Advancing Spatial Reasoning in Large Language Models: An In-Depth Evaluation and Enhancement Using the StepGame Benchmark

Fangjun Li[1], David C. Hogg[1], Anthony G. Cohn[1,2]
[1]University of Leeds, UK [2]Alan Turing Institute, UK

AAAI-24

Introduction

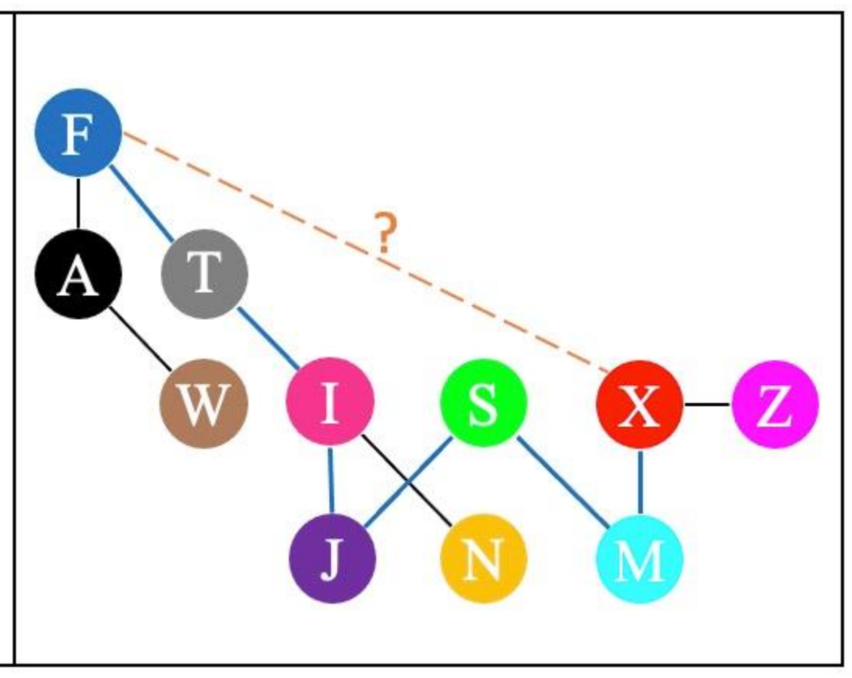
AI has made remarkable progress across various domains, with large language models (LLMs) like ChatGPT gaining substantial attention for their human-like text-generation capabilities. However, spatial reasoning remains a significant challenge, with ChatGPT's performance on spatial benchmarks like StepGame being unsatisfactory. Our analysis of GPT's spatial reasoning on a rectified StepGame benchmark identifies its proficiency in mapping text to spatial relations, yet it struggles with complex reasoning. We provide a flawless solution to the benchmark by combining template-to-relation mapping with logic-based reasoning. To address the limitations of GPT models in spatial reasoning, we deploy Chain-of-Thought (CoT) and Tree-of-Thoughts (ToT) prompting strategies, offering insights into GPT's "cognitive process", and achieving notable improvements in accuracy.

The StepGame Benchmark

Task: multi-hop spatial reasoning in texts

10- hop story:
1. T is slightly off center to the top left and I is slightly off center to the bottom right.
2. M is at the bottom of X.
3. Z presents right to X.
4. N is lower right of I.
5. S is positioned above J and to the right.
6. F is above T at 10 o'clock.
7. A is to the upper left of W.
8. A is at the bottom of F.
9. N is sitting in the right direction of J.
10. M is placed at the lower right of S.

What is the relation of the agent X to the agent F?

