TIDES: Technical Information Discovery and Extraction System

Jihee Kim[1], Subeen Park[1], Hakyung Lee[2], YongTaek Lim[3],
Hyo-won Suh[4], Kyungwoo Song[1*]
[1]Yonsei University [2]Korea Development Bank [3]Datumo [4]ISysE, KAIST
{jihee_sta, sallyna602, kyungwoo.song}@yonsei.ac.kr
hakyunglee0417@gmail.com yongtaek.lim@selectstar.ai
hw_suh@kaist.ac.kr

Abstract

Addressing the challenges in QA for specific technical domains requires identifying relevant portions of extensive documents and generating answers based on this focused content. Traditional pre-trained LLMs often struggle with domain-specific terminology, while fine-tuned LLMs demand substantial computational resources. To overcome these limitations, we propose TIDES, Technical Information Distillation and Extraction System. TIDES is a training-free approach that combines traditional TF-IDF techniques with prompt-based LLMs in a hybrid process, effectively addressing complex technical questions. It uses TF-IDF to identify and prioritize domain-specific words that are rare in other documents and LLMs to refine the candidate pool by focusing on the most relevant segments in documents through multiple stages. Our approach improves the precision and efficiency of QA systems in technical contexts without LLM retraining.

1 Introduction

Technology advances create more complex data every day, making it harder to find the right information efficiently. This problem is especially important in technical fields, where users need exact answers to questions that use domain-specific terms. Large Language Models (LLMs) have made huge progress in answering general questions, almost as well as humans. However, LLMs often fail to capture technical terms correctly in specialized fields (Yang et al., 2023). We can improve this by fine-tuning LLMs (Lee et al., 2023), but training these increasingly large models costs too much time and money.

Traditional Question Answering (QA) systems often fall short in meeting the high standards needed in technical domains. Recent approaches like Question Answering System Architecture (QASA) (Lee et al., 2023) need extensive fine-tuning of smaller models, while Self-RAG (Asai et al., 2024) requires multiple rounds of document retrieval, using too much computing power. Moreover, technical documents contain specialized terms and complex questions that challenge these approaches, which often struggle with.

To address these challenges, we propose the Technical Information Discovery and Extraction System (TIDES), specifically designed to handle technical domains without extensive LLM retraining. TIDES employs a four-stage approach that combines Term Frequency-Inverse Document Frequency (TF-IDF) with advanced prompt engineering within a cognitive reasoning framework. This integration enables efficient filtering of irrelevant documents while maintaining high precision in handling domain-specific terminology. By leveraging carefully designed prompts rather than fine-tuning, TIDES maximizes the capabilities of existing LLMs while significantly reducing computational demands. Its multi-stage filtering architecture also mitigates common LLM limitations such as the "lost in the middle" problem, ensuring reliability and preventing hallucination without costly model adaptation.

TIDES distinguishes itself from existing LLM-based QA systems by its ability to deliver accurate and contextually relevant answers without requiring additional fine-tuning. Our methodology employs a structured four-stage approach, integrating the TF-IDF technique and advanced prompt engineering within a cognitive reasoning framework. By effectively identifying critical keywords in the technical domain, TF-IDF enables the early filtering of irrelevant documents, thereby enhancing the overall accuracy and efficiency of the TIDES model. Such strategic adaptations not only reduce computational demands but also enhance precision in handling domain-specific terminology. Additionally, advanced prompt engineering

*Corresponding author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 21756–21772
November 4-9, 2025

is employed within a cognitive reasoning framework to maximize the LLM's performance, ensuring that the system provides more precise and relevant responses than traditional methods. The code can be available at the GitHub repository: https://github.com/MLAI-Yonsei/TIDES.

2 Related Work

2.1 Document-based QA with LLMs

LLMs have achieved impressive results in extracting answers from unstructured documents (Brown et al., 2020). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020b; Li et al., 2022) enhances factual grounding by integrating information retrieval and text generation. However, even with these advancements, LLMs still produce hallucinations and incorrect facts, especially when handling complex or long documents (Zhao et al., 2024).

To mitigate these issues, researchers have explored iterative retrieval techniques that refine the context in multiple steps. For example, Self-RAG (Asai et al., 2024) updates the retrieved evidence over several iterations, improving accuracy but also increasing computational costs. While effective in controlled settings, iterative methods can hamper scenarios that require immediate deployment without substantial overhead.

Other approaches focus on aggregating information across multiple documents. Pereira, Jayr, et al. (Pereira et al., 2023) decompose questions into sub-questions, retrieve individual answers, and synthesize them, while (Saad-Falcon et al., 2024) extracts and structures document metadata to provide accurate answers.

Specifically, QASA (Lee et al., 2023) extends this idea by using three stages—Associative Selection, Rationale Extraction, and Systematic Composition—to refine answers step-by-step. This framework fine-tunes a T5 model (Raffel et al., 2020) for each stage, enabling the system to extract relevant paragraphs, generate evidence, and synthesize answers systematically. Despite its strengths, QASA has practical limitations. Fine-tuning T5 for each stage requires carefully curated datasets, which are often challenging to collect. Moreover, during the Associative Selection stage, QASA tends to over-identify irrelevant paragraphs, assigning them relevance unnecessarily.

2.2 Domain-Specific QA

When applying QA to specialized domains—such as IT support (Castelli et al., 2020), privacy policies (Keymanesh et al., 2021), legal texts (Abdallah et al., 2023), insurance guidelines (Na et al., 2022), or device manuals (Nandy et al., 2021; Ruiz et al., 2023)—researchers encounter domain-specific jargon and complex reasoning patterns. General-purpose LLMs often misunderstand technical terms or conflate similar concepts, resulting in inaccuracies (Zhang et al., 2024). Although fine-tuning on domain-specific data can alleviate these issues, collecting and preparing such datasets demands significant time and resources (Kratzwald and Feuerriegel, 2019). Repeated queries to proprietary models like GPT-4 to ensure factual consistency raise inference costs and complicate deployment (Tian et al., 2023).

Instead of fine-tuning or repetitive querying, some works integrate lightweight preprocessing steps to enhance efficiency. Techniques that apply TF-IDF (Shrivastava et al., 2022) identify salient terms early on, allowing systems to filter out irrelevant documents before the main reasoning steps. This filtering process reduces noise and streamlines subsequent operations, improving both accuracy and cost-effectiveness.

Our approach, TIDES, builds on these insights by combining the structured reasoning techniques found in multi-stage pipelines like QASA with TF-IDF-based filtering and prompt-based role assignment, all without large-scale fine-tuning. Rather than iteratively refining document sets like Self-RAG, TIDES applies a single, carefully managed retrieval pass. TF-IDF narrows the search space to relevant documents; prompt design encourages the LLM to behave like an expert, focusing on essential technical details and minimizing off-topic content. TIDES thus improves factuality and efficiency, achieving a balance between robust multi-step reasoning and practical, low-overhead adaptation to specialized domains.

