Injecting Context via Situation Working Memory for Logical Reasoning with LLMs

Jieun Kim[1] **Seoha Lim**[1] **YoungHae Choi**[2] **Sung-Bae Cho**[2]
[1]Dept. of Artificial Intelligence and [2]Dept. of Computer Science, Yonsei University
{lilly9928,seoha815,taphy,sbcho}@yonsei.ac.kr

Abstract

Recent advances in large language models (LLMs) have improved logical reasoning by incorporating formal logic or explicit structured representations. However, such methods often lose track of *what is true now* in multi-step reasoning, failing to maintain a coherent global state and its logical consequences. Motivated by Situation Model Theory in cognitive psychology, which views comprehension as constructing and updating a mental model of events along key dimensions (time, space, causality, intention, protagonist), we propose a cognitively inspired method of Situation Working Memory (SituW) for contextual reasoning in LLMs. SituW first builds a situation representation by decomposing text along these five dimensions, and guides LLM inference with the evolving state. Keeping an explicit, dynamically updated situation memory instead of a static logical form encourages globally consistent reasoning over the situation model rather than raw text. Evaluated in both supervised and prompt-based settings, SituW improves accuracy by 23.3%p and 15.93%p while reducing "uncertain" predictions, suggesting that explicit situation modeling supports more globally consistent LLM reasoning.Our code is available at: https://github.com/lilly9928/SituW

1 Introduction

Understanding context in human language processing is not strictly sequential. Instead, humans construct a mental representation of the described situation while comprehending individual sentences. For instance, in figure 1 upon reading *"Peter took the elevator to the fifth floor,"* the reader infers that Peter has arrived at the fifth floor. When encountering *"He went to talk to his professor,"* they naturally deduce that the professor's office is likely on the fifth floor. This inferential process reflects the dynamic nature of human comprehension. In cognitive psychology, such mental representations

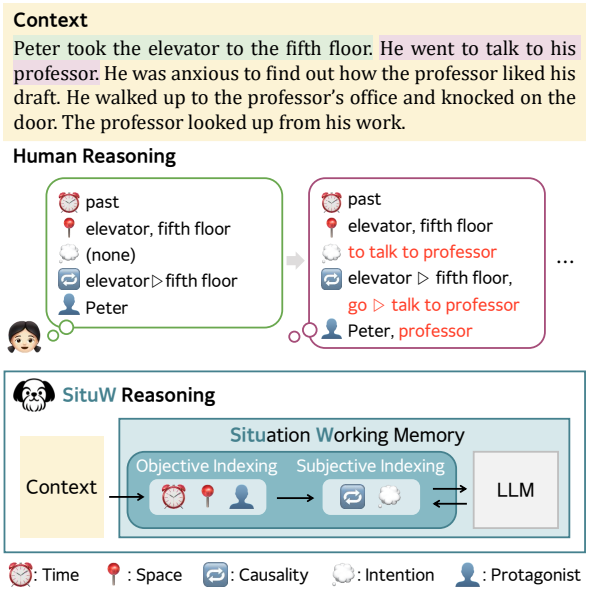

Figure 1: **Motivation for situation working memory**. In sentence understanding and reasoning, humans compress text into a situation model using cues like time, space, causality, intention, and protagonist.

are known as *situation models* (Zwaan and Radvansky, 1998), which structure information along key dimensions such as time, space, causality, intention, and protagonist. Effective reasoning occurs when information from multiple sources is integrated into a unified situation model (Kintsch, 1995).

Despite the remarkable performance of large language models (LLMs) in various reasoning tasks, challenges remain in complex logical inference (Parmar et al., 2024; Sun et al., 2024; Sinha et al., 2019; Kim et al., 2026; Kim and Cho, 2026). A common strategy involves converting intermediate reasoning steps into formal logic (Pan et al., 2023; Ye et al., 2023; Wang et al., 2024a; Gao et al., 2023), but this approach has two key limitations: (1) accurately translating natural language into formal logic while preserving contextual nuances is inherently difficult (Wang et al., 2024a;

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 32467–32480

Sen et al., 2022), and (2) reliance on external symbolic solvers causes errors due to their rigid dependence on precise logical representations (Pan et al., 2023; Chae et al., 2024).

To address these limitations, we propose Situation Working Memory(SituW), a cognitively grounded method that enchances contextual reasoning in LLM. SituW improves logical reasoning by dynamically constructing and using both objective and subjective indexing to capture evolving contextual dependencies. Our method consists of two key phases: (1) situation construction, where the system incrementally builds an objective and subjective indexing while processing a passage, and (2) situation reasoning, where the model leverages the constructed SituW to guide and enhance reasoning. This structured yet flexible method allows SituW to achieve superior performance on various logical reasoning benchmarks. Our main contributions are as follows:

- **Incorporation of situation models into LLM**: To the best of our knowledge, this is among the first work to incorporate situation models from cognitive psychology into LLM-based reasoning, introducing a structured inference method.
- **Beyond the rigidity of symbolic logic**: Unlike formal logic-based methods, our method dynamically constructs and updates a situation working memory, enabling adaptive and context-aware reasoning.
- **Enhanced logical inference**: Using situation modeling, SituW achieves significant improvements over existing symbolic and conventional LLM reasoning approaches.

2 Situation Working Memory

The situation model theory in cognitive psychology explains how humans construct mental representations of the described events during language comprehension (Zwaan and Radvansky, 1998). Building on this foundation, we propose Situation Working Memory (SituW) that enables large language models to maintain and update a situation representation during reasoning. SituW represents each situation using two complementary indexing components: objective indexing and subjective indexing, which together form a dynamic, context-sensitive situation representation. However, it remains unclear *whether current LLMs actually construct such situation-level representations during text processing*. Before detailing the architecture of SituW, we therefore first probe this question empirically by adapting a classic confusable-pair paradigm from cognitive psychology to the LLM setting.

2.1 Confusable Pair Identification as a Test of Situation Modeling

Setup. To investigate this question, we adapt the mental model paradigm introduced in (Garnham, 1981) to the LLM setting. In the original experiment, participants heard sentences presented by an experimenter and were later asked to recognize the sentence they had heard by selecting it from a set of options. Because this auditory recognition procedure cannot be applied directly to LLMs, we reformulate the paradigm as a text-based discrimination task.

