Neuro-symbolic Tuning for Multi-hop Reasoning over Spatial Language

Tanawan Premsri[1], Parisa Kordjamshidi[1]

[1] *Department of Computer Science and Engineering, Michigan State University, MI, USA*

Abstract

Spatial Reasoning is a fundamental aspect of human cognition to perform everyday activities. It is also an essential skill for machines to engage in human-like interactions with the environment. However, recent research shows that even state-of-the-art language models struggle in spatial reasoning, especially in unobserved situations with complex input compositions. This is attributed to not achieving the right level of abstraction required for their generalizability. To alleviate this issue, we propose training the language models with neuro-symbolic techniques to exploit the spatial logical rules of reasoning and provide an additional source of supervision to the models. Training models to adhere to spatial reasoning rules guides them to make more effective abstractions for generalizability and transfer learning. We evaluate our proposed technique on various benchmarks for spatial reasoning over text. Our results based on the multiple language model backbones show the effectiveness of our neuro-symbolic training in domain transfer and complex multi-hop spatial reasoning.

Keywords

Spatial Reasoning, Neuro-symbolic training

1. Introduction

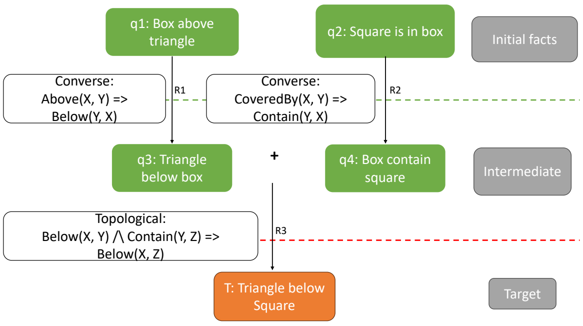
Spatial reasoning is essential for humans cognition and also plays a crucial role in many AI applications, including language grounding [1], computer vision [2, 3], robotics [4, 5, 6] and even more specific fields such as medical domain [7, 8, 9].

Large Language models have been widely applicable in many of problems in these areas and, in some cases, show human level performance [10, 11]. However, recent studies highlight their shortcomings in the spatial reasoning abilities of in multi-hop reasoning over text [12, 13, 14] in many downstream applications [3, 15] which calls for more attention to this topic.

In this paper, we address the issue of spatial reasoning in LMs and their difficulty in obtaining the abstractions required for generalizability in unobserved complex situations employing a generic neuro-symbolic framework. We propose to fine-tune the LMs with a neuro-symbolic technique that exploits the spatial logical rules to guide the level of abstraction captured during training. In particular, we train the models to minimize both the cross-entropy loss and the violation from logical constraints. We demonstrate the effectiveness of our proposed framework in both encoder-based and generative language models. For evaluation, we use three Spatial Question Answering (SQA) benchmarks, SpartQA-HUMAN [16], ReSQ [17], and StepGame [18].

The results show that our proposed method benefits both LM types, especially when multiple hops of reasoning are required. The performance improvements over multiple domains confirm our hypothesis about the effectiveness of neuro-symbolic training on generalizability.

Figure 1: An example of the chain of reasoning questions, Q-*chain*. The factual sentences will turn: The initial and intermediate facts will turn to questions like "Is triangle below square?"



Rules	Constraints in YN
R_1	$AnsYN(q_1) \Rightarrow AnsYN(q_3)$
R_2	$AnsYN(q_2) \Rightarrow AnsYN(q_4)$
R_3	$AnsYN(q_3) \wedge AnsYN(q_4) \Rightarrow AnsYN(t)$
Rules	Constraints in FR
R_1	$AnsFR(q_1, above) \Rightarrow AnsFR(q_3, below)$
R_2	$AnsFR(q_2, coveredby) \Rightarrow AnsFR(q_4, contain)$
R_3	$AnsFR(q_3, below) \wedge AnsFR(q_4, contain) \Rightarrow AnsFR(t, below)$

Table 1
Logical constraints generated based on the Q-*chain* of the example in Figure 1. R_i refer to rule i used in the example. Define AnsYN and AnsFR

2. Training with Spatial Logic

The spatial logical rules used in our framework are based on the developed spatial logical knowledge base in [17]. Examples of such rules are given in the Figure 1. To clarify, Converse rules, $Above(X, Y) \Rightarrow Below(Y, X)$, represents that if an object X is above object Y, therefore, object Y is below object X. The rest of the rules use a similar notation.

