THOUGHT-RETRIEVER: DON'T JUST RETRIEVE RAW DATA, RETRIEVE THOUGHTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have transformed AI research thanks to their powerful *internal* capabilities and knowledge. However, existing LLMs still fail to effectively incorporate the massive *external* knowledge when interacting with the world. Although retrieval-augmented LLMs are proposed to mitigate the issue, they are still fundamentally constrained by the context length of LLMs, as they can only retrieve top-K raw data chunks from the external knowledge base which often consists of millions of data chunks. Here we propose *Thought-Retriever*, a novel model-agnostic algorithm that helps LLMs generate output conditioned on arbitrarily long external data, without being constrained by the context length or number of retrieved data chunks. Our key insight is to let an LLM fully leverage its intermediate thoughts generated when solving past user queries, organizing them in thought memory, and retrieving the relevant thoughts when addressing new queries. Notably, Thought-Retriever can *self-evolve* through continuous user interactions thanks to the growing number and depth of thoughts. Besides algorithmic innovation, we further meticulously prepare a novel benchmark, AcademicEval, which requires an LLM to faithfully leverage ultra-long context to answer queries based on real-world academic papers. Extensive experiments on AcademicEval and two other datasets validate that Thought-Retriever remarkably outperforms state-of-the-art baselines by achieving a 5%-45% higher win rate. More importantly, we further demonstrate 2 exciting findings: (1) Thought-Retriever can indeed help LLM self-evolve after solving more user queries; (2) Thought-Retriever learns to leverage deeper thoughts to answer more abstract user queries.

1 INTRODUCTION AND RELATED WORK

Large language models (LLMs) have revolutionized AI research thanks to their powerful *internal* capabilities Zhao et al. (2023); Wang et al. (2023) and knowledge Peng et al. (2023a), which presents a promising future for building autonomous AI agents. When building LLM agents, researchers further expect LLMs to interact with the world by effectively incorporating the *external knowledge* as their long-term memories, *e.g.*, collected from *facts* Sun et al. (2023) or interactions with *other AI agents* Wu et al. (2023); Kannan et al. (2023). Importantly, the scale of the external knowledge for LLM agents could be arbitrarily large; ultimately, all the digitized information within our universe could serve as the external knowledge for these agents. In practice, when building personalized LLM applications Bill & Eriksson (2023) or LLM-powered domain experts Thirunavukarasu et al. (2023); Liu et al. (2023), *e.g.*, AI doctor, the relevant external knowledge for the LLMs could also easily get extremely large, *e.g.*, billions of tokens. Therefore, our paper aims to raise attention to the pressing research question: *how to effectively and efficiently help LLMs utilize (arbitrarily) rich external knowledge*.

To help LLMs better incorporate external knowledge, existing research mainly falls into two categories: *long-context LLMs* and *retrieval-augmented LLMs (RALMs)*. (1) *Long-context LLMs*, such as MPT MosaicML (2023) and LongChat LM-SYS (2023), aims to expand the LLM's context window, *e.g.*, via novel training algorithms Tay et al. (2022), inference algorithms Xiao et al. (2023), new architectures Peng et al. (2023b); Gu & Dao (2023), or system optimization Xu et al. (2023). Although these methods improve the working memory size of LLM agents, they cannot fundamentally address the issue of interacting with ultra-rich external knowledge using LLM agents, since the

computational complexity is often quadratic to the context length. (2) *RALMs* retrieve pertinent information from external knowledge bases using retrievers, such as BM-25 Robertson et al. (2009), Contriever Izacard et al. (2022), and DRAGON Lin et al. (2023). However, these algorithms are still constrained by LLMs' context length, since they can only retrieve top-K raw data chunks from the external knowledge that fits within an LLM's context limit. (3) *Hierarchical RALMs*, *e.g.*, creating a tree-structured memory for an LLM agent Chen et al. (2023). Despite its potential to help LLMs incorporate more abstract knowledge, the tree construction requires prohibitively significant LLM inference costs when the memory size is large; the constructed tree-structured memory is also rigid, failing to adapt to the specific input of an LLM. Overall, existing methods in attempting to include external knowledge for LLMs still exhibit *fundamental limitations in efficiency and effectiveness*. More discussions about related works can be seen in Appendix A.

Psychological studies Kurzweil (2013); Snell (2012) reveal that human memory is organized hierarchically, which not only aids in retrieving relevant information for problem-solving but also gradually deepens our understanding of the world through continuous processing and summarizing these interactions into complex cognitive thoughts.

Here, we propose *Thought-Retriever*, an LLM-agnostic self-evolving retrieval framework that leverages historical LLM responses to answer new queries. Our key insight is that LLM responses can be transformed into *thoughts* with little computational overhead, and that the thoughts can be organized as a thought memory for the agent to facilitate future tasks. Notably, through continuously interacting with diverse user queries, Thought-Retriever gradually generates more novel thoughts with a larger receptive field, since new data chunks from the external knowledge are incorporated to the thought memory after answering each new query. Therefore, Thought-Retriever gives an LLM agent the potential to *utilize arbitrarily rich external knowledge long-term memories and achieve self-evolution in capabilities*.

Besides algorithmic innovation, we further meticulously prepare a novel benchmark, *AcademicEval*, which requires an LLM to faithfully leverage ultra-long context to answer queries based on real-world academic papers. Extensive experiments on AcademicEval and two other datasets validate that Thought-Retriever remarkably outperforms state-of-the-art baselines by achieving 5%-45% higher win rate. Moreover, we further demonstrate 2 exciting findings: (1) Thought-Retriever can indeed help LLM self-evolve after solving more user queries; (2) Thought-Retriever learns to leverage deeper thoughts to answer more abstract user queries. In summary, our main contributions are as follows:

- Thought-Retriever framework that enables an LLM to efficiently and effectively utilize external knowledge and further self-evolve through continuous interactions.
- AcademicEval[1], a new benchmark for testing LLM's understanding of ultra-long context. Notably, AcademicEval is of high quality, dynamic, and resembles real-world LLM applications.
- Thought-Retriever consistently outperforms all state-of-the-art retrieval-augmented and long-context baselines. We further present two exciting new findings, revealing the self-evolution and the abstraction capability of LLMs.

2 Thought-Retriever: Effectively Equip LLMs with External Knowledge

2.1 Preliminaries

An *external knowledge* base $\mathcal{K} = (K_1, K_2, ..., K_n)$ consists of n data chunks. An LLM L can generate a *thought* $T_i = L(Q_{\text{think}}, \mathcal{K}_i)$ as its response when it is prompted to elaborate its thought process, using query Q_{think}, given a set of reference data chunks $\mathcal{K}_i$. We define the *source* of an LLM's response, *e.g.*, a thought T_i, as the set of data chunks $\mathcal{K}_i$ that are used to generate the response, represented as a mapping $O(T_i) = \mathcal{K}_i$. A key motivation for Thought-Retriever is that an LLM can generate responses based on its past responses; therefore, given a thought T_i, we can recursively trace the source of data chunks with mapping $O(\cdot)$, until we find the *root source* via a mapping $\hat{O}(T_i) = \mathcal{K}_i$, consisting of all the raw data chunks from the external knowledge $\mathcal{K}$ that are used to create the thought T_i.