In our setup, each trial consists of four sentences, and the model's task is to select the confusable pair, that is, the two sentences whose interpretations are most strongly overlapped and, therefore, are more likely to be taken as describing the same underlying event. Our task uses a set of the 24 sentences introduced in (Garnham, 1981); the full set of materials is listed in Appendix B. These sentence sets were originally designed to exploit minimal changes in a key preposition (e.g., *at* vs. *by*, *from* vs. *in*) in order to trigger two distinct event interpretations, thereby inducing competing situation-model representations. An example item and the corresponding instruction prompt are as follows:

Example for Confusable Pair Identification

A confusable pair refers to two expressions that are similar in meaning or used in similar contexts, making them easy for learners to confuse.

Among the options (a), (b), (c), and (d) below, choose the confusable pair.

(a) The girl was given a complete pedicure at the chiropodist's.
(b) The girl was given a complete pedicure by the chiropodist.
(c) The girl had her handbag stolen at the chiropodist's.
(d) The girl had her handbag stolen by the chiropodist.

Results. For comparison with human behavior, we also ran the same paradigm with 11 graduate student participants. As shown in Figure 2, humans achieved an accuracy of 89.28%, while the LLM's performance was substantially lower. These human evaluation results align with those reported in (Garnham, 1981). They indicate that, in identifying written sentences, humans go beyond grammatical analysis and instead integrate linguistic input into a coherent situation model, enabling reliable judgments of event identity even when surface realizations differ.

By contrast, baseline LLMs appear to depend more on surface-level correspondences, suggesting that they solve the task through superficial cues rather than human-like semantic understanding. To test whether making situational structure explicit can mitigate this limitation, we further applied our method to closed-source GPT models: for each trial, the model was first prompted to extract objective indices (time, location, protagonists) and subjective indices (cause, intention), and then choose the confusable pair conditioned on this extracted situation description. This explicit situation-index prompting substantially improved accuracy on GPT models (e.g., 20.8%p for GPT-3.5 and 8.3%p for GPT-4o-mini). Motivated by these findings, we propose Situation Working Memory that enables LLMs to maintain and update event-level context during reasoning through objective and subjective indexing.

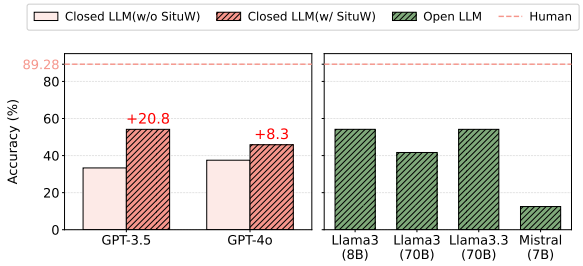

Figure 2: **Accuracy on the confusable pair identification task.** Closed-source GPT models with and without situation-index prompting, open-source LLMs, and humans.

2.2 Situation Memory Construction

Building on this foundation, Situation Working Memory provides a way for large language models to maintain and update a situation representation during reasoning. The situation-level representation maintained by SituW is called situation memory. We implement two pipelines for constructing situation memory: (1) an prompt-based pipeline, and (2) a supervised pipeline based on dataset synthesis and model distillation. In the prompt-based pipeline, situation memory is generated from prompts at inference time. In the supervised pipeline, we first use GPT to synthesize a training dataset and then fine-tune a smaller model to generate situation memory.

Composition. As illustrated in Figure 3, SituW consists of two indexing components: objective indexing and subjective indexing. Objective indexing encodes explicit aspects of a situation, such as time, space, and protagonists. In contrast, subjective indexing captures implicit aspects of a situation, such as causality and intention, which often require interpretation based on prior knowledge and experience.

Situation Working Memory

Objective Indexing

Time	*(None)*
Space	*sugar pines*
Protagonist	*Zhu Hong, Red squirrels, Lina*

Subjective Indexing

Causality	*The low concentration of sugar in sugar pine sap, Red squirrels searching for water or sugar*
Intention	*Searching for water or sugar, To discuss the dietary habits of red squirrels regarding sugar pine sap, To absorb sap from sugar pines*

Figure 3: **Example of situation working memory.** Objective indices (time, space, protagonists) and subjective indices (causality, intention) encode a given context.

Prompt-based Construction. To construct situation memory, we decompose the semantic representation of an input sentence S into two distinct indexing processes: objective indexing and subjective indexing. Each process is executed via a large language model, and contributes complementary components to the final memory representation. Objective indexing extracts entity-independent factual information directly observable in the sentence. Specifically, the LLM identifies three key components: time (t), space (s), and the main protagonist(p). Formally, the objective index O is defined as:

$$O = \text{LLM}_{\text{obj}}(S) = \{t, s, p\}. \quad (1)$$

Subjective indexing, on the contrary, involves inferential reasoning based on the objective elements

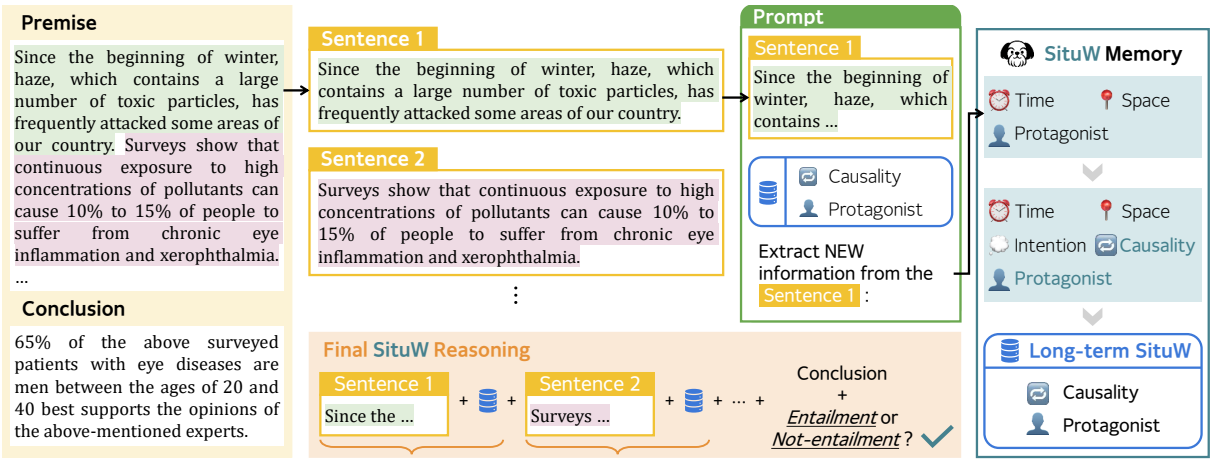

Figure 4: **Overview of situation working memory construction.** The input context is split into sentences and processed sequentially. For each sentence, an LLM extracts newly implied situational elements and incrementally updates situation working memory.

