REASONING LIKE ARISTOTLE: DIAGNOSING AND IMPROVING LANGUAGE MODELS FOR LOGICAL REASONING THROUGH SYLLOGISMS

Xiaoyin Chen, Zach Furie
Duke University
{xiaoyin.chen,zachary.furie}@duke.edu

ABSTRACT

In this paper, we study the reasoning capabilities of language models (LMs) with syllogisms, a basic logical deductive system. By composing syllogisms, we present a pipeline for automatically generating logical questions without any human annotation. Compare to previous benchmarks on logical reasoning, our work introduces a new set of deduction rules by employing syllogism. Despite its limited set of deduction rules, our results show that even the largest GPT-3 is inadequate in answering simple logical questions by following syllogistic rules. Furthermore, by fine-tuning a pretrained language model, we demonstrate empirically that LMs could learn the rules of syllogism and apply them to out-of-distribution samples. Last, by pre-training an LM on our synthetic data, we improve its accuracy by 3% on a dataset of human-written logical questions, compared to a baseline model finetuned only on the downstream dataset. In a zero-shot setting, our pretrained model improves by 15% over a random baseline. Our results suggest a new possibility of unsupervised learning to reason. Code is available at https://github.com/chenyn66/fol_pretrain/

1 INTRODUCTION

Language models (LMs), e.g., BERT and GPT (Devlin et al., 2019; Brown et al., 2020), have achieved tremendous success in many NLP tasks, such as text classification, semantic parsing, and reading comprehension. However, existing pretrained LMs focus on the language modeling objectives with little attention to reasoning capabilities. As a result, in tasks that involve complex and formal reasoning, LMs still fall far behind expectations.

One of such tasks is logical reasoning, the process of applying rigorous logic to draw conclusions from given information. Many studies have provided benchmarks for logical reasoning (Liu et al., 2020; Yu et al., 2020; Clark et al., 2020; Tian et al., 2021). However, some of them do not evaluate logical reasoning independently from other types of reasoning problems Liu et al. (2020); Yu et al. (2020). On the other hand, benchmarks specifically designed for logical reasoning (Clark et al., 2020; Tian et al., 2021) are usually constructed with pre-defined logical templates in first-order logic (FOL), limiting the diversity in the syntax of underlying logical deduction structure. Additionally, those templates are arbitrarily defined without a formal guideline on how to construct meaningful logical questions.

Therefore, we propose a framework for synthesizing logical questions based on syllogism, a classic deductive system. Each syllogism involves two propositions and one conclusion. Although its expressive power is limited compared to FOL, syllogism introduces new deduction rules to previous logical reasoning benchmarks (Sec. 3.2). Additionally, as the set of all logically distinct syllogisms is finite and enumerable, this system provides a formal framework for constructing non-trivial logical questions. With syllogism, we are able to clearly define the problem space of synthetic questions.

Additionally, high-quality logical reasoning datasets (Yu et al., 2020; Liu et al., 2020; Han et al., 2022) are difficult and expensive to acquire. We explore the possibility of learning logical deduction with unsupervised learning. Specifically, we leverage the compositionality of symbolic logical systems, to generate a large number of training problems with arbitrary complexity. With these

synthetic questions, LMs could learn to reason during pre-training and transfer such ability to the downstream application.

We summarize our key findings as follows:

1. **Large Language Models are inadequate in rigorous reasoning:** GPT-3 only correctly answers 46 out of 48 questions in our simplest test case, where all rules are presented in the prompt and test premises are identical to those in the prompt. (Sec. 4.1)

2. **LMs can learn to reason with the rules of syllogism:** With fine-tuning, an LM can achieve near-perfect performance on both in-distribution and out-of-distribution samples. We further show that such ability is robust to different natural language representations. (Sec. 4.2)

3. **Pre-training can improve downstream logical reasoning performance:** We show that per-training with our synthetic questions improves the downstream accuracy by 3% in a finetuned setting and 15% in a zero-shot setting. (Sec. 4.3)

2 Related Work

2.1 Logical Reasoning Benchmark

Recently, there is an increasing research interest in diagnosing logical reasoning ability. ReClor (Yu et al., 2020) and LogiQA (Liu et al., 2020) both used multi-choice questions from graduate admission exams. These questions involve realistic logical reasoning. However, these datasets do not disentangle logical reasoning from other types of reasoning problems, e.g., commonsense reasoning, causing them unsuitable for studying logical reasoning independently. FOLIO (Han et al., 2022) is specifically built with FOL questions written by human experts. However, this dataset is limited in size with only 1435 samples.