To apply training with Spatial Logic, we follow three steps. Firstly, we create example-specific rules based on the given *Spatial Logic*. This process is explained in the example of Figure 1.

We use the resolution tree, which provides the logical implication steps, to infer the answer to the final query from the input context. Note that our synthetic training data (e.g., SpaRTUN) provides the logical representation of the context. We start creating the tree using initial facts in the given context and a forward chaining approach to find the applicable rules. In this way, we obtain the intermediate inferred facts. We denote fact i as q_i and the sequence of all derived intermediate facts, including the target question, as

STRL'24: Third International Workshop on Spatio-Temporal Reasoning and Learning, 5 August 2024, Jeju, South Korea
premsrit@msu.edu (T. Premsri); kordjamshidi@msu.edu (P. Kordjamshidi)

Table 2
Accuracy of SPARTQA-Human and ReSQ with various models. For ReSQ, k is the number of the reasoning steps required to answer the questions. *Unclassified* indicates the cases in which k was a challenge for human annotators to decide.

	SPARTQA-Human		ReSQ				
Model	Ver.1	Ver.2	k=1	k=2	unclassified	All	Line
BERT	54.54	53.57	70.67	56.85	60.66	60.98	1
BERT-T	55.94	58.03	**76.00**	54.79	**61.18**	61.15	2
BERT-T+*Q-Chain* (Our)	**59.44**	**58.92**	72.00	**58.90**	59.90	**61.31**	3
Flan-T5	54.54	60.71	74.67	56.16	61.44	61.80	4
Flan-T5-T	49.65	57.14	81.33	54.79	61.44	62.30	5
Flan-T5-T+*Q-Chain* (Our)	**55.94**	**61.61**	**81.33**	**57.53**	**63.75**	**64.43**	6
GPT3.5 (zero-shot)	58.04	58.03	74.67	60.95	66.58	66.22	7
GPT3.5 (few-shot)	62.23	58.92	84.00	68.49	68.12	70.16	8
GPT3.5 (CoT)	65.73	**71.43**	**86.67**	67.12	68.64	70.49	9
GPT-4 (zero-shot)	**77.62**	68.75	84.00	73.97	**76.86**	**77.05**	10
Llama-3 (zero-shot)	61.54	50.89	80.00	64.38	67.35	68.20	11
Llama-3 (few-shot)	62.94	60.71	82.67	69.86	71.46	72.46	12
Llama-3 (CoT)	67.83	70.54	82.76	**76.03**	67.10	71.15	13

Table 3
Accuracy of StepGame on several models including results of GPT3 reported in [13].

Model	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
BERT	98.51	95.53	91.68	66.71	49.11	41.47	41.47	32.09	28.94	28.16
BERT-T	98.50	95.32	**93.26**	**76.78**	**66.36**	58.76	53.70	46.27	42.71	40.12
BERT-T+*Q-chain* (Our)	**98.70**	**96.45**	93.03	74.58	64.95	**59.04**	**54.38**	**49.23**	**45.36**	**44.05**
GPT3 (few-shot)	55.00	37.00	25.00	30.00	32.00	29.00	21.00	22.00	34.00	31.00
GPT3 (CoT)	61.00	45.00	30.00	35.00	35.00	27.00	22.00	24.00	23.00	25.00
Llama-3 (few-shot)	38.01	27.87	24.15	21.27	19.75	18.03	16.88	15.52	15.17	14.70

Q-Chain.

Secondly, we generate the consistency constraints between q_is given the *Q-Chain*. To explain the consistency constraints between questions, we denote the answer to the YN questions as $AnsYN(q_i)$, which will be True if the answer to q_i is True. We denote the answer to the FR questions as $AnsFR(q_i, relation)$, which will be True if the specified relation exists in the set of answers to q_i. We obtain a set of consistency rules per training example, as shown in Table 1. For example, in Figure 1, if q_1: box above the square," is True, then q_3: "triangle below box," should be True. The corresponding constraints for YN will be $AnsYN(q_1) \Rightarrow AnsYN(q_3)$, and for the FR case, will be $AnsFR(q_1, above) \Rightarrow AnsFR(q_3, below)$.