extracted previously. It targets the extraction of causality (c) and intention (i), which are inherently interpretative and context-sensitive. The subjective index U is therefore conditioned on both the original sentence and its objective structure:

$$U = \text{LLM}_{\text{subj}}(S, O) = \{c, i\}. \tag{2}$$

The final situation memory M is the union of these two sets, capturing both the factual and interpretive dimensions of the situation:

$$M = O \cup U = \{t, s, p, c, i\}. \tag{3}$$

This formulation allows for a modular and interpretable memory structure that can be utilized in downstream tasks that require contextual understanding, temporal reasoning, or agent modeling. The prompts used to extract information are shown in Table 1.

Element	Prompt Template
Time	Extract the time: When does this occur? Reply with a short phrase or None.
Space	Extract the location: Where does this occur? Reply with a short phrase or None.
Causality	Extract the cause: What triggered this? Reply with a short phrase or None.
Intention	Extract the intention: What is the purpose? Reply with a short phrase or None.
Protagonist	Extract the protagonist: Who is involved? Reply with a short phrase or None.

Table 1: **Prompt templates** used to extract situational memory from text.

Supervised Construction. To construct situation memory in a supervised manner, we use a GPT-based reasoning model as a teacher. Given an input context X, we first split it into an ordered sequence of sentences: $X = \{S_1, \ldots, S_n\}$.

For each sentence S_n, it extracts a situation-model snippet m_n consisting of five scales, using the same annotation as defined previously:

$$m_n = \Big(\{t_n, s_n, p_n\}, \{c_n, i_n\}\Big). \tag{4}$$

When moving from sentence S_n to S_{n+1}, we pass forward the proposition and protagonists to maintain coherence between sentences. We denote these carried-over variables as

$$\begin{aligned} \pi_k &= \text{Proposition}(m_n), \\ \rho_k &= \text{Protagonists}(m_n), \end{aligned} \tag{5}$$

and condition the next-step extraction on them. Formally, the teacher generates m_n conditioned on the current sentence, the preceding sentences, the previous memory state, and the carried-over variables, where $\pi_0 = \emptyset$ and $\rho_0 = \emptyset$.

$$m_n = f_{\text{teacher}}(S_k \mid S_{<n}, M_{n-1}, \pi_{n-1}, \rho_{n-1}). \tag{6}$$

These snippets are accumulated into a running memory state via an update rule:

$$M_n = \text{Update}(M_{n-1}, m_n), \qquad M_0 = \emptyset. \tag{7}$$

After processing all sentences, we obtain the final memory M_n.

To construct the final reasoning prompt, we store a memory trace T by concatenating each segmented sentence with its extracted situation-model snippet:

$$T = \{\, S_1 \oplus m_1,\ S_2 \oplus m_2,\ \ldots,\ S_n \oplus m_n \,\},$$

where $\oplus$ denotes the concatenation operator.

An inference module then produces the prediction using the final memory and the memory trace with reasoning prompt (Appendix D):

$$\hat{y} = g_\phi(M_n, X, T). \tag{8}$$

To ensure high-precision pseudo-labels, we keep a generated memory trace only when the final prediction matches the ground-truth label y, which yields the pseudo-labeled dataset,

$$\tilde{\mathcal{D}} = \{(X, T, y) : \hat{y} = y\}. \tag{9}$$

Finally, we fine-tune an open-source large language model on $\tilde{\mathcal{D}}$ using unsloth (Hu et al., 2022).

Situation Memory Construction Prompt

You are reading a logical context sentence-by-sentence.
Return ONLY a valid JSON object.

---

sentence:
{premises}

memory statements:
{situation memory}

memory protagonists:
{situation memory protagonists}

2.3 Situation Reasoning

To extract the final answer in an prompt-based setting, we proceed through the following pipeline. Once situation memory $M = \{t, s, p, c, i\}$ is constructed, we transform each of its components into natural language statements that can be appended to the original context. This step enables downstream models to reason over the enriched context using explicit, interpretable cues grounded in time, space, agency, causality, and intention. Formally, we denote the original context as C, and each memory component $m_i \in M$ is processed in a predefined order: first, objective dimensions (t, s, p), followed by subjective dimensions (c, i). Each component is verbalized using an LLM:

$$C_{j+1} = C_j + \text{LLM}_{\text{verbal}}(m_j), \quad \text{for } j = 0, 1, \ldots, 4. \tag{10}$$

Here, $\text{LLM}_{\text{verbal}}(m_i)$ refers to the natural language description generated from the memory element m_i. After five iterations, we obtain an enriched context C_5 that incorporates all five situational dimensions. The final context C_5 can be integrated with a wide range of reasoning strategies. The model then performs iterative reasoning over C_5 using four steps: *Plan*, *Action*, *Update*, and *Process*.

In a supervised setting, we construct a synthetic training set $\tilde{\mathcal{D}}$. Given an instance in $\tilde{\mathcal{D}}$, we train the model to predict the corresponding target y.

3 Experiments

We evaluate logical reasoning in both supervised and prompt-based settings.

Datasets. We conducted experiments on the following logic-oriented datasets.

PrOntoQA (Saparov and He, 2023) is a synthetic deductive FOL benchmark (modus ponens). Following (Pan et al., 2023), we use the hardest fictional-characters split (5-hop subset).