[1]Code and automatic data collection pipeline will be released.

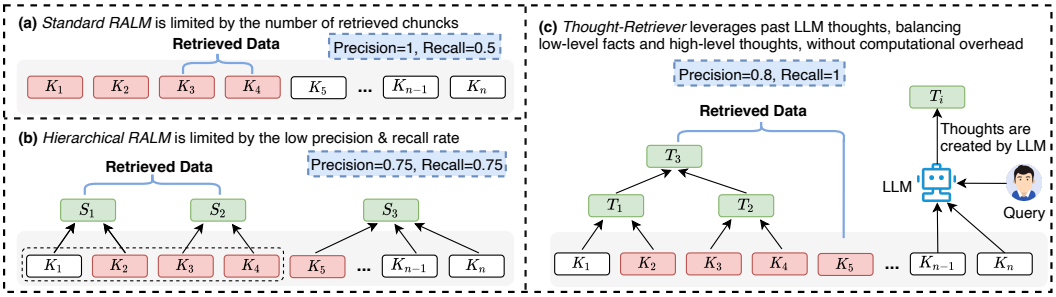


Figure 1: **Why Thought-Retriever helps**. **(a)** A standard RALM is limited by the number of retrieved chunks. The retrieved data fail to cover all the necessary data chunks (red chunks) for a given query. **(b)** A hierarchical RALM could improve the recall at the cost of lower precision. **(c)** Thought-Retriever leverages past LLM thoughts that are collected from answering user queries, with little computational overhead. Thought-Retriever balances low-level and high-level thoughts, leading to high precision and recall.

To measure how effectively an LLM can utilize external knowledge, we propose to extend the retrieval metric, precision and recall, with the root source mapping $\hat{O}(\cdot)$. Assuming that answering a user query Q_{think} requires a set of data chunks $\mathcal{K}_i \in \mathcal{K}$, and an LLM's response is T_i. We have

$$\text{Precision} = \frac{|\mathcal{K}_i \cap \hat{O}(T_i)|}{|\hat{O}(T_i)|}, \quad \text{Recall} = \frac{|\mathcal{K}_i \cap \hat{O}(T_i)|}{|\mathcal{K}_i|} \tag{1}$$

2.2 Motivating Examples

As a motivating example, in Figure 1, we assume $\mathcal{K}_i = \{K_1, K_2, K_3, K_4\}$ is required to answer a user query and an LLM can only fit 2 data chunks in its context window. A standard RALM (Figure 1(a)) can achieve perfect precision by retrieving the correct data chunks; however, it has a lower recall since it does not have the context window to hold all the relevant data chunks.

To address the limited context window of RALM, researchers Chen et al. (2023) proposed hierarchical RALMs (Figure 1(b)), where similar data chunks are summarized into S_i via LLM as a preprocessing step. However, the tree-structured summary structure is rigid, since the summaries S_i are independently generated from the user queries. In Figure 1(b), ideally, chunks $\{K_2, K_3\}$ and $\{K_4, K_5\}$ should be grouped together to answer the user query, where Precision $= 1$, Recall $= 1$ could be achieved; however, the tree construction happened before user query, and the generated tree fail to adapt to the diverse future user query.

To stress the above limitations of existing RALMs, as is shown in Figure 1(a), we propose Thought-Retriever that leverages past LLM thoughts and balances low-level facts and high-level thoughts to answer user queries. In real-world applications, user queries are often sufficiently diverse, leading to numerous diverse thoughts to meet the demands of new user queries. This valuable observation differentiates Thought-Retriever from existing tree-structured RALMs: (1) Thought-Retriever offers a more flexible structure of thoughts that depends on past user queries, and (2) the thoughts leveraged by Thought-Retriever are byproducts from the standard RALM response, making it easy to implement and brings little computational overhead.

2.3 Thought-Retriever Framework

Method Overview. Figure 2 offers an overview of the proposed Thought-Retriever framework, which consists of 2 major components: (1) **Thought retrieval**, where data chunks from external knowledge and thought memory are retrieved; (2) **Answer generation**, where an LLM generates the answer for the user query based on the retrieved data chunks; (3) **Thought generation**, where an LLM further generates thought and its confidence based on the user query and the generated answer; (4) **Thought memory update**, where meaningless and redundant thoughts are removed; the thought memory is updated with the remaining *novel* thoughts, rather than adopting all the *new* thoughts. We summarize the pipeline of Thought-Retriever in Algorithm 1, whose details are shown as follows. Detailed prompts of this section can be found in Appendix B.

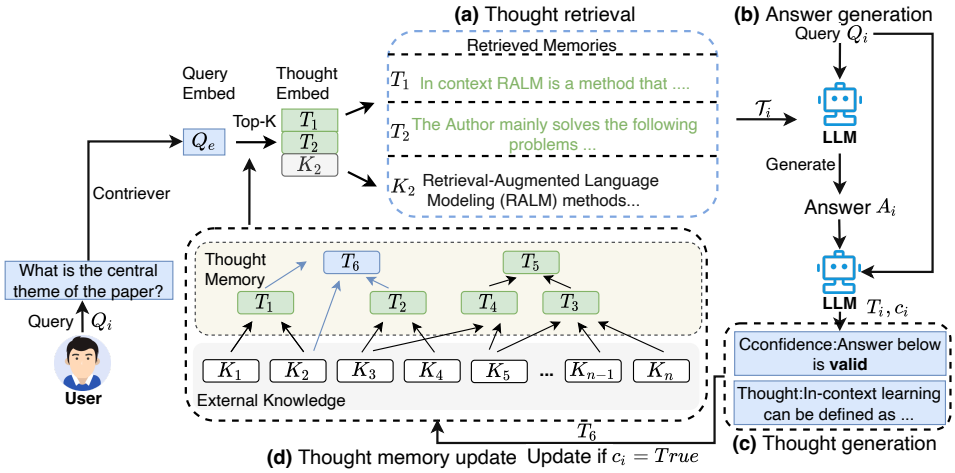


Figure 2: **Thought-Retriever Framework**. **(a) Thought retrieval:** Upon receiving a user query, Thought-Retriever retrieves top-K data chunks from the mixture of external knowledge and thought memory based on embedding similarity; (b) **Answer generation:** The LLM generates the answer for the user query based on the retrieved data chunks; (c) **Thought generation:** The LLM further generates thought and its confidence based on the user query and the generated answer; (d) **Thought memory update:** Meaningless and redundant thoughts are removed and the remaining *novel* thoughts are used to update the thought memory.

Algorithm 1 Thought-Retriever Inference Algorithm

Input: User queries $\mathcal{Q}$, external knowledge $\mathcal{K}$, thought memory $\mathcal{T}$, language model L, retriever R, confidence c.
Output: Answers to user queries $\mathcal{A}$, updated thought memory $\mathcal{T}$.

