Imagine All The Relevance: Scenario-Profiled Indexing with Knowledge Expansion for Dense Retrieval

Sangam Lee[♡] **Ryang Heo**[♡] **SeongKu Kang**[♣] **Dongha Lee**[♡]
[♡]Yonsei University [♣]Korea University
{salee, ryang1119,donalee}@yonsei.ac.kr seongkukang@korea.ac.kr

Abstract

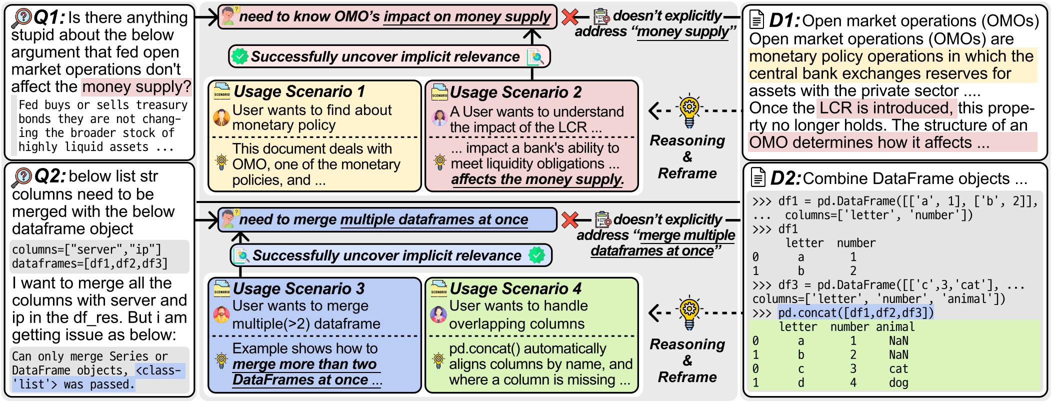
Existing dense retrieval models struggle with reasoning-intensive retrieval task as they fail to capture implicit relevance that requires reasoning beyond surface-level semantic information. To address these challenges, we propose Scenario-Profiled Indexing with Knowledge Expansion (SPIKE), a dense retrieval framework that explicitly indexes implicit relevance by decomposing documents into scenario-based retrieval units. SPIKE organizes documents into scenario, which encapsulates the reasoning process necessary to uncover implicit relationships between hypothetical information needs and document content. SPIKE constructs a scenario-augmented dataset using a powerful teacher large language model (LLM), then distills these reasoning capabilities into a smaller, efficient scenario generator. During inference, SPIKE incorporates scenario-level relevance alongside document-level relevance, enabling reasoning-aware retrieval. Extensive experiments demonstrate that SPIKE consistently enhances retrieval performance across various query types and dense retrievers. It also enhances the retrieval experience for users through scenario and offers valuable contextual information for LLMs in retrieval-augmented generation (RAG).

1 Introduction

Information retrieval (IR) systems are essential for helping users find relevant information within the overwhelming volume of available data. Over the years, dense retrieval (Karpukhin et al., 2020; Khattab & Zaharia, 2020) has emerged as a dominant approach. It employs pre-trained language models (PLMs) to encode queries and documents into shared vector spaces, enabling a deeper understanding of their semantic relationships. Despite these advancements, there remain fundamental challenges in IR.

One of the most significant challenges is that **dense retrieval struggles to capture deeper implicit relevance beyond surface-level semantic.** Recently, BRIGHT (Su et al., 2024), a benchmark for reasoning-intensive retrieval tasks, has been proposed. Unlike traditional IR benchmarks such as BEIR (Thakur et al., 2021) and MTEB (Muennighoff et al., 2022), it requires intensive reasoning to uncover implicit relevance between a query and relevant documents, which cannot be captured through simple keyword or surface-level semantic information. For example, in Figure 1 (Upper), *Q1* asks about the impact on Open Market Operations (OMOs) on money supply, whereas its relevant document *D1* doesn't explicitly address about it. Instead, it discusses OMOs' influence on the Liquidity Coverage Ratio (LCR), which in turn affects money supply. To uncover implicit relevance (OMO → LCR → Money Supply), it is **necessary to reason from the LCR-related information** of *D1*, which discusses regulatory policies and financial mechanisms. However, existing dense retrievers lack the capability to perform reasoning, and thus fail to uncover such implicit relationships. As a result, they struggle with reasoning-intensive retrieval task. (Su et al., 2024).

This limitation becomes even more pronounced when they handle query-document pairs of significantly different formats, such as code and natural language. As shown in Figure 1 (Lower), *D2* provides the necessary information for *Q2* only in the form of code examples (e.g., "pd.concat([df1, df2, df3])"). In such cases, bridging the semantic gap between the


need to know OMO's impact on money supply
doesn't explicitly address "money supply"
Successfully uncover implicit relevance
Reasoning & Reframe
need to merge multiple dataframes at once
doesn't explicitly address "merge multiple dataframes at once"
Successfully uncover implicit relevance
Reasoning & Reframe

Q1: Is there anything stupid about the below argument that fed open market operations don't affect the money supply?

```
Fed buys or sells treasury
bonds they are not chang-
ing the broader stock of
highly liquid assets ...
```

Q2: below list str columns need to be merged with the below dataframe object

```
columns=["server","ip"]
dataframes=[df1,df2,df3]
```

I want to merge all the columns with server and ip in the df_res. But i am getting issue as below:

```
Can only merge Series or
DataFrame objects, <class-
'list'> was passed.
```

Usage Scenario 1
User wants to find about monetary policy
This document deals with OMO, one of the monetary policies, and ...

Usage Scenario 2
A User wants to understand the impact of the LCR ...
... impact a bank's ability to meet liquidity obligations ... ***affects the money supply.***

Usage Scenario 3
User wants to merge multiple(>2) dataframe
Example shows how to ***merge more than two DataFrames at once*** ...

Usage Scenario 4
User wants to handle overlapping columns
pd.concat() automatically aligns columns by name, and where a column is missing ...

D1: Open market operations (OMOs)
Open market operations (OMOs) are monetary policy operations in which the central bank exchanges reserves for assets with the private sector .... Once the LCR is introduced, this property no longer holds. The structure of an OMO determines how it affects ...

D2: Combine DataFrame objects ...

```
>>> df1 = pd.DataFrame([['a', 1], ['b', 2]],
...  columns=['letter', 'number'])
>>> df1
   letter  number
0     a       1
1     b       2
>>> df3 = pd.DataFrame([['c',3,'cat'], ...
columns=['letter', 'number', 'animal'])
>>> pd.concat([df1,df2,df3])
    letter  number animal
0     a       1     NaN
1     b       2     NaN
0     c       3     cat
1     d       4     dog
```

Figure 1: Existing retrieval methods fail to capture implicit relevance which requires intensive reasoning, as they encode document into single vector without any reasoning. In contrast, SPIKE introduces scenario, explicitly modeling how a document establishes relevance to potential information needs.

natural language query and the code-based document requires reasoning over specific parts of the code to uncover implicit relevance. However, existing dense retrieval models are unable to effectively address this discrepancy because they are primarily trained on natural language query-document pairs (Reimers & Gurevych, 2019; Xiao et al., 2023; Meng et al., 2024), making them ill-suited for bridging the gap between them.

In this work, we propose **S**cenario-**P**rofiled **I**ndexing with **K**nowledge **E**xpansion (**SPIKE**), a dense retrieval framework that explicitly indexes potential implicit relevance within documents. The key idea of SPIKE is to reframe document representations into **hypothetical retrieval scenarios**, where each scenario encapsulates the reasoning process required to uncover implicit relevance between a hypothetical information need and the document content. As illustrated in Figure 1, SPIKE organizes document knowledge into hypothetical retrieval scenarios, which are considered alongside the document during the retrieval process. This approach 1) enhances retrieval performance by explicitly modeling how a document addresses hypothetical information needs, capturing implicit relevance between query and document. It also 2) effectively connects query-document pairs across different formats such as code snippets, enabling semantic alignment despite format differences. Additionally, it 3) enhances the retrieval experience for users by providing useful information while also serving as valuable context for LLMs in RAG settings.

Specifically, SPIKE consists of the following three steps: 1) constructing a scenario-augmented training dataset using a teacher LLM to generate high-quality supervision, 2) employing scenario distillation to transfer the reasoning capabilities of teacher LLM into a smaller, more efficient scenario generator, and 3) using the trained scenario generator to formulate structured retrieval scenarios for documents. During inference, SPIKE considers scenario-level relevance alongside document-level relevance to retrieve the relevant documents. Our extensive experiments demonstrate that SPIKE not only improves retrieval performance but also enhances the retrieval experience for users. Additionally, we demonstrate that SPIKE serves as a valuable additional context for LLMs in RAG settings. For reproducibility, our codes are publicly available at the anonymous github repository.[1]

The main contributions of our work are summarized as follows:

- We propose SPIKE, a dense retrieval framework that decomposes documents into scenarios. These scenarios enable effective retrieval by capturing implicit relevance.
- Our extensive experiments show that SPIKE consistently improves performance across diverse retrieval models, query types and document types.
- SPIKE helps users by providing explanations that make retrieved results easier to understand, while also making it easier for LLMs to generate accurate answers in RAG.

[1] https://github.com/augustinLib/SPIKE

2 Related Works

Reasoning-intensive retrieval. Traditional retrieval benchmarks (Thakur et al., 2021; Muennighoff et al., 2022) have largely focused on surface-level information-seeking queries where simple keyword or semantic matching-based retrieval is often sufficient. To address this limitation of traditional retrieval benchmarks, Su et al. (2024) propose BRIGHT, a benchmark that requires reasoning to retrieve relevant documents for a query. Su et al. (2024) and Niu et al. (2024) propose LLM-based query expansion and reranking as potential solutions for reasoning-intensive retrieval task. However, they have several limitations. First, LLM-based query expansion and reranking introduce significant computational overhead. Since these approaches require running inference on inefficient LLMs (> 8B parameters) for every query, they are computationally expensive and lead to high latency. Second, rerankers are dependent on the first-stage retrieval performance. If the first-stage retrieval fails to retrieve relevant documents, even a strong reranker cannot recover them. This dependency prevents rerankers from fully addressing the challenges of reasoning-intensive retrieval. Overall, these limitations highlight the importance of improving first-stage retrieval for reasoning-intensive retrieval tasks, though this area remains largely underexplored.