Lastly, after obtaining the consistency constraints, we minimize the violation of the model from these constraints by adding a corresponding term in the loss function objective. However, we need to obtain a differential form of logic as a surrogate of the original logical constraints violation to do this. We follow the previous research for this goal [19, 20, 21] and use the DomiKnowS framework for the actual implementation [22]. To implement this problem using the DomiKnowS declarative language, we must declare a graph of concepts and relationships and add the logical rules/constraints between them. DomiKnowS offers a Python library and a specific syntax to express the graph and logic. An example of concepts, a symmetric relation, and a constraint using symmetric relation is as follows,

```
story = Concept(name="story")
question = Concept(name="question")
story_contain, = story.contains(question)
answer_class = question(name="answer_class",
↪   ConceptClass=EnumConcept,
                                values=["yes", "no"])
symmetric = Concept(name="symmetric")
s_quest1, s_quest2 =
↪   symmetric.has_a(arg1=question,
↪   arg2=question)
ifL(andL(answer_class.yes('x'),
↪   existsL(symmetric('s', path=('x',
↪   symmetric)))),
    answer_class.yes(path=('s', s_quest2)))
```

We refer the reader to DoinKnowS documentation about the syntax and the semantics of the code [1]. Our main hypothesis is that providing supervision from high-level logical knowledge enables the model to capture higher levels of abstraction, improving generalization to other domains. The advantage of our proposed approach is that it does not require full access to logical knowledge. Any partially available knowledge can be exploited during training without further requirement at inference time. This is crucial since inference-time symbolic reasoning can be time-consuming for real-time applications.

3. Experimental Results

We conduct two sets of experiments on realistic (ReSQ) and synthetic datasets (SpartQA, SpaRTUN, and StepGame). With these experiments, we empirically evaluate the impact of our proposed logic-based fine-tuning on small-scale language models and compare them to very large language models that merely use prompt engineering. We evaluate the performance of our proposed method on two types of language models, encoder-based and generative models. We select BERT as the baseline encoder-based and Flan-T5 as the baseline generative model.

We also report the results of basic fine-tuning with the

[1] https://hlr.github.io/domiknows

SpaRTUN dataset in two, so-called, BERT-T and Flan-T5-T models.

Realistic Domain. ReSQ serves as the realistic SQA domain. As observed in Table 2, using the *Q-chain* is effective for both models (BERT and Flan-T5) with a notable improvement on Flan-T5. Particularly, Flan-T5-T+*Q-chain* (line 6) shows 2% improvement over Flan-T5-T (line 5).

For a deeper understanding of these results, we analyzed the performances on different splits of ReSQ. There are three splits based on the manually annotated depth of reasoning required to answer questions in ReSQ. The first two splits include questions that require one or two hops of reasoning, denoted as k=1, and k=2. The last type is *unclassified*, which covers questions where the depth of reasoning is difficult to determine. Those questions require more of commonsense knowledge. Our observations in Table 2 reveal that our model consistently improves on $k = 2$ but adversely affects BERT's performance on $k = 1$ and the *unclassified* categories. According to this result, we conclude that when more hops of reasoning are required, logic-based tuning demonstrates significant improvement. However, our proposed tuning method is less effective in the *unclassified* class, which requires commonsense knowledge.

On the other hand, LLMs show superior performance on ReSQ compared to all fine-tuning results. The LLMs consistently exhibit around 2% to 13% higher performance compared to Flan-T5+T+*Q-chain* (lines 7 to 13). The performance is much higher on the *unclassified* subset of the dataset, which can be seen even with the *zero-shot* method. This implies that LLM's out-performance is mainly due to their commonsense knowledge rather than their complex reasoning capability, in contrast to our proposed method, which deals with complex multi-hop reasoning.

Nevertheless, we observe that using logic-based fine-tuning yields a higher improvement over Flan-T5 compared to BERT on the unclassified subset. This indicates that the *Q-chain* approach can guide complex reasoning when applied to a model with more commonsense knowledge.

Synthetic Domain with More Complex Logical Reasoning. SpartQA-Human and StepGame are synthetic domains used in our experiments. We consistently observe improvement with our proposed *Q-chain* in this domain, which typically requires many more hops of reasoning. As observed in Table 2, *Q-chain* consistently shows improvement in Flan-T and BERT compared to fine-tuning without it. Moreover, the gap between small PLMs and LLMs is much smaller in this dataset compared to the realistic domain (ReSQ). This is expected since LLMs are better at commonsense than complex reasoning as previously presented.

