedding model from the Europarl corpus. We use Pythia models and $n = 2$ for n-gram analysis, as it captures low-resource language data well. For factual question answering, we use the **TriviaQA** dataset with 95K question-answer pairs. We mine n-gram pairs using the E5 embedding model from the training set and use Pythia models on the 10k testing set. We regard answers as one n-gram and use $n = 5$ for analysis to capture task-specific training data. For reasoning with multiple choices, we use the **MMLU** benchmark covering 57 tasks. We mine n-gram pairs from the testing sets using the E5 model and use Pythia-6.9b-Tulu models. We use $n = 5$ for analysis, similar to TriviaQA.

Searching over Pretraining Data at Scale Given the scale of the LLM pretraining corpus, searching over the whole corpus $\mathcal{D}$ is non-trivial. We utilize the *What's In My Big Data?* **(WIMBD)** platform (Elazar et al., 2024), which is designed to search and retrieve huge text corpora through API calls, facilitating the exploration and analysis.

4. Quantify n-gram Contribution by Gradient

To justify the usage of n-gram pairs and understand the contribution of the task-relevant n-gram pairs to LLM's capability of solving the task, we propose a gradient-based analysis inspired by Han et al. (2023). More specifically, we quantify the contribution of a n-gram by the similarity between its pretraining loss gradient and the task example gradient. For a task-relevant n-gram pair (s^x, s^y), with testing task example (x, y), and $s^x \in x$, $s^y \in y$, the task gradient $g_T(s^y)$ is defined as: $g_T(s^y) = \nabla_\theta \sum_{i=1}^{L-n} \left[-\log P_{\text{LM}}(y_{[i:i+n-1]}|u, x, y_{[1:i-1]}) \mathbb{1}_{y_{[i:i+n-1]}=s^y}\right]$. Here θ denotes all parameters of the LM. To compute the gradient of the n-gram pair (s^x, s^y) in the pretraining corpus $\mathcal{D}$ at the pretraining time, we first retrieve K documents containing the n-gram pair from pretraining corpus $\mathcal{D}$. Here a maximum number of retrieved documents is set for computing efficiency, as back-propagating through a huge number of documents for each n-gram pair is infeasible for our study. Denoting the retrieved documents by $\{d^1, d^2, ..., d^K\}$, the pretraining gradient $g_D(s^y)$ is defined as: $g_D(s^y) = \nabla_\theta \sum_{j=1}^{K} \sum_{i=1}^{L-n} \left[-\log P_{\text{LM}}(d^j_{[i:i+n-1]}|d^j_{[1:i-1]}) \mathbb{1}_{d^j_{[i:i+n-1]}=s^y}\right]$.

In practice, we clip long documents to a maximum length to fit into the GPU memory. Then we compute the contribution of the pretraining data containing task-related n-gram pairs to the task example (x, y) by cosine similarity between the task gradient and pretraining gradient:

$$\beta_p(x, y, \mathcal{D}) = \sum_{(s^x, s^y) \in H_n(T)} C((s^x, s^y), \mathcal{D}) \cos(g_T(s^y), g_D(s^y))$$

Similarly, we can compute the contribution of task-relevant single n-grams $\beta_s(x, y, \mathcal{D})$ by computing the pretraining gradient over a set of documents that only contain s^y. Then we average the per-example contribution scores over the testing set and n-grams, to get an overall contribution of task-relevant n-gram pairs and single n-grams to the task. We plot the average gradient similarity over the testing set in Figure 2, with the solid line representing the n-gram pair data $\beta_p(x, y, \mathcal{D})$ and the dashed line representing the single n-gram data $\beta_s(x, y, \mathcal{D})$.

Across all three datasets, we observe that the n-gram pair data consistently contribute more to the task than the single n-gram data, over different model sizes and types, which confirms our hypothesis that task-relevant n-gram pairs are a good indicator of task-relevant pretraining data. For **WMT** and **TriviaQA**, we observe a U-shape trend in the gradient similarity in general, when model size increases. This indicates that LMs first become less dependent on the pretraining data when the model size grows, then become more dependent. When combined with other results, this can be understood as the model transitioning from memorizing the surface form of pretraining data to being able to compose and generate new content based on the pretraining data. We will revisit this point later.

For **MMLU**, we observe that the instruction-tuned model (right) in fact has a larger gradient similarity than the base model (left), which implies that instruction-tuning improves LMs' ability to utilize the task-related pretraining data to solve difficult tasks.

5. Estimating Data Distribution

In this section, we model the data distribution with the frequency of the previously defined n-gram pairs. Denote $C((s^x, s^y), (x, y)) = \min\{C(s^x, x), C(s^y, y)\}$ to be the number of occurrences of the n-gram pair (s^x, s^y) in a task example (x, y). We can also define the n-gram parallel pair count in a document string d by $C((s^x, s^y), d) = \sum_{s_i \in G_n(d)} \sum_{s_j \in G_n(d)} \mathbb{1}_{s_i = s^x} \mathbb{1}_{s_j = s^y}$. If we define the pretraining data distribution to be over all the possible n-grams, then the empirical data probability of a n-gram pair (s^x, s^y) would be $P_{\mathcal{D}}(s^x, s^y) \propto C((s^x, s^y), \mathcal{D}) = \sum_{d \in \mathcal{D}} C((s^x, s^y), d)$.

5.1. Task-related Data Frequency v.s. Task Performance

In this section, we show a strong positive correlation between the frequency of task-related data in the pretraining corpus and LM's task performance. We also show that some abilities appear to be emergence because of the mismatch between data and model size.

