IE as Cache: Information Extraction Enhanced Agentic Reasoning

Hang Lv[1], Sheng Liang[2], Hongchao Gu[1], Wei Guo[2], Defu Lian[1], Yong Liu[2], Hao Wang[1], Enhong Chen[1*]
[1]University of Science and Technology of China, [2]Huawei Technologies Co., Ltd.

***Abstract*—Information Extraction aims to distill structured, decision-relevant information from unstructured text, serving as a foundation for downstream understanding and reasoning. However, it is traditionally treated merely as a terminal objective: once extracted, the resulting structure is often consumed in isolation rather than maintained and reused during multi-step inference. Moving beyond this, we propose *IE-as-Cache*, a framework that repurposes IE as a cognitive cache to enhance agentic reasoning. Drawing inspiration from hierarchical computer memory, our approach combines query-driven extraction with cache-aware reasoning to dynamically maintain compact intermediate information and filter noise. Experiments on challenging benchmarks across diverse LLMs demonstrate significant improvements in reasoning accuracy, indicating that IE can be effectively repurposed as a reusable cognitive resource and offering a promising direction for future research on downstream uses of IE.**

***Index Terms*—large language model, information extraction, LLM-based agent, reasoning.**

I. Introduction

In the landscape of web search and natural language understanding, Information Extraction (IE) stands as a pivotal technique for parsing the vast ocean of unstructured digital content into structured knowledge [1]–[3] such as entities [4], [5], relations [6], [7], and events [8], [9]. Despite its practical importance, contemporary research predominantly treats IE as an end task—optimizing extraction quality as the final objective—rather than exploring IE as a reusable infrastructure that can be continuously consumed and updated by downstream cognitive processes [10]. In this work, we posit that strategically extracted and maintained information can directly scaffold large language model (LLM) reasoning, especially as LLMs increasingly operate in an agentic manner where they must iteratively read, decide, and act over complex inputs [11].

This perspective is crucial for processing noise-rich, long-form content [12], where LLMs struggle with irrelevant distractors [13] and information decay in middle contexts [14]. In such settings, simply providing raw context is often insufficient; instead, models benefit from a compact intermediate representation that preserves salient evidence while suppressing noise [15]. Addressing these inefficiencies, we draw inspiration from hierarchical computer memory [16] to introduce the cognitive analogy in Figure 1. Just as CPU caches maintain critical data for rapid retrieval, we hypothesize that IE can function as a dynamic memory layer that filters noise and preserves semantically salient content for reasoning.

*Corresponding author: cheneh@ustc.edu.cn

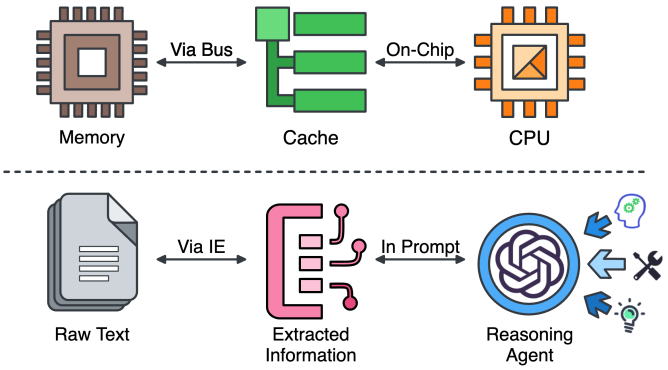

Fig. 1. Cognitive analogy between hierarchical computer memory and text reasoning. Information Extraction functions as a bridge between raw text and the reasoning agent, mirroring the role of a hardware cache.

This analogy leads to our central research question: ***Can IE serve as a cognitive cache to scaffold complex reasoning processes?*** To explore this hypothesis, we propose the *IE-as-Cache* framework, which consists of two coordinated components: *Query-Driven Information Extraction* and *Cache-aware Agentic Reasoning*. Given a user request, the framework first extracts query-relevant content to initialize a compact, structured cache. This cache serves as the primary working memory, enabling the agent to operate on a noise-reduced representation rather than repeatedly rescanning the full text. During the subsequent reasoning process, the model dynamically accesses raw data on-demand to update the cache with new evidence, while strictly controlling redundancy. Our experiments demonstrate the framework's effectiveness on challenging benchmarks, highlighting the potential of IE to significantly improve LLM reasoning as a reusable cognitive resource. To sum up, our contributions are threefold:

- We propose a new viewpoint that repurposes information extraction from an endpoint task to an explicit cognitive cache that actively scaffolds agentic reasoning.
- we develop *IE-as-Cache*, a noval framework that integrates query-driven extraction with cache-aware reasoning and dynamic cache updates for noise-robust inference.
- we conduct extensive experiments across diverse LLMs and three representative task families (question answering, planning, and query-focused summarization), demonstrating consistent improvements in reasoning accuracy.