ProofWriter (Tafjord et al., 2021) tests deductive reasoning under the open-world assumption (True/False/Unknown). We evaluate the depth-5 subset with 600 balanced instances and additionally report results stratified by reasoning depth (0-5).

FOLIO (Han et al., 2024) is an open-domain dataset that requires naturalistic FOL reasoning. We use the complete test set (204 examples).

LogiQA 2.0 (NLI) (Liu et al., 2023a) casts exam-style logic into NLI. We evaluate 3,840 pairs of premise-hypothesis (Entailed vs. Not Entailed).

Baselines. We compare against four representative reasoning baselines. *Standard LLM* directly predicts the answer from the given context and query without explicit intermediate steps. *CoT* (Wei et al., 2022) elicits step-by-step natural language reasoning before producing the final answer. *SymbCoT* (Xu et al., 2024) converts the input into symbolic representations and performs structured logic-guided reasoning. *Logic-LM* (Pan et al., 2023) translates natural language into formal logic expressions and uses a symbolic solver to derive the prediction.

Models. For closed-source backbones, we use GPT-3.5-turbo and GPT-4o-mini (OpenAI, 2023). In the supervised setting, we generate training data with GPT-3-nano and fine-tune instruction-tuned open LLMs (Llama-3.1-Instruct 8B/70B, Llama-3.3-Instruct 70B, Mistral-Instruct 8B, Qwen2.5-Instruct 72B). In the prompt-based setting, we conduct ablations on memory construction and reason-

Method	PrOntoQA	ProofWriter	FOLIO	AVG
GPT-3.5-turbo				
Standard	47.40	40.00	45.09	50.20
CoT	67.80	49.17	**57.35**	57.78
SymbCoT	74.20	53.67	39.71	55.86
Logic-LM	58.80	51.66	50.98	53.71
SituW	**82.20**	**54.67**	53.43	**63.43**
GPT-4o-mini				
Standard	58.20	39.33	61.76	53.16
CoT	72.60	44.16	64.22	61.46
SymbCoT	68.20	62.00	65.69	65.29
Logic-LM	77.40	37.33	62.25	58.99
SituW	**79.00**	**64.33**	**72.06**	**71.79**

Table 2: Performance comparison on logical reasoning benchmarks under prompt-based settings. Best accuracy is in **bold**, second best is underlined.

	0-shot	2-shot		FT		
Size	Direct	Direct	CoT	Vanilla	**SituW**	Δ
Llama-3.1-instruct						
8B	47.78	36.95	44.33	38.92	**60.10**	+ 21.18
70B	61.58	67.98	74.38	38.42	**76.35**	+ 37.93
Llama-3.3-instruct						
70B	64.04	65.02	67.00	35.54	**74.88**	+ 39.34
Mistral-instruct						
7B	47.29	44.33	51.23	51.23	**62.56**	+ 11.33
Qwen2.5-instruct						
72B	67.49	69.95	55.67	71.14	**78.00**	+ 6.86

Table 3: Performance comparison on FOLIO under prompting and supervised settings. Best accuracy is in **bold**, second best is underlined. Δ is the gain over Vanilla FT.

ing modules using open-source backbones such as Gemma3 and Llama3.

3.1 Quantitative Results

Consistent Improvements under prompt-based Evaluation. Table 2 shows the performance of various reasoning methods on three benchmarks: PrOntoQA, ProofWriter, and FOLIO, using GPT-3.5-turbo and GPT-4o-mini. Our method, SituW, achieves the highest average accuracy with both models, 63.43% with GPT-3.5-turbo and 71.79% with GPT-4o-mini. SituW achieves the best results on PrOntoQA and ProofWriter for both backbone models, and also delivers strong gains on FOLIO, outperforming most baselines.

SituW Enables Stronger Supervision. Table 3 shows the performance of the supervised results on FOLIO, comparing the prompting (0-/2-shot) to the fine-tuning (FT). SituW fine-tuning consistently outperforms both prompting baselines and vanilla fine-tuning across all backbones, achieving 60.10 (Llama-3.1 8B), 76.35 (Llama-3.1 70B), 74.88 (Llama-3.3 70B), 62.56 (Mistral 7B), and 78.00 (Qwen2.5 72B). Compared to Vanilla FT, SituW yields substantial absolute gains ranging from +6.86 to +39.34%p, with the largest improvements on the Llama 70B models, where vanilla FT underperforms even a few-shot prompting. This highlights that SituW offers more effective supervision for logical reasoning than standard fine-tuning by constructing, iteratively updating, and reasoning over a sentence-level situation model.

Method	Setting	Vanilla (%)	SituW (%)	Δ
Human	–	**86.63**	–	–
GPT3.5-turbo	Standard	50.49	**54.81**	+ 4.40
	CoT	56.20	**58.30**	+ 2.10
GPT4o-mini	Standard	57.41	**61.97**	+ 4.56
	CoT	61.45	**65.78**	+ 4.33

Table 4: Performance comparison on LogiQA in an prompt-based setting across different models and prompting strategies. Best accuracy is in **bold**.

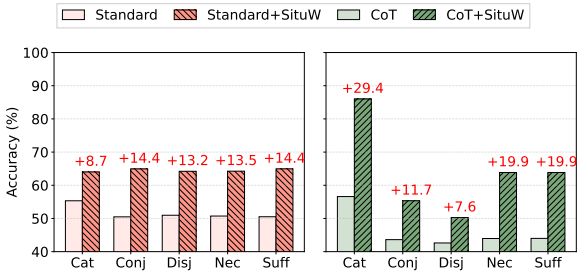

Figure 5: Category-wise accuracy on LogiQA.

Improves Zero-shot and Category-wise Accuracy on LogiQA. Table 4 shows the accuracy of the zero-shot on the LogiQA dataset. Our method, SituW, improves accuracy from 54.81% to 61.97% under the standard prompting setting and from 58.30% to 65.78% with CoT prompting. Figure 5 further breaks down performance by reasoning category, including categorical, conjunction, disjunction, necessity, and sufficiency. In both prompting settings, SituW consistently improves accuracy in all categories. These results suggest that SituW contributes to better handling of various types of logical reasoning in a zero-shot context.