$\mathcal{A} \leftarrow \{\}$
for $Q_i \in \mathcal{Q}$ **do**
  $\mathcal{T}_i \leftarrow R(Q_i, \mathcal{K} \cup \mathcal{T})$ {Thought retrieval}
  $A_i \leftarrow L(Q_i, \mathcal{T}_i)$ {Answer generation}
  $\mathcal{A} \leftarrow \mathcal{A} \cup A_i$
  $T_i, c_i \leftarrow L(Q_i, A_i)$ {Thought generation}
  $\mathcal{T} \leftarrow \mathcal{T} \cup T_i$, if $c_i =$ True {Thought memory update}
end for
return $\mathcal{A}, \mathcal{T}$

Thought Retrieval. After receiving a user query Q_i, Thought-Retriever R retrieves relevant information $\mathcal{T}_i$ from external knowledge $\mathcal{K}$ and previously generated thought memory $\mathcal{T}$ via embedding similarity ranking. This process is formulated as $\mathcal{T}_i \leftarrow R(Q_i, \mathcal{K} \cup \mathcal{T})$.

Answer Generation. Based on the retrieved information $\mathcal{T}_i$, we design a prompt to combine $\mathcal{T}_i$ and user query Q_i and feed the prompt to an LLM L to get the answer A_i. It can be articulated as $A_i \leftarrow L(Q_i, \mathcal{T}_i)$.

Thought Generation. We can generate thoughts via LLM L using the obtained answer A_i and its query Q_i. However, redundant or meaningless thoughts during the generation process may harm the LLM performance. To solve this issue, we design a special prompt so that LLM L can generate thoughts T_i and thought quality confidence c_i based on the user's query Q_i and corresponding answer A_i. This can be described as $T_i, c_i \leftarrow L(Q_i, A_i)$.

Thought Memory Update. The confidence of thought quality c_i is a boolean indicator that determines whether the new generated thought should be updated into the thought memory $\mathcal{T}$. Here, we design that if the LLM is confident about its answer, where c_i is True, $\mathcal{T}$ will be updated.

3 ACADEMICEVAL: NEW BENCHMARK FOR LONG-CONTEXT LLM UNDERSTANDING

Current benchmarks for assessing agent long-context memory utilization involve tasks such as question-answering, long-context summarization, and classification. Despite being well-constructed, they are limited in flexibility and real-world impact and are costly to acquire. To address these issues, we introduce an innovative benchmark, ***AcademicEval***, based on academic papers from arXiv collected on a weekly basis. *AcademicEval* is superior in three aspects: 1) it dynamically collects the most up-to-date data; 2) it acquires high-quality labels at no additional cost; and 3) it allows real-world applications with high impacts. *AcademicEval* comes with two datasets: *abstract* and *related*.

AcademicEval-abstract. This dataset focuses on the summarization of single (*Abstract-single*) or multiple (*Abstract-multi*) academic papers. The agent is presented with one or more papers with the abstract and conclusion sections removed and is tasked with writing an abstract. For *Abstract-single*, the generated abstract is directly compared with the paper's original abstract. For *Abstract-multi*, the generated abstract is compared with a summary of abstracts from all the provided papers, which is generated by an LLM.

AcademicEval-related. This dataset introduces a challenging task for assessing an LLM agent's ability to understand the connections between different segments of its long-context memory. The task is to write a related work section based on the title and abstract of a target paper. The agent needs to use the title and abstract as the query to retrieve memory chunks to complete this task. Each memory chunk depicts an abstract of a paper, where some papers are cited in the related work section of the target paper, while others are random papers from the same broader field. The generated related work is then compared to the original related work of the target paper for evaluation.

Benefits and Contributions. *AcademicEval* offers several advantages over existing benchmark datasets. Firstly, we maintain an up-to-date dataset from arXiv that benefits from the continuous publication of new papers. This dynamic nature eases overfitting and label leakage problems in static benchmarks and enables the evaluation of agent self-adaptability. Secondly, high-quality labels can be generated with no extra cost as opposed to manually crafted datasets that require human effort. Thirdly, our dataset is not only valuable for evaluating LLM agents but also serves as a practical academic tool in the real world to assist researchers in better understanding their fields and boost productivity. We developed a highly automated codebase for dataset construction that will be released soon. We plan to launch a public platform that will enable users to easily create similar datasets or utilize LLMs for academic tasks. The detailed datasets format can be found in Appendix C.

4 EXPERIMENT

4.1 EXPERIMENT SETUP

Additional Datasets. Besides AcademicEval, we further evaluate *Thought-Retriever* against state-of-the-art baselines on two public datasets. (1) **GovReport** Cao & Wang (2022): This dataset comprises 19,466 reports and associated labels prepared by government research agencies to verify if the agent is capable of extracting salient words and useful information from a single lengthy governmental document. (2) **WCEP** Ghalandari et al. (2020): This dataset contains 10,200 entries, each containing multiple news articles associated with an event sourced from the Wikipedia Current Events Portal. It requires the agent to understand and extract useful information from a cluster of documents. Table 1 summarizes the statistics for all the datasets.

Table 1: Overview of Datasets Used

Dataset	Task Type	Avg. len	Cases
AcademicEval			
Abs-single	Single Sum	8,295	100
Abs-multi	Multi Sum	33,637	30
Rel-multi	Multi Related	22,107	30
Public Datasets			
Gov Report	Single QA	8,910	100
WCEP	Multi QA	8,176	30

Baselines. To gain a comprehensive understanding of our thought retriever's performance on agent long-term memory tasks, we have adapted several baselines. All experiments with these baselines are

Table 2: **Thought-Retriever consistently outperforms all the baselines in fact retrieval datasets**. Bold and underline denote the best and second-best results. F1 score evaluates the similarity with the ground truth, higher is better. Win rate compares each method's response with Thought-Retriever, higher is better. Note that the maximum context length is 2,000 tokens for all retriever-based methods and Thought-Retriever employs Contriever as its retriever.

Type	**AcademicEval**						**Public**			
Dataset	**Abstract-single**		**Abstract-multi**		**Related-multi**		**Gov Report**		**WCEP**	
Method	F1	Win Rate	F1	Win Rate	F1	Win Rate	F1	Win Rate	F1	Win Rate
BM25	0.712	21%	0.732	27%	0.714	32%	0.711	30%	0.378	31%
TF-IDF	0.715	20%	<u>0.734</u>	32%	<u>0.731</u>	45%	0.695	35%	0.423	34%
Contriever	0.725	41%	0.728	27%	0.722	27%	0.723	40%	0.411	40%
DPR	<u>0.726</u>	30%	0.731	45%	0.729	27%	0.688	20%	0.401	33%
DRAGON	0.718	36%	0.724	14%	0.719	36%	0.71	40%	<u>0.431</u>	35%
Full Context (left)	0.686	3%	0.699	0%	0.712	7%	0.734	45%	0.407	35%
Full Context (right)	0.685	3%	0.697	0%	0.683	3%	0.720	40%	0.410	41%
OpenOrca-8k	0.701	1%	0.706	7%	0.711	7%	**0.744**	41%	0.369	30%
Nous Hermes-32k	0.704	2%	0.711	16%	0.722	21%	0.748	37%	0.414	37%
Thought-Retriever	**0.743**	**50%**	**0.738**	**50%**	**0.732**	**50%**	<u>0.732</u>	**50%**	**0.438**	**50%**

conducted under the same LLM: Mistral-8x7B with LLM context length of 4, 096.Jiang et al. (2024). Note that we set chunk size=500, K=8, and maximum context length=2,000 tokens for all RALMs.