Document expansion & organization. Prior works have improved retrieval performance by appending pseudo queries (Nogueira et al., 2019; Chen et al., 2024), summaries (Jeong et al., 2021), or keyphrases (Boudin et al., 2020) to the original document. Since these approaches are performed at the indexing stage, they do not introduce additional inference-time overhead. Another line of works replace the original document representations with more effective retrieval units, such as summaries (Sarthi et al., 2024) or propositions (Chen et al., 2023). While these approaches refine document representations, they are limited in handling implicit information that cannot be addressed through simple semantic matching. As a result, they struggle in reasoning-intensive retrieval task. Additionally, the models used in these methods are typically trained only on natural language documents, they cannot be directly applied to non-natural language documents like code snippet.

3 Proposed Method: SPIKE

In this section, we present a dense retrieval framework, **S**cenario-**P**rofiled **I**ndexing with **K**nowledge **E**xpansion (SPIKE), which introduces a scenario-profiled retrieval to explicitly index potential implicit relevance. The overall framework is illustrated in Figure 2.

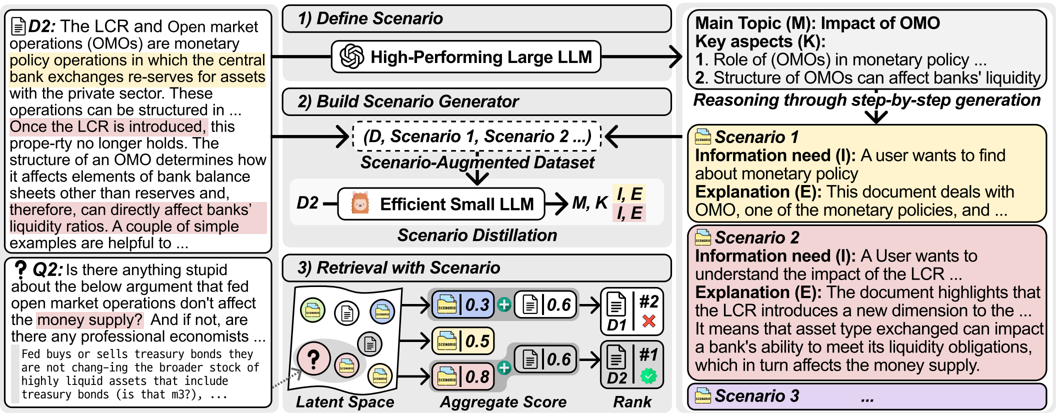
3.1 Scenario: Reasoning format for modeling implicit relevance

The first step of our SPIKE framework is to define the concept of a scenario, which serves as a structured reasoning format used to explicitly model the implicit relevance between a document and potential information needs. SPIKE reframes each document into multiple scenarios, each representing a distinct reasoning path that explains how the document could satisfy a hypothetical information need. Through scenario generation, SPIKE uncovers the diverse forms of implicit relevance a document may hold. To generate meaningful scenarios, we employ an LLM-driven reasoning approach that analyzes the document and constructs each scenario through a step-by-step process involving the following scenario components:

Main topic (M). To construct meaningful scenarios, we first identify the main topic of the document, which serves as a high-level summary of its content. This ensures that subsequent scenario components remain grounded in the document's overall theme, preventing them from diverging too far from its core subject.

Key aspects (K). Then, we extract key aspects that capture the diverse multi-aspects of the document's content. Key aspects provide a more detailed breakdown of the document's content compared to the main topic, capturing the diverse and specific information embedded within the document. By explicitly listing various details at this step, subsequent steps can produce a broader range of information, ensuring diverse scenario coverage.

Information needs (I). The next step is to generate information needs that reflect the potential retrieval intents a document can address. Specifically, we generate diverse infor-


1) Define Scenario
High-Performing Large LLM
2) Build Scenario Generator
(D, Scenario 1, Scenario 2 ...)
Scenario-Augmented Dataset
D2
Efficient Small LLM
M, K
I, E
I, E
Scenario Distillation
3) Retrieval with Scenario
0.3
0.6
0.5
0.8
0.6
#2
D1
#1
D2
Latent Space
Aggregate Score
Rank
Reasoning through step-by-step generation
Scenario 1
Scenario 2
Scenario 3 ...

D2: The LCR and Open market operations (OMOs) are monetary policy operations in which the central bank exchanges re-serves for assets with the private sector. These operations can be structured in ... Once the LCR is introduced, this prope-rty no longer holds. The structure of an OMO determines how it affects elements of bank balance sheets other than reserves and, therefore, can directly affect banks' liquidity ratios. A couple of simple examples are helpful to ...

Q2: Is there anything stupid about the below argument that fed open market operations don't affect the money supply? And if not, are there any professional economists ...

```
Fed buys or sells treasury bonds they
are not chang-ing the broader stock of
highly liquid assets that include
treasury bonds (is that m3?), ...
```

Main Topic (M): Impact of OMO
Key aspects (K):
1. Role of (OMOs) in monetary policy ...
2. Structure of OMOs can affect banks' liquidity

Information need (I): A user wants to find about monetary policy
Explanation (E): This document deals with OMO, one of the monetary policies, and ...

Information need (I): A User wants to understand the impact of the LCR ...
Explanation (E): The document highlights that the LCR introduces a new dimension to the ... It means that asset type exchanged can impact a bank's ability to meet its liquidity obligations, which in turn affects the money supply.

Figure 2: Overview of SPIKE framework. (1) SPIKE define Scenario and generate it with high-performing large LLM. (2) Then, it construct scenario-augmented training set, and use this to optimize the efficient student LLM. During inference, (3) SPIKE considers scenario-level relevance alongside document-level relevance to retrieve the documents.

mation needs that describe situations in which the given document can effectively address hypothetical information requests based on the main topic and each key aspects.

Explanations (E). For each generated information need, we generate an explanation that explicitly convey the connection between the document and the generated information need. These explanations serve as the core component for modeling relevance, as they describe why and how a document provides the necessary information to satisfy a given information need. This step ensures that each scenario captures implicit relevance by explicitly linking the document's content to the generated information need.

By generating components autoregressively, SPIKE uncovers potential implicit relevance within a document, ultimately modeling it through *Explanations (E)*. Leveraging E for retrieval enables SPIKE to overcome limitations of existing dense retrieval models, effectively capturing implicit relevance through explicit reasoning. However, indexing only E may lead to dense vector representations that fail to fully capture the document's overall context. To mitigate this issue, SPIKE incorporates the *Main topic (M)* component along with E, indexing their combination *(M+E)*. The main topic provides a high-level anchor that maintains coherence with the document's overall content, while explanations explicitly model the reasoning processes essential for identifying implicit relevance. This balanced approach ensures that retrieval representations remain both contextually grounded and reasoning-aware, ultimately enhancing retrieval performance more effectively.

3.2 Scenario generator & Scenario Indexing

Since scenario generation requires strong reasoning capabilities, high-performing LLMs like GPT-4o are essential for producing high-quality scenarios. However, applying such models to an entire corpus is computationally expensive and impractical for large-scale corpus. While smaller open-source models are more efficient, they often lack the reasoning ability needed for scenario generation. To address this, we first (1) use a high-performing LLM to construct a scenario-augmented training dataset with high-quality supervision, then (2) employ scenario distillation to train a smaller model, effectively transferring the reasoning capabilities of large LLMs into a small scenario generator.

Scenario-augmented training dataset. To train an effective scenario generator, we first construct a scenario-augmented training dataset $\mathcal{D} = \{(d, \tilde{S}^d)_i\}_{i=1}^N$, where each document d is paired with a sequence of scenarios $\tilde{S}^d = \{\tilde{s}_1, \dots \tilde{s}_k\}$. Since generating high-quality scenarios demands reasoning that goes beyond basic text generation, we leverage high-performing LLM such as GPT-4o to construct a scenario-augmented training dataset.

Scenario distillation. Once the scenario-augmented training dataset is constructed, we train a smaller scenario generator to efficiently produce reasoning-driven scenarios. Specifically, we minimize the following distillation loss:

$$\mathcal{L}_{\text{Distillation}} = -\sum \log P(\tilde{S}^d \mid \mathcal{I}, d; \theta) \tag{1}$$

where θ denotes the parameters of the scenario generator model, trained to generate scenarios $\tilde{S}^d$ given the document d and instruction for scenario generation $\mathcal{I}$.

Scenario indexing. After training the scenario generator, we generate a set of scenarios for each document d in the corpus. From each generated scenario, we extract only the Main topic (M) and Explanation (E) components, as discussed in Section 3.1, which are combined to construct the final scenario representation set $S^d = \{s_1, \ldots s_k\}$. Then each scenario representation is encoded into a dense vector using the same encoder $\mathcal{E}$ used for document representations. These vectors are used to build a scenario-profiled index alongside the standard document index. This additional scenario-profiled index allows the retrieval system to capture implicit relevance more effectively by leveraging reasoning-derived scenario representations during retrieval. Since scenario generation and indexing are performed offline, this approach imposes no additional inference-time burden, unlike query expansion (Su et al., 2024) or LLM-based reranking methods (Niu et al., 2024).