We can estimate the probability of task T related data appearing in the pretraining corpus as the probability of any

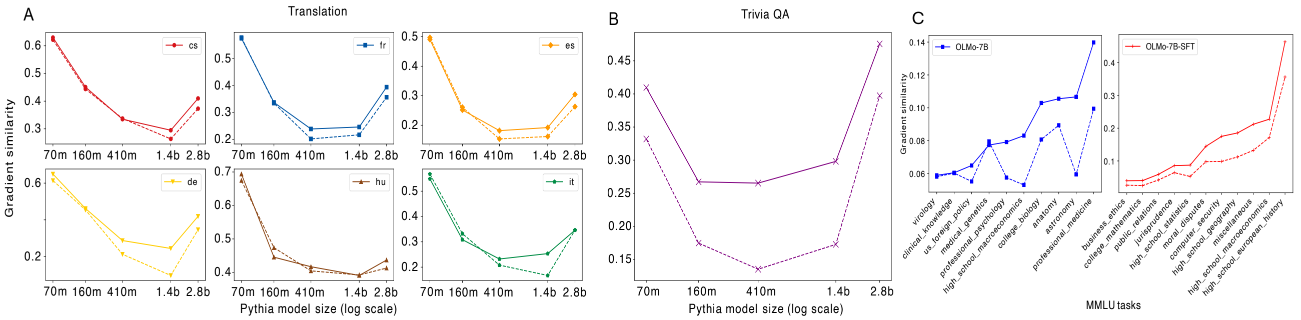


Figure 2: Cosine similarity between n-gram task gradient and pretraining gradient. A: Gradient v.s. Pythia model size with **WMT**; B: Gradient v.s. Pythia model size with **TriviaQA**; C: Gradient v.s. different tasks in **MMLU** with base (left), and instruction-tuned (right) OLMo-7B. Solid: Single Instances. Dotted: X-Y Pairs.

of the task-relevant n-gram pairs appearing:

$$P_{\mathcal{D},n}(T) = \sum_{(s^x,s^y)\in H_n(T)} P_{\mathcal{D}}(s^x, s^y) \propto \sum_{(s^x,s^y)\in H_n(T)} C((s^x, s^y), \mathcal{D})$$

We then investigate the relationship between LMs' task capabilities and task-related pretraining data distribution by plotting task performances against the sum of all task-related n-gram pair counts. For **WMT**, we count relevant n-gram pairs for each language and use BLEU as the performance metric. For **TriviaQA**, we group testing examples by n-gram pair matching count and use the exact match rate. Figure 3 shows that task performance generally increases with the number of task-related n-gram pairs when the model size is large enough. We observe a threshold model size (around 400M) below which performance remains near-random regardless of n-gram pair count. At the threshold, we see a sudden phase change, or "emergent abilities," where performance jumps from near-random to reasonable, with more significant jumps for more n-gram pairs found in pretraining. This suggests emergent abilities require both a large enough model size and sufficient relevant pretraining data.

5.2. Data Distribution v.s. Language Model Distribution

In this section, we measure the memorization of LMs by the similarity between the LM distribution and the pretraining data distribution, while we measure the generalization of LMs by the novelty of the generation, in terms of distribution difference or the amount of novel n-grams.

We can decompose a pair of input-output text (x, y) into many such n-gram pairs. Then suppose the n-gram pairs are mutually independent, the empirical data probability of a pair of translated sentences (x, y) can be decomposed into:

$$\begin{aligned} P_{\mathcal{D},n}(x, y) &= \prod_{s_i\in G_n(x)} \prod_{s_j\in G_n(y)} P_D(s_i, s_j)^{\mathbb{1}_{(s_i,s_j)\in H_n(T)}} \\ &\propto \exp\Big(\sum_{(s_i,s_j)\in H_n(T)} C((s^x, s^y), (x, y)) \log C((s^x, s^y), \mathcal{D})\Big) \quad (1) \end{aligned}$$

To avoid excessive searches over the pretraining corpus, we suppose the marginal distribution $P_{\mathcal{D},n}(x)$ is similar for all input x's from task T. Then we approximate the conditional distribution as $P_{\mathcal{D},n}(y|x) \propto P_{\mathcal{D},n}(x, y)$. Similarly, we can decompose the LM probability $P_{\text{LM}}(\hat{y}|u, x)$ into n-grams as follows:

$$\begin{aligned} \tilde{P}_{\text{LM},n}(\hat{y}|u, x) = \exp \sum_{i=1}^{L-n} [&\log P_{\text{LM}}(\hat{y}_{[I:i+n-1}|u, x, \hat{y}_{[1:i-1]}) \\ &\sum_{s^x\in G_n(x)} \mathbb{1}_{(s^x,\hat{y}_{[i:i+n-1]})\in H_n(T)}] \quad (2) \end{aligned}$$

Then we can compare the (empirical) data distribution and the LM distribution. Since the sample space of text sequence $\hat{y}$ is too large, even after decomposed into n-grams, the common distribution similarity measure requiring the whole distribution over the sample space would be infeasible. So we choose to fit a linear regression instead from the log data probability $\log P_{\mathcal{D},n}(\hat{y}|x)$ to the log LM probability $\log \tilde{P}_{\text{LM},n}(\hat{y}|u, x)$ for each training example $(x, \hat{y})$. In this way, we take into consideration the possible mismatch in scale between these two distributions, but reserve the distribution shape. We measure the closeness of these two distributions by the R^2 score of the regression.