First, we consider 2 heuristic-based retrievers: (1) **BM25** Robertson et al. (2009): A widely-used ranking function in information retrieval. (2) **TF-IDF** Ramos et al. (2003): A statistical measure that evaluates the importance of a word in a memory. Second, we select 3 deep learning-based retrievers: (3) **Contriever** Izacard et al. (2022): leveraging contextualized embeddings and neural networks to understand and retrieve relevant memory chunks. (4) **DPR** Karpukhin et al. (2020): retrieving memory chunks by encoding chunks and queries into dense vectors. (5) **DRAGON** Lin et al. (2023): employing contrastive learning to train its ability to retrieve memory chunks. Third, we consider full context window baselines with document truncation: (6) **Full Context (left)** Chen et al. (2023): This approach uses the initial segment of a document, truncated to fit within a 4,096-token window. Focusing on the first 4,096 tokens, it prioritizes early content in the document. (7) **Full Context (right)** Chen et al. (2023): In contrast to Full Context (left), it utilizes the final segment of a document, also truncated to a 4,096-token window. Lastly, we selected two long-context LLMs: (8) **OpenOrca-8k** Mukherjee et al. (2023): fine-tuned on the Mistral 7B model using the OpenOrca dataset. At its release time, it was ranked the best model among all models smaller than 30B on Hugging Face, with a maximum context length of 8,192 tokens. (9) **Nous Hermes-32k** Shen et al. (2023): trained on Mixtral8x7B MoE LLM. It boasts a maximum context length of 32,768 tokens. Note that we do not compare with MEMWALKER Chen et al. (2023), since it is costly to run and cannot scale to tasks with many data chunks. We use Contriever as Thought-Retriever's retriever.

Evaluation Metrics. Our evaluation approach encompasses both automated and AI-based assessments: (1) **F1** Zhang* et al. (2020): This metric computes the semantic similarity between the generated text and the ground truth reference. A F1 score closer to 1 indicates a higher alignment with the reference text, signifying better quality of the generated content. (2) **Win Rate**: Alongside F1, we incorporate feedback from AI evaluator for a more comprehensive assessment. Here, we choose Platypus2 Instruct (70B) as our AI evaluator. This evaluation process involves presenting various responses to LLMs evaluator, who then ranks the quality of the responses. The percentage represents the frequency of a response being chosen over our thought retriever. A rate below 50% suggests that our thought retriever is outperforming the compared baseline.

4.2 Interaction with Factual Knowledge

This section is to verify the performance of *Thought-Retriever* when the external knowledge comes from interaction with facts. We report the performance of our model and baselines in Table 2. Major observations are as follows:

First, in both *AcademicEval* and public benchmarks, *Thought-Retriever* significantly outperforms most baselines on two metrics. For example, it achieves 5%-45% win rate advantage across all datasets. This suggests that thoughts formed through interaction with the environment can effectively enhance an agent's performance in different tasks.

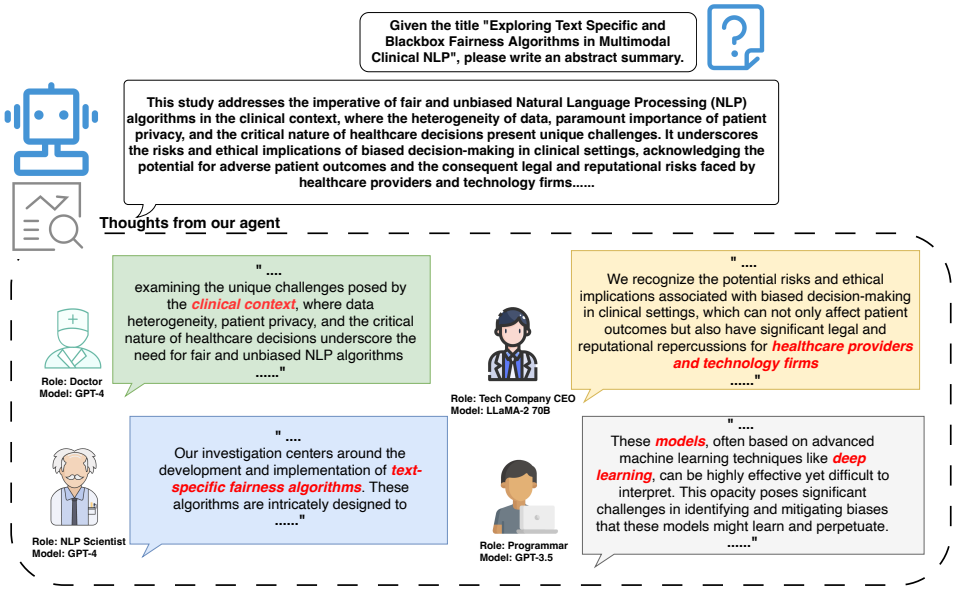


Figure 3: This figure presents a case study in which our agent communicates with four other agents, each an expert in a different field. These expert agents are not only assigned specific roles (e.g., doctor) but also provide relevant background texts to substantiate their expertise. Our agent is then able to rapidly learn from their thoughts and incorporate them as external knowledge.

Second, we observe that the performance of methods that use the entire text directly have many features on two different benchmarks differs greatly, which contain Full Context baselines and long-context LLMs baselines. However, the performance of retriever-based methods are stable across two benchmarks. This is due to two reasons: (1) AcademicEval is a more challenging benchmark. It contains ”multi-modal” information, such as tables, different chapters, different symbol formats, etc. Directly putting this complicated information in a context makes it difficult for the LLM agent to process and analyze. For retriever-based methods, they extract the most important information for respond the query from the entire memory, so they can filter out the influence of some redundant information and get better results; (2) Some long-context LLMs may have continuously train on the public benchmarks, which causes the leak of the label and the overfit of the model. Contrast to this, AcademicEval is a good benchmark for evaluate the zero-shot performance of LLM agent and has no risk of label leakage and overfitting. Since the benchmark is formed using papers from arXiv, it is dynamic and always up-to-date, benefiting from the continuous publication of new papers.