3 Method

We provide TIDES, a methodology that adapts the QASA three-stage framework's cognitive reasoning processes to technical domain QA systems. Distinctively, TIDES uses a prompt-based approach without requiring fine-tuning of existing LLMs, reducing computational resources and enhancing adaptability to domain-specific queries. During

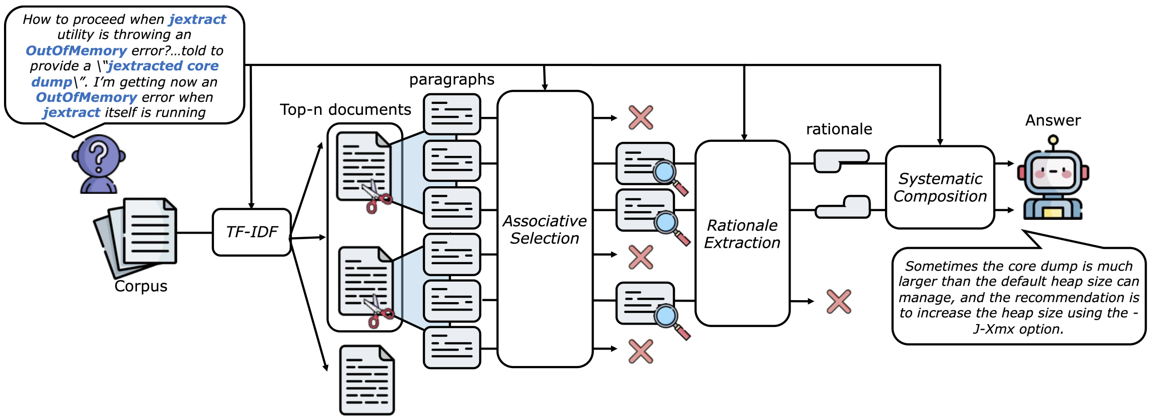

Figure 1: TIDES, Technical Information Discovery and Extraction System. The workflow starts with the analysis of technical documents using TF-IDF to identify relevant content. Non-relevant documents are discarded, and the remaining documents are segmented into paragraphs. These paragraphs undergo associative selection to filter out non-relevant content further. In the rationale generation phase, key evidence is extracted from the relevant paragraphs. Finally, systematic composition combines the extracted evidence into a coherent and concise answer to the technical question.

development, we iteratively tested intermediate prompts in ChatGPT to identify ambiguity or overconfidence, ensuring that our final instructions effectively guide the LLM to produce concise and accurate responses. To improve efficiency, we include a preliminary TF-IDF step to filter out irrelevant information, streamlining the cognitive reasoning tasks that follow.

3.1 TF-IDF

TF-IDF quantifies the importance of words by considering their frequency within a document and their rarity across a corpus D. It assigns higher weights to terms frequent in a specific document but rare elsewhere, effectively highlighting domain-specific terminology while minimizing the influence of common words.

This weighting approach benefits technical-domain QA, which differs from open-domain search as users often need exact terms like error codes and CLI flags in answers. A query such as “0x800F081F update failure” fails to be satisfied by a passage that merely paraphrases the term; the exact token sequence is critical to correctness. Sparse retrieval models like TF-IDF give priority to these rare, domain-specific tokens via the IDF component, whereas dense embeddings may smooth them away in vector space (Kamalloo et al., 2023). TF-IDF thus serves as a domain-adaptive filter that identifies relevant paragraphs without requiring fine-tuning.

Within TIDES, TF-IDF filters out irrelevant documents by measuring relevance to a given question q. To reduce noise and allow TF-IDF to focus on domain-specific terminology, preprocessing steps include converting text to lowercase, removing stop words using NLTK, and handling punctuation. These steps ensure that common and irrelevant terms do not dominate the ranking process.

After applying TF-IDF, we rank all documents in the corpus D and retain the top n most relevant documents. This refinement reduces the corpus to $\tilde{D} = \{\tilde{d}_1, \ldots, \tilde{d}_n\}$, where $n \leq N$. In our implementation, we set n = 30 based on empirical testing that showed an optimal performance-to-computation ratio, as metrics plateau beyond this value while runtime continues to increase almost linearly (see Section 4.6 for detailed analysis).

This refined set $\tilde{D}$ enables TIDES's three-stage process to operate effectively, enhancing both efficiency and accuracy by focusing only on pertinent documents and reducing computational load.

3.2 Associative Selection

The Associative Selection step begins by segmenting the refined corpus $\tilde{D}$ into paragraphs $P = \{p_1, \ldots, p_M\}$ for enhanced analysis of question q. We assess each paragraph p_j through binary classification of relevance, yielding $\tilde{P} = \{\tilde{p}_1, \ldots, \tilde{p}_m\}$, where $m \leq M$, retaining only pertinent paragraphs for evidence generation.

As shown in Figure 2, TIDES uses tailored prompts that improve LLM performance without fine-tuning. These prompts elicit expert-like reasoning in IT and computer science, capturing key technical details and justifications for paragraph se-

Backbone	Method	Associative Selection			Final Answer				
		Precision	**Recall**	**F1**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**BERT Score**	**Token F1 Score**
Flan-T5-xl	QASA	1.08	12.50	1.98	6.61	0.62	0.47	77.4	10.35
GPT-3.5 Turbo	QASA	0.36	50.00	0.71	8.50	2.52	5.90	75.70	10.25
	Self-RAG	-	-	-	8.73	1.39	5.77	79.93	10.95
	TIDES	**0.40**	**100.00**	**0.81**	**16.12**	**6.09**	**10.53**	**81.18**	**19.19**
GPT-4 Turbo	-	-	-	-	5.60	1.67	4.08	74.45	6.16
	QASA	0.25	71.43	0.50	5.75	1.12	3.84	78.31	6.48
	Self-RAG	-	-	-	7.25	0.94	4.65	78.43	6.62
	TIDES	**1.14**	**83.30**	**2.25**	**13.95**	**3.92**	**8.53**	**80.90**	**14.75**
Llama-3.1	-	-	-	-	6.96	2.32	5.32	77.63	7.71
	QASA	0.19	66.67	0.37	5.49	1.39	3.85	77.76	6.89
	Self-RAG	-	-	-	7.76	2.93	5.45	78.20	8.46
	TIDES	**0.38**	**100.00**	**0.75**	**11.33**	**3.39**	**7.55**	**79.34**	**12.31**

Table 1: Performance comparison of QA methods (QASA, Self-RAG, TIDES) integrated with different language models (Flan-T5-xl, GPT-3.5 Turbo, GPT-4 Turbo, Llama-3.1) on the TechQA dataset. Results show both Associative Selection metrics (Precision, Recall, F1) and Final Answer quality metrics (ROUGE scores, BERT Score, Token F1). TIDES consistently achieves the highest performance (in **bold**) across most metrics for all models, particularly enhancing GPT-4 Turbo's capabilities.

lection. The LLMs provide both binary relevance classifications and concise rationales for their decisions. Table 10 in Appendix B provides complete prompts.

3.3 Rationale Extraction

The rationale extraction step generates evidence $E = \{e_1, e_2, \ldots, e_m\}$, from selected paragraphs $\tilde{P}$. These evidence highlight key content relevant to the question q and form the basis for deriving the final answer a.

We extract evidence e_k from each paragraph $\tilde{p}_k \in \tilde{P}$ using tailored prompts (Figure 2) with a built-in refinement check. This check instructs the model to output "no" for paragraphs lacking relevant information, allowing us to remove them from the evidence set. This integrated filtering approach eliminates irrelevant paragraphs without additional processing, improving evidence accuracy.

3.4 Systematic Composition

The Systematic Composition is engineered to synthesize the final answer a from the evidence E accumulated in previous stages. This step involves formulating a concise and coherent response, emphasizing crucial extracted keywords while eliminating redundant text. Also, the prompts are specifically designed to include a "No Answer" option.