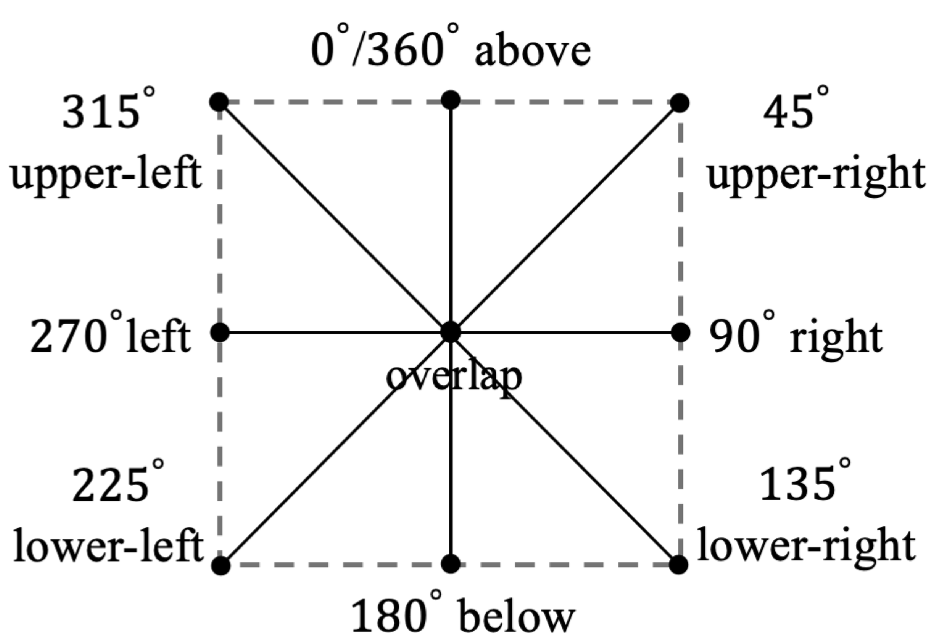


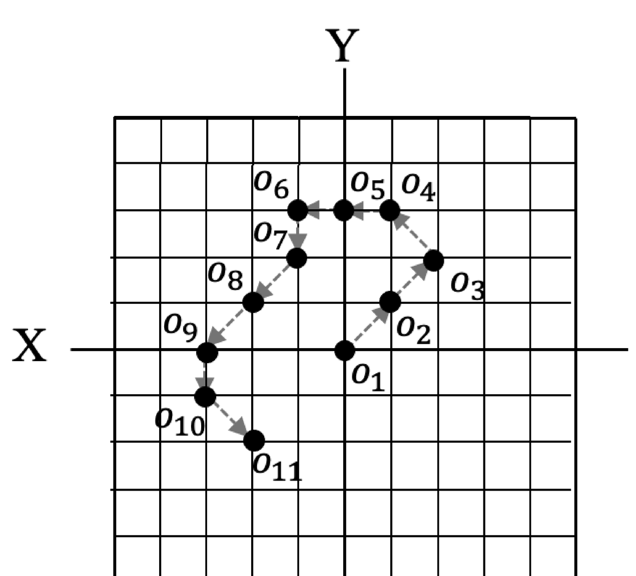
Solution to StepGame

Sentence-to-Relation Mapping + ASP Reasoner

Sentences	Template	ASP Facts
Y and I are parallel, and Y is on top of I. →	Y_above_I →	above("Y", "I").
F is on the left side of and below Q. →	F_lowerleft_Q →	down_left("F", "Q").
J is at O's 6 o'clock. →	J_below_O →	below("J", "O").
A is directly north east of B. →	A_upperright_B →	up_right("A", "B").
What is the relation of the agent B to the agent J?	query_B_J	query("B", "J").

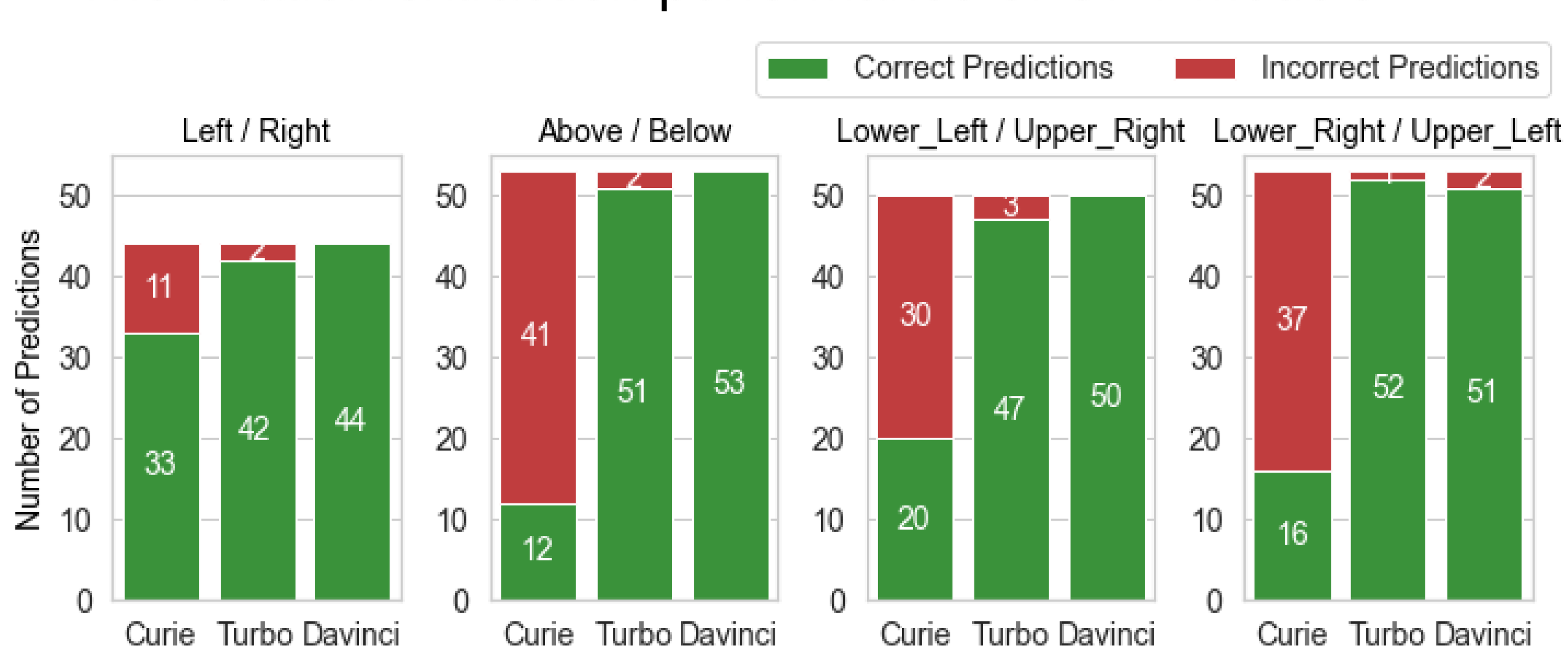
The ASP module calculates the location of o_i to o_j by adding the offsets $v(o_i, o_j)$.