Meanwhile, other benchmarks have been proposed by automatically generating logical questions Clark et al. (2020); Tian et al. (2021); Saeed et al. (2021). For example, Clark et al. (2020) focused on a specific set of FOL expressions, namely, conjunctive implication with negation. Tian et al. (2021) used pre-defined templates as base logic expressions to compose natural language inference problems involving logic. Some other studies focused on investigating the question that if LMs are actually reasoning. Zhang et al. (2022) showed that models may take shortcuts by exploiting statistical patterns. Gubelmann & Handschuh (2022) demonstrated that LMs may rely on shallow heuristics to perform logical reasoning. Although this study also used syllogism, it is limited to a small subset as its main purpose differs from ours. To the best of our knowledge, our work is the first extensive study of LMs' reasoning performance on syllogism.

2.2 Reasoning via Pre-training

Recent work has attempted to improve the reasoning ability of LMs by constructing input context with richer dependencies. Yasunaga et al. (2022) leveraged hyperlinks and references from Wikipedia to improve the model's multi-hop reasoning ability. Deng et al. (2021) constructed a self-supervised multi-hop reasoning task with distant supervision. Both methods focus on improving multi-hop reasoning performance, the ability to aggregate information from multiple pieces of evidence. Instead, we focus on logical reasoning skills in our study. Pi et al. (2022) shared a similar idea with us by constructing training data via abstract logical programs. However, their input context is in abstract symbols, e.g., $p \rightarrow q$, which is unrealistic. Instead, we translate our questions into natural language. Jiao et al. (2022); Ouyang et al. (2022) formed input context by constructing semantic graphs from documents. This approach essentially differs from ours, which is based on syllogism. Last, Betz et al. (2021) used syllogistic arguments to improve the pre-training of autoregressive LM and tested on downstream text generation tasks. Instead, our study focuses on the performance of logical reasoning tasks.

3 APPROACH

3.1 TASK DEFINITION

To evaluate the logical reasoning ability of LMs, we developed a pipeline to automatically generate logical questions by composing syllogisms. Each problem is a triple *(context, conclusion, answer)*, where both *context* are premises, *conclusion* is the logical statement to verify, and *answer* is either true if the *conclusion* is entailed by the *context* or false if the *conclusion* is a contradiction. *Context* and *conclusion* are expressed in natural language by verbalizing with templates. As we only focus on logical deduction, we construct our data such that the conclusion can be determined with the information presented in the context only. Doing so disentangles logical reasoning from other types of reasoning problems, e.g. commonsense reasoning, which involves world knowledge.

3.2 SYLLOGISM

Syllogism is a common deductive reasoning system first defined by Aristotle. A syllogism contains two premises, one conclusion, and three terms whose relations are expressed by the premises. Each premise and conclusion states one of four possible relations between two terms: *All A are B (A), No A is B (E), Some A are B (I), Some A are not B (O)*[1]. There are 256 logically distant syllogisms and 24 of them are valid (Lehman, 1973). Our methods use all 24 valid syllogisms as the basis for constructing questions. In Table 1, we show two examples of valid syllogisms:

All A are B	Some A are not B
All C are A	All A are C
All C are B	Some C are not B

Table 1: Two valid syllogisms. The first two rows are the two premises and the last row is the conclusion entailed by the premises.

Multiple syllogisms can be composed into a more complex reasoning problem by using the conclusion of the previous syllogism as one premise for the later syllogism. For example, the two syllogisms in Table 1 can be composed into: *All A are B, All C are A, Some C are D*, therefore *Some B are not D*. We say a problem has depth *d* if it is composed of *d* syllogisms.

Note that syllogism is a proper subset of first-order logic, as all four possible premises can be expressed in FOL form: $\forall x(A(x) \rightarrow B(x))$, $\forall x(A(x) \rightarrow \neg B(x))$, $\exists x(A(x) \wedge B(x))$, $\exists x(A(x) \wedge \neg B(x))$. Therefore, our work can complement previous studies on LM's ability of FOL reasoning by introducing a new set of distinct deduction rules. To the best of our knowledge, all previous benchmarks do not contain syllogism.