4.3 Interaction with Other LLM Agents

Forming thoughts can be a lengthy process. When a new agent lacks relevant memory or external knowledge, it is challenging to develop high-quality thoughts and memories from scratch. Consequently, we aim to investigate whether Thought-Retriever can help the agent quickly learn from other agents who have already formed expert knowledge.

To answer this question, we design an experiment on Abstract-single and the goal of the agent is to write an abstract summary based on its title. Our agent builds its memories based on interaction with with other agents, which include different roles of an LLM or different LLMs as shown in Fig 3. To verify the effectiveness of Thought-Retriever under this setting, we design four different comparison cases: (a) Based on Contriever, retrieve and respond to the original context of the article as a golden case; (b) Feed the query directly to the LLM to get the responses; (c) Let other agents provide some relevant data based on query, then use these data as raw memories of our agent, and finally retrieve and get response based on Contriever; (d) Replace Contriever with Thought-Retriever in the setting of (c). We performed AI evaluation on the responses obtained in four cases, and the results showed that rank of them from good to bad is: a, d, c, b. Moreover, the response quality of (d) is very close to that of (a). These observations demonstrate the good performance of Thought-Retriever under this setting.

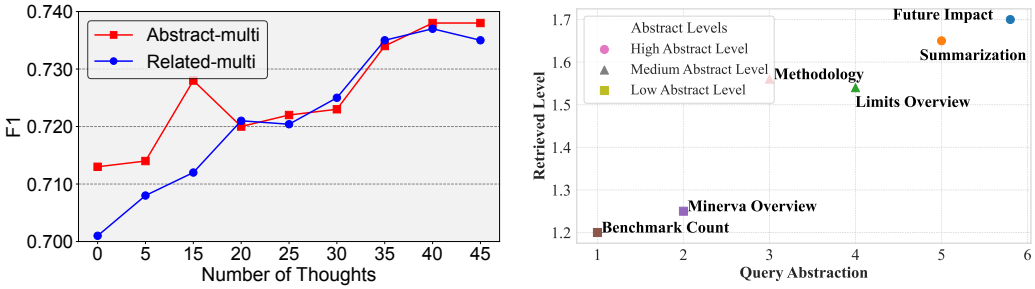


Figure 4: **(a).** It illustrates the agent's performance across various datasets as the number of thoughts increases. **(b).** Deeper thoughts help abstract queries. Specifically, it illustrates the correlation between six questions, categorized by their level of abstraction as evaluated by GPT-4 (x-axis), and the abstraction level of the corresponding retrieved information (y-axis). The questions are grouped into three categories: high abstraction (top 2 questions), medium abstraction, and low abstraction, respectively. Keywords from each question are displayed next to their corresponding data points for clarity.

4.4 New Findings from Thought-Retriever

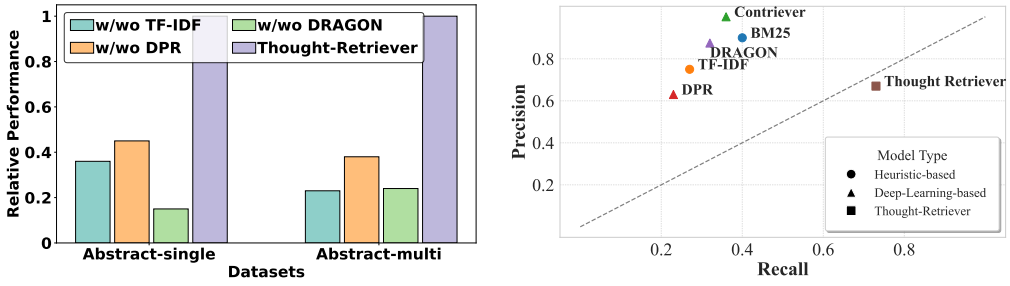
Thought-Retriever helps LLM self-evolve after solving more user queries - a new type of scaling law. To investigate the relationship between the performance of Thought-Retriever and the number of thoughts, we design an experiment using varying numbers of thoughts on Abstract-multi and Related-multi of AcademicEval. As depicted in Fig. 4(a), there is a distinct trend of increasing F1 scores correlating with the growing number of thoughts, which indicates improved performance. Therefore, more interactions with the users enable Thought-Retriever to assist LLMs in self-evolving and developing deeper understandings, demonstrating a new type of scaling law Kaplan et al. (2020).

Thought Retriever learns to leverage deeper thoughts to answer more abstract user queries. We conduct a cases study to explore the relationship between the abstraction levels of queries and the retrieved information. Specifically, we created a set of questions with varying levels of abstraction and ranked them according to their abstraction level using GPT-4 (exact queries can be found in Appendix D). For retrieved information abstraction level, we first assigned all the raw segments of text from external knowledge base an abstraction level of 1. The abstraction thought is then calculated as the average abstraction level of al lthe segments it retrieves, plus one. For example, a thought based solely on the external knowledge base would have an abstraction level of 2.

To explore the relationship between the abstraction levels of queries and the retrieved information, we conducted a case study. Specifically, we created a set of questions with varying levels of abstraction and ranked them according to their abstraction level using GPT-4. We assigned all memory chunks from the external knowledge base an abstraction level of 1. The abstraction level of a thought is then calculated as the average abstraction level of all the segments it retrieves, plus one. For example, a thought based solely on the external knowledge base would have an abstraction level of 2. If it also incorporates other thoughts, its abstraction level would be higher. As shown in Fig. 4(b), where the y-axis represents the abstraction level of the question and the x-axis represents the average abstraction level of all information retrieved by our method. It can be observed that more abstract questions tend to retrieve information with higher abstraction levels.

4.5 Ablation Study

We conduct a series of experiments to investigate the impact of various retrievers. (1) **w/wo TF-IDF**: In this variant, we replace the retriever in our method with TF-IDF to assess its effectiveness compared to our current retriever. (2) **w/wo DPR**: Here, we substitute the retriever in our method with DPR to evaluate its performance relative to our existing retriever. (3) **w/wo DRAGON**: We replace our method's retriever with DRAGON to assess its performance in comparison to our current retriever. We report the evaluation results on Abstract-single and Abstract-multi datasets in Fig. 5(a). It is clear



(a) (b)

Figure 5: **(a).** Thought-Retriever performs better than all the other variants across two datasets. **(b).** Thought- Retriever significantly performs better in balancing recall and precision (The dotted line indicates the exact balance between precision and recall. The closer the dotted line is, the better the balance is). The traditional retriever-based method achieves high precision but low recall. Thought Retriever balances precision with recall, which maintains good precision when the recall is very high.

from these comparisons that our method consistently outperforms all the variants, suggesting that Contriever is most suitable for Thought-Retriever.

4.6 Qualitative Analysis based on Precision and Recall

In our motivation example in Sec 2.2, we show cases where traditional methods fell short in achieving good recall and precision values. In this section, we conduct a case study on our Related-multi dataset, where we demonstrate that when compared to other baselines, our Thought-Retriever significantly performs better in balancing recall and precision.