3.2 Qualitative Results

More Confident, Less 'Uncertainty' Responses. In Figure 6, the left panel shows the accuracy of

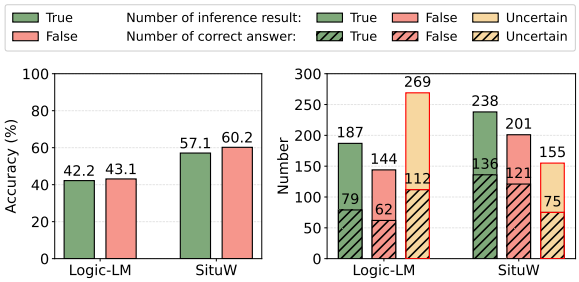

Figure 6: Response distribution and accuracy on LogiQA. Left: class-wise accuracy (%). Right: number of predicted labels (solid) and correct predictions (striped).

Objective	Causality	Intention	Accuracy (%)
✓			54.38
✓	✓		54.41
✓		✓	54.93
✓	✓	✓	**64.81**

Table 5: Ablation results of situation components.

true/false inference, while the right panel visualizes the number of predictions for each label (true/-false/uncertain; solid bars) and the corresponding number of correct predictions (striped bars). Logic-LM achieves a lower overall accuracy and produces a relatively higher proportion of 'uncertain' predictions. By contrast, SituW makes more true/-false decisions and achieves higher accuracy, with fewer 'uncertain' outputs. This pattern is consistent with the idea that converting statements into logical expressions can omit or weaken some contextual cues. Moreover, the reduced rate of 'uncertain' predictions with SituW suggests that incorporating situation-level understanding helps the model interpret sentences more reliably than relying primarily on structural parsing alone.

Full Components Needed for Coherent Situation Reasoning. Table 5 shows the contribution of each component through an ablation study. Using only the objective yields 54.38% accuracy. Adding the causality term alone provides virtually no improvement (54.41, 0.03%p), while adding the intention term alone gives a small gain (54.93, 0.55%p). In contrast, enabling both causality and intention simultaneously results in a large jump to 64.81% (10.43%p over the base objective). This non-additive gain indicates strong complementarity: the causality constraint is most effective when coupled with intention/goal alignment, and vise versa, suggesting that all components are required for a coherent, task-relevant situation representation.

Cons \ Inf	Gemma3-27B	Llama3-70B	GPT3.5
Gemma3-4B	62.19	65.15	54.78
Gemma3-27B	60.65	63.73	54.78
Llama3-8B	60.43	63.33	54.69
Llama3-70B	61.45	**64.85**	54.87

Table 6: Cross-model construction–inference pairing. Accuracy (%) for open-source LLMs across construction (Cons) and inference (Inf) combinations.

Method	Accuracy (%)	Tokens		Time (s)
		in	out	
Direct	44.16	132	13.33	0.1072
CoT	<u>58.10</u>	318	176	0.4824
SymbCoT	55.86	4655	426	0.4187
SituW	**63.43**	4790	425	0.57

Table 7: Performance comparison of methods in accuracy and efficiency. Best accuracy is in **bold**, second best is <u>underlined</u>.

Construction/Inference Combinations for Prompt-based Reasoning. Table 6 shows additional experiments in which we combine open-source LLMs for the construction of a situation model and for reasoning/inference in an prompt-based setting. Rows represent the construction model, and columns represent the reasoning model, which facilitates analysis of how different pairings affect accuracy. In general, Llama3-70B as the inference model delivers the strongest performance, achieving the best score when paired with Llama3-70B for construction. Gemma3-27B inference is slightly weaker, but remains relatively stable across construction choices. These results show that, beyond closed-source models, combining open-source models alone can also yield strong performance in an prompt-based setting.

Accuracy-efficiency Trade-off. Table 7 compares methods in terms of average accuracy, token usage, and runtime in ProntoQA, ProofWriter, and FOLIO. Direct prompting is the most efficient (132 input tokens, 13.33 output tokens, 0.107 sec) with the lowest computational cost, but also achieves the lowest accuracy (44.16%). CoT substantially improves accuracy to 58.10% with a moderate increase in cost (318/176 tokens, 0.482 sec). SymbCoT attains comparable accuracy (55.86%) but requires far more input tokens (4655) and similar

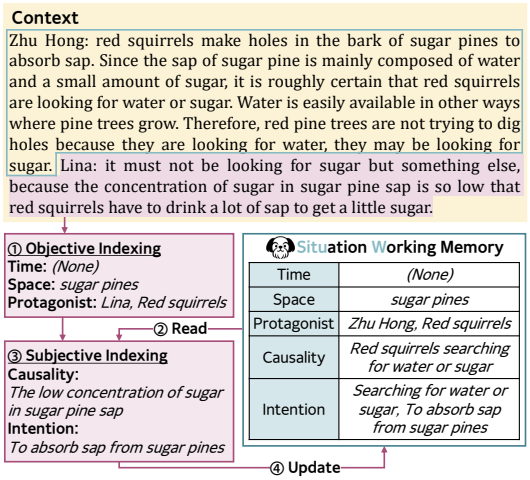

Figure 7: Overview of SituW's workflow.

runtime (0.419 sec), reflecting the overhead of symbolic conversion. Our approach achieves the best overall accuracy: SituW reaches 63.43% with 4790 input tokens and 0.57 sec.

Visualization of SituW. Figure 7 illustrates the end-to-end workflow of prompt-based SituW. Given a context, SituW first performs objective indexing to extract explicit situational attributes such as time, space (location), and protagonists. It then performs subjective indexing to infer latent but decision-critical factors, including causal relations (e.g., why an event happens) and intentions/goals (e.g., what an agent is trying to achieve). These extracted fields are written into a structured situation working memory with fixed slots (time, space, protagonist, causality, intention), and the memory is updated iteratively as more evidence is read. The resulting working memory provides a compact, structured state representation that conditions subsequent reasoning/planning, enabling the model to focus on relevant variables and their causal-goal structure rather than re-processing the full context.