3.3 Retrieval with scenario

During retrieval, we produce the final ranked list Y_{final}. To this end, we first compute the relevance scores between the query and each document, as well as its associated scenarios with pre-built indexes. Let q be the query and $\mathcal{E}$ the dense retrieval model. For a given document d with associated scenario set S^d, we compute relevance scores as follow:

$$r_d = \text{sim}(\mathcal{E}(q), \mathcal{E}(d)), \quad r_s = \text{sim}(\mathcal{E}(q), \mathcal{E}(s)), \quad \forall\, s \in S^d \tag{2}$$

where $\text{sim}(\cdot)$ denotes cosine similarity, r_d represents the relevance score for the document, and r_s represents the relevance score for the scenario. Then, we select the maximum relevance score among the scenarios associated with each document and compute the final relevance score as a weighted sum of the document and scenario scores:

$$r_{\text{final}}(d) = \alpha\, r_d + (1 - \alpha) \max_{s \in S^d} \{r_s\} \tag{3}$$

where α is the relevance weight, which is the hyperparameter that controls the effect of document and scenario. Finally, we produce Y_{final} by calculating final relevance score $r_{\text{final}}(d)$ for all documents and sorting them in descending order.

3.4 Efficient retrieval strategy

While SPIKE's scenario-profiled index enhances retrieval effectiveness, a naive implementation could introduce significant latency by scoring every scenario for all documents. Such an approach would cause additional overhead and latency to scale proportionally with the corpus size (N). To ensure practical efficiency, we employ an efficient retrieval strategy. For a given query, SPIKE first identifies a candidate set of top-k' documents using only the document scores (r_d). Subsequently, scenario scores (r_s) are computed exclusively for this limited subset. This approach ensures that the additional computation is bounded by the hyperparameter k' (where $k < k' \ll N$) and does not scale with the corpus size.

4 Experiments

In this section, we conduct our experiments to answer the following research questions:

- **RQ1:** Can SPIKE effectively enhance the retrieval performance?
- **RQ2:** Can SPIKE's scenarios serve as useful information for real-world users?
- **RQ3:** Can SPIKE's scenarios serve as an effective additional context in a RAG setting?

	Natural language					Code				Math			Avg.	Improv.
	Bio.	Earth.	Econ.	Psy.	Sus.	Rob.	Stack.	Leet.	Pony	Aops	TheoQ.	TheoT.		
Dense retrieval models (< 1B)														
BGE	12.0	24.2	16.6	17.4	**13.3**	**12.2**	9.5	26.7	5.6	**6.0**	13.0	6.9	13.6	**+5.9%**
+SPIKE	**13.2**	**26.4**	**17.0**	**18.1**	13.2	11.5	**13.3**	**27.1**	**6.4**	4.8	**13.0**	**8.5**	**14.4**	
SBERT	15.5	20.1	16.6	**22.6**	15.3	8.4	9.5	26.4	6.9	5.3	**20.0**	10.8	14.8	**+6.1%**
+SPIKE	**18.2**	**23.1**	**17.9**	21.3	**15.5**	**9.0**	**13.4**	**26.7**	**8.1**	**5.4**	19.3	**11.2**	**15.7**	
Dense retrieval models (> 1B)														
E5-Mistral	18.8	26.0	15.5	15.8	18.5	16.4	9.8	28.7	4.8	**7.1**	**26.1**	26.8	17.9	**+20.7%**
+SPIKE	**25.9**	**33.0**	**18.2**	**20.6**	**20.6**	**18.4**	**16.2**	**29.4**	**17.5**	7.0	23.4	**28.4**	**21.6**	
SFR	19.5	26.6	17.8	19.0	19.8	16.7	12.7	27.4	2.0	**7.4**	**24.3**	26.0	18.3	**+18.6%**
+SPIKE	**23.6**	**31.7**	**19.9**	**26.0**	**21.2**	**17.8**	**17.6**	**28.6**	**17.3**	6.5	22.8	**27.5**	**21.7**	
GRIT	25.0	**32.8**	19.0	19.9	18.0	17.3	11.6	29.8	**22.0**	8.8	25.1	21.1	20.9	**+4.3%**
+SPIKE	**27.8**	29.0	**20.0**	**20.4**	**19.0**	**19.2**	**16.7**	**32.0**	18.3	**9.2**	**25.2**	**24.9**	**21.8**	
Qwen	30.9	36.2	17.7	24.6	14.9	13.5	19.9	25.5	14.4	**27.8**	**32.9**	**32.9**	24.3	**+3.3%**
+SPIKE	**32.4**	**41.2**	**23.7**	**25.7**	**24.7**	**16.0**	**23.7**	**26.3**	**16.7**	12.5	27.1	31.0	**25.1**	

Table 1: The retrieval performance of existing retrieval models and our SPIKE framework on the BRIGHT benchmark with the original query. We report nDCG@10 for all datasets. Avg. denotes the average score across 12 datasets and Improv. denotes the improvement rate of the average score. The best score on each model is shown in bold.

4.1 Experimental settings

Datasets & Evaluation metric. We use BRIGHT benchmark (Su et al., 2024) to assess retrieval performance of SPIKE. By evaluating on BRIGHT, we assess how much SPIKE improves the performance of dense retrieval models that previously struggled on reasoning-intensive retrieval task, demonstrating its effectiveness. As done in previous works, we evaluate retrieval performance using nDCG@10. More details are provided in Appendix C.1.

Backbone models. To demonstrate that SPIKE can be applied effectively across different dense retrievers, we evaluate performance of SPIKE with 6 representative dense retrievers. Specifically, we conduct experiments with BGE-Large (Xiao et al., 2023), SBERT (Reimers & Gurevych, 2019), E5-Mistral-7B (Wang et al., 2023), SFR-Embedding-Mistral (Meng et al., 2024), GRIT (Muennighoff et al., 2024) and gte-Qwen1.5 (Li et al., 2023).

Implementation details. We construct the scenario-augmented training dataset by randomly sampling 300 documents per dataset from StackExchange split of BRIGHT (Su et al., 2024) and all datasets in BEIR (Thakur et al., 2021), resulting in a total of 8,100 documents. Note that there is not any exposure to test queries or their relevance annotations, ensuring that retrieval evaluation remains entirely independent of the training process. For each sampled document, we use GPT-4o to generate scenarios, ensuring that the dataset contains high-quality scenarios that facilitate deeper reasoning over a document's contents. For the scenario generator, we use Llama-3.2-3B-Instruct (Dubey et al., 2024) as the backbone model and fine-tune it using LoRA (Hu et al., 2021). For our main experiments, we set the relevance weight in Equation (3) to 0.7. We use the efficient retrieval strategy mentioned in Section 3.4 for all experiments, setting the value K' at 1000. For reproducibility, we also provide more details about implementation details in Appendix C.2.

4.2 SPIKE improves retrieval performance (RQ1)

Main result on BRIGHT. Table 1 shows the retrieval performance of dense retrieval models on the BRIGHT benchmark, comparing their original performance against their SPIKE-enhanced versions. Across different retrieval models and datasets, SPIKE consistently improves average retrieval performance, demonstrating its effectiveness in capturing implicit relevance through scenario-profiled indexing. These improvements are observed across both small (<1B) and large (>1B) dense retrieval models, highlighting the ability to generalize across different retrieval architectures. Notably, SPIKE yields substantial gains for models with relatively weaker baseline performance, such as E5-Mistral and SFR, where retrieval accuracy improves by over 18%, emphasizing its potential to bridge reasoning gaps in weaker LLM-based retrieval models. Moreover, the results show that SPIKE provides

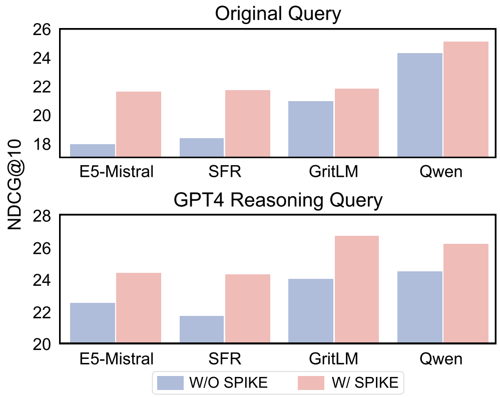


Figure 3: Average nDCG@10 scores on BRIGHT for different query types.

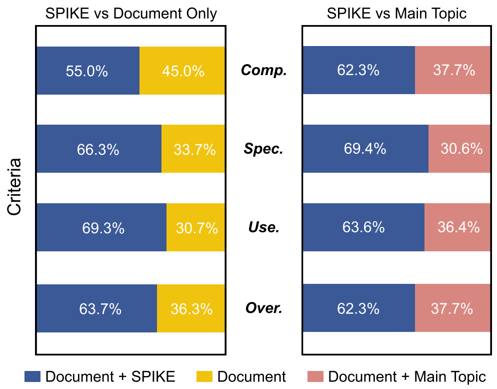


Figure 4: Human evaluation of pairwise comparisons for retrieval results.

significant benefits in datasets involving non-natural language content, such as code-based datasets. Standard dense retrieval models struggle in these datasets due to the inherent discrepancy between non-natural language documents and natural language queries. SPIKE mitigates this gap and connects between them by leveraging scenario-profiled retrieval, which explicitly describes in natural language how a non-natural language document addresses diverse information needs. In the math domain, SPIKE also shows performance gain. In **TheoT.**, where documents feature LaTeX-formatted mathematical expressions, our scenario-profiled approach substantially improves retrieval performance by clarifying implicit reasoning steps. In contrast, the performance gains in **Aops.** and **TheoQ.** are less pronounced, primarily because these datasets present documents in a question-like format. Since these documents are already written in a question-like format, it becomes challenging to generate additional scenarios that meaningfully enrich them.

Results with reasoning-augmented queries. We additionally evaluate performance using reasoning-augmented queries, obtained by prompting LLM to reformulate the original queries with explicit step-by-step reasoning (Su et al., 2024). This setup allows us to assess how SPIKE performs when queries already contain explicit reasoning steps. In this experiment, we use GPT-4 reasoning queries provided by the BRIGHT benchmark, which are reformulated from the original queries in BRIGHT. For further details, please refer to the Appendix C.3. Figure 3 compares average nDCG@10 scores for four retrieval models under two query types: original queries and GPT-4 reasoning queries, each evaluated with and without SPIKE enhancement. First, using SPIKE with original query yields performance comparable to or better than using GPT-4 reasoning queries without SPIKE. Notably, this result highlights the efficiency of SPIKE, as it achieves similar performance to GPT-4-based query reformulation while using a much smaller 3B generator. Second, when GPT-4 reasoning queries are used, incorporating SPIKE further improves retrieval performance across all models. This consistent improvement demonstrates that SPIKE remains robust across various query types, including those augmented with GPT-4's chain-of-thought reasoning. In other words, the scenarios generated by SPIKE help capture implicit relevance in reasoning-intensive tasks, regardless of the query format.