The left two panels of Figure 4, for WMT and TriviaQA, R^2 score first decreases then slightly increases as model size increases, suggesting that smaller models learn a distribution more similar to pretraining data and are more data dependent. The decreasing similarity to data distribution and gradient similarity suggests that memorization plays a

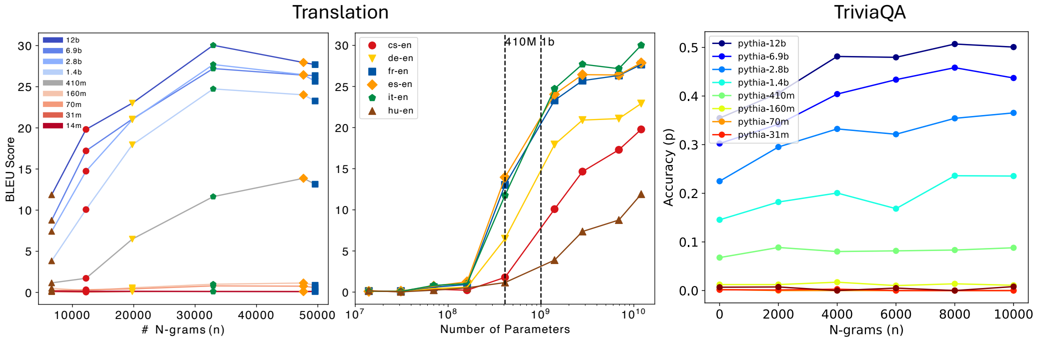


Figure 3: BLEU score v.s. total n-gram pair count in the Pile (left) and Pythia model parameters (middle) for different languages in **WMT**. Right: **TriviaQA** exact match score v.s. total n-gram pair count in the Pile with different Pythia model sizes.

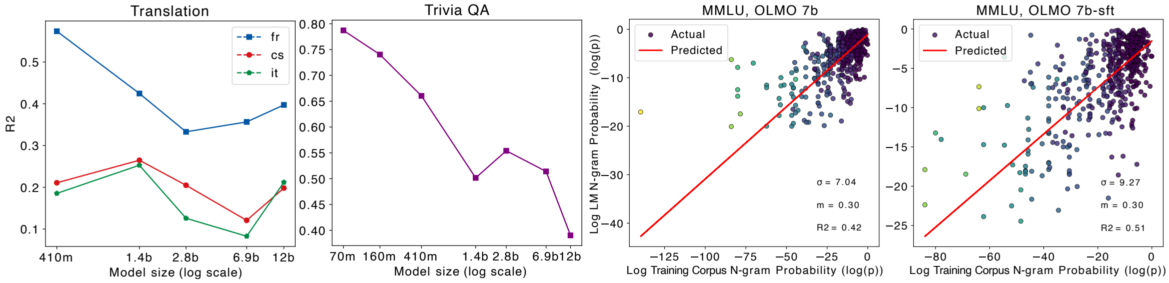


Figure 4: R^2 score of linear regression from data distribution to LM distribution v.s. Pythia model size for different languages in **WMT** (leftmost), **TriviaQA** (middle left). We show the scatter plots for linear regression on all **MMLU** data combined, as the number of examples with sufficient counts in each task is insufficient. The R^2 score is 0.42 with the base OLMo-7B model (middle right) and 0.51 with the instruction-tuned OLMo-7B model (rightmost).

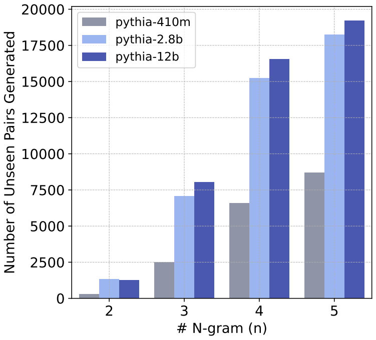


Figure 5: Number of unique n-gram pairs generated on **WMT** v.s. the value of n for n-gram length, with different Pythia models.

smaller role while generalization strengthens task performance. Larger models generate more novel n-gram pairs not seen in pretraining, supporting this transition. Emergent abilities, positively related to task-relevant data amount, can be understood as LMs transitioning from pure memorization to generalization. Interestingly, there is a slight increase in distribution and gradient similarity for sufficiently large models, implying that while generalization strengthens, memorization also increases, allowing LMs to better utilize training data.

In the right two panels of Figure 4, for **MMLU**, we observe that the instruction-tuned LM is closer to the pretraining data distribution than the base LM distribution, which is surprising. This echoes the gradient similarity results in Figure 2, that the instruction-tuned model shows more dependency on relevant pretraining data. We have extensively searched the instruction tuning dataset, Tulu, to make sure there are no task-relevant n-gram pairs, thus rule out the effect of memorizing the instruction tuning set. As reported in Groeneveld et al. (2024), the performance of the instruction-tuned OLMo model is significantly better on MMLU than the base model. This indicates that instruction-tuning boosts the existing capabilities that LMs have already learned from pretraining, instead of learning new capabilities.

6. Conclusion

In this paper, we propose a scalable method to trace LLMs' capabilities back to the pretraining data by searching for pretraining data at the n-gram level and enforcing semantic similarity within n-gram pairs using embedding models. This approach enables an extensive search across the pretraining corpus while allowing direct interpretation of the n-grams. Experiments with Pythia and OLMO models on various tasks reveal that task-relevant n-gram pairs play a crucial role in model performance, with small models tending to memorize and larger models demonstrating enhanced generalization. This analysis is a first step in comprehensively analyzing the origins of LLM capabilities.

References

S. Arora and A. Goyal. A theory for emergence of complex skills in language models, 2023.

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

C. Callison-Burch, P. Koehn, C. Monz, and J. Schroeder, editors. *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, Mar. 2009. Association for Computational Linguistics. URL https://aclanthology.org/W09-0400.