4 Related Work

Logical Reasoning with LLMs. Chain-of-Thought(Wei et al., 2022) improves the reasoning ability of LLMs by guiding them through a step-by-step process. However, it struggles with tasks that require strict logical adherence(Han et al., 2024; Parmar et al., 2024; Sinha et al., 2019). Recent studies address this by using LLMs as semantic parsers, converting natural language into logical expressions processed by external reasoning engines(Wang et al., 2024b; Pan et al., 2023). Although this reduces logical errors, it introduces dependencies and potential information loss. To overcome these issues, recent methods integrate symbolic reasoning within LLMs, enabling internal logical processing even when expressions are imprecise(Sun et al., 2024; Xu et al., 2024).

LLMs with External Memory. LLMs achieve strong performance in many tasks (Team, 2025, 2024), but still struggle with multi-step reasoning and grounding over long-context documents (Plaat et al., 2026). To mitigate these limitations, recent work augments LLMs with external memory (Hu et al., 2023; Wang et al., 2024c). For multi-hop reasoning, symbolic-memory approaches (Hu et al., 2023), (Wang et al., 2024c) either use database-backed memory, which can be domain-dependent (Hu et al., 2023), or build predicate–object working memory with rule/fact bases (Wang et al., 2024c). Although the latter is related to our approach, it mainly applies when text can be reliably converted into rules and may lose information during symbolic conversion. Scratchpad-based methods (Nye et al., 2021) improve step-by-step computation, but are less directly applicable to general reasoning. On the contrary, our method targets broader reasoning settings without requiring domain-specific schemas or strict rule-based transformations.

5 Concluding Remarks

In this paper, we proposed a method that integrates situation models from cognitive psychology into LLM-based reasoning. By dynamically constructing a situation memory, it enables adaptive and context-aware inference, addressing key limitations of formal logic-based methods. Experimental results show significant improvements across multiple benchmarks, enhancing LLMs' intrinsic reasoning abilities without relying on external solvers. It is effective in both prompt-based and supervised settings: in the prompt-based setting, it serves as a plug-and-play inference method via prompt-based situation-memory construction, while in the supervised setting, we distill situation-memory traces as structured supervision to improve fine-tuning stability and generalization over vanilla fine-tuning. In general, these results highlight it as a practical and effective method for robust contextual reasoning in LLMs.

6 Limitations

Although SituW demonstrates strong performance on logical reasoning tasks, several limitations remain. First, the iterative process of situation memory construction and reasoning can substantially increase token usage due to repeated generation steps, leading to higher computational cost, especially in long-context settings where memory grows over time. Second, the multi-step framework introduces additional inference latency, as situation memory construction and subsequent reasoning require multiple sequential decoding passes. Third, because our approach relies on LLM-generated intermediate representations, errors or hallucinations in these representations may propagate through the reasoning process and lead to incorrect downstream outcomes. Improving runtime efficiency while maintaining reasoning quality, as well as reducing error propagation in intermediate steps, remains an important direction for future work.

Acknowledgments

This work was supported by IITP grant funded by the Korea government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); No. RS-2022-II220113, Developing a Sustainable Collaborative Multi-modal Lifelong Learning Framework).

References

Hyungjoo Chae, Yeonghyeon Kim, Seungone Kim, Kai Tzu-iunn Ong, Beong-woo Kwak, Moohyeon Kim, Sunghwan Kim, Taeyoon Kwon, Jiwan Chung, Youngjae Yu, and Jinyoung Yeo. 2024. Language models as compilers: Simulating pseudocode execution improves algorithmic reasoning in language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 22471–22502.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: program-aided language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 10764–10799.

Alan Garnham. 1981. Mental models as representations of text. *Memory & Cognition*, 9:560–565.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander R. Fabbri, Wojciech Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. FOLIO: natural language reasoning with first-order logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 22017–22031.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. Chatdb: Augmenting llms with databases as their symbolic memory. *CoRR*, abs/2306.03901.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Jieun Kim and Sung-Bae Cho. 2026. Neuro-symbolic reasoning with multiple large language models combined by first-order logic. In *Hybrid Artificial Intelligent Systems*, pages 227–238, Cham. Springer Nature Switzerland.

Jieun Kim, Yujin Jeong, and Sung-Bae Cho. 2026. Visual–linguistic abductive reasoning with LLMs for knowledge-based visual question answering. In *Findings of the Association for Computational Linguistics: (EACL) 2026*, pages 6529–6544.

Walter Kintsch. 1995. How readers construct situation models for stories: The role of syntactic cues and causal inferences. *Coherence in spontaneous text*, 1995:139–160.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023a. Logiqa 2.0 - an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023b. Logiqa 2.0 - an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *CoRR*, abs/2112.00114.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for

faithful logical reasoning. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 3806–3824.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13679–13707.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Bäck. 2026. Multi-step reasoning with large language models, a survey. *ACM Comput. Surv.*, 58(6):160:1–160:35.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *International Conference on Learning Representations (ICLR)*.

Prithviraj Sen, Breno W. S. R. de Carvalho, Ryan Riegel, and Alexander G. Gray. 2022. Neuro-symbolic inductive logic programming with logical neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8212–8219.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4505–4514.

Hongda Sun, Weikai Xu, Wei Liu, Jian Luan, Bin Wang, Shuo Shang, Ji-Rong Wen, and Rui Yan. 2024. Determlr: Augmenting llm-based logical reasoning from indeterminacy to determinacy. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9828–9862.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics (ACL)*, pages 3621–3634.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261.

Llama Team. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024a. Can llms reason with rules? logic scaffolding for stress-testing and improving llms. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7523–7543.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024b. Symbolic working memory enhances language models for complex rule application. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 17583–17604.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024c. Symbolic working memory enhances language models for complex rule application. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 17583–17604.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13326–13365.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. Satlm: Satisfiability-aided language models using declarative prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Rolf A. Zwaan and Gabriel A. Radvansky. 1998. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2):162–185.

A Experimental Setup: Datasets

To evaluate the validity of the situation model in actual reasoning processes, we conducted experiments on three datasets of logical reasoning.

PrOntoQA (Saparov and He, 2023) is a synthetic dataset designed to evaluate the deductive reasoning abilities of LLMs, focusing on first-order logic (FOL) with modus ponens. Following prior work (Pan et al., 2023), we use the version of the hardest fictional characters, specifically the 5-hop subset, which requires multi-step inference. Each instance involves verifying the truth value of a new fact in a question-answer format.