4.3 SPIKE enhances retrieval experience for real-world users (RQ2)

To examine the effectiveness of the additional information provided by SPIKE, we conduct a human evaluation. Specifically, we compare different retrieved results across four criteria: *Comprehensibility (Comp.)*, *Specificity (Spec.)*, *Usefulness (Use.)* and *Overall (Over.)*. These criteria are designed to assess different aspects of user satisfaction with the retrieved results (see Appendix C.4 for detailed descriptions). Figure 4 presents the results of two human evaluation settings: (1) comparing standard document retrieval (Document Only) with SPIKE, and (2) comparing SPIKE with one of its variants, Document + Main Topic. Across all evaluation criteria, SPIKE consistently outperforms both baselines. The gains are particularly notable in *Specificity* and *Usefulness*, where reasoning-derived information pro-

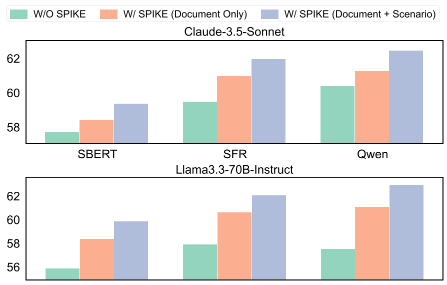


Figure 5: Average QA performance in RAG. Document only uses retrieved documents as context; +Scenario additionally uses scenario information.

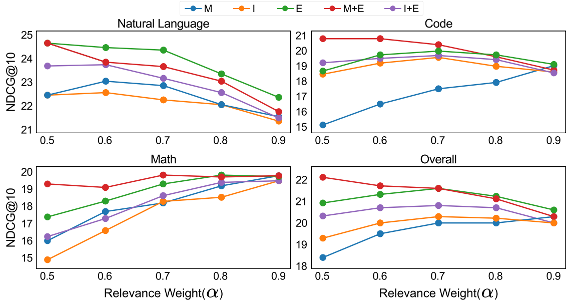


Figure 6: Retrieval performance of SPIKE across different scenario components and relevance weights for each document types, using E5-Mistral-7B as the retrieval model.

vides clearer and more practically helpful signals by explicitly expressing implicit relevance. Furthermore, SPIKE offers a clear advantage in *Comprehensibility*, demonstrating that users find retrieved results easier to interpret when supported by the scenario context. Especially, the comparison between SPIKE and the Document + Main Topic setting (i.e., SPIKE w/o Explanation) highlights the critical role of the Explanation (E) component. While the Main Topic provides useful context, the explanation substantially enhances document comprehension and practical decision-making by explicitly expressing implicit relevance. **This finding clearly implies that reasoning-derived explanations significantly improve the clarity and usefulness of retrieved results for real-world users.**

4.4 SPIKE boosts RAG performance with additional context (RQ3)

As SPIKE 's scenario-profiled information helps users better understand about retrieved contents, it can also serve as effective context for retrieval-augmented generation (RAG) by providing additional information alongside the retrieved content. To verify this, we evaluate the QA performance of Claude-3.5-sonnet and Llama3.3-70B-Instruct when augmented with documents retrieved by different retrievers, comparing its performance with and without the additional context from SPIKE. Specifically, we use reference answers provided in the BRIGHT benchmark and follow the evaluation process of Su et al. (2024), where a evaluation model scores the generated answers based on their alignment with the references. For detailed experimental settings, please refer to Appendix C.5. Figure 5 presents the average QA performance in different retrievers and generation models in the RAG setting. First, we observe that using documents retrieved via SPIKE as context consistently improves QA accuracy across all retrievers and generators. This improvement is attributed to enhanced retrieval performance, which allows more relevant documents to be retrieved, thereby providing better context for the generator to produce accurate answers. Moreover, further performance gains are achieved when SPIKE 's scenario contexts are provided alongside the retrieved documents. These results suggest that the scenario context not only contributes to retrieval performance but also directly enriches the retrieved content, offering stronger and more structured support for answer generation in retrieval-augmented settings.

5 Analysis

Ablation study on scenario components. As shown in Figure 6, we analyze the effectiveness of different scenario-generation components within SPIKE across various document types: natural language, code, and math. Among the individual components (M, I, E), E achieves the highest retrieval performance, showing the importance of explicitly modeling reasoning to capture implicit relevance. Additionally, combining E with other components (M+E, I+E) led to further improvements, achieving higher retrieval performance compared to when each component was used individually. Notably, the combination of M+E yielded the best overall performance, further emphasizing the benefit of combining main topic identification with reasoning-derived explanations, as discussed in Section 3.1. These results highlight

	Natural language					Code				Math			Avg.	Improv.
	Bio.	**Earth.**	**Econ.**	**Psy.**	**Sus.**	**Rob.**	**Stack.**	**Leet.**	**Pony**	**Aops**	**TheoQ.**	**TheoT.**		
E5-Mistral	18.8	26.0	15.5	15.8	18.5	16.4	9.8	**28.7**	4.8	**7.1**	**26.1**	26.8	17.9	-
+SPIKE ($C = 4000$)	25.0	28.5	17.9	20.0	20.0	**17.7**	14.1	28.6	9.7	6.1	23.5	**27.9**	19.9	11.5%
+SPIKE ($C = 20000$)	**25.8**	**35.9**	**19.5**	**24.3**	**21.3**	16.1	**14.1**	28.0	**25.6**	6.9	23.6	27.8	**22.4**	**25.4%**
SFR	19.5	26.6	17.8	19.0	19.8	16.7	12.7	27.4	2.0	**7.4**	**24.3**	26.0	18.3	-
+SPIKE ($C = 4000$)	22.0	29.0	19.0	25.5	22.0	**17.6**	**16.2**	**27.9**	9.6	6.2	22.7	26.7	20.4	11.5%
+SPIKE ($C = 20000$)	**25.0**	**34.3**	**22.8**	**29.9**	**22.4**	16.6	16.1	27.8	**24.2**	7.1	23.3	**27.5**	**23.1**	**26.3%**
GRIT	25.0	32.8	19.0	19.9	18.0	17.3	11.6	29.8	22.0	8.8	25.1	21.1	20.9	-
+SPIKE ($C = 4000$)	23.0	27.6	18.4	20.2	19.5	18.3	15.8	**31.5**	17.5	**8.9**	**25.6**	25.9	21.0	0.7%
+SPIKE ($C = 20000$)	**30.4**	**34.9**	**21.7**	**24.3**	**21.1**	**18.6**	**17.0**	29.4	**24.4**	8.8	25.3	**27.0**	**23.6**	**13.0%**
Qwen	30.9	36.2	17.7	24.6	14.9	13.5	19.9	25.5	14.4	**27.8**	**32.9**	32.9	24.3	-
+SPIKE ($C = 4000$)	31.3	40.4	22.8	24.4	24.5	16.2	22.8	24.6	11.1	13.6	27.8	33.3	24.4	0.5%
+SPIKE ($C = 20000$)	**34.6**	**42.9**	**23.2**	**30.6**	**28.1**	**19.4**	**24.1**	**25.6**	**27.0**	13.5	28.6	**34.0**	**27.6**	**13.8%**

Table 2: Retrieval performance when scenarios are generated by a scenario generator trained only on the BEIR corpus without the BRIGHT corpus. We report nDCG@10 for all datasets. Avg. denotes the average score across 12 datasets and Improv. denotes the improvement rate of the average score. C denotes the number of documents used in the scenario-augmented training dataset. The best score on each model is shown in bold.

that modeling implicit relevance through reasoning and indexing it for use in retrieval is more effective than relying solely on summaries or information needs.

Ablation study on relevance weight. Figure 6 also illustrates the impact of the relevance weight α in Equation (3), which balances scenario-level and document-level relevance scores. Across nearly all components, retrieval performance peaked at $\alpha = 0.7$. However, different document type exhibited distinct trends. In natural language and code documents, M+E performance improves as α decreases indicating that the information provided by scenarios is more crucial than the original document content in these document types. Conversely, in math documents, performance improves steadily as α increases, suggesting that document-level content (e.g., LaTeX equations) is more critical for effective retrieval. Given these analysis, we select $\alpha = 0.7$ as the primary configuration, as it provides the most consistent performance improvements across different types of documents. The full results of the ablation study on scenario components and relevance weight are provided in Appendix B.3.

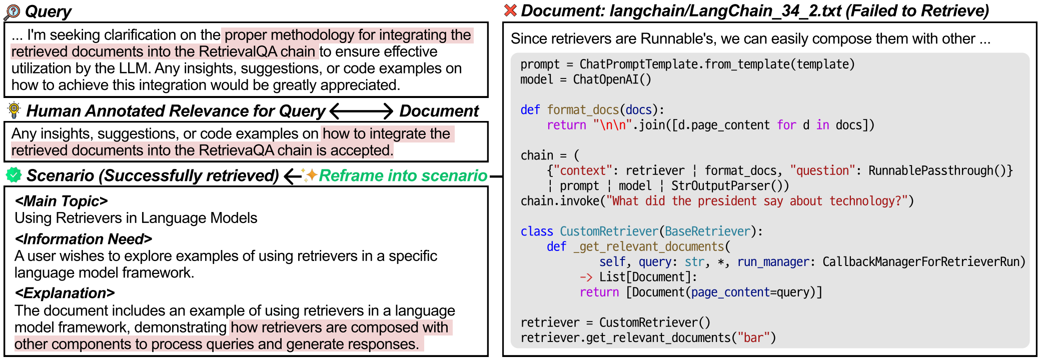
Zero-shot generalization of scenario generator. To evaluate the generalizability of our scenario generator, we conduct a zero-shot experiment where the generator was trained exclusively on the BEIR corpus without exposure to the BRIGHT corpus. Table 2 presents the retrieved results under this setting. Notably, BEIR does not fully cover the range of domains present in BRIGHT, particularly code and math. Despite this domain gap, the version of SPIKE trained only on the BEIR corpus consistently improves retrieval performance across nearly all datasets in BRIGHT. Surprisingly, even when trained solely on BEIR, it sometimes exhibits higher performance than when utilizing a scenario generator trained on the BRIGHT corpus (refer to Table 1). Specifically, we observe substantial gains in code domain and TheoT. dataset, which are not covered by the BEIR corpus. These results demonstrate the strong out-of-domain generalization capability of our scenario-based retrieval framework.