In the experiment, the related work section of the case study includes 22 papers, which is regarded as ground-truth of retrieval. We aimed to assess how well different retrievers could cover these 22 papers, given a limitation of retrieving only 8 chunks of information at a time. We plotted the findings in Fig. 5(b) where the x-axis is the recall value and the y-axis represents the precision. It can be observed that all traditional retrieval methods displayed significantly low recall values. This is primarily attributed to the top-K retrieval limit, which, in this scenario, could only encompass at most 8 out of the 22 papers. In comparison, Thought-Retriever demonstrates a notable improvement in recall value. This is because it leverages thoughts which is constructed from multiple papers, thereby allowing Thought-Retriever to achieve a much higher recall. More importantly, Thought-Retriever also exhibits moderately high precision compared to other retrievers. This suggests that, despite a minor trade-off, Thought-Retriever does not significantly compromise its ability to retrieve most relevant information.

5 Conclusion

We propose Thought-Retriever to to effectively and efficiently help LLMs utilize rich external knowledge. It represents a breakthrough in enhancing LLMs by facilitating dynamic access to vast external knowledge without the limitations of context length. It introduces an innovative strategy that leverages ”intermediate thoughts” from past interactions, allowing the system to evolve and refine its understanding continuously. Demonstrating superior performance across various datasets, including the novel AcademicEval benchmark, Thought-Retriever showcases its potential to revolutionize AI systems, making them more adaptive and capable of real-time, context-aware responses. This advancement not only promises significant improvements in industries like customer service, healthcare, and legal advisory but also lays the groundwork for future research aimed at achieving a more general AI, pushing the boundaries of technology’s role in society.

References

Desirée Bill and Theodor Eriksson. Fine-tuning a llm using reinforcement learning from human feedback for a therapy chatbot application, 2023.

Shuyang Cao and Lu Wang. Hibrids: Attention with hierarchical biases for structure-aware long document summarization, 2022.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*, 2023.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. A large-scale multi-document summarization dataset from the wikipedia current events portal. *arXiv preprint arXiv:2005.10070*, 2020.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2022.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models. *arXiv preprint arXiv:2309.10062*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Ray Kurzweil. *How to create a mind: The secret of human thought revealed*. Penguin, 2013.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*, 2023.

Zhe Liu, Chunyang Chen, Junjie Wang, Mengzhuo Chen, Boyu Wu, Xing Che, Dandan Wang, and Qing Wang. Make llm a testing expert: Bringing human-like interaction to mobile gui testing via functionality-aware decisions. *arXiv preprint arXiv:2310.15780*, 2023.

LM-SYS. Longchat-13b-16k. https://github.com/DachengLi1/LongChat, June 2023. GitHub Repository.

MosaicML. Taking language models to the next level with mpt-7b. https://fusionchat.ai, 2023.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023a.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023b.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*, 2023.

Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pp. 29–48. Citeseer, 2003.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. Taskbench: Benchmarking large language models for task automation. *arXiv preprint arXiv:2311.18760*, 2023.

Bruno Snell. *The discovery of the mind*. Courier Corporation, 2012.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (llm)? aka will llms replace knowledge graphs? *arXiv preprint arXiv:2308.10168*, 2023.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2022.

Arun James Thirunavukarasu, Shathar Mahmood, Andrew Malem, William Paul Foster, Rohan Sanghera, Refaat Hassan, Sean Zhou, Shiao Wei Wong, Yee Ling Wong, Yu Jeat Chong, et al. Large language models approach expert-level clinical knowledge and reasoning in ophthalmology: A head-to-head cross-sectional study. *medRxiv*, pp. 2023–07, 2023.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*, 2023.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*, 2023.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*, 2023.

A Additional Related Works

How to utilize the long memory of LLM agents to respond to user's query has gained increasing importance. Recent advancements predominantly focus on two approaches: scaling LLMs' parameters to support extended memory windows and selectively extracting information from lengthy memories, a process known as RALM.

Long-context LLMs. In response to the challenge of long-memory processing in LLM agents, the most intuitive strategies involve expanding the LLM's memory window, enabling it to process longer inputs. These methods typically include training larger, more advanced models MosaicML (2023); LM-SYS (2023), fine-tuning existing language models to handle wider windows Tay et al. (2022), and applying positional encoding to extend the memory window size Xiao et al. (2023). However, these methods have shortcomings in their high costs associated with model training and a lack of flexibility, as they do not address the fundamental issue of long memory. For instance, to process even longer memories, it becomes necessary to engage in additional parameters or model training, which is both rigid and resource-intensive.

Retrieval-Augmented Language Models. Given the limitations of the long-memory LLMs, RALM emerges as a notable alternative, distinguished by its flexibility and lower costs. Current RALM methods retrieve pertinent information from extensive memory chunks using a variety of techniques, such as retrievers based on token embeddings Izacard et al. (2022); Lin et al. (2023), keyword searches Robertson et al. (2009), or a fine-tuned reranker Ram et al. (2023), among others. While these approaches have demonstrated promising results, they still face considerable challenges. The effectiveness of these methods is still constrained by the memory capacity of LLMs; they can only retrieve an amount of information that fits within the LLM's window size. As a result, facing the growing memory length of LLM agents, these methods frequently fall short. Recently, to ease the memory window limitation of RALM, some researchers propose methods based on memory summarization, which covers more memory information and retrieves fewer memory chunks. For example, Howard et al. Chen et al. (2023) summarize memory chunks into a hierarchical tree structure and make the agent navigate the effective information according to the tree structure. However, these methods require deliberately building a rigid tree structure, which is not flexible when the agent responds to different queries. In addition, a large amount of navigation by the agent also generates huge costs.

In this paper, we address the aforementioned challenges by introducing a Thought-Retriever framework based on RALM, which utilizes the user's query to summarize retrieved agent memory segments into 'thoughts'.

B Prompt Utilization

Here, we present the detailed prompt instructions we used in our framework for thought retrieval, answer generation, and thought generation and update, specifically on AcademicEval-abstract and AcademicEval-related.

Prompts for AcademicEval-abstract. As shown in Figure B, we first retrieve information based on the query for writing a summarization. Then, we ask the LLM agent to write an abstract based on the retrieved information. Lastly, we prompt it to evaluate if the answer is meaningful and determine whether we should save it as a new thought in our thought memory.

Prompts for AcademicEval-related. In Figure 7, we present the prompts used for AcademicEval-related. Initially, we supply a query to write a related work section for introducing the paper, based on its abstract for information retrieval. Subsequently, we request the LLM agent to generate a related work paragraph, specifically drawing from the original abstract and the information retrieved. Here, we also provide an example to aid the LLM agent in comprehending the task. Finally, we employ the same prompt to assess whether the response is meaningful and decide if it should be saved as a new thought in our thought memory.

AcademicEval-Abstract

Retrieval Instruction:
"Please craft an abstract summarizing the key points from the provided text. The abstract should be in appropriate length, and include the main theme, significant findings or arguments, and conclusions of the text. Ensure it captures the essence of the the content in a clear, succinct manner."