LLM + ASP

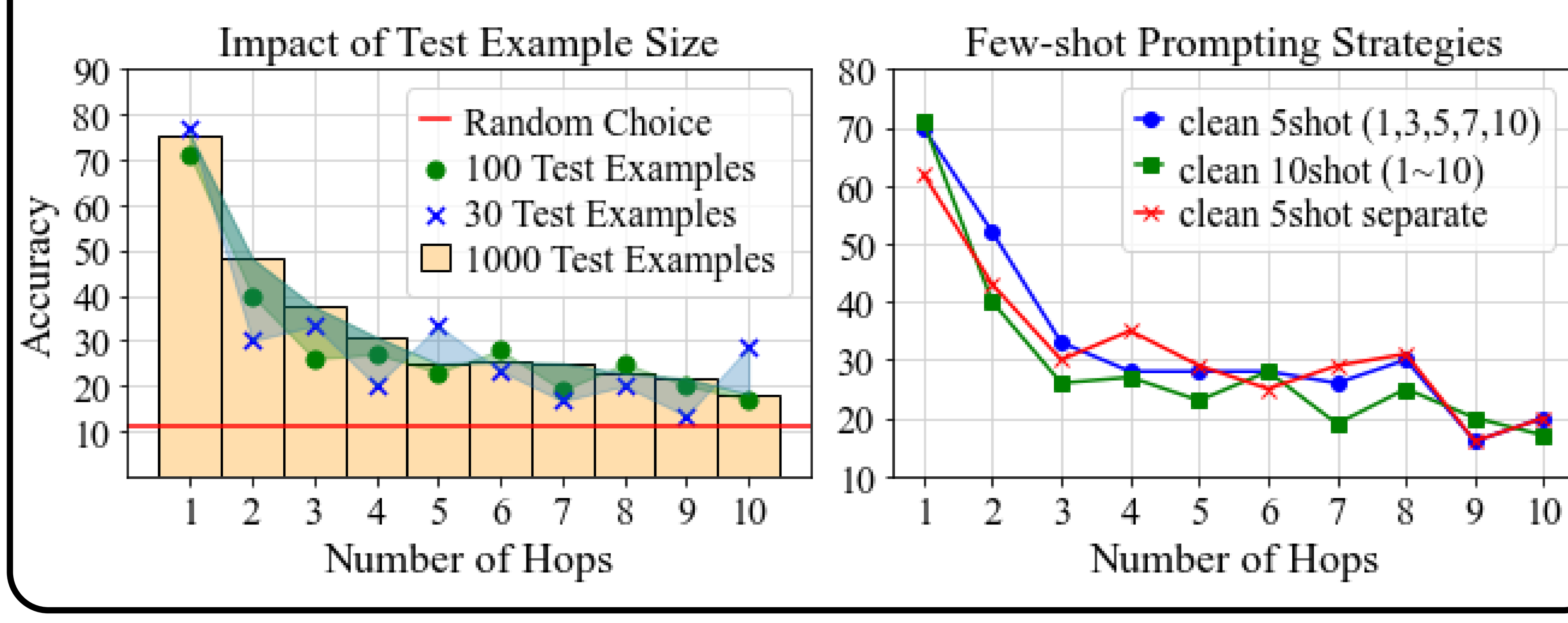
The relation extraction performance of GPT models.



