Reasoning Ability Emerges in Large Language Models as Aggregation of Reasoning Paths: A Case Study With Knowledge Graphs

Xinyi Wang [1] William Yang Wang [1]

Abstract

This study focuses on the emergence of reasoning abilities in large language models (LLMs). While LLMs have shown remarkable capabilities in complex reasoning tasks, the exact origin of this ability and its relationship to pre-training and fine-tuning stages remain unclear. Previous research has explored in-context learning but has not fully addressed reasoning abilities such as logical reasoning or math deduction. The paper proposes investigating reasoning in LLMs through reasoning over knowledge graphs. The experiments demonstrate the importance of the pre-training sequence in enabling effective reasoning. The findings suggest that LLMs acquire reasoning abilities during pre-training rather than fine-tuning. Furthermore, training LLMs with next-token prediction enables them to aggregate relevant reasoning paths and derive new conclusions. The empirical results support the explanation of LLMs predicting unseen facts using a path ranking algorithm.

1. Introduction

Recently, significant advancements have been observed in large language models (LLMs), exemplified by GPT4 (Bubeck et al., 2023), which have demonstrated remarkable capabilities in performing intricate reasoning tasks involving known or given facts. These tasks include first-order logical reasoning and solving mathematical word problems. Nevertheless, the precise timing in the training process and the manner in which this reasoning ability emerges remains uncertain. It is unclear whether the reasoning ability originates from the pre-training phase or the instruction tuning phase, and how LLMs acquire the skill of reasoning from the relatively simple objective of predicting the next word.

LLMs are renowned for acquiring astonishing capabilities through extensive pre-training. In-context learning is one such capability, where LLMs learn to follow given demonstrations during inference. While several studies have attempted to explain in-context learning using either Bayesian inference (Jiang, 2023; Hahn & Goyal, 2023; Xie et al., 2022; Wang et al., 2023) or gradient descent (von Oswald et al., 2022; Dai et al., 2022; Akyürek et al., 2022; Li et al., 2023), few studies focus on elucidating the emergence of reasoning ability—the capacity to draw novel conclusions from existing knowledge. Prystawski & Goodman (2023) propose that the chain-of-thought technique is effective when the training data exhibits localized structure with respect to dependencies between variables. However, their experiments equate reasoning to conditional probability estimation of boolean variables with intermediate variables, which can be considered overly abstract compared to real-world reasoning processes such as logical reasoning or mathematical deduction.

In this paper, we aim to comprehend the development of reasoning ability in LLMs by examining a straightforward yet general case of reasoning: reasoning over knowledge graphs. A knowledge graph stores facts in the form of triples (e_1, r, e_2), where e_1 and e_2 represent entities connected by the relationship r. Knowledge graphs can be incomplete, lacking certain relations between existing entities. These missing relations can typically be inferred from the known facts stored in the knowledge graph, often employing logical rules. For instance, the relation (A, `isGrandChildof`, C) can be derived from the triples (A, `isSonOf`, B) and (B, `isSonOf`, C).

We can consider the pre-training corpus of LLMs as a natural language serialization of an extensive knowledge graph. Subsequently, instruction tuning allows for direct responses to human-provided queries. Remarkably, without explicitly encoding reasoning rules, LLMs can accurately deduce previously unseen facts from known ones to answer a given query. In our experiments, we find that the construction of the pre-training sequence plays a crucial role. Pre-training is most effective when we serialize the knowledge graph as ”chains” of steps or random walk paths on the graph. On the other hand, pre-training on disconnected factual sentences leads to significantly inferior reasoning ability. Our

[1]Department of Computer Science, University of California, Santa Barbara. Correspondence to: Xinyi Wang <xinyi$_{w}ang@ucsb.edu$>.

Work presented at the ES-FoMo Workshop at ICML 2023.

experimental results also suggest that the reasoning ability of LLMs predominantly stems from pre-training rather than fine-tuning.