Ablation study on scenario-augmented training dataset. To investigate the impact of scaling the scenario-augmented training dataset on retrieval performance, we conducted an ablation study on the size of the scenario-augmented training dataset used for training our scenario generator. Table 2 presents the results of this ablation study performed solely with the BEIR corpus. Even when utilizing only 4,000 documents, SPIKE demonstrates superior performance compared to the baseline without SPIKE. Furthermore, scaling the scenario-augmented training dataset yields a significantly greater performance improvement. Notably, as previously mentioned, this performance surpasses the retrieval performance achieved when trained on the BRIGHT dataset. These results indicate that performance of SPIKE consistently improves as the scenario-augmented training dataset scales, suggesting that SPIKE could benefit from even larger and more diverse datasets to further enhance its capabilities to capture implicit relevance across various domains.

Ablation study on efficient retrieval strategy. To validate the effectiveness of the efficient retrieval strategy introduced in Section 3.4, we conducted an ablation studys on the candidate set size, k'. Table 3 shows the retrieval performance as k' is varied. The results indicate

	Natural language					Code				Math			Avg.
	Bio.	Earth.	Econ.	Psy.	Sus.	Rob.	Stack.	Leet.	Pony	Aops	TheoQ.	TheoT.	
E5-Mistral	18.8	26.0	15.5	15.8	18.5	16.4	9.8	28.7	4.8	7.1	26.1	26.8	17.9
+SPIKE ($k' = 1000$)	25.9	33.0	18.2	20.6	20.6	18.4	16.2	29.4	17.5	7.0	23.4	28.4	21.6
+SPIKE ($k' = 10000$)	25.7	33.0	18.2	20.6	20.6	18.4	16.2	29.4	17.2	7.0	23.5	28.2	21.5
+SPIKE ($k' = N$)	25.7	33.0	18.2	20.6	20.6	18.4	16.2	29.4	17.2	7.0	23.5	28.2	21.5

Table 3: Retrieval performance on BRIGHT for different values of K', using E5-Mistral-7B as the retrieval model. Retrieval performance is virtually unchanged regardless of the value of K', whether K' is set to 1000, 10000, or N (i.e., exhaustive search).


Query
Human Annotated Relevance for Query ⟷ Document
Scenario (Successfully retrieved)
Reframe into scenario
✗ Document: langchain/LangChain_34_2.txt (Failed to Retrieve)

... I'm seeking clarification on the proper methodology for integrating the retrieved documents into the RetrievalQA chain to ensure effective utilization by the LLM. Any insights, suggestions, or code examples on how to achieve this integration would be greatly appreciated.

Any insights, suggestions, or code examples on how to integrate the retrieved documents into the RetrievaQA chain is accepted.

<Main Topic>
Using Retrievers in Language Models

<Information Need>
A user wishes to explore examples of using retrievers in a specific language model framework.

<Explanation>
The document includes an example of using retrievers in a language model framework, demonstrating how retrievers are composed with other components to process queries and generate responses.

Since retrievers are Runnable's, we can easily compose them with other ...

```
prompt = ChatPromptTemplate.from_template(template)
model = ChatOpenAI()

def format_docs(docs):
    return "\n\n".join([d.page_content for d in docs])

chain = (
    {"context": retriever ¦ format_docs, "question": RunnablePassthrough()}
    ¦ prompt ¦ model ¦ StrOutputParser())
chain.invoke("What did the president say about technology?")

class CustomRetriever(BaseRetriever):
    def _get_relevant_documents(
            self, query: str, *, run_manager: CallbackManagerForRetrieverRun)
        -> List[Document]:
        return [Document(page_content=query)]

retriever = CustomRetriever()
retriever.get_relevant_documents("bar")
```

Figure 7: A StackOverflow example from BRIGHT where standard dense retrieval fails due to the absence of explicit semantic connections, while SPIKE retrieves the correct document by capturing implicit relevance through scenario-profiled retrieval.

that performance remains nearly identical to the naive method (i.e., scoring all scenarios) as long as k' is sufficiently large, while significantly reducing the computational load. This result demonstrates that our efficient retrieval strategy successfully provides high efficiency without compromising the retrieval effectiveness of the SPIKE framework.

Case study on non-natural language documents. Figure 7 shows an example from the StackOverflow in BRIGHT. In this case, standard dense retrieval, which relies solely on the document, fails to retrieve the relevant result. In contrast, SPIKE successfully retrieves it. The query seeks guidance on integrating retrieved documents into a RetrievalQA chain, but the relevant document does not explicitly contain it. Instead, it provides a code snippet demonstrating how retrievers interact with other components. Standard dense retrieval fails as it relies on surface-level similarity without reasoning, making it unable to bridge the gap between the query and the document. SPIKE overcomes this limitation by leveraging scenarios that highlight latent connection between the query and the document. This case highlights SPIKE 's ability to retrieve documents that require reasoning to establish implicit relevance, making it particularly effective for non-natural language content like code snippets where relevant information is not explicitly stated.

6 Conclusion

This paper proposes SPIKE, a novel dense retrieval framework designed to enhance retrieval effectiveness by explicitly modeling implicit relevance. By reframing documents into structured retrieval scenarios, SPIKE addresses the limitations of existing dense retrieval models that struggle with reasoning-intensive tasks. Our extensive experiments demonstrate that SPIKE not only enhances retrieval performance across various domains, including natural language, code, and math, but also improves usability in real-world retrieval systems and serves as an effective context for RAG-based applications. Furthermore, our analysis showed that SPIKE exhibits robust out-of-domain generalization capabilities.

Acknowledgements

This work was supported by the IITP grants funded by the Korea government (MSIT) (No. RS-2020-II201361; RS-2024-00457882, AI Research Hub Project; IITP-2025-RS-2020-II201819).

References

Florian Boudin, Ygor Gallina, and Akiko Aizawa. Keyphrase generation for scientific document retrieval. In *Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://api.semanticscholar.org/CorpusID:220047513.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. Dense x retrieval: What retrieval granularity should we use? In *Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:266163052.

Yanfei Chen, Jinsung Yoon, Devendra Singh Sachan, Qingze Wang, Vincent Cohen-Addad, MohammadHossein Bateni, Chen-Yu Lee, and Tomas Pfister. Re-invoke: Tool invocation rewriting for zero-shot tool retrieval. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:271709437.

David R. Cheriton. From doc2query to doctttttquery. 2019. URL https://api.semanticscholar.org/CorpusID:208612557.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Bap tiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Laurens Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Ken-591 neth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Babu Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Ro main Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Ben Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm’an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu

Yang, and Zhiwei Zhao. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. URL https://api.semanticscholar.org/CorpusID:271571434.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL https://api.semanticscholar.org/CorpusID:235458009.

Soyeong Jeong, Jinheon Baek, chaeHun Park, and Jong C. Park. Unsupervised document expansion for information retrieval with stochastic text generation. *ArXiv*, abs/2105.00666, 2021. URL https://api.semanticscholar.org/CorpusID:233481135.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, pp. 6769–6781, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*, pp. 39–48, 2020. URL https://doi.org/10.1145/3397271.3401075.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfrembedding-mistral: enhance text retrieval with transfer learning. *Salesforce AI Research Blog*, 3, 2024.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:252907685.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*, 2024.

Tong Niu, Shafiq Joty, Ye Liu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Judgerank: Leveraging large language models for reasoning-intensive reranking. *ArXiv*, abs/2411.00142, 2024. URL https://api.semanticscholar.org/CorpusID:273798418.

Rodrigo Nogueira, Wei Yang, Jimmy J. Lin, and Kyunghyun Cho. Document expansion by query prediction. *ArXiv*, abs/1904.08375, 2019. URL https://api.semanticscholar.org/CorpusID:119314259.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. *ArXiv*, abs/2401.18059, 2024. URL https://api.semanticscholar.org/CorpusID:267334785.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan Ö. Arik, Danqi Chen, and Tao Yu. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *ArXiv*, abs/2407.12883, 2024. URL https://api.semanticscholar.org/CorpusID:271270735.

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663, 2021. URL https://api.semanticscholar.org/CorpusID:233296016.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *ArXiv*, abs/2401.00368, 2023. URL https://api.semanticscholar.org/CorpusID:266693831.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.

Contents of Appendix

A Comparison with existing document expansion works

A.1 Experimental settings

As discussed in Section 2, prior works have improved document representations by expanding them with pseudo queries (Nogueira et al., 2019; Chen et al., 2024) or summaries (Jeong et al., 2021). To evaluate whether SPIKE offers a more effective approach in reasoning-intensive retrieval tasks, we conduct additional comparative experiments against these traditional document expansion methods. However, directly comparing SPIKE to existing methods is not entirely fair, as prior approaches typically rely on models such as DocT5Query (Cheriton, 2019) or PEGASUS-Large (Zhang et al., 2019), which not only lack the ability to handle non-natural language documents like code but also fall significantly short in overall language understanding capabilities compared to the LLM used in SPIKE. To ensure a fair comparison, we use the same LLM backbone (Llama3.2-3B-Instruct) that is used for SPIKE 's scenario generator, and apply it to generate pseudo queries and summary for each document. Specifically, we generate three pseudo queries for the pseudo query document expansion and one summary for the summary document expansion per document, and concatenate them with the document representation.