N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

S. Chan, A. Santoro, A. Lampinen, J. Wang, A. Singh, P. Richemond, J. McClelland, and F. Hill. Data distributional properties drive emergent in-context learning in transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18878–18891. Curran Associates, Inc., 2022.

Y. Chen, C. Zhao, Z. Yu, K. McKeown, and H. He. Parallel structures in pre-training data yield in-context learning. *arXiv preprint arXiv:2402.12530*, 2024.

Y. Elazar, A. Bhagia, I. Magnusson, A. Ravichander, D. Schwenk, A. Suhr, P. Walsh, D. Groeneveld, L. Soldaini, S. Singh, H. Hajishirzi, N. A. Smith, and J. Dodge. What's in my big data?, 2024.

V. Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

V. Feldman and C. Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2881–2891. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1e14bfe2714193e7af5abc64ecbd6b46-Paper.pdf.

L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

D. Groeneveld, I. Beltagy, P. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. H. Jha, H. Ivison, I. Magnusson, Y. Wang, S. Arora, D. Atkinson, R. Authur, K. Chandu, A. Cohan, J. Dumas, Y. Elazar, Y. Gu, J. Hessel, T. Khot, W. Merrill, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, V. Pyatkin, A. Ravichander, D. Schwenk, S. Shah, W. Smith, N. Subramani, M. Wortsman, P. Dasigi, N. Lambert, K. Richardson, J. Dodge, K. Lo, L. Soldaini, N. A. Smith, and H. Hajishirzi. Olmo: Accelerating the science of language models. *Preprint*, 2024.

X. Han, D. Simig, T. Mihaylov, Y. Tsvetkov, A. Celikyilmaz, and T. Wang. Understanding in-context learning via supportive pretraining data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12660–12673, 2023.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

H. Ivison, Y. Wang, V. Pyatkin, N. Lambert, M. Peters, P. Dasigi, J. Jang, D. Wadden, N. A. Smith, I. Beltagy, and H. Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2, 2023.

M. Jiang, K. Z. Liu, M. Zhong, R. Schaeffer, S. Ouyang, J. Han, and S. Koyejo. Investigating data contamination for pre-training language models. *arXiv preprint arXiv:2401.06059*, 2024.

J. Kirchenbauer, G. Honke, G. Somepalli, J. Geiping, D. Ippolito, K. Lee, T. Goldstein, and D. Andre. Lmd3: Language model data density dependence. *arXiv preprint arXiv:2405.06331*, 2024.

I. Magar and R. Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, 2022.

B. Prystawski, M. Y. Li, and N. Goodman. Why think step by step? reasoning emerges from the locality of experience. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=rcXXNFVlEn`.

L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, V. Hofmann, A. H. Jha, S. Kumar, L. Lucy, X. Lyu, N. Lambert, I. Magnusson, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, A. Ravichander, K. Richardson, Z. Shen, E. Strubell, N. Subramani, O. Tafjord, P. Walsh, L. Zettlemoyer, N. A. Smith, H. Hajishirzi, I. Beltagy, D. Groeneveld, J. Dodge, and K. Lo. Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2024.

S. Srivastava, A. PV, S. Menon, A. Sukumar, A. Philipose, S. Prince, S. Thomas, et al. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. *arXiv preprint arXiv:2402.19450*, 2024.

X. Wang, W. Zhu, M. Saxon, M. Steyvers, and W. Y. Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL `https://openreview.net/forum?id=BGvkwZEGt7`.

X. Wang, A. Amayuelas, K. Zhang, L. Pan, W. Chen, and W. Y. Wang. Understanding the reasoning ability of language models from the perspective of reasoning paths aggregation. *arXiv preprint arXiv:2402.03268*, 2024.

Y. Wang, H. Ivison, P. Dasigi, J. Hessel, T. Khot, K. Chandu, D. Wadden, K. MacMillan, N. A. Smith, I. Beltagy, and H. Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023b. URL `https://openreview.net/forum?id=w4zZNC4ZaV`.

S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=RdJVFCHjUMI`.

C. Zhang, D. Ippolito, K. Lee, M. Jagielski, F. Tramèr, and N. Carlini. Counterfactual memorization in neural language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=67o9UQgTD0`.

Appendix

A. Limitations

While this paper provides valuable insights into how large-scale pretraining corpus contributes to the emergent abilities of LLMs through n-gram search, there are a few limitations that we want to list out. First, the main model we use, Pythia, and the main pretraining corpus, Pile, is slightly outdated, and has been outperformed by many new open-source LLMs. Most open-source LLMs lack corresponding pretraining data and have limited model sizes, hindering scaling effect studies. The recently released Neo LLMs and Matrix pretraining corpus offer better experimental opportunities for future work. The current WIMBD system has limitations in searching larger corpora like Dolma (3T tokens) and Matrix (4.7T tokens), requiring improved searching and retrieval methods. The quality of task-relevant n-gram pairs is highly sensitive to the filtering method, and while the current embedding similarity-based approach is effective, better filtering methods could significantly enhance the analysis, which is left for future research.

B. Broader Impacts

The insights and methodologies developed in this paper have several significant implications for the broader field of artificial intelligence, particularly in the development and deployment of large language models (LLMs). Understanding the balance between memorization and generalization within LLMs is crucial for both advancing the theoretical foundation of machine learning and addressing practical concerns related to their use.

Enhanced Model Interpretability: By extending the definition of memorization and examining how LLMs utilize their pretraining data, our research contributes to a deeper understanding of the internal mechanics of these models. This improved interpretability can help researchers and practitioners diagnose and mitigate issues related to data bias, model robustness, and unexpected behaviors in AI systems.