By training LLMs with the next-token prediction objective, the models are expected to learn the marginal distribution of text sequences. However, when provided with appropriate training data in the form of text sequences, we hypothesize that LLMs are capable of aggregating pertinent reasoning paths to derive new conclusions. Empirical evidence supports our hypothesis, as the predicted distributions of LLMs on unseen facts closely align with the predicted distribution obtained through the path ranking algorithm (Lao et al., 2011) — a weighted aggregation of all potential reasoning paths.

2. Problem setting

Consider a knowledge graph $\mathcal{G} = \{(e_1^i, r^i, e_2^i)\}_{i=1}^n$ consisting of n triples, such that $r^i(e_1^i, e_2^i)$ for all i. Let $\mathcal{R}$ denote the set of all possible relations and $\mathcal{E}$ denote the set of all entities. Let $C(e)$ denote all the children of e. i.e. the set of $e' \in \mathcal{E}$ such that $r(e, e')$ for some $r \in \mathcal{R}$. Let $C_r(e)$ denote all the children of e with relation r. i.e. the set of $e' \in \mathcal{E}$ such that $r(e, e')$. Our goal is to predict a set of unseen triples $\mathcal{T} = \{(e_1^j, r^j, e_2^j)\}_{j=1}^m$, $e_1^j, e_2^j \in \mathcal{E}$, $r^j \in \mathcal{R}$, by training a Transformer based generative language model from scratch on the given knowledge graph $\mathcal{G}$.

To translate a triple into a sentence, we consider assigning k templates $\tau_r^1, \tau_r^2, ..., \tau_r^k$ to each relation $r \in \mathcal{R}$. Then each time when we sample a triple, we randomly translate it into a sentence using one of the templates. For example, the triple (e_1, `locatedIn`, e_2) can be translated to "e_1 `is located in` e_2" or "e_2 `is where` e_1 `is situated`". We consider two ways of translating the entities into natural language. One is to translate the entity into its meaningful name directly. To avoid learning spurious correlations through the lexical form of the entity names, we also consider adding each entity as a new token to the Transformer.

3. Pre-training v.s. fine-tuning

We consider two ways of constructing pre-training data:

- Randomly sample $l \sim U[1, L_{max}]$ triples from the knowledge graph $\mathcal{G}$.
- Randomly sample a start entity $e \sim U(\mathcal{E})$, then perform random walk on $\mathcal{G}$ from e by sampling the next node using $e' \sim U(C(e))$.

We use L_{max} to denote the maximum number of sentences in a paragraph and $U(\cdot)$ to denote the uniform distribution. Then we translate each triple into a sentence and concatenate all the sentences to become a paragraph. The paragraphs are then concatenated together and separated by the special end-of-sequence token to form text chuncks of the same length. The training loss would be the normal next-token prediction loss (we denote the language model parameters by θ):

$$\mathcal{L}_{LM}(\theta) = \sum_t \log P_\theta(w_{t+1}|w_{1:t})$$

To create an instruction-tuning dataset, we first curate a set of logical reasoning rules $\mathcal{H}$ for each relation in the test set. For example, a rule $h \in \mathcal{H}$ can be h : (e_1, `locatedIn`, e_2) $\leftarrow$ (e_1, `neighborOf`, e_3) $\wedge$ (e_3, `locatedIn`, e_2). Let $\mathcal{H}[r]$ denote the set of rules that implies relation r. Here we only consider logic rules with conjunctive form: $\forall \{e_i\}_{i=0}^l \subset \mathcal{E}, r(e_0, e_l) \leftarrow r_1(e_0, e_1) \wedge r_2(e_1, e_2) \wedge ... \wedge r_l(e_{l-1}, e_l)$. We abbreviate such rule by $h = [r_1, r_2, ..., r_l]$. In this case, each rule can be achieved by some random walk paths. Note the logic rules may not always hold, but they should be a good representation of the correct reasoning paths most of the time.

Based on these rules, we construct instruction-tuning examples in the form of a binary query followed by a valid reasoning path and a binary answer. For example, "`Oceania is the continent where Palau is situated. True or False? Palau is located in Micronesia. Oceania is the continent where Micronesia is situated. So Oceania is the continent where Palau is situated. True.`" We manually create some negative queries to balance the training and testing label.