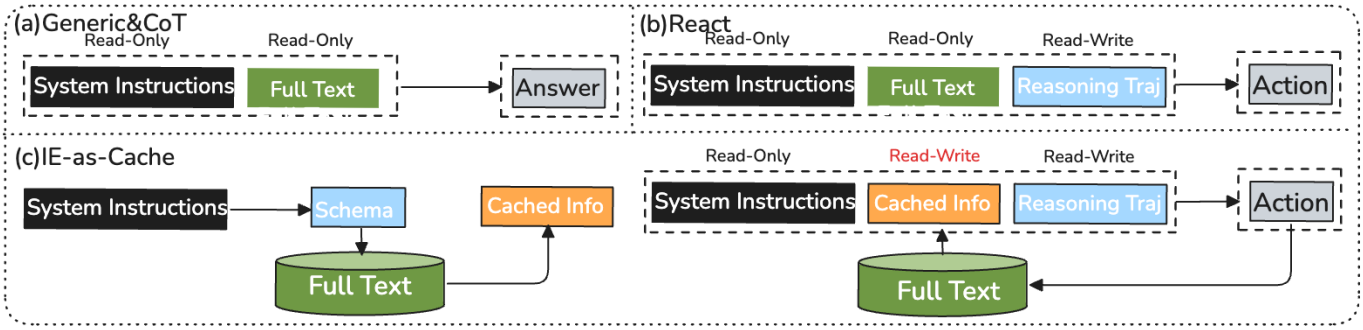

Fig. 2. Comparison of context mechanisms. Unlike (a) and (b) which treat raw text as a static read-only block, (c) IE-as-Cache introduces a dynamic Read-Write Cache. This mechanism filters noise by keeping the full text external and only maintaining semantically critical information in the active memory.

II. Related Work

*a) **Information Extraction as Foundational Infrastructure**:* The evolution of information extraction (IE) has progressed from task-specific solutions to more unified frameworks. Early work laid the foundation with core tasks like Named Entity Recognition [4], Relation Extraction [6], and Event Extraction [8], each requiring specialized architectures and limited to specific domains. The paradigm shift came with Unified Information Extraction [17], which demonstrated that schema-guided prompting could harmonize these diverse extraction tasks into structured templates, substantially reducing task fragmentation. However, despite this unification, these approaches have predominantly treated IE as an *end task*, emphasizing extraction quality as the final deliverable, which limits its use as a *reusable cognitive resource* for ongoing reasoning and multi-step cognitive processes.

*b) **Operationalizing IE for Knowledge Systems**:* While traditional research has used IE to construct knowledge graphs and extract structured information for specific tasks [18], [19], its role in supporting dynamic, multi-step reasoning in large language models (LLMs) remains underexplored [20]. Some efforts leverage IE outputs for domain-specific applications, such as scientific knowledge discovery and fact-based search [21], [22], organizing extracted facts as relatively stable artifacts for downstream querying. However, these pipelines typically treat IE as a one-time process, and the extracted structures are rarely maintained or updated for iterative reasoning, limiting their flexibility and scalability for general-purpose LLM reasoning [23].

*c) **Structured Reasoning Architectures**:* Recent advances in prompting highlight the critical role of structured information in enhancing LLM reasoning capabilities. Studies show that *template structures* can significantly affect reasoning fidelity [24]. The Chain-of-Table paradigm [25] leverages tabular representations for numerical reasoning, while [26] formalizes *table-structured thought* for multi-task generalization. However, these methods largely emphasize *output formatting* and intermediate traces, leaving a gap in optimizing the *input*. Our work addresses this gap by utilizing IE as a cognitive cache to maintain a compact, noise-resistant representation that can be retrieved and updated during multi-step reasoning.