Results of LLMs for relation extraction + ASP Reasoner

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Map+ASP	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Curie+ASP	46	43	42	59	67	67	57	56	58	61
Davinci+ASP	**100**	**100**	99	**100**	**100**	99	**100**	**100**	**100**	**100**
SOTA	92.6	89.9	89.1	93.8	92.9	91.6	91.2	90.4	89.0	88.3

Evaluation of GPT-3.5 Turbo on StepGame

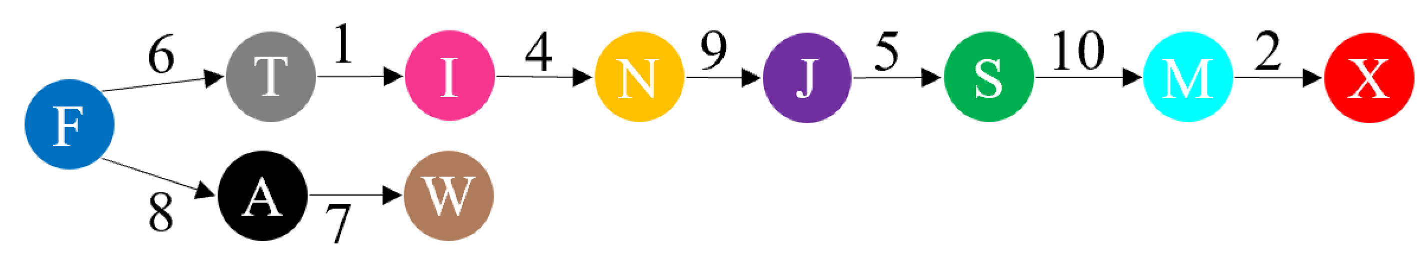


Methods

Our CoT approach decomposes each step of thought c_i to incorporate a coherent and detailed reasoning process.

At reasoning step i, $\boldsymbol{c_i} = [\boldsymbol{c_i^{link}}, \boldsymbol{c_i^{map}}, \boldsymbol{c_i^{calcu}}]$

- $\boldsymbol{c_i^{link}}$: guide LLMs to examine all relations in story ($R = [r^1, \dots, r^j, \dots, r^k]$) and select candidate r^j for each i
- $\boldsymbol{c_i^{map}}$: map r^j to simple relation description o_i is to the v of o_{i+1}
- $\boldsymbol{c_i^{calcu}}$: calculate the coordinate of o_{i+1} with r^j, $o_{i+1} = o_i + v(r^j) = (x_{o_i}, y_{o_i}) + (x_v, y_v) = (x_{o_{i+1}}, y_{o_{i+1}})$



Our ToT approach is designed to enhance the chain building process, allowing LLMs to consider different pathways.

Require: LLM, input x
1: $S_0 \leftarrow Init(x)$
2: $i \leftarrow 1$
3: **while** no $s_f \in S_{i-1}$ has arrived at o_t **do**
4: $S_i' \leftarrow \{s \cdot c | c \in G(s, j) \wedge ChainExtn(c) \wedge s \in S_{i-1}\}$
5: **if** $S_i' = \emptyset$ **then return** failure
6: $S_i \leftarrow select(b, \{\langle s, y \rangle | s \in S_i' \wedge y = \sum_1^n \sigma(V(s))\})$
7: $i = i + 1$
8: **end while**
9: return $Link(s_f)$

Results - Accuracy

		k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Turbo	base	62	43	30	35	29	25	29	31	16	20
	CoT	/	34	40	36	28	28	26	31	25	24
	ToT	/	/	35	35	25	45	15	40	40	35
Davinci	base	77	42	21	26	25	30	23	23	22	22
	CoT	/	48	53	46	46	48	40	45	41	32
	ToT	/	/	65	50	45	60	50	50	55	50
GPT-4	base	**100**	70	55	45	40	25	40	35	35	25
	CoT	/	**80**	75	**95**	85	85	**90**	80	60	65
	ToT	/	/	**85**	85	**90**	**90**	85	**90**	**100**	**95**