4 Result

We evaluate the effectiveness of TIDES across two distinct datasets: TechQA (Castelli et al., 2020) and E-Manual QA (Nandy et al., 2021). TechQA consists of IT support questions, each paired with 50 relevant documents, while the E-Manual QA dataset includes question-answer pairs from electronic device manuals, focusing on two subsets: smart TV remote controls and Samsung Galaxy S10. These datasets provide diverse technical contexts to evaluate TIDES's ability to handle domain-specific and complex queries effectively.

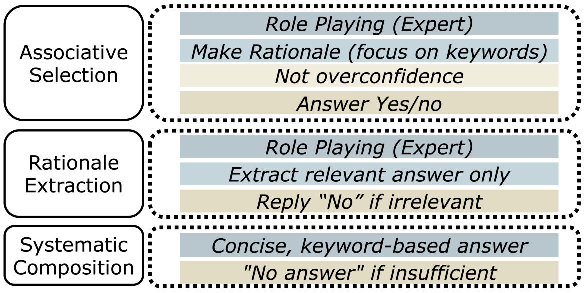

Figure 2: The workflow involves three stages: In the first stage, experts generate rationales by focusing on keywords and maintaining a balanced confidence level while providing yes/no answers. In the second stage, experts extract only the relevant answers and respond with "No" if the content is irrelevant. In the final stage, experts formulate concise, keyword-based answers and state "No answer" if the information is insufficient.

We compare four configurations across all backbone models (Flan-T5-xl, GPT-3.5 Turbo, GPT-4 Turbo, Llama-3.1-8b): Baseline (backbone model without augmentation), QASA, Self-RAG (a simplified zero-shot version), and TIDES. Notably, GPT-3.5 Turbo's standalone performance could not

Backbone	Method	Final Answer				
		ROUGE-1	**ROUGE-2**	**ROUGE-L**	**BERT Score**	**Token F1 Score**
GPT-4 Turbo	-	30.94	12.42	19.93	**85.51**	15.77
GPT-4 Turbo	QASA	19.92	6.00	12.00	83.50	9.56
GPT-4 Turbo	Self-RAG	21.24	5.41	12.18	83.23	10.08
GPT-4 Turbo	TIDES	**40.53**	**20.54**	**27.99**	78.32	**22.90**
Llama-3.1	-	28.42	13.32	20.34	85.88	15.39
Llama-3.1	QASA	11.69	3.98	8.32	82.42	6.45
Llama-3.1	Self-RAG	29.84	14.38	21.99	85.96	15.82
Llama-3.1	TIDES	**39.91**	**20.05**	**29.93**	**88.47**	**24.38**

Table 2: Performance comparison of different QA methods on the E-Manual dataset's smart TV remote questions (50 examples). The TIDES method significantly outperforms other methods across most metrics.

Backbone	Method	Final Answer				
		ROUGE-1	**ROUGE-2**	**ROUGE-L**	**BERT Score**	**Token F1 Score**
GPT-4 Turbo	-	22.66	6.94	15.13	84.35	12.79
GPT-4 Turbo	QASA	26.65	9.13	17.11	85.70	14.21
GPT-4 Turbo	Self-RAG	21.30	5.44	12.19	83.23	10.08
GPT-4 Turbo	TIDES	**45.06**	**21.29**	**33.77**	**87.46**	**27.16**
Llama-3.1	-	34.15	16.61	23.60	86.28	17.91
Llama-3.1	QASA	16.70	6.07	11.25	83.13	7.98
Llama-3.1	Self-RAG	33.65	**16.73**	23.09	85.91	16.85
Llama-3.1	TIDES	**34.93**	16.63	**24.47**	**87.58**	**19.92**

Table 3: Performance comparison of different QA methods on the E-Manual dataset's S10 questions (50 examples). The TIDES method significantly outperforms other methods across most metrics.

be fully evaluated due to its maximum token limit, restricting its ability to process longer documents without truncation.

Evaluation metrics include Precision, Recall, F1 Score, BERTScore (Zhang* et al., 2020), which measures semantic similarity, ROUGE Score (Lin, 2004) for assessing overlap with reference answers, and Token F1 for evaluating exact token matches. Experimental details are provided in Appendix C.

4.1 Quantitative Analysis

4.1.1 Performance on TechQA Dataset

Table 1 shows that TIDES consistently demonstrates improvements over both QASA and Self-RAG across all backbone models on the TechQA dataset.

TIDES with GPT-4 Turbo shows doubled ROUGE and Token F1 scores compared to baseline, efficiently extracting relevant, concise answers. The transition from QASA to TIDES yields notable gains in ROUGE-L and BERTScore, producing more contextually accurate responses. TIDES also effectively addresses GPT-3.5 Turbo's document length constraints, showing adaptability to technical QA challenges.

TIDES exhibits strong performance in identifying relevant paragraphs and providing precise answers, particularly for queries requiring nuanced contextual understanding.

4.1.2 Performance on E-Manual QA Dataset

Smart TV Remote Questions:

Table 2 presents the results for the smart TV remote subset. TIDES achieves substantial improvements across all metrics, showing better performance than QASA and Self-RAG in both ROUGE and Token F1 scores. Notably, when paired with Llama-3.1, TIDES records a ROUGE-L score of 29.93 and a Token F1 score of 24.38, demonstrating effective extraction of critical details from technical instructions.

However, the BERTScore was slightly lower than other metrics due to reference answers relying heavily on verbatim instructions, limiting semantic overlap. Despite this, TIDES consistently delivers accurate and concise responses tailored to technical queries.

S10 Questions:

The results for the Samsung Galaxy S10 subset are summarized in Table 3. TIDES maintains strong performance across all metrics, showing significant improvements compared to QASA and Self-RAG. For instance, using GPT-4 Turbo, TIDES achieves a ROUGE-1 score of 45.06 and a

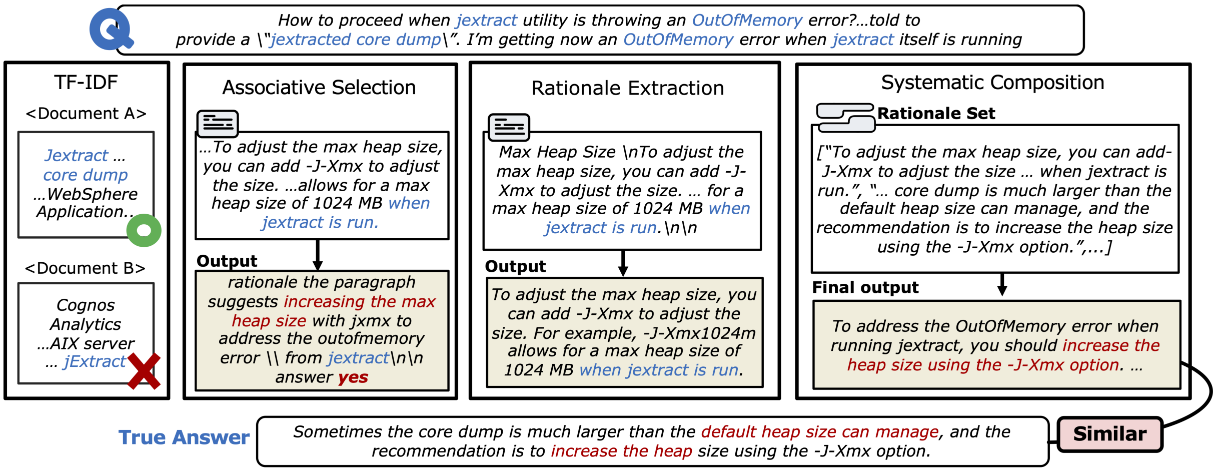

Figure 3: Qualitative analysis of TIDES's performance on a question with an available answer from the TechQA dataset. The example demonstrates the model's ability to identify relevant paragraphs, extract key information, and generate an accurate and concise answer by combining evidence from multiple sources.