III. IE as Cache

In this section, we introduce *IE-as-Cache*, a framework that repurposes Information Extraction (IE) as a cognitive cache for agentic reasoning. Given a user query Q and a raw text source T, the goal is to produce a final answer A while reducing the model's reliance on repeatedly reading long, noise-rich inputs. The key idea is to treat the raw text as external storage and maintain a compact, read-write cache C inside the agent's working context. This framework consists of two coordinated components: *Query-Driven Information Extraction* for initializing the cache, and *Cache-Aware Agentic Reasoning* for using and updating it during multi-step inference. Algorithm 1 summarizes the overall procedure.

A. Query-Driven Information Extraction

Effective cache initialization requires the extraction targets to adapt to the user's intent. Conventional IE pipelines typically rely on fixed schemas (e.g., predefined entity/relation sets). Such fixed schemas are often mismatched with open-domain reasoning, where the appropriate extraction granularity depends on the semantics of the query [27], [28]. We therefore consider two candidates for query-conditioned extraction.

Paradigm 1: Monolithic Extraction

$$E = \mathcal{F}_{\text{LLM}}(Q, T), \tag{1}$$

where an LLM directly maps the query Q and raw text T to extractions E. While simple, this formulation entangles *schema identification* (what to extract) with *content extraction* (which facts to extract), often leading to noisy, inconsistent, or overly specific structures [29].

Paradigm 2: Schema-Decoupled Extraction

$$S = \mathcal{F}_{\text{LLM}}^{\text{schema}}(Q), \tag{2}$$

$$E = \mathcal{F}_{\text{LLM}}^{\text{extract}}(Q, S, T), \tag{3}$$

where we first generate a query-conditioned extraction schema S, and then perform schema-guided extraction to obtain E. Intuitively, S specifies the extraction structure aligned with the query (e.g., slot names or table columns), and the schema-guided extractor focuses the model on extracting query-relevant content from T under this structure. In practice, we

represent E as a compact set of structured entries (e.g., a table-like record set), which is more suitable as an intermediate cache than free-form text.

Our preliminary experiments showed that this two-stage approach significantly outperforms monolithic extraction, so we adopt it for initializing the cache. We then form the agent's initial prompt by combining the user query Q with the extracted structure E, which serves as the initial cache for subsequent reasoning.

B. Cache-Aware Agentic Reasoning

While §III-A provides an initial cache, the core of *IE-as-Cache* is how an agent reasons with C and maintains it dynamically. As illustrated in Figure 2, we shift from a static context paradigm to a cache-centric design.

*1) **Architectural Comparison: Standard agent vs. IE-as-Cache**:* The fundamental distinction between our approach and standard reasoning agents (e.g., ReAct [30]) lies in how the context window is allocated.

Standard ReAct Architecture (Figure 2b): A standard agent places the raw text T into the context window as a read-only block and appends observations and reasoning traces during interaction. For long-form inputs, this design introduces substantial redundancy and distractors, and the agent repeatedly scans the raw text at each step. As a result, the working context is dominated by uncompressed input, and the dynamic component is primarily the reasoning trajectory.

IE-as-Cache Architecture (Figure 2c): We instead treat the full text T as external storage and keep a compact, structured cache C as the primary read-write object in working memory. The agent reasons over C, and only accesses T on demand. Importantly, the cache is not a static summary: it is explicitly maintained and updated as reasoning progresses, preserving only semantically salient, query-relevant information in the agent's immediate focus.

*2) **Reason–Act Interface**:* At each step, the agent produces a reasoning trace and, when needed, triggers an action:

$$(r, a) \leftarrow \mathcal{F}_{\text{LLM}}^{\text{reason}}(Q, C). \tag{4}$$

We implement `seek_information` as a lightweight action within a standard agentic loop, enabling step-specific evidence acquisition and cache maintenance with minimal engineering changes. The overall workflow remains compatible with existing ReAct-style agents (Algorithm 1).

*3) **On-Demand Extraction as Memory Access**:* When the agent issues $\texttt{seek\_information}(q_t)$, it does not retrieve and append raw snippets. Instead, we reuse the schema-decoupled extractor to perform *on-demand extraction* over the raw text:

$$E' \leftarrow \mathcal{F}_{\text{LLM}}^{\text{extract}}(Q, S, T, q_t), \tag{5}$$

where q_t provides a step-specific focus indicating what information should be extracted to resolve the current uncertainty. This design aligns with the cache analogy: the agent accesses external storage and reads back only structured, query-relevant content rather than unstructured observations.