3.3 DATA GENERATION

In this section, we describe our pipeline of translating syllogism symbols into natural language. Our pipeline is completely automatic by employing hand-crafted templates. Two steps are involved to verbalize a syllogism 1) connective translation and 2) predicate translation. First, to convert logical relations, we write templates with placeholders for the predicates. Then, we sample from a large vocabulary list to substitute the placeholders. For negative samples, we simply negate the conclusions, as negations of syllogism relations are also syllogism relations[2]. To evaluate the effect of semantics on reasoning performance, we designed two different types of templates for translation. We refer to them as **NOUN** template and **ADJ** template accordingly. In Table 2 we show two different translations of the same underlying logical question.

NOUN template follows the classic interpretation of syllogism, where each term is an entity. Table 1 show one example of this template. We write 33 templates in total with at least 8 for each relation. For example, an alternative interpretation for *All A are B* could be *A is always B*. To substitute the entities, we sample from a preconstructed vocabulary of size 1000 containing nouns and noun phrases. We ignore the semantics of terms. One example of generated questions is:

[1] Letters in parentheses are shorthands for each relation.

[2] For example, the negation *All A are B* is *Some A are not B*.

NOUN	ADJ
A subset of fire is owner. All fire is carton of milk. If something is carton of milk, then it is not reputation.	Thrilling and intellectual person exists. Everyone who is thrilling is soulless. For all people, if he is soulless then he is not glamorous.
Sometimes owner is not reputation.	Someone is intellectual and not glamorous.

Table 2: An example question of depth 2 generated by our pipeline, verbalized by two different templates

Instruction Prompt	Accuracy
Baseline, No Instructions	0.954
Answer questions about syllogisms.	0.961
Answer questions about syllogisms, ignoring semantics.	0.979
Answer questions about syllogisms in first-order logic form.	46/48

Table 3: GPT-3 ICL performance on syllogistic reasoning. Both training and test problems have depth 1. The first three rows are performances on problems in natural language. The last row shows the performance on FOL representations.

Premises: Gene is not line. There is no choice that is not gene. Conclusion: Not all choice is line.

ADJ template provides an alternative verbalization of syllogisms by treating each predicate as a description of a property. In order to guarantee the consistency of the resulting questions, we limit the scope of quantifiers to people. With this template, *All A are B* can be interpreted as *Every A person is B*. We write 66 templates in total with at least 13 for each relation. We construct a vocabulary of size 973 by sampling from adjectives for describing people and moods. Similarly, terms are substituted while disregarding their meanings. One example of generated questions is:

Premises: It is impossible for a celestial person to be apprehensive. There is someone who is celestial and inconsiderate. Conclusion: Some people are inconsiderate and not apprehensive.

4 Experiments

In this section, we discuss three sets of experiments we conducted, one for each conclusion we presented in Section 1.

4.1 Can GPT-3 robustly reason about syllogisms with in-context learning?

As GPT-3 (Brown et al., 2020) and other large language models (LLMs) have demonstrated superior performance on many applications with in-context learning (ICL), we are interested in investigating if they can solve logical problems composed by syllogism.

Experimental Details For all GPT-3 experiments, we use the *text-davinci-003*, the most capable and largest model with 175B parameters in the GPT-3 family. We construct the prompts with a simple format that each few-shot example consists of two line 1) "Story: $< context >$ Conclusion: $< conclusion >$", 2) "Answer: $< True/False >$". We balance the number of positive and negative samples in both train and test data. Unless otherwise specified, we use the ADJ template for problem construction. We also perform modest prompt engineering as shown in Table 3.

Results First, we test if GPT-3 can learn syllogisms and apply them to test samples. We use 24 training problems of depth one (one of each type of syllogism) and 1000 problems of the same depth. To disentangle the effect variation of templates, we use a fixed template for each of the four relations both in training and test samples. Under this setting, the only variance in the problems is the sampled terms.

As shown in Table 3, with the best prompt, GPT-3 achieves a 97.9% accuracy on the test set. This is surprising given that all possible deduction rules are given in the prompt and the training and test

samples are almost identical except for the difference in substituted terms. Presumably, a human can easily achieve 100% accuracy by simply following the rules in the samples.