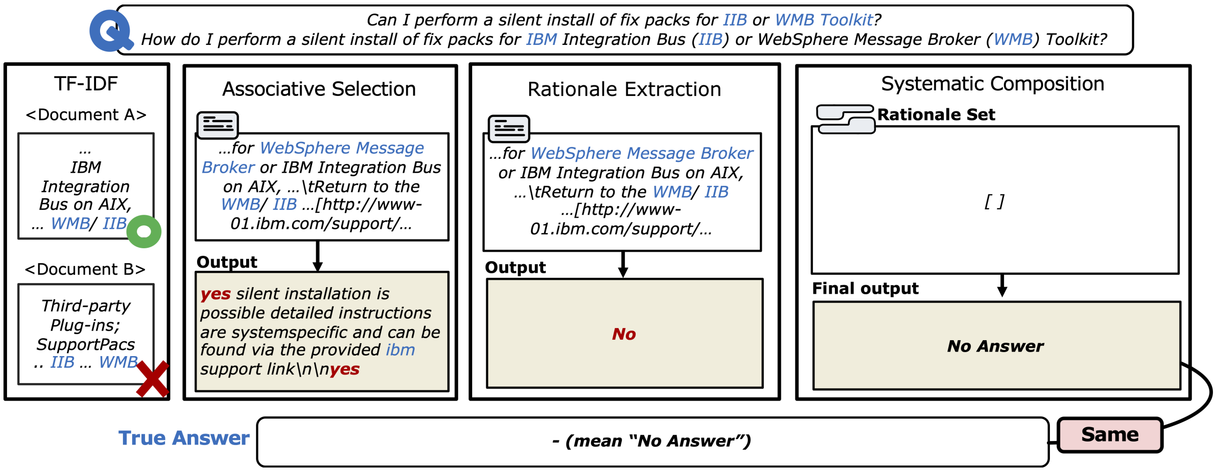

Figure 4: Qualitative analysis of TIDES's performance on a question without a valid answer in the given corpus in the TechQA dataset. The example demonstrates the model's ability to recognize the lack of sufficient information to generate a response, correctly identifying when the provided paragraphs do not contain the necessary evidence to answer the question and responding with "No Answer" accordingly.

Token F1 score of 27.16, both the highest among all tested configurations. Llama-3.1 also demonstrates robust performance, with TIDES showing results comparable to or better than Self-RAG across most metrics. While the ROUGE-2 score for TIDES with Llama-3.1 was slightly lower than Self-RAG, it performs better in ROUGE-L and Token F1, indicating effective balancing of relevance and precision.

These results indicate TIDES's capability to handle diverse technical contexts, effectively balancing computational efficiency and accuracy in producing relevant responses.

4.2 Qualitative Analysis

4.2.1 TechQA Dataset

To further evaluate TIDES, we conducted a qualitative analysis on TechQA (Figures 3, 4). For questions with valid answers, TIDES effectively identified relevant paragraphs and extracted key information to produce concise, accurate responses by combining evidence from multiple sources. In some cases, minor irrelevant content remained, leading to slightly verbose answers.

For questions without valid answers, TIDES consistently recognized the lack of sufficient information and returned "No Answer", correctly avoiding speculative or unsupported responses.

Question	How can I Encrypt SD card?
Ground-truth Answer	You can encrypt your optional memory card (not included) to protect its data... From Settings, tap **Biometrics and security >Encrypt or decrypt SD card**... Tap **Encrypt SD card** and follow the **prompts to encrypt all data on your memory card.**
GPT4-Turbo Answer	I'm sorry, but the provided text does not contain specific instructions on how to encrypt an SD card... If you're using an Android device, you might find the option to encrypt your SD card in the device's security settings... It's recommended to consult the user manual...
TIDES Answer	To encrypt your SD card, go to Settings, then tap on **'Biometrics and security.'**... Select **'Encrypt or decrypt SD card,'** choose **'Encrypt SD card,'** and follow the on-screen **prompts to encrypt all data on your memory card.**

Table 4: Comparison of model responses on the E-manual QA s10 dataset for "How can I encrypt an SD card?" TIDES provides precise, relevant answers, while GPT-4 Turbo offers a more general response.

Question	How to pair the TV with the Samsung Smart Remote?
Ground-truth Answer	**When you turn on the TV for the first time, the Samsung Smart Remote pairs to the TV automatically. If the Samsung Smart Remote does not pair to the TV automatically, point it at the remote control sensor of the TV**, and **then press and hold the and buttons simultaneously for 3 seconds or more. The Samsung Smart Remote may not be supported depending on the model or geographical area.**
GPT4-Turbo Answer	To find Caption Settings on your Samsung TV, follow these steps: 1. Press the Home button on your Samsung Smart Remote to open the Smart Hub. 2. Use the directional buttons on the remote to navigate to "Settings" and select it ... 3. In the Settings menu, navigate to "General." ...
TIDES Answer	To pair the TV with the Samsung Smart Remote, follow these steps: 1. **When you turn on the TV for the first time, the Samsung Smart Remote pairs to the TV automatically.** 2. **If the Samsung Smart Remote does not pair to the TV automatically, point it at the remote control sensor of the TV**. 3. **Press and hold the Return and Play/Pause buttons simultaneously for 3 seconds or more.** Note: The images, buttons, and functions of the Samsung Smart Remote may differ depending ... **The Samsung Smart Remote may not be supported depending on the model or geographical area.**

Table 5: Comparison between Ground-truth, GPT-4 Turbo, and TIDES responses for the question "How to pair the TV with the Samsung Smart Remote?" from the E-manual QA smart TV remote dataset. TIDES closely matches the Ground-truth with precise, relevant instructions, while GPT-4 Turbo deviates with unrelated content.

4.2.2 E-manual Dataset

Table 4 compares responses to "How can I encrypt an SD card?". TIDES accurately extracted relevant instructions from the electronic manual to generate a clear step-by-step answer, while GPT-4 Turbo produced generic suggestions lacking specificity.

Table 5 shows results for "How to pair the TV with the Samsung Smart Remote?". TIDES closely matched the ground truth with precise, actionable steps. In contrast, GPT-4 Turbo failed to provide relevant instructions and offered unrelated content. These results demonstrate TIDES's ability to process domain-specific technical documents and deliver accurate, contextually appropriate answers.

4.3 Ablation Study

We evaluated several similarity metrics (TF-IDF, BM25, Jaccard, Cosine) using the TechQA validation set. We measured how often correct documents appeared within the top 10 ranked results for each question. As shown in Table 6, TF-IDF consistently identified the correct documents in 81.81% of cases where answers existed, significantly outperforming other metrics. Based on these results, we adopted TF-IDF as our primary document filtering method.