Pri vacy and Security Considerations: Our findings have direct implications for privacy and security in AI. Demonstrating how LLMs memorize and potentially recall training data underscores the need for rigorous data handling and anonymization techniques. It raises awareness about the risks of inadvertent leakage of sensitive information, thereby informing policy and best practices for data usage in training large models.

Economic and Societal Impact: As LLMs become more integral to various industries, understanding their capabilities and limitations can have significant economic and societal implications. Our research can help businesses and policymakers make informed decisions about deploying these models, ensuring they are used ethically and effectively. This, in turn, can lead to more reliable and trustworthy AI systems, fostering greater public trust and acceptance.

C. Related Work

Understanding LLMs' capabilities from training data Because of the scale of the data and model sizes, most work on understanding LLMs attempts to examine how LLMs gain their capabilities from synthetic experiments or on a small scale (Arora and Goyal, 2023). Prystawski et al. (2023) and Wang et al. (2024) discuss how the reasoning ability of language models is a consequence of their pretraining data. Prystawski et al. (2023) discusses how chain-of-thought reasoning is effective in autoregressive language models because of local structure within pretraining data, and Wang et al. (2024) derives novel conclusions from known facts by aggregating reasoning paths seen in pretraining data. On the other hand, Xie et al. (2022) and Wang et al. (2023a) discuss how in-context learning is a by-product of learning the pretraining data distribution. They both suggest that language models learn to implicitly infer a latent variable from the given prompt, as the pretraining data is generated from some unknown latent variable. Additionally, Chan et al. (2022) propose that the distributional properties of training data drive emergent in-context learning behaviors in large language models. Chen et al. (2024) also highlights the significance of parallel structures in pretraining data for the emergence of in-context learning.

However, the small-scale nature of such analysis is antithetical to the commonly believed main driving factor behind the performance of LLMs: scaling. Recently, Kirchenbauer et al. (2024) proposes to provide statistical evidence of the dependence of a target model capabilities on subsets of its training data, by estimating the data distribution with an embedding-induced kernel. However, their estimation is based on a very small portion of the pretraining data (around 0.3%) as computing the embeddings of a huge dataset is very non-trivial. To get a better estimation of the whole distribution of the pretraining data, Elazar et al. (2024) construct a retrieval system, **WIMBD**, that can efficiently search n-gram phrases over hundreds and thousands of GBs of pretraining data. However, it is unclear what insights of the LLMs trained on these datasets can be obtained from such searches.

New methods and analysis to investigate these capabilities at scale and to understand the role of scaling are needed to obtain useful insights into real-world LLMs. In this work, we aim to provide an in-depth analysis of the origin of the general zero-shot capabilities of LLMs, by performing full searches across the whole pretraining corpus with the WIMBD framework.

Memorization v.s. generalization The phenomenon of machine learning models being able to perfectly memorize the training data has been studied in many previous works. Most of them define LLM memorization as exactly recalling the training examples by designed prompting, including the memorization of rare long-tail data, like private information (Zhang et al., 2023), and the contamination of testing sets (Jiang et al., 2024). Carlini et al. (2022) found that the exact copy and pasting behaviors are more prevalent in larger LMs.

Several papers have studied the interplay between memorization and generalization of training data. Feldman (2020) prove that memorizing the training data is in fact required for optimal generalization on testing data. Many works along this line (Feldman and Zhang, 2020; Zhang et al., 2023) extend the original definition of memorization by quantifying the extent of memorizing a training example with the performance difference when including and excluding this specific example in training data. However, this definition is impractical for large-scale analysis of pretrained LLMs as it would require retraining an LLM from scratch to analyze one data point. In this paper, we propose a new definition of memorization by using n-gram counts, which is more suitable for large-scale analysis with LLMs.