Interestingly, we note that GPT's performance improves as we explicitly prompt the model to ignore the semantics. This is probably because the semantics of the conclusion may affect the model prediction. As our data is purely synthetic, many samples may contradict commonsense, which may cause to model to derive a wrong conclusion. Dasgupta et al. (2022) presented a more detailed study of this content effect. To further disentangle the content effect, we leverage the symbolic representations of FOL. We generate questions in purely symbolic form with all predicates denoted by *A, B, C* in a fixed order. With this construction, there are only 48 possible problems at depth 1[3]. Again, we use 24 training problems to guarantee that all rules are presented in the prompt. We test the model on all 48 problems. We observe that GPT-3 still fails to achieve perfect accuracy, indicating that the model is unable to reason formally and semantics is not the only reason for incorrect reasoning over natural language.

Compositional Generalization As the above experiments are trained and tested on the same depth, we extend our study to investigate GPT's compositional generalization performance. For this experiment, we prompt the model with 34 examples, 24 base rules plus 10 examples at depth d. Then we test the model at depth $2-8$. We use ADJ template with randomly sampled terms. It is worth noting that due to the budget, we use 100 test examples for each depth, potentially causing these results to have high variance. Nevertheless, as shown in Figure 1, model performance deteriorates as the problem depth increases, indicating that GPT is unable to compose rules in the prompt to perform multiple reasoning steps.

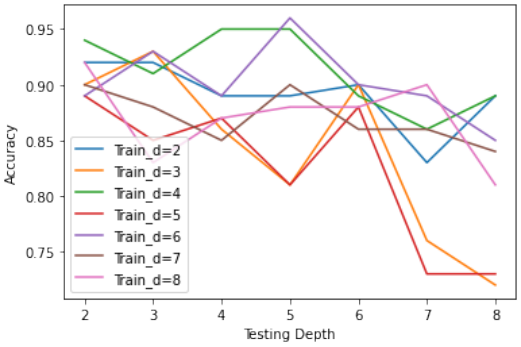

Figure 1: Compositional generalization performance of GPT-3.

4.2 Can Pretrained Language Models Learn Syllogisms through Fine-tuning?

Next, we test pretrained language models' (PLMs) capabilities on logical reasoning through fine-turning.

Experimental Details For experiments, we use pretrained RoBERTa-Large (Liu et al., 2019) from Hugging Face (Wolf et al., 2020). Questions are fed to the model as: *[CLS] context [SEP] conclusion [SEP]*. For extra complexity, we shuffle the order of premises in the context. We take the embedding of the *[CLS]* token from the last layer and project it to a scalar through a linear layer, following a standard binary classification approach. We balance the number of positive and negative samples. Similar to the previous compositional generalization setting, we finetune RoBERTa for 10 epochs on 5000 examples of depth d, with a learning rate of 1e-5 and a linear warm-up schedule for 1 epoch, followed by linear decay. We test on problems of depth 1-8, with 5000 data for each depth, and take the average over five runs. We use ADJ template by default.

Results Figure 2 shows the results of finetuned RoBERTa. First, we observe that the model reaches nearly perfect accuracies on test problems with a similar depth as the training problems. However, we see that the performance starts to drop as the test complexity start to differ from that of training. Especially, this deterioration happens in both directions, from easier problems to more complex ones and vice versa. This result implies that the model still fails to generalize compositionally.

One possible fix is to train the model on multiple depths so it can observe various composition forms in the training data. Accordingly, we finetune another RoBERTa on problems of depth 1-6 and test it on depth 7-16. With this setting, the model success to generalize to test data with greater depths, as shown in Table 4.

[3]There are 24 valid syllogisms, times 2 for true or false.

	Test_d =1	Test_d =2	Test_d =3	Test_d =4	Test_d =5	Test_d =6	Test_d =7	Test_d =8
Train_d = 1	**1.0000**	0.9998	0.9832	0.9032	0.7883	0.6724	0.6030	0.5729
Train_d = 2	0.9972	**0.9997**	0.9986	0.9874	0.9488	0.8827	0.7982	0.7351
Train_d = 3	0.9074	0.9869	**0.9979**	0.9994	0.9992	0.9976	0.9964	0.9952
Train_d = 4	0.8091	0.9545	0.9878	**0.9963**	0.9986	0.9995	0.9979	0.9772
Train_d = 5	0.7831	0.9379	0.9824	0.9948	**0.9982**	0.9991	0.9988	0.9966
Train_d = 6	0.7707	0.9292	0.9748	0.9923	0.9964	**0.9979**	0.9991	0.9996
Train_d = 7	0.7726	0.9268	0.9769	0.9918	0.9961	0.9985	**0.9993**	0.9997
Train_d = 8	0.7480	0.9088	0.9615	0.9818	0.9932	0.9961	0.9985	**0.9990**

Figure 2: Results of finetuned RoBERTa-Large, both training and test questions are in ADJ template. Cells are color-coded by their relative performance in the table.