The result is further supported when assessing the proposed method on StepGame. As can be observed in Table 3, the fine-tuning method consistently demonstrates significant positive differences in all reasoning steps compared to LLMs. The struggle of GPT3 on reasoning over StepGame is also investigated in [13]. The reported results from this paper are in Table 3. Our proposed method consistently improves by 1%—4% on a higher number of reasoning hops (k = 6 to k = 10), similar to the observation in ReSQ. These results confirm our primary hypothesis that our proposed method equips the models with a higher level of logical abstraction to conduct higher reasoning steps.

References

[1] Y. Zhang, Q. Guo, P. Kordjamshidi, Towards navigation by reasoning over spatial configurations, 2021. arXiv:2105.06839.

[2] Y. Zhang, P. Kordjamshidi, Lovis: Learning orientation and visual signals for vision and language navigation, 2022. arXiv:2209.12723.

[3] F. Liu, G. Emerson, N. Collier, Visual spatial reasoning, Transactions of the Association for Computational Linguistics 11 (2023) 635–651.

[4] E. A. Sisbot, L. F. Marin, R. Alami, Spatial reasoning for human robot interaction, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2007, pp. 2281–2287.

[5] E. Yadollahi, M. A. Monteiro, A. Paiva, Learning spatial reasoning in virtual vs. physical games with robots, in: Proceedings of the 11th International Conference on Human-Agent Interaction, HAI '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 162–170. URL: https://doi.org/10.1145/3623809.3623830. doi:10.1145/3623809.3623830.

[6] Y. Zhang, P. Kordjamshidi, LOViS: Learning orientation and visual signals for vision and language navigation, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 5745–5754. URL: https://aclanthology.org/2022.coling-1.505.

[7] J. Atif, C. Hudelot, G. Fouquier, I. Bloch, E. D. Angelini, From generic knowledge to specific reasoning for medical image interpretation using graph based representations., in: IJCAI, 2007, pp. 224–229.

[8] S. Datta, Y. Si, L. Rodriguez, S. E. Shooshan, D. Demner-Fushman, K. Roberts, Understanding spatial language in radiology: Representation framework, annotation, and spatial relation extraction from chest x-ray reports using deep learning, Journal of Biomedical Informatics 108 (2020) 103473. URL: https://www.sciencedirect.com/science/article/pii/S1532046420301027. doi:https://doi.org/10.1016/j.jbi.2020.103473.

[9] S. Gong, Y. Zhong, W. Ma, J. Li, Z. Wang, J. Zhang, P.-A. Heng, Q. Dou, 3dsam-adapter: Holistic adaptation of sam from 2d to 3d for promptable medical image segmentation, 2023. arXiv:2306.13465.

[10] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[12] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, P. Fung, A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, 2023. URL: https://arxiv.org/abs/2302.04023. arXiv:2302.04023.

[13] Z. Yang, A. Ishay, J. Lee, Coupling large language models with logic programming for robust and general reasoning from text, 2023. arXiv:2307.07696.

[14] R. Mirzaee, P. Kordjamshidi, Disentangling extraction and reasoning in multi-hop spatial reasoning, 2023. arXiv:2310.16731.

[15] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. Guibas, F. Xia, Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, 2024. arXiv:2401.12168.

[16] R. Mirzaee, H. R. Faghihi, Q. Ning, P. Kordjmashidi, Spartqa: : A textual question answering benchmark for spatial reasoning, 2021. arXiv:2104.05832.

[17] R. Mirzaee, P. Kordjamshidi, Transfer learning with synthetic corpora for spatial role labeling and reasoning, 2022. arXiv:2210.16952.

[18] Z. Shi, Q. Zhang, A. Lipani, Stepgame: A new benchmark for robust multi-hop spatial reasoning in texts, 2022. arXiv:2204.08292.

[19] T. Li, V. Gupta, M. Mehta, V. Srikumar, A logic-driven framework for consistency of neural models, 2019. arXiv:1909.00126.

[20] A. Asai, H. Hajishirzi, Logic-guided data augmentation and regularization for consistent question answering, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5642–5650. URL: https://aclanthology.org/2020.acl-main.499. doi:10.18653/v1/2020.acl-main.499.

[21] H. Wang, M. Chen, H. Zhang, D. Roth, Joint constrained learning for event-event relation extraction, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 696–706. URL: https://aclanthology.org/2020.emnlp-main.51. doi:10.18653/v1/2020.emnlp-main.51.

[22] H. R. Faghihi, Q. Guo, A. Uszok, A. Nafar, E. Raisi, P. Kordjamshidi, Domiknows: A library for integration of symbolic domain knowledge in deep learning, 2021. arXiv:2108.12370.