Similarity Metric	Correct Documents in Top 10 (%)
TF-IDF	**81.81**
BM25	27.27
Cosine Similarity	63.64
Jaccard Similarity	45.45

Table 6: Comparison of similarity metrics on the TechQA validation dataset. The TF-IDF metric achieved the highest performance, with the correct document appearing in the top 10 results in 81.81% of cases where a correct answer existed.

4.4 Time complexity

Despite multiple stages, TIDES reduces computational overhead through early filtering and segmentation. TF-IDF and Associative Selection progressively reduce input size before reasoning, alleviating the quadratic complexity ($O(N^2)$) of transformer-based LLMs. Splitting documents of length N into n chunks of size L reduces complexity to $O(n \cdot L^2)$, which is especially beneficial for long technical documents.

This design also mitigates the "lost in the middle" problem (Liu et al., 2024a), where LLMs struggle with mid-context content. By working on smaller segments, TIDES ensures consistent access to relevant information throughout the document.

(a) GPT-3.5 Turbo: QASA vs. 3-stage TIDES

Model configuration	Runtime (s)
GPT-3.5 Turbo + QASA	222.62
GPT-3.5 Turbo + **TIDES (3-stage)**	134.68

(b) LLAMA-3.1: 3-stage TIDES vs. Parallel TIDES

Pipeline variant	Runtime (s)
LLAMA-3.1 + TIDES (3-stage)	307.15
LLAMA-3.1 + **TIDES-Parallel**	19.76

Table 7: Processing latency on the TechQA dataset (seconds, lower is better). Each value is the average runtime for six randomly selected questions. (a) Replacing the baseline QASA retrieval pipeline with the proposed 3-stage TIDES pipeline cuts latency by 39 % when using GPT-3.5 Turbo. (b) Further optimising TIDES by parallelising the Selective Association and Rationale Extraction stages yields a 15.5× speed-up on LLAMA-3.1. Blue rows highlight the faster configuration in each comparison.

Empirical results confirm these advantages. The "GPT-3.5 Turbo + QASA" baseline in Table 7 (a) refers to our implementation of the QASA structure (Lee et al., 2023) using prompted GPT-3.5 Turbo instead of the original fine-tuned T5 models. On TechQA, TIDES reduced GPT-3.5 Turbo's runtime from 222.62s to 134.68s (39% reduction), with accurate responses maintained. On Llama-3.1, we implemented a parallelized version of TIDES, parallelizing Associative Selection and Rationale Extraction, achieving a 15.5× speedup (307.15s to 19.76s, Table 7 (b)). This was achieved by combining the Associative Selection and Rationale Extraction stages into a single LLM call and processing multiple documents concurrently using an `asyncio` task manager with a thread pool executor, limiting concurrency to five to eight documents. This highlights TIDES's ability to substantially improve computational efficiency while preserving answer quality.

In summary, TIDES demonstrates that thoughtful multi-stage pipeline design can overcome LLM efficiency limitations, reducing overall time and ensuring reliable long-context processing.

4.5 Statistical Validation

Metric	Paired t-test (p)	Wilcoxon signed-rank (p)
BERT Score	0.12	**0.02**
ROUGE-1	0.15	0.09
ROUGE-2	0.89	0.21
ROUGE-L	0.23	0.07
Token F1	0.12	0.07

Table 8: Statistical comparison of TIDES vs. Self-RAG on Llama-3.1 using paired t-test and Wilcoxon tests. Only BERT Score shows significant improvement (p < 0.05, **bold**) under Wilcoxon test, with consistent improvements in other metrics.

We evaluated the statistical significance of performance differences between TIDES and Self-RAG on the Llama-3.1 backbone using paired t-tests and Wilcoxon signed-rank tests across metrics. As shown in Table 8, TIDES achieved significant improvements in BERT-Score F1 (p = 0.023), demonstrating its strength in generating semantically accurate responses. While other metrics did not show statistically significant differences, the consistent performance highlights TIDES's robustness and effectiveness in document-based QA.

4.6 Effect of Retrieval Size (n)

To justify the fixed choice n=30, we varied $n \in \{1, 5, \dots, 45\}$ on the first five questions of the Smart TV Remote dataset. Table 9 shows a clear quality–latency trade-off: metrics improve up to $n \approx 25-30$ and plateau thereafter, while runtime grows almost linearly with n. Hence n=30 offers the best overall balance used throughout the paper.

n	Token-F1	BERT	R-1	R-2	R-L	Time (s)
1	86.23	21.63	11.67	16.01	12.17	**0.73**
5	89.62	49.77	32.45	36.99	29.31	1.33
10	89.36	46.18	27.80	32.82	28.23	1.12
15	89.15	46.50	27.46	34.03	28.75	1.53
20	90.29	54.88	41.17	45.05	34.10	3.05
25	**90.83**	59.07	40.52	47.07	36.03	4.41
30	90.24	**60.56**	**43.37**	**48.21**	**36.29**	4.78
35	88.06	48.90	31.90	37.01	30.20	6.48
40	89.26	54.30	36.82	43.82	35.21	5.30
45	89.67	53.30	34.81	39.09	31.16	6.25

Table 9: Performance-latency trade-off for retrieval size n on Smart TV Remote questions. Quality metrics (Token-F1, BERT, ROUGE-1/2/L) improve up to $n \approx 30$ before plateauing; runtime increases with n. **Bold** indicates best per metric. Results are averaged over 5 questions.

We further validated TIDES's generalizability on medical domain data using the Long Health (Adams et al., 2024) dataset. Complete results are provided in Appendix D.

5 Conclusion

TIDES enhances traditional QASA by incorporating a specialized step for effective terminology management, significantly improving accuracy in identifying relevant paragraphs. By integrating TF-IDF with prompt-based LLMs, TIDES refines natural language understanding in technical domains like TechQA, enabling the system to handle complex queries with greater precision. This approach not only reduces processing time and costs through streamlined operations but also ensures consistent and concise responses. Consequently, TIDES marks an advancement in applying LLMs to specialized technical data, offering a robust solution that optimizes overall QA performance while addressing the specific challenges of technical terminology.

Limitations

The TIDES model shows significant potential, but to enhance its generalizability and applicability, it must be validated across a broader range of datasets, particularly by integrating domain-specific terminologies and reasoning. This will improve both the accuracy and interpretability of its responses. Additionally, refining similarity metrics to better suit specialized domains is crucial, as more nuanced measures will allow TIDES to retrieve and process relevant information with greater precision.

Because TIDES's retrieval stage is tuned for technical-domain QA using TF-IDF and associative selection, other domains (e.g., biomedical or legal) may benefit from alternative retrieval methods—such as domain-specific embeddings or custom indexing—to capture relevant concepts. Likewise, adapting TIDES to tasks beyond question answering (for instance, summarization, classification, or multi-hop reasoning) will require careful prompt redesign and potentially new pipeline components. Finally, errors in early stages (e.g., misclassified paragraphs or missed extractions) can propagate through the multi-stage pipeline, occasionally amplifying mistakes in the final output. Future work will involve systematic empirical evaluations across diverse domains and tasks, the exploration of hybrid retrieval strategies, and the development of adaptive prompting techniques and verification mechanisms to strengthen TIDES's robustness and versatility.

Acknowledgements

This work was supported by the project "Development of Core Technologies for AI-based Management and Maintenance of Gas-Oil Plants" (No. 21ATOGC161932-01) funded by the Ministry of Land, Infrastructure and Transport (MLIT) of the Republic of Korea; supported by the project "Development of AI-based Technologies for Digital Transformation and Information Extraction in Engineering Design" (No. RS-2022-00143813) funded by the Ministry of Trade, Industry and Energy (MOTIE) of the Republic of Korea; supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2024-00457216); and supported by a grant (22183MFDS431) from Ministry of Food and Drug Safety in 2025.