ProofWriter (Tafjord et al., 2021) is a benchmark for deductive reasoning under the open-world assumption (OWA), where each pair (Context, Question) is labeled as {True, False, Unknown}. We use the depth-5 subset, requiring multi-step inference, and follow prior work in sampling 600 balanced instances for evaluation. Additionally, we select instances from the OWA subset, categorized by reasoning depth (0–5 steps), to compare performance at different levels of reasoning depth.

FOLIO (Han et al., 2024) is an open-domain dataset designed to evaluate complex first-order logic reasoning in natural language. The problems align with real-world knowledge and feature highly natural linguistic expressions. For evaluation, we use the full test set of 204 examples, which require intricate logical inference.

LogiQA 2.0 (NLI) (Liu et al., 2023b) recasts the original LogiQA passages into a Natural-Language-Inference setting to probe complex logical reasoning. Given a premise drawn from a civil-service-exam passage and a hypothesis sentence, the task is to determine whether the hypothesis is *Entailed* or *Not Entailed*. A single premise can be paired with several alternative hypotheses, yielding a total of 3,840 premise-hypothesis pairs.

B Error Analysis

Model	Accuracy	Reasoning Errors (%)
GPT3.5-turbo	54.81	73.63
GPT4o-mini	61.97	88.45

Table 8: Model performance comparison.

In Table 8, we further analyze incorrectly predicted cases and measure the proportion that an LLM judge still identifies as having a correct memory component, which we classify as reasoning errors. This proportion reaches 73.63% for GPT-3.5-turbo and 88.45% for GPT-4o-mini. These findings suggest that most failures arise from errors in downstream reasoning rather than memory extraction, with only about 26% and 12% errors, respectively, attributable to clear memory extraction failures.

Error Type	GPT4o-mini	GPT3.5-turbo
Entity/ Protagonist	128	308
Intention	483	145
Normalization / Consistency	18	55
Time / Space / Intention	24	10
Cause / Causality	0	2
Negation	3	2

Table 9: Distribution of memory construction errors across categories.

We categorize SituW's failure cases into several error types. As summarized in Table 8, the most frequent errors for both GPT-3.5-turbo and GPT-4o-mini involve Intention and Entity/Protagonist, followed by Normalization/Consistency and mixed Time/Space/Intention errors, while Cause/-Causality and Negation errors are relatively rare. Section B.1 presents a representative Entity/Protagonist error.

B.1 Example: Entity/Role Confusion Error

```
[Context]
If an official from the government environmental
     protection department wants to speak at
    this non-governmental environmental
    protection seminar, he must be a member of a
     non-governmental environmental protection
    organization. However, the meeting
    stipulated that as a member of a non-
    governmental environmental protection
    organization, he is not allowed to be both
    an official of the government environmental
    protection department and a speaker at this
    non-governmental environmental protection
    seminar.

[Question]
If the above assertion is true, among the
    speakers at this seminar, there are no
    members of non-governmental environmental
    protection organizations.

[Gold label]
not-entailment

[Predicted]
entailment

[Protagonists (from memory)]
["official", "government environmental
    protection department", "non-governmental
    environmental protection seminar", "non-
    governmental environmental protection
    organization", "he"]
```

The error arises from conflating distinct roles, including speaker, official, and member of a non-governmental environmental protection organization, into a single undifferentiated protagonist set. This conflation causes the constraint that "a speaker who is both an official and a member is not allowed" to be misread as the broader claim that "no speaker is a member," thus producing an entailment prediction instead of the correct non-entailment label.

Examples of confusable pair identification questions from (Garnham, 1981).

In each set of sentences, a and b are the confusable pair and c and d are the non-confusable pair.

1. (a) The girl was given a complete pedicure at the chiropodist's.
   (b) The girl was given a complete pedicure by the chiropodist.
   (c) The girl had her handbag stolen at the chiropodist's.
   (d) The girl had her handbag stolen by the chiropodist.

2. (a) The clerk was going to have the wood cut up for him at the joiner's.
   (b) The clerk was going to have the wood cut up for him by the joiner.
   (c) The clerk was informed about the rail strike at the joiner's.
   (d) The clerk was informed about the rail strike by the joiner.

3. (a) The judge got his contact lenses from the optician.
   (b) The judge got his contact lenses in the optician's.
   (c) The judge answered a telephone call from the optician.
   (d) The judge answered a telephone call in the optician's.

4. (a) The leader purchased three loaves from the baker.
   (b) The leader purchased three loaves in the baker's.
   (c) The leader got a punch on the nose from the baker.
   (d) The leader got a punch on the nose in the baker's.

5. (a) The salesman was given two pounds for the coat at the pawnbroker's.
   (b) The salesman was given two pounds for the coat by the pawnbroker.
   (c) The salesman was bumped into on the step at the pawnbroker's.
   (d) The salesman was bumped into on the step by the pawnbroker.

6. (a) The teacher wanted her clock repaired at the watchmaker's.
   (b) The teacher wanted her clock repaired by the watchmaker.
   (c) The teacher was saved from falling down at the watchmaker's.
   (d) The teacher was saved from falling down by the watchmaker.

7. (a) The politician bought his wife a ring from the jeweller.
   (b) The politician bought his wife a ring in the jeweller's.
   (c) The politician received congratulations from the jeweller.
   (d) The politician received congratulations in the jeweller's.

8. (a) The bishop bought a new pipe from the tobacconist.
   (b) The bishop bought a new pipe in the tobacconist's.
   (c) The bishop heard an account of the state of the nation from the tobacconist.
   (d) The bishop heard an account of the state of the nation in the tobacconist's.

9. (a) The servant had a tooth removed at the dentist's.
   (b) The servant had a tooth removed by the dentist.
   (c) The servant was injured on the knee at the dentist's.
   (d) The servant was injured on the knee by the dentist.

10. (a) The student bought some spring cabbage from the greengrocer.
    (b) The student bought some spring cabbage in the greengrocer's.
    (c) The student heard about the concert from the greengrocer.
    (d) The student heard about the concert in the greengrocer's.