Test Depth	7	8	9	10	11	12	13	14	15	16
Accuracy	1	1	1	1	0.9996	0.9958	0.9952	0.9944	0.9968	0.9984

Table 4: Finetuned RoBERTa-Large on depth 1-6, test on depth 7-16.

Generalization to Different Domains Additionally, We test the model's robustness to different types of templates. Here, we want to examine if the model can parse the semantics and map them to the same underlying logical relations. As such, we finetune RoBERTa with NOUN template and test it with ADJ template. Results in Figure 4 show that such transformation hurts the performance, however, the models still perform significantly over a random baseline. As both the connectives and terms are interpreted differently by the two templates, our results suggest that LMs are able to transfer the logical rules learned in one domain to another.

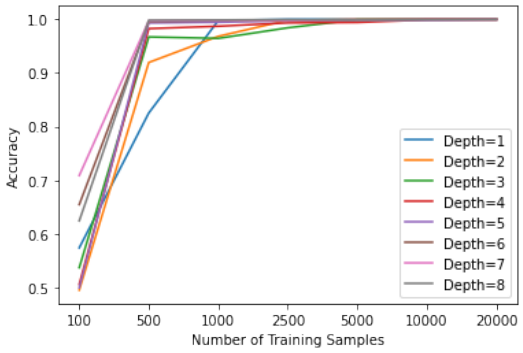

Figure 3: In-distribution test accuracy vs. number of training samples.

Data Efficiency Another concern is how much data the models need to learn the logic. Figure 3 shows that RoBERTa needs around 500-1000 to master the test data. At the first glance, it seems odd that problems with less depths required more training samples than more complex problems. Our explanation is a more complex problem presents more rules to the model than a simpler problem, effectively allowing the model to learn more rules per sample. Notably, RoBERTa requires 1000 samples to robustly reason about depth one problem, a disproportional amount compared to 24 underlying rules.

4.3 Can pre-training on syllogisms help downstream logical reasoning performance?

Ultimately, our goal is to transfer the logical reasoning ability learned on synthetic data to real-life problems. In this section, we test the effect of pre-training with logical problems on the downstream logical question answering (QA) task. We use FOLIO (Han et al., 2022) a logical QA dataset written by human experts based on FOL deduction. The task can be formatted in the same way as our synthetic data, with one exception that its prediction has three target labels: *True, False, Unknown*. Models need to predict *Unknown* when the conclusion is neural to the premises. This dataset contains 1004 training samples and 204 test samples.

	Test_d =1	Test_d =2	Test_d =3	Test_d =4	Test_d =5	Test_d =6	Test_d =7	Test_d =8
Train_d = 1	**0.9311**	0.9228	0.8678	0.7518	0.7272	0.6936	0.6819	0.6552
Train_d = 2	0.9195	**0.9391**	0.9324	0.9089	0.8920	0.8512	0.8472	0.8035
Train_d = 3	0.8478	0.9421	**0.9470**	0.9479	0.9419	0.9251	0.9146	0.8714
Train_d = 4	0.7725	0.9360	0.9574	**0.9633**	0.9602	0.9599	0.9443	0.9151
Train_d = 5	0.6938	0.9128	0.9528	0.9677	**0.9686**	0.9685	0.9544	0.9351
Train_d = 6	0.6265	0.8694	0.9106	0.9685	0.9681	**0.9709**	0.9570	0.9393
Train_d = 7	0.5702	0.7990	0.8313	0.9463	0.9637	0.9736	**0.9597**	0.9380
Train_d = 8	0.5321	0.7358	0.7332	0.8756	0.9429	0.9726	0.9586	**0.9345**

Figure 4: Results of finetuned RoBERTa-Large, training questions are in NOUN template and test questions are in ADJ template. Cells are color-coded by their relative performance in the table.