References

Abdelrahman Abdallah, Bhawna Piryani, and Adam Jatowt. 2023. Exploring the state of the art in legal qa systems. *Journal of Big Data*, 10(1):127.

Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. 2024. Longhealth: A question answering benchmark with long clinical documents. *arXiv preprint arXiv:2401.14490*.

Wasi Ahmad, Jianfeng Chi, Yuan Tian, and Kai-Wei Chang. 2020. PolicyQA: A reading comprehension dataset for privacy policies. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 743–749, Online. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. 2024. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, 6(8):852–863.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Hans Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott Mccarley, Mike Mccawley, et al. 2020. The techqa dataset. In *Annual Meeting of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Louie Giray. 2023. Prompt engineering with chatgpt: a guide for academic writers. *Annals of biomedical engineering*, 51(12):2629–2633.

Abraham Ittycheriah, Martin Franz, Wei-Jing Zhu, Adwait Ratnaparkhi, and Richard J Mammone. 2000. Ibm's statistical question answering system. In *TREC*.

Ehsan Kamalloo, Xinyu Zhang, Odunayo Ogundepo, Nandan Thakur, David Alfonso-hermelo, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Evaluating embedding APIs for information retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 518–526, Toronto, Canada. Association for Computational Linguistics.

Moniba Keymanesh, Micha Elsner, and Srinivasan Parthasarathy. 2021. Privacy policy question answering assistant: A query-guided extractive summarization approach. *arXiv preprint arXiv:2109.14638*.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. Chatgpt: Jack of all trades, master of none. *Information fusion*, 99:101861.

Bernhard Kratzwald and Stefan Feuerriegel. 2019. Putting question-answering systems into practice: Transfer learning for efficient domain customization. *ACM Transactions on Management Information Systems (TMIS)*, 9(4):1–20.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. 2023. Qasa: Advanced question answering on scientific articles. In *Proceedings of the 40th International Conference on Machine Learning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *Preprint*, arXiv:2202.01110.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Siyu Liu, Tongqi Wen, ASL Subrahmanyam Pattamatta, and David J Srolovitz. 2024b. A prompt-engineered large language model, deep learning workflow for materials classification. *Materials Today*, 80:240–249.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Farhad Moghimifar, Lizhen Qu, Yue Zhuo, Mahsa Baktashmotlagh, and Gholamreza Haffari. 2020. CosMo: Conditional Seq2Seq-based mixture model for zero-shot commonsense question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5347–5359, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Seon-Ok Na, Young-Min Kim, and Seung-Hwan Cho. 2022. Insurance question answering via single-turn dialogue modeling. In *Proceedings of the Second Workshop on When Creative AI Meets Conversational AI*, pages 35–41.

Abhilash Nandy, Soumya Sharma, Shubham Maddhashiya, Kapil Sachdeva, Pawan Goyal, and NIloy Ganguly. 2021. Question answering over electronic devices: A new benchmark dataset and a multi-task learning based QA framework. In *Findings of the*

Association for Computational Linguistics: EMNLP 2021, pages 4600–4609, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Jayr Pereira, Robson Fidalgo, Roberto Lotufo, and Rodrigo Nogueira. 2023. Visconde: Multi-document qa with gpt-3 and neural reranking. In *European Conference on Information Retrieval*, pages 534–543. Springer.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. `https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf`. OpenAI technical report.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140).

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ellen Riloff and Michael Thelen. 2000. A rule-based question answering system for reading comprehension tests. In *ANLP-NAACL 2000 workshop: reading comprehension tests as evaluation for computer-based language understanding systems*.

Eneko Ruiz, María Inés Torres, and Arantza del Pozo. 2023. Question answering models for human–machine interaction in the manufacturing industry. *Computers in Industry*, 151:103988.

Jon Saad-Falcon, Joe Barrow, Alexa Siu, Ani Nenkova, Seunghyun Yoon, Ryan A. Rossi, and Franck Dernoncourt. 2024. PDFTriage: Question answering over long, structured documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 153–169, Miami, Florida, US. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5).

Saptarshi Sengupta, Shreya Ghosh, Preslav Nakov, and Prasenjit Mitra. 2023. Can you answer this?–exploring zero-shot qa generalization capabilities in large language models (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16318–16319.

Rajesh Shrivastava, Abinash Pujahari, Simar Preet Singh, and Tanmay Bhowmik. 2022. Efficient question answering in chatbot using tf-idf and cosine similarity. In *Innovations in Information and Communication Technologies: Proceedings of ICIICT 2022*. Springer.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. Legal prompt engineering for multilingual legal judgement prediction. *arXiv preprint arXiv:2212.02199*.

Jiaqi Wang, Enze Shi, Sigang Yu, Zihao Wu, Chong Ma, Haixing Dai, Qiushi Yang, Yanqing Kang, Jinru Wu, Huawen Hu, et al. 2023. Prompt engineering for healthcare: Methodologies and applications. *arXiv preprint arXiv:2304.14670*.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Bo Qiao, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023. Empower large language model to perform better on industrial domain-specific question answering. In *Proceedings* *of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 294–312, Singapore. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yichi Zhang, Zhuo Chen, Yin Fang, Yanxi Lu, Li Fangming, Wen Zhang, and Huajun Chen. 2024. Knowledgeable preference alignment for LLMs in domain-specific question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 891–904, Bangkok, Thailand. Association for Computational Linguistics.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

Jie Zhou, Pei Ke, Xipeng Qiu, Minlie Huang, and Junping Zhang. 2023. Chatgpt: potential, prospects, and limitations. *Frontiers of Information Technology & Electronic Engineering*, pages 1–6.

A Detailed Overview of Related Work

A.1 Large-scale Language Models

Pre-trained language models have been explored continuously because of their effectiveness in diverse natural language tasks and their generalization performance. Word2vec (Mikolov et al., 2013) is a pre-trained word embedding to learn the representation for each word. However, Word2vec is hard to express the contextual information, and CoVe (McCann et al., 2017) and ELMO (Peters et al., 2018) utilize recurrent neural network structures to incorporate the context. With the development of Transformers, Transformer-based pre-trained language models, such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018), have started to emerge. BERT utilizes the bi-directional encoder structure to focus on the left-to-right and right-to-left contextual information, while GPT adopts a decoder structure and causal masking to mimic the generative nature. In addition to encoder and decoder-like structures, T5 (Raffel et al., 2020) and BART (Lewis et al., 2020a) adopt encoder-decoder structures to enhance the performance with diverse pretext tasks. However, the encoder-decoder structure requires twice the number of parameters. Besides, it has been known that the decoder-like structure, including GPT, shows superior performance on diverse generative tasks, such as translation and question-answering, compared to the encoder-like structure. Therefore, recent large-scale language models such as Llama-3 (Touvron et al., 2023) adopt a decoder-like structure.