Algorithm 1 IE-as-Cache Framework

Require: Query Q, raw text T, max steps H
Ensure: Answer A

Query-Driven Extraction (Initialize Cache)
$S \leftarrow \mathcal{F}_{\text{LLM}}^{\text{schema}}(Q)$ ▷ query-conditioned schema
$E \leftarrow \mathcal{F}_{\text{LLM}}^{\text{extract}}(Q, S, T)$ ▷ schema-guided extraction
$C \leftarrow \texttt{InitCache}(Q, E)$
Cache-Aware Agentic Reasoning
$t \leftarrow 0$
while $t < H$ **do**
  $(r, a) \leftarrow \mathcal{F}_{\text{LLM}}^{\text{reason}}(Q, C)$
  if $a = \texttt{final}(A)$ **then**
    return A
  end if
  if $a = \texttt{seek\_information}(q_t)$ **then**
    $E' \leftarrow \mathcal{F}_{\text{LLM}}^{\text{extract}}(Q, S, T, q_t)$ ▷ on-demand extraction
    $C \leftarrow \mathcal{F}_{\text{LLM}}^{\text{update}}(Q, S, C, E')$ ▷ merge/prune
    if $\texttt{NeedCheck}(r, C)$ **then** ▷ optional
      $C \leftarrow \mathcal{F}_{\text{LLM}}^{\text{check}}(Q, S, C, r)$ ▷ self-check
    end if
  end if
  $t \leftarrow t + 1$
end while
$A \leftarrow \mathcal{F}_{\text{LLM}}^{\text{answer}}(Q, C)$ ▷ fallback at step limit
return A

*4) **Cache Update: Action → Cache Maintenance**:* After obtaining the newly extracted E', we update the cache via

$$C \leftarrow \mathcal{F}_{\text{LLM}}^{\text{update}}(Q, S, C, E'). \tag{6}$$

The update operator synthesizes newly extracted content with existing cache entries. To keep this step lightweight, we perform cache maintenance in a single LLM call: the model (i) merges compatible information into existing entries when appropriate, (ii) removes redundant or off-target content, and (iii) prunes low-utility details to keep the cache compact. This maintenance step is the key difference from standard agents: it replaces "Action → Observation" with an explicit "Action → Cache Update" cycle, so the agent reasons over a compact, noise-resistant representation throughout multi-step inference.

*5) **Self-Check & Fallback**:* Optionally, the agent performs a lightweight self-check to revise the cache or refine the next information request, without changing the main loop (Algorithm 1). The reasoning process terminates when the agent outputs $\texttt{final}(A)$ or reaches a maximum of H steps; if the step limit is reached, we generate the answer from the current cache as a fallback:

$$A \leftarrow \mathcal{F}_{\text{LLM}}^{\text{answer}}(Q, C). \tag{7}$$

Overall, *IE-as-Cache* turns information extraction into a lightweight, dynamically maintained cache that keeps only query-relevant structure in the working context and updates it on demand, helping agents reason over long, noise-rich text without repeatedly re-reading the full input and reducing distraction during multi-step inference.

IV. EXPERIMENT

A. Experiment Setup

*a) **Tasks**:* To rigorously evaluate generalizability, we focused on benchmarks involving lengthy, unstructured raw text that demand complex aggregative reasoning. Our selection criteria strictly target scenarios where information must be actively synthesized from noise-rich contexts, moving beyond the simple span retrieval or pre-structured inputs typical of standard benchmarks. Based on this, we employed:

- **Logical QA (TACT [31]):** A dataset requiring logical deduction over long-form text. It serves as a testbed for the agent's ability to aggregate scattered evidence and perform multi-hop reasoning without pre-defined schemas or tables.
- **Agentic Planning (Calendar Scheduling [32]):** A complex constraint satisfaction task where the agent must manage and resolve conflicting schedules derived purely from raw natural language descriptions, simulating personal assistant scenarios.
- **Query-Focused Summarization (QMSUM [33]):** A dataset for summarizing specific spans from extensive multi-turn meeting transcripts based on user queries. This tests the framework's capacity to distill and synthesize key information from high-noise dialogue environments.

This diverse suite allows us to verify our method's effectiveness across **Question Answering**, **Planning**, and **Summarization** tasks within noise-rich contexts.