	Finetuned on FOLIO	Zero-shot
RoBERTa	0.68 ± 0.02	0.5
Pretrained with Symbols	0.68 ± 0.01	0.48 ± 0.02
Pretrained with Natural Language	0.71 ± 0.01	0.65 ± 0.01

Table 5: Finetuned and zero-shot performance on FOLIO of pre-training with syllogisms. Results are average over five runs. Zero-shot performance of RoBERTa is based on random guessing, see text for more details.

Experimental Details We generate questions up to depth 6[4] with both NOUN and ADJ templates. With these questions, We pretrain RoBERTa-Large with 42000 samples for two epochs. Then, we discard the prediction head of the pretrained model and finetune it with a new linear layer for prediction. We follow the same parameter settings in Han et al. (2022) and report the performance from the best checkpoint. For the baseline, we compare to RoBERTa finetuned only on FOLIO with the exact parameter settings. Additionally, we compare to the framework proposed by Pi et al. (2022), where pre-training is done with questions in symbolic form.

Results Table 5 shows the results of pre-training with syllogisms. After pre-training with our synthetic problems, The finetuned performance on FOLIO improves by 3%. This indicates that the model is able to transfer the logical rules learned during pre-training to downstream problems. Furthermore, pre-training with pure symbols does not improve performance, suggesting that the naturalness of synthetic questions does affect transferability.

Zero-shot Transfer Additionally, we test the transfer performance in the zero-shot setting. To adapt the pretrained model for FOLIO, we create a subset of only True or False questions, which contains 135 questions. As such, a random baseline has 50% accuracy. We test our pretrained model directly on this subset without any further finetuning. As shown in Table 5, our pretrained model improves by 15% from the baseline. Similarly to the finetuning results, pre-training with symbols can not improve the downstream reasoning performance.

5 Analysis

Here we discuss the highlights of our results.

5.1 Large Language Models are inadequate in rigorous reasoning

From Sec. 4.1, we conclude that GPT-3 is unable to robustly apply syllogism through in-context learning only. Especially, as syllogism is one of the most basic logical deductive systems and only covers a small subset of FOL, failing in this simple benchmark raises a concern about LLMs' ability

[4]We additionally test other depths and find 6 yields the best result.

to perform formal logical reasoning. Recently, many advanced prompting methods have been proposed for improving reasoning (Wei et al., 2022; Creswell et al., 2022; Nye et al., 2021). Inspired by these methods, we perform an additional experiment to prompt GPT to produce an intermediate representation in FOL symbols before generating the final output. We find that doing so reduces the accuracy from 97.9% to 83.4%. Unfortunately, we are unable to test other prompting methods due to budget. We expect the model performance on our dataset could be improved by employing these advanced prompts. However, as these methods focus on multi-step reasoning, we do not expect them can address GPT's inconsistency on depth-one problems.

5.2 Finetuned language models can Learn Syllogisms and generalize to out-of-distribution data

In Sec. 4.2, we show that a finetuned RoBERTa is able to consistently apply syllogistic rules and generalize to more complex questions. The model can also generalize to a new domain, as shown by our synthetic domain transfer task. This indicates that PLMs are able to incorporate their semantics knowledge from language modeling with the logical reasoning abilities learned from fine-tuning.

5.3 Pre-training with synthetic data can improve downstream logical reasoning performance

In Sec. 4.3, a RoBERTa model pretrained on our synthetic questions improves its accuracy by 3% on FOLIO dataset. As FOLIO is a human-written dataset, our results show that reasoning ability from synthetic data could potentially transfer to real-life applications. Importantly, as logical reasoning dataset are constructed with expensive human annotations, our pipeline opens a possibility for unsupervised learning for logical reasoning. Moreover, as shown by our ablation study that pre-training with questions in a more natural form achieves better transferability, we expect the downstream performance could be further improved by employing a pipeline that can automatically produce more realistic questions.

One limitation of our framework is that templates have limited variety in natural language. For example, when negating a term, our templates can only append *not* to the prefix, however, in natural language, negations are often expressed by antonyms, e.g., from *happy* to *sad* instead of *not happy*. To overcome this limitation, we test with using LLMs for verbalization and using back translation to augment the data. However, none of these attempts yield a satisfactory result, we leave the question of automatically generating natural questions to future work.