Because of the generalization performance of LLMs, there have been diverse research works to utilize LLMs for QA tasks. LLMs are known to possess a wealth of knowledge because they have been trained on large-scale datasets spanning various topics and domains (Sengupta et al., 2023). With such abilities, many studies are building QA systems utilizing the inferential capabilities of LLMs without additional training (Moghimifar et al., 2020). Prompt engineering is one of the most promising methods utilizing the inherent knowledge of LLMs. Prompt engineering assists in enabling LLMs to produce more accurate outputs in the direction desired by the user for the input query (Giray, 2023). Due to its ability to achieve decent performance without additional training, it is desired to be utilized for QA in various fields such as healthcare (Wang et al., 2023), legal judgment (Trautmann et al., 2022), and manufacturing (Liu et al., 2024b). However, while LLMs are generally well-generalized for tasks including QA, they have several critical limitations. First, they cannot answer questions about data beyond a specific time or personal data (Zhou et al., 2023). Furthermore, some studies point out that while LLMs like ChatGPT demonstrate strong capabilities across many tasks, they may not perform best on specific tasks (Kocoń et al., 2023). To overcome these limitations, we propose a methodology called TIDES to enhance the performance of LLMs in QA tasks.

A.2 Document-based QA

Traditional QA systems were composed of various ways including keyword-based search, rule-based search, and statistical-based search. First, there is TF-IDF (Salton and Buckley, 1988), which evaluates the importance of keywords in documents to find highly relevant answers to questions, considering both word frequency and inverse document frequency. Also, Riloff, Ellen, and Michael Thelen. (Riloff and Thelen, 2000) propose a heuristic rule-based QA system for reading comprehension tests. TREC (Ittycheriah et al., 2000) answers the question with the maximum entropy modeling method. However, Rajpurkar, Pranav, et al. (Rajpurkar et al., 2018) point out that the performance of previous models significantly lags behind human labeling in terms of performance, proposing the SQuAD dataset (Rajpurkar et al., 2016).

With the BERT (Devlin et al., 2019) emerging, there have been attempts to utilize it for QA. Datasets from various domains have been proposed to evaluate whether BERT can solve QA tasks such as commonsense (Talmor et al., 2019), policy (Ahmad et al., 2020), and manufacturing (Castelli et al., 2020). BERT-based models have demonstrated good performance in QA across various domains, but they also have a crucial bound. Their training involves predicting the starting and ending indices of answers within documents, limiting their ability to provide comprehensive answers to questions from various document parts.

A.3 Document-based QA with LLMs

As LLMs advance, several studies aim to overcome the aforementioned limitations by leveraging the reasoning abilities of LLMs to find answers to questions within documents (Brown et al., 2020). Next, Retrieval Augmented Generation (RAG) is proposed in various works (Lewis et al., 2020b). RAG is a method that combines information retrieval with text generation to enhance the quality of generated responses (Li et al., 2022). The strength of RAG lies in its ability to enhance the quality and reliability of generated responses by leveraging retrieved information. However, despite the expectation that RAG generates plausible answers, limitations exist, such as hallucination and inconsistency not being guaranteed (Zhao et al., 2024).

To overcome these limitations, recent QA with LLMs aims to aggregate information spread across multiple documents or paragraphs to provide comprehensive answers. Pereira, Jayr, et al. (Pereira et al., 2023) propose a method for decomposing the input question into simpler sub-questions, retrieving answers for each sub-question, and then synthesizing this information to generate a comprehensive answer. Also, there is a work that suggests a methodology that provides accurate answers through structured retrieval of extracted metadata from documents (Saad-Falcon et al., 2024).

Specifically, QASA proposes a QA method dividing it into three stages - Associative Selection, Rationale Extraction, and Systematic Composition. They fine-tune T5 (Raffel et al., 2020) at each stage, using different datasets, and then answer the question. First, at Associative Selection, QASA determines whether each paragraph of the document is relevant to the input question. Then, in Rationale Extraction, the relevance of each paragraph to the question is extracted using LLM. At last, at Systematic Composition, the LLM generates comprehensive answers utilizing the extracted paragraphs and rationales from Rationale Extraction. By fine-tuning T5 for each stage with a small dataset, QASA demonstrates improved performance compared to InstructGPT (Ouyang et al., 2022), a model ten times larger than T5 in parameters.

However, we empirically identify several shortcomings with QASA. Firstly, collecting datasets for fine-tuning T5 for each stage of QASA is not a trivial task. Also, we observe that in the Associative Selection stage, QASA tends to provide answers indicating relevance even from many irrelevant paragraphs. To address these issues, we propose QASA with several improvements. Firstly, in the Associative Selection stage, we encourage QASA to not only assess the relevance between the question and the paragraph but also inquire about the reason for the relevance. As a result, our model avoids overly affirmative responses, indicating that many paragraphs are relevant. Furthermore, in the second stage, we also double-check for irrelevant paragraphs, and these modifications result in reducing the overall response time by nearly 90%.

A.4 Domain Specific Question Answering

As document-based QA methodologies advance, there are also many studies focused on building QA systems for specific domains including private policy (Keymanesh et al., 2021), legal (Abdallah et al., 2023), insurance (Na et al., 2022) and manufacturing (Ruiz et al., 2023). There are several notable characteristics of using domain-specific QA. At First, collecting domain-specific datasets can be extremely challenging, making it difficult for users to fine-tune models and build QA systems (Kratzwald and Feuerriegel, 2019). Also, domain-specific vocabulary in questions may not be understood by generalized models or specialized inference processes may be required depending on the domain (Zhang et al., 2024). At last, in domain-specific document QA, since it must be able to substitute for expert judgment, the factuality of LLM becomes even more crucial (Augenstein et al., 2024).

To manage the above issues, various works have been proposed in QA systems. Tian, Katherine, et al. (Tian et al., 2023) propose a methodology that fine-tunes a model for factuality. Also, there is a work that asks the same question multiple times, which requires the same answer. However, there are shortcomings with using the two aforementioned studies to supplement the factuality of LLM at the QA task. Firstly, fine-tuning the model requires significant cost and time in data mining, and training. Moreover, repeatedly querying the model incurs substantial costs when using models provided as APIs, such as GPT. Instead, there are also works aiming at cost-efficiently improving the robustness of domain-specific terms and factuality. First, there is a branch that generates answers by decoding the outputs of layers preceding the final output layer rather than the last output layer. Additionally, there are studies focused on reducing the pool of potential answer candidates that can be generated. Shrivastava, Rajesh, et al. (Shrivastava et al., 2022) propose an effective chatbot system applying TF-IDF to prune answerable data, thus reducing the number of potential answers. Also, there is a work that overcomes the lack of domain-specific words using TF-IDF. In this context we find that QASA performs poorly on questions containing specialized terminology. To address this issue, motivated by previous works, for the first stage, we apply TF-IDF to compare the similarity between the question and specific paragraphs, using only the highly similar documents for QA.

B Prompts

Step	Prompt
Associative Selection	Instruction: You are an **expert** in IT and computer science. Provide a **rationale** in 20 words or less for how the paragraph relates to the question, focusing on key technical details. Finally, give a clear yes or no answer **avoiding overconfidence**. Question: (question) Context: (paragraph) Output: (rationale + (yes || no))
Rationale Extraction	Instruction: You are an **expert** in IT and computer science. Given a question and a context, if the context contains information that directly answers the question or provides clear supporting evidence, **extract** only the relevant section. Do not include any additional explanations or comments. If **no relevant information is found, simply respond "No"**. Question: (question) Context: (selected paragraph) Output: (extract rationales || No)
Systematic Composition	Instruction: Compose a **concise answer** to the question using the most relevant keywords and phrases from the rationales. Aim for a natural response while still aligning closely with the rationales. If the rationales do not sufficiently address the question, **respond with "No answer"**. Question: (question) Context: (evidence set) Output: (answer)

Table 10: Prompts used in the TIDES framework for each step of the QA process. The instructions assign an expert role to the model, encouraging it to generate rationales, avoid overconfidence, and respond with "no" to unfounded sentences, ultimately optimizing LLM's performance in identifying relevant information, generating accurate rationales, and composing comprehensive answers.