*b) **Models**:* We verify our framework across a diverse spectrum of model families and parameter scales. Our selection includes two state-of-the-art proprietary models, `GPT-4o` and `GPT-4o-mini` , alongside three open-source models catering to different computational constraints: the `LLaMA3.1-8B-Instruct` [34] and the `Qwen-3` series, ranging from the mid-sized `8B` to the lightweight `0.6B` [35].

*c) **Evaluation Metrics**:* We follow the standard evaluation protocols of each benchmark. For TACT and Calendar Scheduling, we report Exact Match (EM), which directly measures whether the produced answer satisfies the target decision/constraints. For QMSUM, we report `ROUGE-1` [36], reflecting query-relevant content coverage in the generated summary. In addition to end-task performance, we further analyze the intermediate extraction quality on TACT using ROUGE and Sentence-T5 similarity [37] against the provided ground-truth references (Table II), enabling a direct connection between cache quality and final reasoning accuracy.

*d) **Baselines**:* To comprehensively evaluate the IE-as-Cache framework, we benchmark it against four established paradigms:

- **Generic (Standard Prompting):** The model receives the instruction and full raw context, generating the answer directly without intermediate steps.
- **Chain of Thought (CoT) [38]:** A reasoning-enhanced baseline that encourages the model to generate explicit reasoning traces over the raw text ("Let's think step by step") before deriving the final answer.

TABLE I
MAIN EXPERIMENTAL RESULTS ACROSS THREE DATASETS. WE REPORT EXACT MATCH SCORE FOR TACT AND CALENDAR, AND ROUGE-1 SCORE FOR QMSUM. BEST RESULTS ARE MARKED IN **BOLD**.

Data	Method	GPT-4o	4o-mini	Llama8B	Qwen8B	Qwen.6B
TACT	Generic	42.74	32.26	14.52	25.81	6.45
	CoT	52.42	37.90	25.00	45.97	15.32
	React	57.26	52.42	27.42	57.26	12.90
	IE as Tool	61.29	45.16	31.45	26.29	12.10
	IE as Cache	**71.77**	**58.87**	**38.71**	**62.90**	**33.06**
	w/o update	65.32	56.45	37.10	62.90	25.81
Calendar	Generic	49.10	27.20	12.40	9.50	4.10
	CoT	56.40	33.30	**18.00**	13.50	8.20
	React	57.00	30.90	8.50	10.50	8.50
	IE as Cache	**65.20**	**38.40**	15.70	**17.90**	**10.10**
QMSUM	Generic	32.03	32.92	28.76	31.82	27.12
	CoT	31.70	31.53	27.54	29.30	28.44
	React	29.38	29.59	24.27	27.61	28.64
	IE as Cache	**35.21**	**34.80**	**32.14**	**33.20**	**30.67**

- **ReAct [30]:** An agentic baseline where the model interleaves reasoning traces with action execution. To ensure a fair comparison, we designed task-specific action spaces tailored to the unique characteristics of each dataset to facilitate effective interaction with the environment.
- **IE-as-Tool [31]:** A specialized pipeline that first converts text into a structured table and then executes Pandas queries. *Note:* Due to its strict dependency on tabularizable logic, this baseline is exclusively applied to the TACT dataset, as it is incompatible with the unstructured, dynamic nature of Calendar Scheduling and QMSUM.

*e) **Experiment Setup**:* Unless otherwise specified, we use greedy decoding with temperature set to 0 for all models. For proprietary models (e.g., `GPT-4o` and `GPT-4o-mini`), we run each setting three times and report the average performance to account for residual nondeterminism in serving. For open-source models, we fix the random seed and use the same inference configuration across all methods.

B. Main Results

Table I presents the comprehensive performance comparison across three diverse datasets. Overall, the proposed *IE-as-Cache* framework achieves state-of-the-art performance in most experimental settings, consistently outperforming both standard prompting baselines (Generic, CoT) and agentic baselines (ReAct, IE-as-Tool) across model scales ranging from 0.6B to huge GPT-4o.

*a) **Overall Trends**:* Across all three task families, *IE-as-Cache* shows the most consistent gains when inputs are long and noisy, and when reasoning requires aggregating multiple cues. These improvements are particularly significant for smaller models, indicating that an explicit cache helps mitigate limited robustness in long-context reasoning by concentrating key information. We also observe that purely agentic interaction (ReAct) is not always beneficial: frequent context switching, driven by raw text and incremental observations, can introduce noise, whereas a compact cache offers a more stable representation for multi-step reasoning.