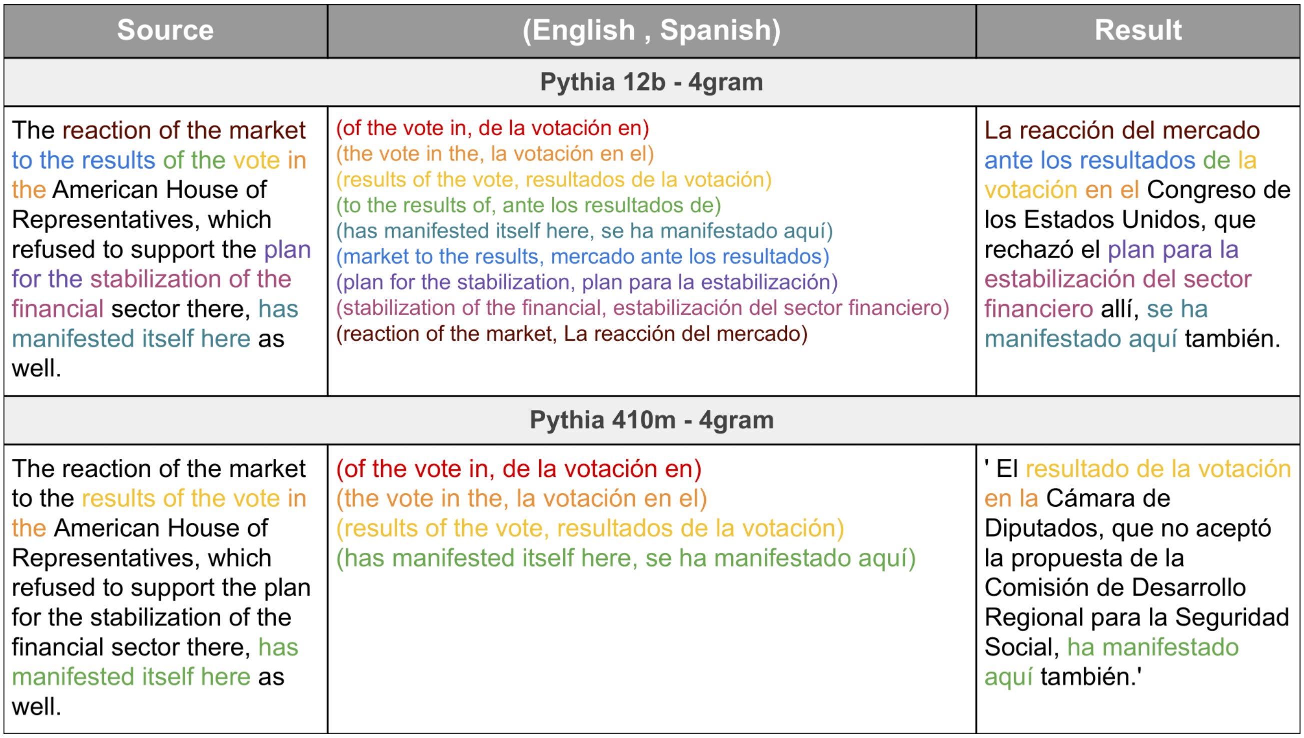
D. Experiment Details

We perform our experiments on 8 GPU 40G A100 working stations. Below is the license information for the datasets we used:

- Pile: MIT license. URL: `https://github.com/EleutherAI/the-pile/tree/master`
- Tulu: ODC-BY license. URL: `https://huggingface.co/datasets/allenai/tulu-v2-sft-mixture`
- WMT-09: published with the WMT workshop. URL: `https://www.statmt.org/wmt09/translation-task.html`
- TriviaQA: Apache License 2.0. URL: `https://nlp.cs.washington.edu/triviaqa/`
- MMLU: MIT license. URL: `https://github.com/hendrycks/test`

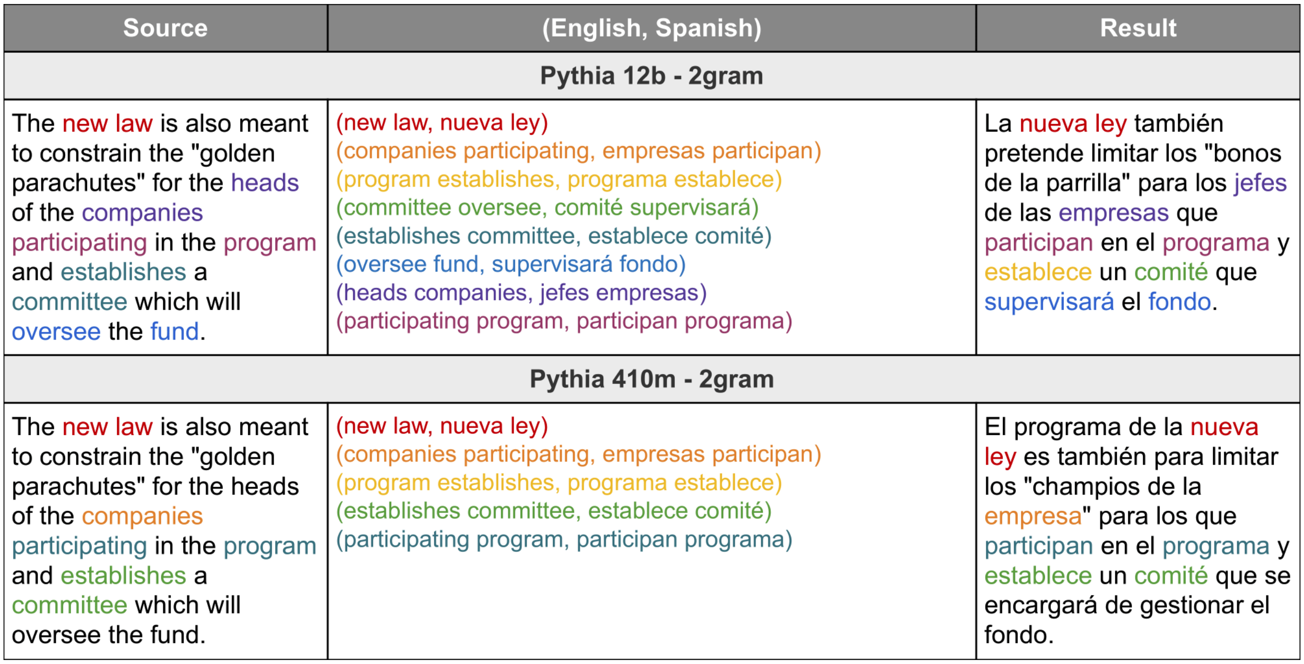
E. n-gram Pair Examples

In this section, we present some representative examples collected from the analysis for the different tasks evaluating, including Translation and Question-Answering (MMLU, TriviaQA). In order to show examples from the different experiments, we show examples with different model sizes and number of n-grams.