A.2 Results

Table 4 presents the performance comparison between SPIKE, pseudo query-based document expansion, and summary-based document expansion. First, SPIKE consistently improves average performance across all dense retrievers, whereas traditional document expansion methods leveraging pseudo queries or summary yield smaller gains compared to SPIKE, and even lead to performance degradation. In particular, document expansion approach that leveraging pseudo queries fails to provide any improvements and even degrades performance across all dense retrievers. These results suggest that the additional context used in prior document expansion methods may introduce noise, especially in reasoning-intensive retrieval tasks where surface-level semantic information is insufficient.

	Natural language					**Code**				**Math**			**Avg.**	**Improv.**
	Bio.	**Earth.**	**Econ.**	**Psy.**	**Sus.**	**Rob.**	**Stack.**	**Leet.**	**Pony**	**Aops**	**TheoQ.**	**TheoT.**		
E5-Mistral	18.8	26.0	15.5	15.8	18.5	16.4	9.8	28.7	4.8	**7.1**	**26.1**	26.8	17.9	-
+PQ.	18.6	28.3	13.2	12.7	15.4	14.6	14.3	29.9	3.3	4.8	19.7	16.5	15.9	-11.2%
+Sum.	17.8	24.9	16.5	18.5	**22.6**	16.3	15.8	**31.3**	0.6	5.3	22.5	19.8	17.7	-1.1%
+SPIKE	**25.9**	**33.0**	**18.2**	**20.6**	20.6	**18.4**	**16.2**	29.4	**17.5**	7.0	23.4	**28.4**	**21.6**	**+20.7%**
SFR	19.5	26.6	17.8	19.0	19.8	16.7	12.7	27.4	2.0	**7.4**	**24.3**	26.0	18.3	-
+PQ.	17.9	28.5	16.2	17.7	17.6	15.1	15.8	29.4	3.7	5.8	19.6	19.2	17.2	-6.0%
+Sum.	20.6	27.6	18.5	23.3	**24.6**	16.7	17.4	**31.6**	1.1	5.6	22.7	22.3	19.3	5.5%
+SPIKE	**23.6**	**31.7**	**19.9**	**26.0**	21.2	**17.8**	**17.6**	28.6	**17.3**	6.5	22.8	**27.5**	**21.7**	**+18.6%**
GRIT	25.0	**32.8**	19.0	19.9	18.0	17.3	11.6	29.8	**22.0**	8.8	25.1	21.1	20.9	-
+PQ.	20.3	24.6	16.0	18.9	18.5	18.8	13.3	**33.4**	7.6	7.4	23.4	19.2	18.4	-12.0%
+Sum.	25.4	31.9	19.9	**23.2**	**22.5**	16.0	**17.5**	33.0	3.0	8.8	22.8	18.1	20.2	-3.3%
+SPIKE	**27.8**	29.0	**20.0**	20.4	19.0	**19.2**	16.7	32.0	18.3	**9.2**	**25.2**	**24.9**	**21.8**	**+4.3%**
Qwen	30.9	36.2	17.7	24.6	14.9	13.5	19.9	25.5	14.4	**27.8**	**32.9**	**32.9**	24.3	-
+PQ.	31.1	42.6	21.7	**28.4**	**25.1**	14.5	**26.6**	**27.1**	**17.4**	8.7	22.9	24.9	24.3	+0.0%
+Sum.	**36.6**	**46.7**	8.9	14.4	18.9	5.0	24.4	22.6	0.6	2.5	24.8	20.8	18.8	-22.6%
+SPIKE	32.4	41.2	**23.7**	25.7	24.7	**16.0**	23.7	26.3	16.7	12.5	27.1	31.0	**25.1**	**+3.3%**

Table 4: Performance comparison of different document expansion methods on reasoning-intensive retrieval tasks. We compare SPIKE against pseudo query-based and summary-based document expansion approaches. PQ. denote the pseudo query-based approach and Sum. denote the summary-based approach.

B More experiment & analysis result

B.1 Reasoning-augmented query result

To provide a more comprehensive view of how SPIKE interacts with reasoning-augmented queries, we present the full results of experiments using GPT-4-generated reasoning queries from the BRIGHT benchmark. Table 5 reports nDCG@10 scores across all 12 datasets in BRIGHT, comparing standard retrieval models with and without SPIKE. These results confirm that SPIKE consistently improves retrieval performance even when applied to queries that already include explicit reasoning. It demonstrates not only the robustness of SPIKE across query types, but also its potential to be effectively integrated into future methods aimed at enhancing reasoning-intensive retrieval performance.

	Natural language					**Code**				**Math**			**Avg.**	**Improv.**
	Bio.	**Earth.**	**Econ.**	**Psy.**	**Sus.**	**Rob.**	**Stack.**	**Leet.**	**Pony**	**Aops**	**TheoQ.**	**TheoT.**		
E5-Mistral	29.6	43.6	20.1	26.7	**15.6**	11.8	17.7	29.1	**9.0**	5.3	25.6	**35.7**	22.5	**+8.4%**
+SPIKE	**37.4**	**46.2**	**22.3**	**27.1**	15.5	**13.4**	**23.2**	**30.2**	8.5	**6.8**	**28.0**	34.8	**24.4**	
SFR	26.2	39.1	21.5	28.3	**19.5**	13.4	16.8	28.4	1.5	7.1	25.9	33.2	21.7	**+10.7%**
+SPIKE	**30.1**	**40.2**	**24.7**	**29.6**	17.8	**15.2**	**22.7**	**30.0**	**9.4**	**8.8**	**28.0**	**34.5**	**24.3**	
GRIT	33.1	**38.9**	22.3	28.8	24.1	17.4	17.7	31.8	**11.7**	6.7	26.3	29.5	24.0	**+4.2%**
+SPIKE	**29.8**	35.1	**23.8**	**29.1**	**24.2**	**18.4**	**22.0**	**34.2**	11.4	**8.3**	**30.0**	**33.6**	**25.0**	
Qwen	35.8	**43.0**	24.3	**34.3**	24.4	15.6	19.7	25.4	5.2	4.6	28.0	33.7	24.5	**+6.9%**
+SPIKE	**37.5**	42.2	**26.6**	33.4	**24.5**	**17.6**	**26.6**	**28.4**	**4.3**	**7.3**	**31.0**	**35.0**	**26.2**	

Table 5: The retrieval performance of existing retrieval models and our SPIKE framework on the BRIGHT benchmark with GPT4 reasoning query provided in Su et al. (2024).

B.2 RAG Experimental Result

We provide the full result tables for the RAG experiments conducted in Section 4. Table 6 presents the complete results using Claude-3.5-sonnet as the generation model and GPT-4o for answer evaluation. To further verify the robustness of our findings under different model configurations, we also include results where Llama3.3-70B-Instruct is used as the generation model (Table 7). This another setup allows us to assess whether the benefits of SPIKE 's additional context hold consistently across different generation model.

Ret.	**Bio.**		**Earth.**		**Econ.**		**Psy.**		**Rob.**		**Stack.**		**Sus.**		**Avg.**	
	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.
SBERT	58.4	-	62.4	-	52.7	-	56.7	-	53.3	-	66.8	-	53.5	-	57.7	-
+SPIKE	58.8	**61.2**	64.0	**64.9**	**55.7**	54.4	57.0	**57.0**	53.8	**55.6**	65.0	**66.1**	54.7	**56.9**	58.4	**59.4**
SFR	60.8	-	64.1	-	54.8	-	60.2	-	54.1	-	67.3	-	55.4	-	59.5	-
+SPIKE	60.5	**64.8**	68.4	**70.1**	54.8	**55.6**	**62.1**	61.5	**56.3**	55.8	67.6	**68.6**	57.1	**57.3**	61.0	**62.0**
Qwen	62.2	-	68.6	-	55.9	-	60.8	-	50.5	-	67.7	-	56.9	-	60.4	-
+SPIKE	62.9	**65.7**	**69.4**	68.7	56.7	**58.8**	**60.8**	60.5	55.0	**56.1**	66.9	**68.5**	57.4	**58.8**	61.3	**62.5**
Oracle	*67.9*		*73.3*		*66.1*		*73.2*		*71.0*		*77.0*		*64.0*		*70.3*	

Table 6: Full RAG performance results using Claude-3.5-sonnet as the generation model and GPT-4o for evaluation. doc. denotes document only, which only use retrieved document as context and +sce denotes +Scenario which additionally uses scenario information as context.

B.3 Analysis result

Table 8 presents the full results of our analysis, as discussed in Section 5.

Ret.	Bio.		Earth.		Econ.		Psy.		Rob.		Stack.		Sus.		Avg.	
	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.	doc.	+sce.
SBERT	56.5	-	57.8	-	53.2	-	55.0	-	53.0	-	64.1	-	51.9	-	55.9	-
+SPIKE	59.1	**61.0**	63.2	**65.1**	**56.3**	55.0	**57.6**	56.9	52.3	**57.1**	64.9	**66.5**	55.4	**57.4**	58.4	**59.9**
SFR	57.6	-	60.2	-	53.7	-	59.1	-	56.4	-	69.3	-	49.4	-	57.9	-
+SPIKE	60.8	**65.0**	66.7	**69.6**	55.3	**57.2**	**62.2**	62.0	**56.1**	55.8	67.6	**67.8**	55.6	**57.5**	60.6	**62.1**
Qwen	60.6	-	62.4	-	51.7	-	57.5	-	51.7	-	65.2	-	53.3	-	57.5	-
+SPIKE	63.2	**65.8**	68.7	69.4	57.1	**58.2**	**61.8**	61.2	52.5	**58.4**	66.2	**68.7**	58.1	**58.3**	61.1	**62.9**
Oracle	*66.4*		*73.8*		*65.0*		*73.0*		*72.6*		*76.3*		*63.7*		*70.1*	

Table 7: Full RAG performance results using Llama3.3-70B-Instruct as the generation model and GPT-4o for evaluation. doc. denotes document only, which only use retrieved document as context and +sce denotes +Scenario which additionally uses scenario information as context.