Answer Generation Instruction:
" Please craft an abstract summarizing and connecting the key points from the provided Text. The text should be composed of content extracted from different papers, potentially spanning varied disciplines, but all addressing overlapping themes or subjects. The abstract should be of appropriate length (around 300 words), encompassing the main theme, significant findings or arguments, and conclusion of the Text. Ensure the abstract captures the essence of the content in a clear, succinct manner, providing a coherent summary that bridges the various papers."

Thought Generation and Thought Memory Update Instruction:
"Input: Given question:{question}, given answer:{answer}. Based on the provided question and its corresponding answer, perform the following steps:
Step 1: Determine if the answer is an actual answer or if it merely indicates that the question cannot be answered due to insufficient information. If the latter is true, just output 'idk' without any extra words.
Step 2: If it is a valid answer, succinctly summarize both the question and answer into a coherent knowledge point, forming a fluent passage.
"

Figure 6: Prompts used in Thought Retriever Framework on AcademicEval-Abstract

C Details of AcademicEval

We provide the data format for datasets in our proposed AcademicEval benchmark in Table C. For AcademicEval-abstract, in the single document setting, each case includes the paper title, abstract, and main content, excluding the abstract and conclusion. For the multiple document setting, we combine five such entries into one. For AcademicEval-related, each paper includes a title, its abstract, and a label indicating whether it is the original paper, the original paper's related work, or just a random paper under the same broader field.

D Specific Queries of Abstract Level

This section lists the specific queries utilized in our case study in Section 4.4, demonstrating how Thought Retriever leverages deeper thoughts for more abstract user queries. Each query is categorized by its general level of abstraction, ranked according to its abstraction level as assessed by GPT-4, and detailed with its exact content in Table 4.

	Attribute	Description
Abstract-Single	'title'	The title of the academic paper.
	'abstract'	The abstract of the academic paper.
	'main_content'	The content of the paper excluding the abstract and the conclusion.
Abstract-Multiple	'title 1'	The title of the first academic paper.
	'abstract 1'	The abstract of the first academic paper.
	'main_content 1'	The content of the first paper excluding the abstract and the conclusion.
	...	...
	'title 5'	The title of the fifth academic paper.
	'abstract 5'	The abstract of the fifth academic paper.
	'main_content 5'	The content of the fifth paper excluding the abstract and the conclusion.
Related-multi	'title'	The title of the academic paper.
	'abstract'	The abstract of the academic paper.
	'label'	The label indicates if this paper is the "original paper", "related work", or "random paper"

Table 3: AcademicEval Data Format

Abstraction	Rank (GPT-4)	Query
High	6 (Most Abstract)	"What are the broader future implications of user-centric utility in NLP model evaluation?"
High	5	"Please craft an abstract summarizing the key points from the provided text."
Medium	4	"What are some of the limitations of this study?"
Medium	3	"What are the key methods introduced in this paper?"
Low	2	"Please explain the term Minerva to me."
Low	1 (Least Abstract)	"How many benchmarks are used to test the model's long context understanding ability in this paper?"

Table 4: Queries Used in Abstraction Level Case Study

E EXAMPLE OUTPUT OF LLMS

We have listed examples of outputs using different methods on AcademicEval-abstract-single. Specifically, in Table 5, we provide the original paper title and abstract, as well as the generated abstracts through our Thought Retriever, DPR, and TF-IDF methods respectively. We also use GPT-4 as an evaluator. It gives the following comment:

"The abstract generated by Thought Retriever is the most closely aligned with the original abstract. It covers the key points mentioned in the original text, such as the critique of the leaderboard paradigm for focusing primarily on performance metrics at the expense of other important factors like compactness, fairness, and energy efficiency. It also addresses the need for more transparency on leaderboards, including the reporting of practical statistics like model size, energy efficiency, and inference latency, to better reflect the utility of models to practitioners. This option encapsulates the essence of the original abstract by discussing the divergence between what is incentivized by leaderboards and what is practical and useful for the NLP community, advocating for a more holistic approach to evaluating NLP models that aligns with the community's diverse needs and values."

F DISCUSSION

Transformative Impact and Real-World Applications. The thought retriever represents a paradigm shift in AI systems, transforming them from static repositories of knowledge to dynamic, intelligent frameworks that interact and learn. Its unique architecture not only processes and retrieves information but also evolves with each user interaction, effectively 'thinking' and adapting over time. Such an intelligent system is crucial for scenarios where real-time learning and context-aware responses are vital. For instance, existing AI service systems could be significantly enhanced by incorporating our approach. By storing original guidelines and regulations as part of the external knowledge base and recording each human query and its results as thoughts, these systems can evolve into more intelligent entities capable of continuous improvement and learning. This adaptive capability makes the thought retriever an invaluable tool for dynamic and ever-changing industrial environments, where quick decision-making based on historical data and evolving information is crucial. In sectors like customer service, healthcare, and legal advisory, where personalized and informed responses are key, the thought retriever can provide more accurate, context-aware, and efficient solutions. Its ability to continuously learn and adapt from user interactions positions it as a groundbreaking tool for transforming how industries interact with and utilize AI technology.

Future Research. Inspired by human thinking, our thought retriever represents a solid step toward general AI agents. Building on this foundation, future research could address several key challenges. Firstly, scalability and efficiency in processing increasingly complex datasets will be crucial. This involves not only enhancing computational power but also refining algorithms for greater precision and speed. Secondly, understanding and mimicking human-like reasoning remains a pivotal goal. This includes grasping nuances in language, emotion, and cultural contexts, pushing the boundaries of what AI can comprehend and respond to. Moreover, ensuring ethical considerations in AI decision-making is significant. As the retriever evolves, its impact on privacy, security, and societal norms must be rigorously evaluated and guided. Finally, exploring new domains of application, such as personalized education, mental health analysis, and advanced robotics.