TABLE II
IMPACT OF EXTRACTION QUALITY ON FINAL REASONING ACCURACY (EM) ON THE TACT DATASET (GPT-4O). *w/ gold schema* REPRESENTS AN ORACLE SETTING USING GROUND-TRUTH EXTRACTION SCHEMAS. SIMILARITY IS CALCULATED VIA SENTENCE-T5-LARGE.

Method	ROUGE-1	ROUGE-L	Similarity	EM
IE as Tool	27.68	21.65	84.90	61.29
IE as Cache	31.04	24.27	87.11	71.77
w/ gold schema	50.35	39.55	91.97	73.39

*b) **Performance on Logical QA**:* On the TACT dataset, which features long contexts filled with irrelevant distractors, our framework demonstrates the most significant improvements. With the strong `GPT-4o` backbone, *IE-as-Cache* achieves an EM score of **71.77**, surpassing the specialized baseline `IE-as-Tool` (61.29) by a substantial margin of over 10 points. Notably, this advantage is even more pronounced in smaller open-source models. For instance, on `Llama3.1-8B`, our method improves performance from 14.52 (Generic) and 27.42 (ReAct) to **38.71**. This indicates that the dynamic cache mechanism effectively filters out noise that typically confuses LLMs in long-context logical reasoning, a capability that rigid pipeline approaches fail to match.

*c) **Performance on Agentic Planning**:* For the Calendar Scheduling task, which requires strict constraint satisfaction, *IE-as-Cache* continues to lead, particularly with capable proprietary models. Using `GPT-4o`, our method achieves **65.20** EM, significantly outperforming ReAct (57.00). This suggests that maintaining a high-density cache of "scheduled events" helps the agent better detect conflicts than simply reading raw descriptions. We observe that smaller models (e.g., `Qwen3-0.6B`, `Llama3.1-8B`) struggle significantly with this task across all methods due to the inherent difficulty of reasoning with multiple constraints. However, even in these low-resource scenarios, our framework generally maintains a performance edge (e.g., 17.90 vs 13.50 for `Qwen3-8B`). *Note:* One exception is observed with `Llama-3.1`, where CoT slightly outperforms our method. We attribute this to the model's specific sensitivity to the prompt structure in zero-shot extraction scenarios.

*d) **Performance on Summarization**:* In the query-focused summarization task involving multi-turn meeting transcripts, *IE-as-Cache* proves to be the most robust approach. Interestingly, we observed that the standard `ReAct` agent often underperforms compared to simple `Generic` or `CoT` baselines on this dataset (e.g., ReAct 29.38 vs Generic 32.03 on `GPT-4o`). This is likely because ReAct's frequent context switching fragments the narrative flow of the meeting. In contrast, our framework consistently improves upon the baselines across all models (e.g., reaching **35.21** with `GPT-4o` and **33.20** with `Qwen3-8B`). This confirms that our "Action $\rightarrow$ Extraction $\rightarrow$ Cache Update" cycle successfully distills scattered dialogue turns into a coherent representation without losing the global context required for summarization.

C. Ablation and Further Analysis

To identify the sources of our performance gains, we conduct an ablation study on the cache update mechanism and a deeper analysis of the intermediate extraction quality.

*a) **Impact of Dynamic Cache Update**:* We first examine the necessity of the *Write-Through Cache Update* mechanism. As shown in the "*w/o update*" row of Table I, removing the agent's ability to dynamically refresh the cache results in a consistent performance drop across all models. For instance, on TACT with GPT-4o, the EM score declines from **71.77** to **65.32**. This degradation highlights that static extraction is insufficient for multi-hop reasoning. Complex tasks require an iterative process where the agent updates its information state based on intermediate reasoning results, confirming that our dynamic maintenance of the cognitive cache is essential for handling evolving contexts.

*b) **Correlation between Extraction and Reasoning**:* To further understand *why* IE-as-Cache outperforms the baselines, we analyze the quality of the intermediate extracted content on the TACT dataset (Table II). We benchmark our extracted information against the ground-truth tabular references provided in TACT, utilizing ROUGE scores to measure lexical overlap and Sentence-T5-Large to quantify semantic similarity.

We observe a positive correlation between extraction quality and final reasoning accuracy. *IE-as-Cache* achieves higher semantic similarity (87.11) compared to *IE-as-Tool* (84.90), which directly translates to a +10.48% improvement in final EM. This confirms that preserving semantically critical information is a prerequisite for accurate downstream reasoning.