C Experiment details

C.1 Dataset

C.1.1 BRIGHT

BRIGHT includes 1,398 real-world queries covering diverse domains such as economics, psychology, robotics, mathematics, and software programming. These queries are carefully designed to reflect challenging scenarios that demand deep comprehension and reasoning to retrieve relevant documents. Su et al. (2024) categorizes datasets into groups such as StackExchange, Coding, and Theorem-based collections. Specifically, individual datasets are classified as follows:

- StackExchange: Biology (**Bio.**), Earth Science (**Earth.**), Economics (**Econ.**), Psychology (**Psy.**), Robotics (**Rob.**), Stack Overflow (**Stack.**), Sustainable Living (**Sus.**),
- Coding: Leetcode (**Leet.**), Pony (**Pony**)
- Theorem-based: Aops (**AoPS**), TheoremQA-Question (**TheoQ.**), TheoremQA-Theorem (**TheoT.**)

Our work adopts a different classification based on document type, categorizing datasets into Natural Language, Code, and Math to better capture the retrieval challenges associated with different content structures. Specifically, individual datasets are classified as follows:

- Natural Language: Biology (**Bio.**), Earth Science (**Earth.**), Economics (**Econ.**), Psychology (**Psy.**), Sustainable Living (**Sus.**)
- Code: Leetcode (**Leet.**), Pony (**Pony**), Robotics (**Rob.**), Stack Overflow (**Stack.**),
- Math: Aops (**AoPS**), TheoremQA-Question (**TheoQ.**), TheoremQA-Theorem (**TheoT.**)

During the evaluation process, for instruction-following models used in our experiments, we directly utilized the instructions provided by Su et al. (2024).

C.1.2 BEIR

BEIR is a benchmark comprising diverse information retrieval tasks, consisting of 18 datasets across various domains such as Wikipedia, scientific publications, and others. In this work, we use the corpus from 15 publicly available BEIR dataset (MS MARCO, TREC-COVID, NFCorpus, NQ, HotpotQA, FiQA-2018, ArguAna, Touche-2020, CQADupStack, Quora, DBPedia, SCIDOCS, FEVER, Climate-FEVER, and SciFact) to train our scenario generator.

C.2 Implementation details

C.2.1 Scenario-augmented training dataset

Scenario generation for constructing the scenario-augmented training dataset was performed using GPT-4o with greedy decoding to ensure consistent and high-quality outputs. Additionally, to guarantee that each scenario component was generated without omission,

Comp.	α	Natural language					Code				Math			Avg.
		Bio.	Earth.	Econ.	Psy.	Sus.	Rob.	Stack.	Leet.	Pony	Aops	TheoQ.	TheoT.	
M	0.0	18.9	20.6	13.7	18.2	12.2	6.5	11.8	17.1	2.7	1.4	11.1	18.7	12.7
	0.1	21.0	22.7	14.4	19.7	13.3	7.7	11.7	18.9	3.6	1.6	13.4	19.3	13.9
	0.2	22.4	24.5	15.0	20.5	14.5	8.7	12.3	21.7	4.0	2.0	15.3	20.6	15.1
	0.3	23.1	26.7	16.1	21.0	17.0	9.6	12.7	23.9	5.2	2.2	16.5	23.0	16.4
	0.4	23.9	27.9	17.6	21.9	17.9	9.9	13.1	25.3	7.1	2.7	18.5	23.3	17.4
	0.5	24.6	28.8	18.2	22.5	18.6	10.9	13.0	27.1	9.2	3.4	20.0	24.7	18.4
	0.6	25.2	30.0	18.7	22.0	19.7	12.5	14.0	28.8	10.8	4.5	21.5	27.1	19.5
	0.7	24.8	29.3	19.5	21.4	19.4	14.0	14.1	29.3	12.5	5.5	22.7	17.3	20.0
	0.8	24.7	28.2	18.0	19.6	20.1	15.1	13.9	29.1	13.6	6.3	24.1	27.1	20.0
	0.9	23.6	28.6	18.0	18.2	19.5	16.4	13.6	29.2	16.7	6.7	25.8	27.0	20.3
I	0.0	22.9	22.6	15.1	17.2	12.5	10.2	14.6	19.6	7.3	1.3	12.0	13.7	14.1
	0.1	23.9	24.4	16.5	18.9	14.2	11.1	14.8	22.3	8.2	2.0	13.9	15.6	15.5
	0.2	25.2	26.3	17.7	20.1	15.2	12.2	14.7	24.9	10.0	2.5	16.0	16.2	16.7
	0.3	26.1	27.7	17.5	20.6	16.0	12.6	14.3	26.2	12.0	3.4	18.3	16.9	17.6
	0.4	26.8	28.8	18.0	20.3	16.7	12.9	13.7	27.9	14.4	4.0	19.9	18.2	18.4
	0.5	26.5	30.0	17.9	20.7	17.3	14.7	14.7	28.7	15.9	4.9	20.8	19.1	19.3
	0.6	25.9	31.1	18.5	20.1	17.6	15.5	15.5	29.3	16.7	5.4	21.8	22.7	20.0
	0.7	24.9	30.6	18.3	19.3	18.3	16.7	15.7	29.4	16.5	6.0	22.3	25.2	20.3
	0.8	18.8	18.7	29.1	18.7	18.6	16.5	14.4	29.3	15.7	6.7	23.7	25.1	20.2
	0.9	23.5	28.6	18.1	18.1	18.7	17.1	14.0	28.7	14.7	7.2	25.1	26.1	20.0
E	0.0	27.2	27.6	16.2	20.7	14.0	8.3	12.6	20.2	15.9	0.4	10.9	14.0	15.7
	0.1	27.7	28.6	16.7	21.7	14.5	9.2	13.0	22.6	16.3	0.6	13.0	14.8	16.6
	0.2	28.0	30.1	16.9	22.3	15.2	10.1	13.5	25.2	16.5	1.1	15.5	18.8	17.8
	0.3	28.7	30.9	17.4	23.8	16.8	11.3	14.3	26.7	16.2	1.4	17.2	23.2	19.0
	0.4	29.2	32.0	17.7	24.4	18.3	12.7	14.4	27.7	17.3	2.4	18.5	24.9	20.0
	0.5	29.4	33.0	17.8	24.0	19.2	13.6	15.1	28.8	17.3	3.5	20.6	27.9	20.9
	0.6	29.0	32.6	19.0	22.6	19.3	15.0	15.3	30.4	18.2	4.7	21.9	28.2	21.3
	0.7	28.3	32.3	19.2	22.2	19.7	16.1	15.6	30.4	18.0	5.1	23.8	28.5	21.6
	0.8	26.5	32.0	18.5	20.3	19.6	16.8	15.0	29.8	17.0	6.5	24.8	28.1	21.2
	0.9	25.0	30.1	17.9	19.4	19.7	16.6	14.0	29.3	16.6	7.1	25.5	26.5	20.6
M+E	0.0	25.9	29.9	15.7	21.3	17.9	15.9	15.8	23.4	17.9	1.2	14.5	18.5	18.2
	0.1	26.4	30.2	16.2	22.1	19.1	16.6	16.5	24.8	18.0	1.9	16.2	19.9	19.0
	0.2	27.4	31.5	16.4	23.0	19.6	17.1	16.7	26.7	18.0	2.4	17.7	21.2	19.8
	0.3	28.1	32.2	16.8	23.7	20.3	17.8	16.0	28.4	18.1	3.3	18.5	23.3	20.5
	0.4	28.0	34.0	17.0	23.9	19.9	18.3	16.7	29.4	18.1	4.3	20.7	23.7	21.2
	0.5	27.9	34.4	17.5	23.5	20.4	18.6	16.8	29.2	18.6	6.5	25.8	25.8	22.1
	0.6	26.6	32.9	17.7	21.4	21.0	18.9	16.9	29.1	18.3	6.2	22.3	28.9	21.7
	0.7	25.9	33.0	18.2	20.6	20.6	18.4	16.2	29.4	17.5	7.0	23.4	28.4	21.6
	0.8	25.0	32.0	18.2	20.1	20.4	17.9	14.9	29.5	15.9	7.5	24.5	27.1	21.1
	0.9	23.9	28.8	18.1	18.7	19.6	16.9	14.1	28.6	15.1	7.5	25.2	26.6	20.3
I+E	0.0	24.0	25.9	14.5	21.9	15.3	13.2	14.7	22.5	9.9	1.2	11.7	13.0	15.6
	0.1	24.3	27.8	14.8	22.4	16.0	13.4	15.0	25.0	11.7	1.4	13.4	14.8	16.7
	0.2	25.4	29.4	15.9	22.1	18.0	14.7	14.8	26.7	13.1	1.4	15.0	16.8	17.8
	0.3	26.2	31.0	16.5	22.7	18.5	15.9	15.0	27.7	14.0	2.2	16.6	19.3	18.8
	0.4	26.9	32.0	16.9	23.0	19.2	16.8	16.1	27.7	14.7	3.0	18.8	21.1	19.7
	0.5	27.1	32.3	17.6	21.9	19.5	16.7	16.1	28.7	15.3	4.3	20.3	24.0	20.3
	0.6	27.0	32.5	18.1	21.7	19.6	16.9	16.0	29.4	15.7	4.9	21.6	25.5	20.7
	0.7	25.7	31.8	18.2	20.9	19.6	17.5	16.0	29.7	15.5	5.7	22.8	26.3	20.8
	0.8	24.8	31.0	17.9	20.2	19.2	17.5	14.8	30.3	14.9	7.0	24.0	27.3	20.7
	0.9	23.2	29.2	17.8	18.5	19.0	16.7	14.2	29.0	14.2	6.9	25.0	26.6	20.0

Table 8: Full ablation results presented in Section 5.

we leveraged OpenAI's structured output feature, ensuring that all output scenarios should be JSON format. For the prompt used in this process, please refer to Table 9 in C.6.

For each individual document, we did not set a fixed number of scenarios to be generated. Instead, we employ an adaptive approach, where the number of generated scenarios is determined based on the content of each document. Documents with more extensive content resulted in a larger number of generated scenarios, while those with less content generated fewer scenarios.