	Original / Generated Content
Paper Title	**Utility is in the Eye of the User: A critique of NLP Learderboards**
Original Abstract	Benchmarks such as GLUE have helped drive advances in NLP by incentivizing the creation of more accurate models. While this leaderboard paradigm has been remarkably successful, a historical focus on performance-based evaluation has been at the expense of other qualities that the NLP community values in models, such as compactness, fairness, and energy efficiency. In this opinion paper, we study the divergence between what is incentivized by leaderboards and what is useful in practice through the lens of microeconomic theory. We frame both the leaderboard and NLP practitioners as consumers and the benefit they get from a model as its utility to them. With this framing, we formalize how leaderboards – in their current form – can be poor proxies for the NLP community at large. For example, a highly inefficient model would provide less utility to practitioners but not to a leaderboard, since it is a cost that only the former must bear. To allow practitioners to better estimate a model's utility to them, we advocate for more transparency on leaderboards, such as the reporting of statistics that are of practical concern (e.g., model size, energy efficiency, and inference latency).
Thought Retriever	The paper "Utility is in the Eye of the User: A Critique of NLP Leaderboards" provides a critique of the existing belief that leaderboard metrics accurately reflect the utility of Natural Language Processing (NLP) models. The critique emphasizes the need to consider multiple factors beyond just predictive accuracy when evaluating NLP models, including energy efficiency, inference latency, model size, and practicality. The paper argues that leaderboard values and practitioner values differ, and calls for increased transparency on leaderboards, including the reporting of practical statistics, to better align leaderboard metrics with the preferences and values of the NLP community.The research highlights the importance of a more holistic approach to NLP evaluation that considers the diverse needs and values of the NLP community. Practical applications of this research include encouraging the development of more comprehensive NLP evaluation metrics, advocating for increased transparency on leaderboards with the reporting of practical statistics, and emphasizing the need to consider the diverse needs and values of the NLP community. The research also emphasizes the difference between leaderboard incentives and practical utility, suggesting an analysis of the incentives and utility functions of both leaderboards and NLP practitioners. The findings suggest that future NLP research may focus on developing more comprehensive evaluation metrics, promoting transparency on leaderboards, challenging the traditional focus on predictive accuracy, and creating NLP models and evaluation metrics that better serve the needs of a diverse range of users. The paper provides practical guidance for improving the evaluation of NLP models, encouraging a more holistic approach that aligns with the preferences and values of the NLP community. In summary, the paper argues that current NLP leaderboards prioritize predictive accuracy, but overlook other important factors such as energy efficiency and inference latency. The paper calls for increased transparency on leaderboards, including the reporting of practical statistics, to better align leaderboard metrics with the preferences and values of the NLP community. The research emphasizes the need for a more holistic approach to NLP evaluation that considers the diverse needs and values of the NLP community, and encourages the development of more comprehensive evaluation metrics that better serve the needs of a diverse range of users.

Table 5: Comparison of Abstracts

	Original / Generated Content
DPR	The text discusses the importance of incorporating cost, such as model size, energy-efficiency, training time, and inference latency, into leaderboard rankings for NLP models, which currently prioritize accuracy above all else. This one-dimensional focus on accuracy can result in a SOTA model providing high utility to a leaderboard but zero utility to a practitioner, as the model may come with costs that are not considered in the leaderboard rankings. The text suggests that model creators could voluntarily submit these cost statistics, but without a requirement, there is no explicit incentive for them to do so. The text also proposes the idea of creating a leaderboard for a specific type of user, with a desired threshold and reporting the performance on the worst-off group to incentivize creators to improve worst-case performance. Additionally, the text mentions the use of revealed preference theory to estimate each model's utility to practitioners, by examining which models they actually use. The text also suggests that leaderboards could make usage statistics for models public, and offer an interface that allows users to change the weighting on each metric and dynamically re-rank the models, effectively creating a personalized leaderboard for each user.The text also discusses the concept of creating a leaderboard for a specific type of user, with a desired threshold and reporting the performance on the worst-off group to incentivize creators to improve worst-case performance. This is in line with Rawlsian fairness, which prioritizes the welfare of the worst-off group in a society. The text also highlights the importance of considering the specific use-case and application of the models, as different applications may place different values on certain metrics such as latency or accuracy.In conclusion, the text argues that current leaderboards for NLP models are cost-ignorant and primarily focus on accuracy, which can result in models that provide high utility to the leaderboard but zero utility to practitioners. The text suggests that incorporating cost into leaderboard rankings, creating leaderboards for specific types of users, and utilizing revealed preference theory to estimate each model's utility to practitioners, could result in more useful and practical rankings for NLP practitioners. Additionally, the text highlights the importance of considering the specific use-case and application of the models, as different applications may place different values on certain metrics.
TF-IDF	The development and evaluation of natural language processing (NLP) models have been a focus of recent research, with studies highlighting issues such as brittleness, bias, and lack of robustness in these systems. The creation of datasets like SQuAD 2.0 addresses the challenge of unanswerable questions, while researchers have also identified biases in NLP systems that lead to better performance on gender-stereotypical inputs and inanimate objects. A key issue in NLP model evaluation is the alignment of priorities between leaderboard rankings and practitioners' preferences. While both groups value accuracy, practitioners are also sensitive to out-of-distribution data, fairness, and real-world applicability. Leaderboards, however, may not account for these factors, leading to differences in utility functions. A study of these differences reveals several limitations in contemporary leaderboard design. First, leaderboards only increase utility when improvements in accuracy also increase rank, whereas practitioners can benefit from any accuracy improvement. Second, leaderboards do not account for prediction costs, such as model size and energy efficiency, which are important considerations for practitioners. These findings suggest that leaderboard design should be reevaluated to better align with practitioners' preferences and real-world requirements. One approach could be to incorporate prediction costs and fairness measures into leaderboard rankings. Another possibility is to develop alternative evaluation methods that better reflect the complexities of NLP model usage in practice. In conclusion, current NLP model evaluation methods have limitations that can lead to brittle, biased, and inefficient models. By addressing these issues and better aligning leaderboard design with practitioners' preferences, researchers can work towards developing more robust, fair, and applicable NLP systems.

AcademicEval-Related

Retrieval Instruction:
"Could you please write a related work for introducing this paper? it's abstract is: {paper abstract}"

Answer Generation Instruction:
"Given the abstract and related work of a research article, along with a sample material, write a paragraph about its related work. Use the following as guidance:
Abstract: This research paper investigates the impact of climate change on global agricultural productivity. The study employs a comprehensive dataset of temperature and precipitation changes over the past century, combined with historical crop yield data. Through advanced statistical modeling and machine learning techniques, the research identifies significant correlations between temperature and precipitation fluctuations and variations in crop yields. Furthermore, it predicts future scenarios of agricultural productivity under different climate change scenarios, providing valuable insights for policymakers and stakeholders in the agricultural sector to develop adaptive strategies.

Related Work: Previous studies in the field have explored the relationship between climate change and agriculture but have primarily focused on specific regions or crops. Smith et al. (2017) conducted a comprehensive analysis of the impact of temperature on wheat yields in North America, highlighting the vulnerability of wheat crops to warming temperatures. Additionally, Johnson et al. (2019) investigated the effects of changing precipitation patterns on rice production in Southeast Asia, emphasizing the importance of water management in mitigating climate-related risks to agriculture. While these studies contribute valuable insights, our research extends their scope by considering a global perspective and employing advanced modeling techniques to provide more accurate predictions of future agricultural productivity under climate change scenarios.

Based on the abstract of this article and related materials, write a paragraph about its related work:
Abstract: {abstract}; Related materials: {context}
"

Thought Generation and Thought Memory Update Instruction:
"Input: Given question:{question}, given answer:{answer}. Based on the provided question and its corresponding answer, perform the following steps:
Step 1: Determine if the answer is an actual answer or if it merely indicates that the question cannot be answered due to insufficient information. If the latter is true, just output 'idk' without any extra words.
Step 2: If it is a valid answer, succinctly summarize both the question and answer into a coherent knowledge point, forming a fluent passage.
"