6 Conclusion

We introduce a novel pipeline that composes syllogism into logical questions and translates them into natural language. With this pipeline, we use synthetic logical questions to test the logical reasoning ability of large language models with in-context learning and pretrained language models with fine-tuning. Our results indicate that one of the most capable LLMs, GPT-3, failed to robustly apply syllogisms and to generalize compositionally, even when all rules are given. We further show that a finetuned model is able to learn reasoning rules and generalize them to new domains. Last, we show that pre-training with our synthetic data can improve model performance in answering human-written logical questions.

Our work suggests the possibility of improving logical reasoning through un/self-supervised learning with synthetic data. As syllogism has limited expressive power, generalizing this framework to FOL is a natural extension of our work. To this end, this paper demonstrates the potential of leveraging symbolic systems as scaffolds for learning to reason.

Author Contributions

Both team members ran trials on GPT-3, wrote this paper, and worked on the code to generate the datasets (code prototype by Zach, final code by Xiaoyin). Xiaoyin also ran the trials on RoBERTa.

REFERENCES

Gregor Betz, Christian Voigt, and Kyle Richardson. Critical thinking for language models. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pp. 63–75, Groningen, The Netherlands (online), June 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.iwcs-1.7.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as Soft Reasoners over Language, May 2020. URL http://arxiv.org/abs/2002.05867. arXiv:2002.05867 [cs].

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning, May 2022. URL http://arxiv.org/abs/2205.09712. arXiv:2205.09712 [cs].

Ishita Dasgupta, Andrew K Lampinen, Stephanie CY Chan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*, 2022.

Xiang Deng, Yu Su, Alyssa Lees, You Wu, Cong Yu, and Huan Sun. ReasonBERT: Pre-trained to Reason with Distant Supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6112–6127, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.494. URL https://aclanthology.org/2021.emnlp-main.494.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv:1810.04805 [cs].

Reto Gubelmann and Siegfried Handschuh. Uncovering more shallow heuristics: Probing the natural language inference capacities of transformer-based pre-trained language models using syllogistic patterns. *arXiv preprint arXiv:2201.07614*, 2022.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir Radev. FOLIO: Natural Language Reasoning with First-Order Logic, September 2022. URL http://arxiv.org/abs/2209.00840. arXiv:2209.00840 [cs].

Fangkai Jiao, Yangyang Guo, Xuemeng Song, and Liqiang Nie. MERIt: Meta-Path Guided Contrastive Learning for Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3496–3509, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.276. URL https://aclanthology.org/2022.findings-acl.276.

Anne Lehman. Two sets of perfect syllogisms. *Notre Dame Journal of Formal Logic*, 14(3):425–429, 1973.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3622–3628, Yokohama, Japan, July 2020. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-6-5. doi: 10.24963/ijcai.2020/501. URL https://www.ijcai.org/proceedings/2020/501.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

Siru Ouyang, Zhuosheng Zhang, and hai zhao. Logic pre-training of language models, 2022. URL https://openreview.net/forum?id=1gEb_H1DEqZ.

Xinyu Pi, Qian Liu, Bei Chen, Morteza Ziyadi, Zeqi Lin, Yan Gao, Qiang Fu, Jian-Guang Lou, and Weizhu Chen. Reasoning Like Program Executors, January 2022. URL http://arxiv.org/abs/2201.11473. arXiv:2201.11473 [cs].

Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. RuleBERT: Teaching soft rules to pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1460–1476, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.110. URL https://aclanthology.org/2021.emnlp-main.110.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. Diagnosing the First-Order Logical Reasoning Ability Through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3738–3747, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.303. URL https://aclanthology.org/2021.emnlp-main.303.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain of Thought Prompting Elicits Reasoning in Large Language Models, October 2022. URL http://arxiv.org/abs/2201.11903. arXiv:2201.11903 [cs].

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining Language Models with Document Links, March 2022. URL http://arxiv.org/abs/2203.15827. arXiv:2203.15827 [cs].

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. RECLOR: A READING COMPREHENSION DATASET REQUIRING LOGICAL REASONING. pp. 26, 2020.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. On the Paradox of Learning to Reason from Data, May 2022. URL http://arxiv.org/abs/2205.11502. arXiv:2205.11502 [cs].