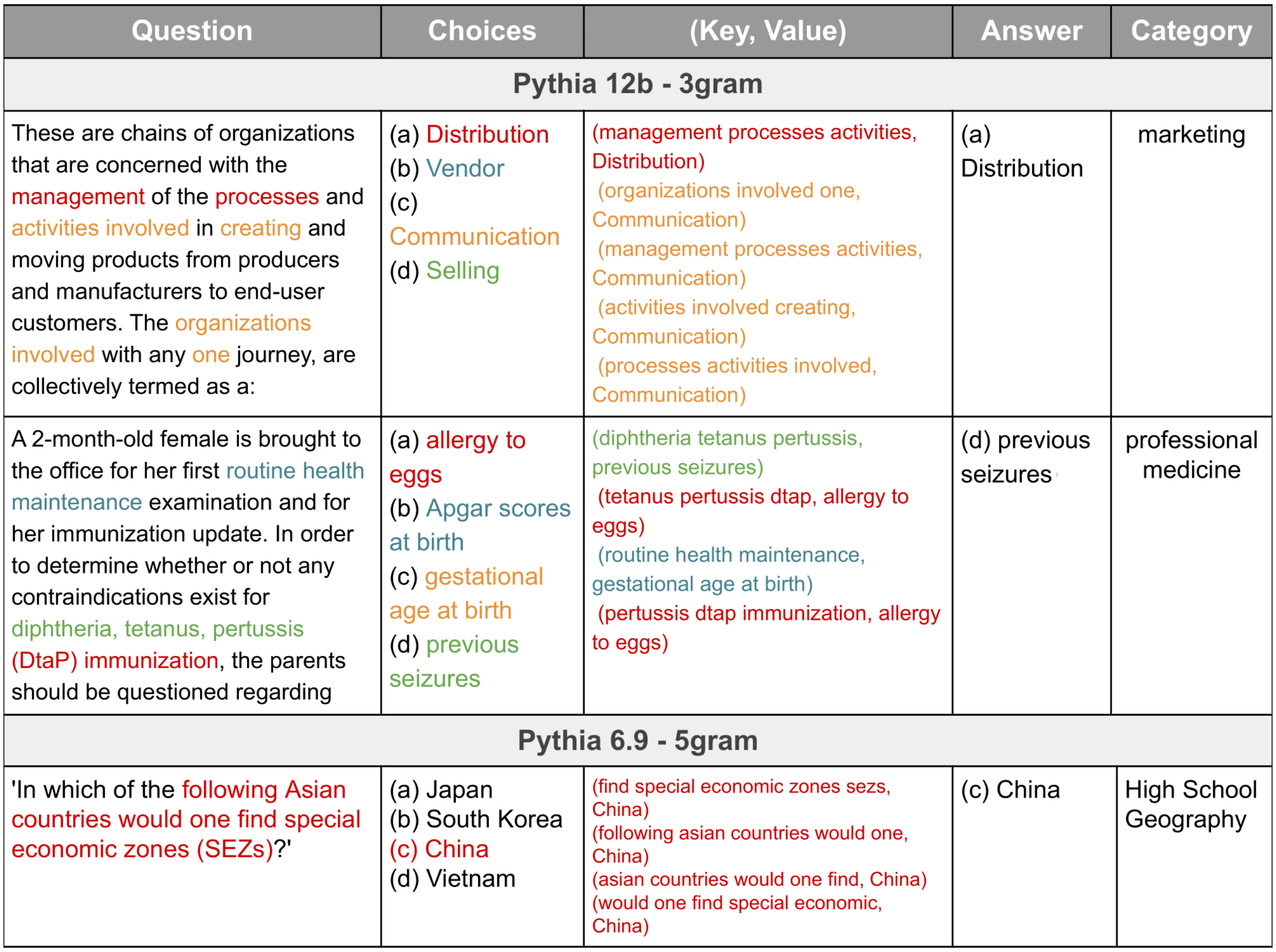
Source	(English , Spanish)	Result
Pythia 12b - 4gram		
The reaction of the market to the results of the vote in the American House of Representatives, which refused to support the plan for the stabilization of the financial sector there, has manifested itself here as well.	(of the vote in, de la votación en) (the vote in the, la votación en el) (results of the vote, resultados de la votación) (to the results of, ante los resultados de) (has manifested itself here, se ha manifestado aquí) (market to the results, mercado ante los resultados) (plan for the stabilization, plan para la estabilización) (stabilization of the financial, estabilización del sector financiero) (reaction of the market, La reacción del mercado)	La reacción del mercado ante los resultados de la votación en el Congreso de los Estados Unidos, que rechazó el plan para la estabilización del sector financiero allí, se ha manifestado aquí también.
Pythia 410m - 4gram		
The reaction of the market to the results of the vote in the American House of Representatives, which refused to support the plan for the stabilization of the financial sector there, has manifested itself here as well.	(of the vote in, de la votación en) (the vote in the, la votación en el) (results of the vote, resultados de la votación) (has manifested itself here, se ha manifestado aquí)	' El resultado de la votación en la Cámara de Diputados, que no aceptó la propuesta de la Comisión de Desarrollo Regional para la Seguridad Social, ha manifestado aquí también.'

Figure 6: Examples of mined pairs for the translation Task (English to Spanish) using Pythia Models with 4-Gram analysis. Models evaluated include those with 12 billion and 410 million parameters.