*c) **Efficiency of Dynamic Schema**:* We also introduced an oracle setting, "*w/ gold schema*", where extraction is guided by ground-truth schemas. Crucially, the results reveal a diminishing return: while the oracle setting yields significantly higher lexical overlap (ROUGE-1: 50.35 vs 31.04), the gain in final reasoning accuracy is remarkably marginal (EM: 73.39 vs 71.77). This suggests that our *IE-as-Cache* framework is highly efficient. Even without the perfect lexical matching of the gold schema, our query-driven dynamic extraction successfully captures the vast majority of *reasoning-critical* information. It constructs a "sufficiently good" cognitive cache that supports near-optimal reasoning performance, demonstrating the robustness of our generated extraction plans.

V. CONCLUSION

This paper repurposes Information Extraction (IE) from a static terminal objective to a dynamic *cognitive infrastructure* for LLMs. We introduce the *IE-as-Cache* framework, which manages extracted information as a read-write memory layer, effectively mitigating noise and context decay during multi-step reasoning. Our findings confirm that this mechanism significantly enhances performance across diverse benchmarks, improving both reasoning accuracy and efficiency. Ultimately, this study demonstrates that IE can be effectively utilized as a reusable cognitive resource, offering a promising direction for future research in the downstream applications of information extraction, particularly in noise-rich and unstructured contexts.

VI. Acknowledgement

This work was supported by the National Natural Science Foundation of China (Nos. U23A20319 and 62441239), as well as the Anhui Province Science and Technology Innovation Project (Nos. 202423k09020010 and 202423k09020011).

References

[1] R. Gaizauskas and Y. Wilks, "Information extraction: Beyond document retrieval," *Journal of documentation*, vol. 54, no. 1, pp. 70–105, 1998.

[2] S. Sarawagi *et al.*, "Information extraction," *Foundations and Trends® in Databases*, vol. 1, no. 3, pp. 261–377, 2008.

[3] A. McCallum, "Information extraction: Distilling structured data from unstructured text," *Queue*, vol. 3, no. 9, pp. 48–57, 2005.

[4] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," in *Named Entities: Recognition, classification and use*. John Benjamins publishing company, 2009, pp. 3–28.

[5] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE transactions on knowledge and data engineering*, vol. 34, no. 1, pp. 50–70, 2020.

[6] S. Pawar, G. K. Palshikar, and P. Bhattacharyya, "Relation extraction: A survey," *arXiv preprint arXiv:1712.05191*, 2017.

[7] Q. Zhang, M. Chen, and L. Liu, "A review on entity relation extraction," in *2017 second international conference on mechanical, control and computer engineering (ICMCCE)*. IEEE, 2017, pp. 178–183.

[8] W. Xiang and B. Wang, "A survey of event extraction from text," *IEEE Access*, vol. 7, pp. 173 111–173 137, 2019.

[9] J. Liu, Y. Chen, K. Liu, W. Bi, and X. Liu, "Event extraction as machine reading comprehension," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020.

[10] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, and E. Chen, "Large language models for generative information extraction: A survey," 2024.

[11] J. Wu, J. Zhu, and Y. Liu, "Agentic reasoning: Reasoning llms with tools for the deep research," 2025. [Online]. Available: https://arxiv.org/abs/2502.04644

[12] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler, "Long range arena: A benchmark for efficient transformers," 2020. [Online]. Available: https://arxiv.org/abs/2011.04006

[13] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. Chi, N. Schärli, and D. Zhou, "Large language models can be easily distracted by irrelevant context," 2023. [Online]. Available: https://arxiv.org/abs/2302.00093

[14] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," 2023. [Online]. Available: https://arxiv.org/abs/2307.03172

[15] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, X. Sun, L. Li, and Z. Sui, "A survey on in-context learning," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024.

[16] C. Packer, S. Wooders, K. Lin, V. Fang, S. G. Patil, I. Stoica, and J. E. Gonzalez, "Memgpt: Towards llms as operating systems," 2024. [Online]. Available: https://arxiv.org/abs/2310.08560

[17] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, and H. Wu, "Unified structure generation for universal information extraction," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022.