To empirically verify whether pre-training or instruction tuning is more important to the emergence of reasoning ability, we use the hardest version (S3) of the Countries knowledge graph (Bouchard et al., 2015). The COUNTRIES dataset contains 2 relations (`locatedIn` and `neighborOf`) and 227 entities, including countries, regions, and subregions. It is carefully designed to test logical rule learning and reasoning capabilities. The test set contains unseen queries of the form (`Egypt`, `locatedIn`, ?), and the answer is the corresponding region `Africa`. The location information of the neighboring countries of the testing countries and the location information of neighbors of the neighboring countries are also removed from the knowledge graph.

To test the reasoning ability of the pre-trained language model, we format the test triples as sentence completion tasks. For example, the triple (e_1, `locatedIn`, e_2) will be translated to the prompt "e_1 `is located in`". If the completion is a subregion of the ground truth answer (e.g. `South Africa` to `Africa`), we will also count it as a correct prediction.

To test the reasoning ability of the instruction-tuned language model, we turn the test triples into binary queries of the form "`Oceania is the continent where Palau is situated. True or False?`". For each triple, we create two positive queries and two negative queries.

We manually create a small set of logical rules for reasoning the `locatedIn` relation as follows:

```
[locatedIn, locatedIn]

[neighborOf, locatedIn]

[neighborOf, locatedIn, locatedIn]

[neighborOf, neighborOf, locatedIn]

[neighborOf, neighborOf, locatedIn,
locatedIn]
```

In Table 1, we show the dev and test accuracy of a GPT-style language model trained with different pre-training and fine-tuning strategies. The model we use has the same architecture as the 124M version GPT2 (Radford et al., 2019), and adopts the same tokenizer. Results show that direct fine-tuning without any pre-training performs significantly worse than pre-training with random walk data without any instruction-tuning. The pre-trained models also perform significantly better after being fine-tuned with instruction data. This shows that the reasoning ability of language models comes from unsupervised pre-training instead of fine-tuning.

In terms of the pre-training strategy, we can see that random walk performs significantly better than random sampling, both before and after fine-tuning. This suggests that how the pre-training sequences are constructed is crucial and would affect the final reasoning ability of the model. It can also be observed that the models perform much better when we translate the entities into new tokens instead of their real names. This is likely because the information of each entity is well separated and concentrated into one new token instead of diffused into several different tokens.

4. Reasoning mechanism

We also investigate how language models reason by looking deeper into the predicted distributions. As shown in Table 1, the sampling of pre-training sequences is very important for reasoning. So we hypothesize that language models' appeared reasoning ability is aggregating the seen reasoning paths in the training set. For example, when the model tries to complete the query "e_1 `is located in`", it recalls the most possible reasoning paths starting from e_1 and ending with `locatedIn`. It aggregates the probability of these reasoning paths leading to different answers as the predicted probability of each answer.

We assume that the model only selects paths that follow proper logic rules. We define the probability of following a specific logic rule $h \in \mathcal{H}$ between e_1 and e_2 to be the sum of the probability of all possible reasoning paths from e_1 to e_2 following the rule h:

$$P(e_2|e_1,h) = \begin{cases} \mathbb{1}_{e_1=e_2} & \text{if } h=\emptyset \\ \sum_{e' \in C_{r_1}(e_1)} P(e'|e_1,r_1)P(e_2|e',h') & \text{if } h \neq \emptyset \end{cases}$$

where $\mathbb{1}$ is the inidcator function, $h = [r_1, r_2, ..., r_l]$ and $h' = [r_2, r_3, ..., r_l]$. For a uniform random walk, $P(e'|e_1,r_1) = 1/|C(e_1)|$. The simplest way of aggregating the probabilities of following different rules is by summing them together:

$$P_s(e_2|e_1,r) = \sum_{h \in \mathcal{H}[r]} P(e_2|e_1,h) \qquad (1)$$

We also consider assigning a different weight to each rule and aggregating the probabilities of different rules by a weighted sum:

$$P_w(e_2|e_1,r) = \sum_{h \in \mathcal{H}[r]} w(h)P(e_2|e_1,h) \qquad (2)$$

To learn the weight for each rule, we adopt a similar method as the Path Ranking algorithm (Lao et al., 2011). We view the probability of following each rule $P(e_2|e_1,h)$ as a feature and then sample some positive and negative triples from the graph to train a logistic regression model for a target relation $r \in \mathcal{R}$. Here the positive triple (e_1, r, e_2) mean $r(e_1, e_2)$ holds while for the negative triple $r(e_1, e_2)$ does not hold. The loss function of the logistic regression is:

$$\mathcal{L}_r(w) = -\sum_i \left[y_i \ln p_i + (1-y_i)\ln(1-p_i)\right] + \lambda|w|$$

where $p_i = \frac{e^{P_w(e_2|e_1,r)}}{1+e^{P_w(e_2|e_1,r)}}$ and $y_i = \mathbb{1}_{r(e_1,e_2)}$. $\lambda|w|$ is a regularization term, and we can take any appropriate norm on w.

Note that we can obtain a similar triple probability by prompting the language model with the natural language translation of e_1 and r. We denote this probability by $P_\theta(e_2|e_1,r)$. To compare P_s, P_w, and P_θ, we first normalize them over all the entities $\mathcal{E}$ and then compute the KL divergence between them for each testing triple.

In Table 2, we can see the KL divergence between the aggregated path distribution and the language model predicted distribution is significantly lower when pre-trained with random walk data. This verifies our hypothesis that the language model is aggregating the reasoning paths probabilities seen in the pre-training. Since the pre-training sequence distribution of the random walk data matches the assumed

Entity translation	Pre-training	Instruction-tuning	Dev	Test
Meaningful name	Random sample	✗	67.2	58.7
	Random walk	✗	100	95.8
	✗	✓	93.8	79.2
	Random sample	✓	97.9	90.6
	Random walk	✓	100	95.6
New tokens	Random sample	✗	95.8	83.3
	Random walk	✗	95.8	95.8
	✗	✓	87.5	83.3

Table 1. Accuracy on Countries (S3) knowledge graph.

Entity translation	Pre-training	Instruction-tuning	KL$[P_s, P_\theta]$	KL$[P_w, P_\theta]$
New tokens	Random sample	✗	3.59 $\pm$ 1.60	2.10 $\pm$ 1.77
	Random walk	✗	0.52 $\pm$ 0.32	0.29 $\pm$ 0.17

Table 2. KL divergence between language model prediction (P_θ) and aggregated paths (P_s, P_w), averaged over the Countries (S3) test set. The standard deviations are also reported.

random walk distribution in path aggregation, the random walk pre-trained model produces a similar distribution as the aggregated distributions. We can also observe that the weighted aggregation matches the language model distribution better than the unweighted aggregation. This shows that the language model is able to focus on the more evidential rules instead of the unimportant ones.

5. Conclusion

In conclusion, this study delved into the emergence of reasoning abilities in large language models (LLMs), with a particular focus on reasoning over knowledge graphs. The findings shed light on the origins of LLMs' remarkable reasoning capabilities, showcasing the importance of pre-training in acquiring these skills. The construction of the pre-training sequence, such as organizing it as ”chains” or random walks on the graph, was found to significantly impact the effectiveness of reasoning. Importantly, the study revealed that LLMs reason over known facts by aggregating relevant reasoning paths.

References

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Bouchard, G., Singh, S., and Trouillon, T. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Symposia*, 2015.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Dai, D., Sun, Y., Dong, L., Hao, Y., Sui, Z., and Wei, F. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

Hahn, M. and Goyal, N. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

Jiang, H. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

Lao, N., Mitchell, T., and Cohen, W. W. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 529–539, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL `https://aclanthology.org/D11-1049`.

Li, Y., Ildiz, M. E., Papailiopoulos, D., and Oymak, S. Transformers as algorithms: Generalization and implicit model selection in in-context learning. *arXiv preprint arXiv:2301.07067*, 2023.

Prystawski, B. and Goodman, N. D. Why think step-by-step? reasoning emerges from the locality of experience. *arXiv preprint arXiv:2304.03843*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

Wang, X., Zhu, W., and Wang, W. Y. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, 2023.

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