C.2.2 *Scenario generator*

The scenario generator is optimized using AdamW with a learning rate of 2e-5, a linear warmup scheduler, weight decay of 0.1, and a batch size of 4 with gradient accumulation of 2. Optimization is conducted for a maximum of 10 epochs, with early stopping based on evaluation loss and a patience of 4. The LoRA hyperparameters are set as follows: r = 32, alpha=64, dropout = 0.1. After training the scenario generator, we applied it to the entire BRIGHT corpus to generate scenarios for each dataset.

C.3 Reasoning-augmented query

BRIGHT provides reasoning-augmented queries, which are reformulated versions of the original queries generated using various large language models. These queries incorporate explicit step-by-step reasoning, aiming to clarify the user intent and better guide retrieval models. Among the available variants, we adopt the GPT-4–generated reasoning queries, which achieved the best performance in the experiments reported in the Su et al. (2024).

C.4 Criteria for human evaluation

We randomly sample 100 examples from the BRIGHT test set and ask three human judges per example to compare different retrieval contexts following four criteria:

- **Specificity**: Which search result better provides information that is more specific in detail?
- **Comprehensibility**: Which search result is more easily understandable and clear, allowing you to grasp the overall content at a glance?
- **Usefulness**: Which search result is more practically helpful in solving user problems or aiding decision-making in the query?
- **Overall**: Which search result do you prefer overall when reviewing search results?

We show the interface for the evaluation in Figure 8.

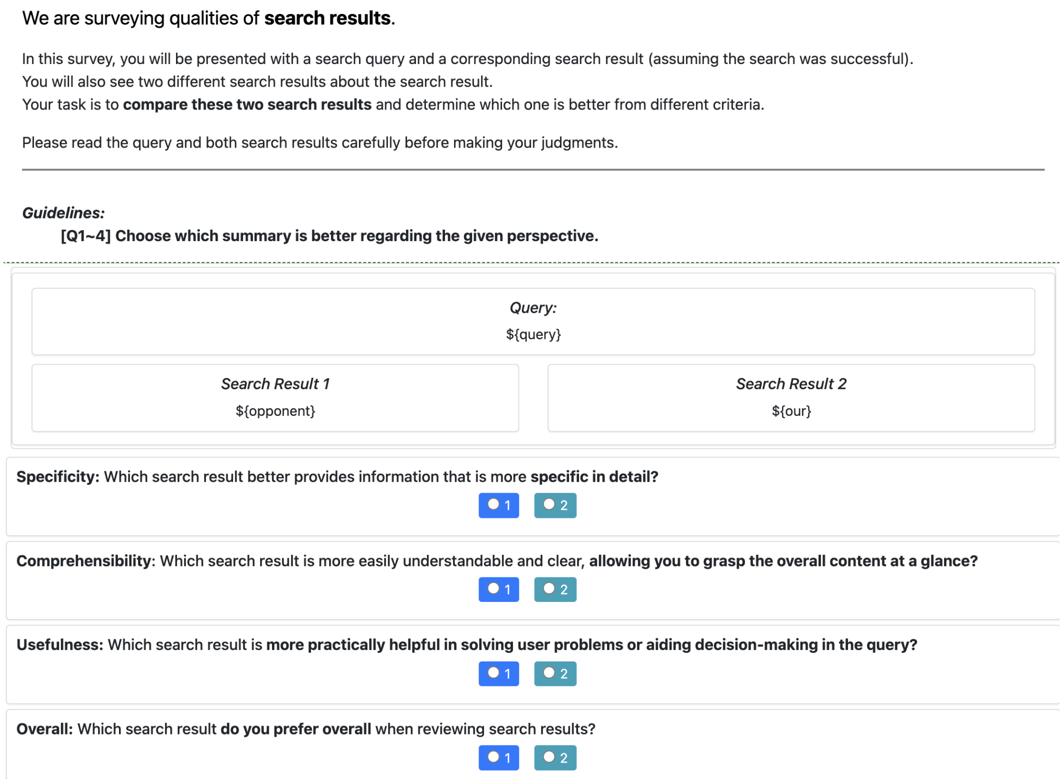
C.5 Experimental setting for RAG experiment

In Section 4.4, we investigate whether SPIKE can serve as an effective context provider in retrieval-augmented generation (RAG) by offering additional information alongside retrieved documents. In this experiment, we use the queries and reference answers from the BRIGHT dataset, following the experimental setup proposed by Su et al. (2024), and use GPT-4o as the evaluation model to score the generated answers. Specifically, in the oracle setting, the full set of gold documents corresponding to each query is provided as context, while the retrieval setting uses the top-10 documents retrieved by the retrieval model. The prompts used for question answering and evaluation also follow those introduced in Su et al. (2024).

C.6 Prompt

We present four types of prompts used in our experiments:

- **Construct scenario-augmented dataset**: The prompt designed for constructing scenario-augmented dataset is shown in Table 9
- **Scenario generator instruction**: The prompt designed for training scenario generator is shown in Table 10
- **RAG answering**: The prompt designed for answering in RAG setting is shown in Table 11
- **Evaluate RAG**: The prompt designed for evaluating answer in RAG setting is shown in Table 12



We are surveying qualities of **search results**.

In this survey, you will be presented with a search query and a corresponding search result (assuming the search was successful).
You will also see two different search results about the search result.
Your task is to **compare these two search results** and determine which one is better from different criteria.

Please read the query and both search results carefully before making your judgments.

Guidelines:

[Q1~4] Choose which summary is better regarding the given perspective.

Query:
${query}

Search Result 1
${opponent}

Search Result 2
${our}

Specificity: Which search result better provides information that is more **specific in detail?**

○ 1 ○ 2

Comprehensibility: Which search result is more easily understandable and clear, **allowing you to grasp the overall content at a glance?**

○ 1 ○ 2

Usefulness: Which search result is **more practically helpful in solving user problems or aiding decision-making in the query?**

○ 1 ○ 2

Overall: Which search result **do you prefer overall** when reviewing search results?

○ 1 ○ 2

Figure 8: The interface for human evaluation

Prompt for constructing scenario-augmented dataset

[Task Description]

You are an advanced language model specializing in knowledge extraction and user need modeling. Your task is to extract hypothetical user scenarios from a given dataset document, ensuring that the generated information needs reflect the document’s overall insights and knowledge, rather than isolated details.

Step 1: Document Analysis:
Summarize the key points of the document in a structured manner. This step should not be a direct extraction but should synthesize the document’s core concepts, key arguments, and insights, avoiding specific code snippets, variable names, or minor details.
Content:
- Main Topic: Briefly describe the primary subject of the document
- Key Aspects: Summarize the core concepts, insights, or knowledge presented

Step 2: Generate Possible Information Needs:
Based on the document analysis, generate a diverse set of possible information needs that can be satisfied by the document, ensuring that they **focus on high-level insights, generalizable knowledge, or core principles conveyed by the document rather than specific implementation details (e.g., function names, variable names, or isolated sections)**.
Guidelines:
- The information needs must align with the document’s main message and core knowledge, not minor details.
- Focus on concepts, reasoning, and insights rather than localized facts.
- Ensure that they **focus on high-level insights, generalizable knowledge, or core principles conveyed by the document rather than specific implementation details (e.g., function names, variable names, or isolated sections)**.
- Ensure that the needs **capture different aspects of the document’s knowledge** rather than concentrating on a single part.

Format:
- Each information need is started with ”A User wants to know”
- Generate a python list of information needs. (e.g. [”information need 1”, ”information need 2”, ”information need 3”])

Step 3: Generate Explanation for Each Information Need:
For each information need, explain how the document fulfills that need, ensuring that explanations are generalized and conceptual rather than overly detailed. Avoid focusing on function names, variable names, or specific lines unless absolutely necessary for clarity.

Format:
- Generate JSON format with the following components:
- Key: information need
- Value: explanation for the information need

[Text Content]

...

Table 9: The prompt for constructing scenario-augmented dataset

Scenario generator instruction

[Task Description]
You are an advanced language model specializing in knowledge extraction and user need modeling. Your task is to extract hypothetical user scenarios from a given dataset document, ensuring that the generated information needs reflect the document's overall insights and knowledge, rather than isolated details.

Content:
- Main Topic: Briefly describe the primary subject of the document
- Key Aspects: Summarize the core concepts, insights, or knowledge presented
- Information Needs: Generate a diverse set of possible information needs that can be satisfied by the document
- Explanation: Explain how the document fulfills that need, ensuring that explanations are generalized and conceptual rather than overly detailed.

Format:
- Generate JSON format

[Text Content]
...

Table 10: Scenario generator instruction

RAG answering Prompt

[Task Description]
Problem:
question

Document:
document

Based on the provided documents, write an answer to the problem.

Table 11: The prompt for RAG answering

RAG evaluation Prompt

[Task Description]

```
———- PROBLEM START ———-
problem
———- PROBLEM END ———-
———- STUDENT ANSWER START ———-
predicted answer
———- STUDENT ANSWER END ———-
———- REFERENCE ANSWER START ———-
gold answer
———- REFERENCE ANSWER END ———-
Criteria:
0 - The student’s answer is completely irrelevant or blank.
10 - The student’s answer addresses about 10% of the reference content.
20 - The student’s answer addresses about 20% of the reference content.
30 - The student’s answer addresses about 30% of the reference content.
40 - The student’s answer addresses about 40% of the reference content.
50 - The student’s answer addresses about 50% of the reference content.
60 - The student’s answer addresses about 60% of the reference content.
70 - The student’s answer addresses about 70% of the reference content.
80 - The student’s answer addresses about 80% of the reference content.
90 - The student’s answer addresses about 90% of the reference content.
100 - The student’s answer addresses about 100% of the reference content.
Use the following format to give a score:
REASON:
Describe why you give a specific score
SCORE:
The score you give, e.g., 60
Do not say anything after the score
```

Table 12: The prompt for RAG evaluation