[18] B. Zhang and H. Soh, "Extract, define, canonicalize: An llm-based framework for knowledge graph construction," 2024. [Online]. Available: https://arxiv.org/abs/2404.03868

[19] X. Zou, "A survey on application of knowledge graph," in *Journal of Physics: Conference Series*, vol. 1487, no. 1. IOP Publishing, 2020, p. 012016.

[20] F. Xu, Q. Hao, Z. Zong, J. Wang, Y. Zhang, J. Wang, X. Lan, J. Gong, T. Ouyang, F. Meng *et al.*, "Towards large reasoning models: A survey of reinforced reasoning with large language models," *arXiv preprint arXiv:2501.09686*, 2025.

[21] Z. Song, G.-Y. Hwang, X. Zhang, S. Huang, and B.-K. Park, "A scientific-article key-insight extraction system based on multi-actor of fine-tuned open-source large language models," *Scientific Reports*, vol. 15, no. 1, p. 1608, 2025.

[22] N. Zhang, X. Xu, L. Tao, H. Yu, H. Ye, S. Qiao, X. Xie, X. Chen, Z. Li, and L. Li, "DeepKE: A deep learning based knowledge extraction toolkit for knowledge base population," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, W. Che and E. Shutova, Eds. Abu Dhabi, UAE: Association for Computational Linguistics, Dec. 2022.

[23] D.-H. Zhu, Y.-J. Xiong, J.-C. Zhang, X.-J. Xie, and C.-M. Xia, "Understanding before reasoning: Enhancing chain-of-thought with iterative summarization pre-prompting," 2025. [Online]. Available: https://arxiv.org/abs/2501.04341

[24] J. He, M. Rungta, D. Koleczek, A. Sekhon, F. X. Wang, and S. Hasan, "Does prompt formatting have any impact on llm performance?" 2024. [Online]. Available: https://arxiv.org/abs/2411.10541

[25] Z. Wang, H. Zhang, C.-L. Li, J. M. Eisenschlos, V. Perot, Z. Wang, L. Miculicich, Y. Fujii, J. Shang, C.-Y. Lee, and T. Pfister, "Chain-of-table: Evolving tables in the reasoning chain for table understanding," 2024. [Online]. Available: https://arxiv.org/abs/2401.04398

[26] Z. Sun, N. Deng, H. Yu, and J. You, "Table as thought: Exploring structured thoughts in llm reasoning," 2025. [Online]. Available: https://arxiv.org/abs/2501.02152

[27] Y. Jiao, M. Zhong, S. Li, R. Zhao, S. Ouyang, H. Ji, and J. Han, "Instruct and extract: Instruction tuning for on-demand information extraction," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023.

[28] S. Sekine, "On-demand information extraction," in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 2006, pp. 731–738.

[29] S. Liang, H. Lv, Z. Wen, Y. Wu, Y. Zhang, H. Wang, and Y. Liu, "Adaptive schema-aware event extraction with retrieval-augmented generation," 2025. [Online]. Available: https://arxiv.org/abs/2505.08690

[30] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," 2023. [Online]. Available: https://arxiv.org/abs/2210.03629

[31] A. Caciularu, A. Jacovi, E. B. David, S. Goldshtein, T. Schuster, J. Herzig, G. Elidan, and A. Globerson, "Tact: Advancing complex aggregative reasoning with information extraction tools," 2024.

[32] H. S. Zheng, S. Mishra, H. Zhang, X. Chen, M. Chen, A. Nova, L. Hou, H.-T. Cheng, Q. V. Le, E. H. Chi, and D. Zhou, "Natural plan: Benchmarking llms on natural language planning," 2024. [Online]. Available: https://arxiv.org/abs/2406.04520

[33] M. Zhong, D. Yin, T. Yu, A. Zaidi, M. Mutuma, R. Jha, A. H. Awadallah, A. Celikyilmaz, Y. Liu, X. Qiu, and D. Radev, "Qmsum: A new benchmark for query-based multi-domain meeting summarization," 2021. [Online]. Available: https://arxiv.org/abs/2104.05938

[34] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, and A. M. et al., "The llama 3 herd of models," 2024.

[35] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, "Qwen3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2505.09388

[36] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[37] J. Ni, G. Hernandez Abrego, N. Constant, J. Ma, K. Hall, D. Cer, and Y. Yang, "Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models," in *Findings of the Association for Computational Linguistics: ACL 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022.

[38] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023. [Online]. Available: https://arxiv.org/abs/2201.11903