11. (a) The writer collected his manuscript from the publisher.

(b) The writer collected his manuscript in the publisher's.
(c) The writer got an invitation to a wedding from the publisher.
(d) The writer got an invitation to a wedding in the publisher's.

12. (a) The matron got three pork chops from the butcher.
(b) The matron got three pork chops in the butcher's.
(c) The matron heard an account of the cricket match from the butcher.
(d) The matron heard an account of the cricket match in the butcher's.

13. (a) The worker had her prescription made up at the chemist's.
(b) The worker had her prescription made up by the chemist.
(c) The worker was made to look foolish at the chemist's.
(d) The worker was made to look foolish by the chemist.

14. (a) The chairman bought a cup for the prize from the silversmith.
(b) The chairman bought a cup for the prize in the silversmith's.
(c) The chairman heard about the epidemic from the silversmith.
(d) The chairman heard about the epidemic in the silversmith's.

15. (a) The vicar went to have a suit fitted at the tailor's.
(b) The vicar went to have a suit fitted by the tailor.
(c) The vicar was given an old lampshade at the tailor's.
(d) The vicar was given an old lampshade by the tailor.

16. (a) The actress was given a shampoo and set at the hairdresser's.
(b) The actress was given a shampoo and set by the hairdresser.
(c) The actress's brother was discovered at the hairdresser's.
(d) The actress's brother was discovered by the hairdresser.

17. (a) The woman was given a thorough examination at the doctor's.
(b) The woman was given a thorough examination by the doctor.
(c) The woman was told about the new traffic scheme at the doctor's.
(d) The woman was told about the new traffic scheme by the doctor.

18. (a) The secretary bought some curtains from the draper.
(b) The secretary bought some curtains in the draper's.
(c) The secretary got an invitation to a party from the draper.
(d) The secretary got an invitation to a party in the draper's.

19. (a) The husband bought some sugar from the grocer.
(b) The husband bought some sugar in the grocer's.
(c) The husband received a message from the grocer.
(d) The husband received a message in the grocer's.

20. (a) The policeman had a bouquet made up at the florist's.
(b) The policeman had a bouquet made up by the florist.
(c) The policeman was attacked from behind at the florist's.
(d) The policeman was attacked from behind by the florist.

21. (a) The hostess bought a mink coat from the furrier.
(b) The hostess bought a mink coat in the furrier's.
(c) The hostess received a telegram from the furrier.
(d) The hostess received a telegram in the furrier's.

22. (a) The farmer bought his copy of "Farmer's Weekly" from the newsagent.
(b) The farmer bought his copy of "Farmer's Weekly" in the newsagent's.
(c) The farmer borrowed a pound from the newsagent.
(d) The farmer borrowed a pound in the newsagent's.

23. (a) The academic took a book to be bound at the bookbinder's.

(b) The academic took a book to be bound by the bookbinder.
(c) The academic was taken to an exhibition at the bookbinder's.
(d) The academic was taken to an exhibition by the bookbinder.

24. (a) The detective was advised not to go and inspect the house at the estate-agent's.
(b) The detective was advised not to go and inspect the house by the estate-agent.
(c) The detective was tipped off at the estate-agent's.
(d) The detective was tipped off by the estate-agent.

C Human Evaluation

We conducted the human evaluation after obtaining informed consent from all participants. The consent form explained that participation was voluntary and we would collect and analyze only anonymized evaluation data (i.e., no personally identifiable information was stored or used). Participants were recruited on a voluntary basis and were not financially compensated. The evaluation was then carried out using the interface shown in Figure 8.

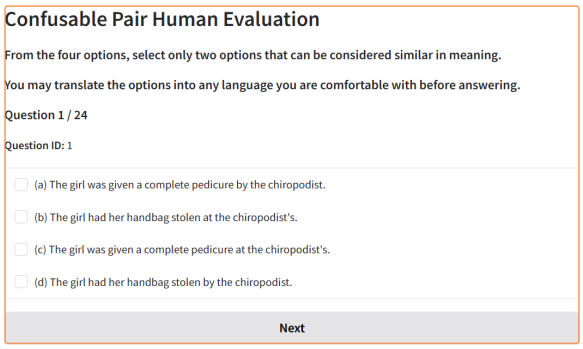

Figure 8: Interface for human evaluation.

D Additional Prompts

D.1 Prompt Used in the Memory Step

```
You are reading a logical context sentence-
by-sentence. Return ONLY a valid JSON object.
No extra text.

Given:
- sentence: the next sentence to read
- memory_statements: list of normalized
  propositions accumulated so far
- memory_protagonists: list of protagonists
  accumulated so far

Extract NEW information from the sentence:
1) time: explicit time expressions in the
   sentence; if none, ["none"]
2) space: explicit location expressions;
   if none, ["none"]
3) intention: explicit goal/purpose;
   if none, ["none"]
4) proposition: a short normalized propo-
   sition for the sentence
   - keep negation
   - keep quantifiers roughly
     (every/each/are/is)
5) protagonists_in_sentence: entities/kinds
   mentioned in the sentence
   (can include ones already in memory)

Output JSON schema:
{
  "time": [...],
  "space": [...],
  "intention": [...],
  "proposition": "...",
  "protagonists_in_sentence": [...],
  "confidence": 0.0-1.0
}

sentence:
[[SENTENCE]]

memory_statements:
[[MEM_STATEMENTS]]

memory_protagonists:
[[MEM_PROTAGONISTS]]
```

D.2 Prompt for Final Reasoning from Memory

```
You are a careful reasoner.
Use the MEMORY (accumulated statements +
reading log) to determine whether the
QUESTION's conclusion is:
- True: entailed by the MEMORY
- False: contradicted by the MEMORY
- Uncertain: neither entailed nor
  contradicted (insufficient information)

You must output exactly ONE of:
True, False, Uncertain.

Then end with exactly:
Final Answer: <True/False/Uncertain>

MEMORY_STATEMENTS:
[[STATEMENTS]]

READING_LOG:
[[READING]]

QUESTION (conclusion to evaluate):
[[QUESTION]]

OPTIONS:
[[OPTIONS]]
```