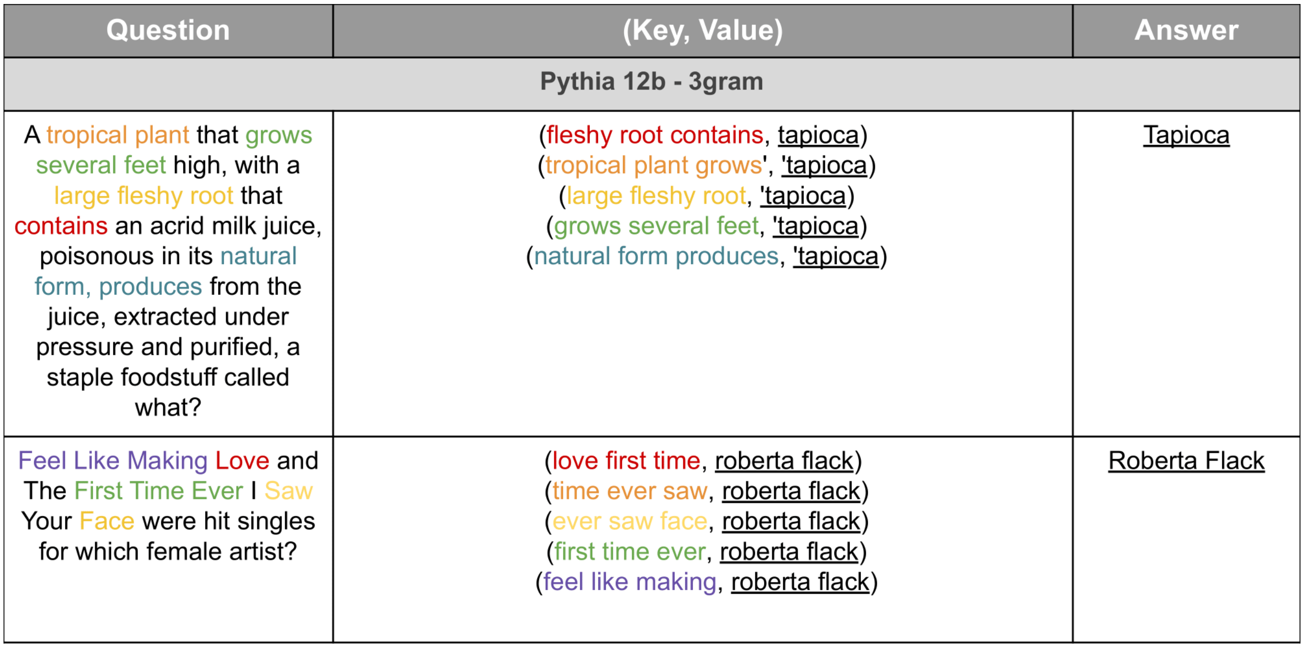
Source	(English, Spanish)	Result
Pythia 12b - 2gram		
The new law is also meant to constrain the "golden parachutes" for the heads of the companies participating in the program and establishes a committee which will oversee the fund.	(new law, nueva ley) (companies participating, empresas participan) (program establishes, programa establece) (committee oversee, comité supervisará) (establishes committee, establece comité) (oversee fund, supervisará fondo) (heads companies, jefes empresas) (participating program, participan programa)	La nueva ley también pretende limitar los "bonos de la parrilla" para los jefes de las empresas que participan en el programa y establece un comité que supervisará el fondo.
Pythia 410m - 2gram		
The new law is also meant to constrain the "golden parachutes" for the heads of the companies participating in the program and establishes a committee which will oversee the fund.	(new law, nueva ley) (companies participating, empresas participan) (program establishes, programa establece) (establishes committee, establece comité) (participating program, participan programa)	El programa de la nueva ley es también para limitar los "champios de la empresa" para los que participan en el programa y establece un comité que se encargará de gestionar el fondo.

Figure 7: Examples of mined pairs for the translation Task (English to Spanish) using Pythia Models with 2-Gram analysis. Models evaluated include those with 12 billion and 410 million parameters.



Question	Choices	(Key, Value)	Answer	Category
Pythia 12b - 3gram				
These are chains of organizations that are concerned with the management of the processes and activities involved in creating and moving products from producers and manufacturers to end-user customers. The organizations involved with any one journey, are collectively termed as a:	(a) Distribution (b) Vendor (c) Communication (d) Selling	(management processes activities, Distribution) (organizations involved one, Communication) (management processes activities, Communication) (activities involved creating, Communication) (processes activities involved, Communication)	(a) Distribution	marketing
A 2-month-old female is brought to the office for her first routine health maintenance examination and for her immunization update. In order to determine whether or not any contraindications exist for diphtheria, tetanus, pertussis (DtaP) immunization, the parents should be questioned regarding	(a) allergy to eggs (b) Apgar scores at birth (c) gestational age at birth (d) previous seizures	(diphtheria tetanus pertussis, previous seizures) (tetanus pertussis dtap, allergy to eggs) (routine health maintenance, gestational age at birth) (pertussis dtap immunization, allergy to eggs)	(d) previous seizures	professional medicine
Pythia 6.9 - 5gram				
'In which of the following Asian countries would one find special economic zones (SEZs)?'	(a) Japan (b) South Korea (c) China (d) Vietnam	(find special economic zones sezs, China) (following asian countries would one, China) (asian countries would one find, China) (would one find special economic, China)	(c) China	High School Geography

Figure 8: Examples of mined pairs for the MMLU Task using Pythia Models with 2-Gram and 5-gram analysis. Models include 12 billion and 6.9 billion parameters.



Question	(Key, Value)	Answer
Pythia 12b - 3gram		
A tropical plant that grows several feet high, with a large fleshy root that contains an acrid milk juice, poisonous in its natural form, produces from the juice, extracted under pressure and purified, a staple foodstuff called what?	(fleshy root contains, tapioca) (tropical plant grows', 'tapioca) (large fleshy root, 'tapioca) (grows several feet, 'tapioca) (natural form produces, 'tapioca)	Tapioca
Feel Like Making Love and The First Time Ever I Saw Your Face were hit singles for which female artist?	(love first time, roberta flack) (time ever saw, roberta flack) (ever saw face, roberta flack) (first time ever, roberta flack) (feel like making, roberta flack)	Roberta Flack

Figure 9: Examples of mined pairs for the TriviaQA Task using Pythia Models (12b) with 3-Gram analysis.