C Experimental details

To evaluate the effectiveness and robustness of TIDES across various types of technical documentation, we conducted experiments using two datasets:

- **TechQA** (Castelli et al., 2020): This dataset consists of IT support questions sourced from IBM's internal forums, each paired with relevant technical answers. It primarily focuses on enterprise-level IT problem-solving scenarios. If the documentation cannot substantiate an answer, the question is marked as "Non-Answerable" with an "N" label.

- **E-Manual QA** (Nandy et al., 2021): This dataset comprises question-answer pairs extracted from electronic manuals of consumer electronic devices such as smartphones and smart TVs. It includes a wide range of user-centered technical questions, featuring both curated questions and real user inquiries from online forums.

For measuring TIDES's enhancements, we applied the following LLMs: Flan-T5-xl(Raffel et al., 2020), GPT-3.5 Turbo, GPT-4 Turbo and Llama-3.1. Additionally, we compared the performance of TIDES against traditional QASA and a simplified version of Self-RAG. Unlike the standard Self-RAG, which typically involves multiple retrieval iterations and fine-tuning, our version operates in a zero-shot setting with a single retrieval pass.

In all experiments, we successfully skipped the fine-tuning step by addressing the problem through prompt engineering without any additional training. However, to ensure fidelity to the questions, we adjusted the temperature setting to zero in experiments involving TIDES.

We employed a wide range of metrics for evaluating the effectiveness and accuracy of the models.

- **Precision**, **Recall**, and **F1 Score**: To assess the accuracy and comprehensiveness of the obtained responses.
- **BERTScore** (Zhang* et al., 2020): To assess the semantic resemblance of the produced and given responses.
- **ROUGE Score** (Lin, 2004): To evaluate the similarity of n-grams in the generated responses compared to the reference answers to understand fluency and relevance.
- **Token F1**: To evaluate precision in technical contexts by measuring accuracy at the token level

D Long Health Dataset

The LongHealth (Adams et al., 2024) dataset is a benchmark designed to evaluate LLMs' capabilities in processing extensive clinical documentation. It consists of 20 detailed fictional patient cases across various diseases, with each case containing 5,090-6,754 words. The benchmark includes 400 multiple-choice questions across different tasks including information extraction, negation understanding, and chronological information sorting.

D.1 Performance on Longhealth Dataset

Model	Accuracy (%)
TIDES (Llama-3.1)	68.50
SelfRAG (Llama-3.1)	45.25
QASA (Llama-3.1)	52.25
Llama-3.1	22.00

Table 11: Performance comparison on LongHealth dataset's Task 3, which evaluates models' ability to identify unanswerable questions in clinical documentation. All models use Llama-3.1 as the base architecture. TIDES demonstrates superior performance (68.50% accuracy), significantly outperforming the baseline model (22.00%) as well as other retrieval-augmented approaches. Higher accuracy indicates better capability to avoid hallucination when information is insufficient.

LongHealth's Task 3 evaluates a model's ability to recognize insufficient information and avoid hallucination - a critical capability in clinical contexts. As shown in Table 11, TIDES (68.50%) outperforms the base Llama-3.1 model (22.00%) in identifying unanswerable questions. This improvement demonstrates how TIDES's evidence evaluation mechanisms effectively prevent hallucination when information is not available, which is particularly critical in healthcare applications where incorrect information can have serious consequences.

TIDES excels at identifying unanswerable questions, outperforming both baseline LLMs and retrieval-augmented approaches. For example, Table 12 illustrates a case where the question "What chronic neuroinflammatory disease does Mrs. Sample have?" - for which the correct answer is "Cannot be answered with the provided documents" - is posed. In this case, TIDES correctly responds that the information is insufficient, while the vanilla Llama-3.1 model hallucinates a specific diagnosis. This demonstrates TIDES's strong evidence evaluation and ability to avoid speculation, particularly valuable in high-stakes domains like healthcare.

Question	What chronic neuroinflammatory disease does Mrs. Sample have?
Ground-truth Answer	Cannot be answered with provided documents.
Llama-3.1 Answer	Mrs. Sample has multiple sclerosis, a chronic neuroinflammatory disease.
TIDES(Llama-3.1) Answer	The provided documents do not contain sufficient information to determine Mrs. Sample's condition. Therefore, the question cannot be answered.

Table 12: TIDES successfully avoids hallucination by correctly recognizing when no sufficient evidence exists to answer the question. Example from the LongHealth dataset.

D.2 Detailed Analysis on Long Health

Table 13 demonstrates TIDES's three-stage reasoning process on a clinical question from the Long Health dataset, highlighting how each stage contributes to the final conclusion.

Question	Which chronic, neuroinflammatory disease does Mrs. Sample suffer from?
Stage 1: Associative Selection	
Original Paragraph	**Medical History:** Mr. Wells initially presented with right upper abdominal pain, which led to the discovery of multiple intrahepatic masses in liver segments IV, VII/VIII. Subsequent investigations confirmed the diagnosis of HCC. He also suffers from chronic obstructive pulmonary disease (COPD)...
Relevance Rationale	Mentions Mr. Paul Wells' medical history, including COPD and HCC
Decision	Relevant (YES) [Note: Incorrectly classified]
Stage 2: Rationale Extraction	
Extracted Evidence	Mr. Wells initially presented with right upper abdominal pain, which led to the discovery of multiple intrahepatic masses in liver segments IV, VII/VIII. Subsequent investigations confirmed the diagnosis of HCC. He also suffers from chronic obstructive pulmonary disease (COPD), emphysema, and respiratory insufficiency requiring home oxygen therapy.
Stage 3: Systematic Composition	
Final Answer	There is no information provided about Mrs. Sample in the provided documents. The information is about Mr. Wells and Mrs. Anderson.
Ground-truth	Question cannot be answered with provided documents

Table 13: TIDES's three-stage process on a challenging question from the Long Health dataset. In Stage 1, a paragraph about Mr. Wells is incorrectly identified as relevant based on his COPD diagnosis (a chronic condition, though not neuroinflammatory). In Stage 2, this evidence about Mr. Wells is extracted. Despite these intermediate errors, in Stage 3, TIDES correctly synthesizes that there is no information about Mrs. Sample's condition in the documents. This demonstrates the system's ability to recover from intermediate errors through its multi-stage verification approach.

This example highlights a key strength of TIDES's multi-stage architecture: error resilience. Even when Stage 1 incorrectly classifies paragraphs about Mr. Wells as relevant (likely because COPD is a chronic condition, albeit not neuroinflammatory), and Stage 2 extracts evidence about his condition, the final Systematic Composition stage correctly synthesizes that the documents contain no information about Mrs. Sample's neuroinflammatory disease. The system's final answer aligns with the ground truth that this question cannot be answered with the provided documents, demonstrating TIDES's capacity to identify unanswerable questions—a critical capability in clinical contexts where acknowledging information gaps is essential for preventing dangerous hallucinations.