IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 May 2025, accepted 27 June 2025, date of publication 15 July 2025, date of current version 28 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3589499

RESEARCH ARTICLE

Bridging the Question–Answer Gap in Retrieval-Augmented Generation: Hypothetical Prompt Embeddings

DOMEN VAKE[1,2], JERNEJ VIČIČ[1,3], AND ALEKSANDAR TOŠIĆ[1,2]

[1]Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, 6000 Koper, Slovenia
[2]InnoRenew CoE, 6310 Izola, Slovenia
[3]Research Centre of the Slovenian Academy of Sciences and Arts, Fran Ramovš Institute of the Slovenian Language, 4274 Žirovnica, Slovenia

Corresponding author: Domen Vake (domen.vake@famnit.upr.si)

This work was supported by European Union's Horizon 2020 under Grant 101135012.

ABSTRACT Retrieval-Augmented Generation (RAG) systems synergize retrieval mechanisms with generative language models to enhance the accuracy and relevance of responses. However, bridging the style gap between user queries and relevant information in document text remains a persistent challenge in retrieval-augmented systems, often addressed by runtime solutions (e.g., Hypothetical Document Embeddings (HyDE)) that attempt to improve alignment but introduce extra computational overhead at query time. To address these challenges, we propose Hypothetical Prompt Embeddings (HyPE), a framework that shifts the generation of hypothetical content from query time to the indexing phase. By precomputing multiple hypothetical prompts for each data chunk and embedding the chunk in place of the prompt, HyPE transforms retrieval into a question-question matching task, bypassing the need for runtime synthetic answer generation. This approach does not introduce latency but also strengthens the alignment between queries and relevant context. Our experimental results on six common datasets show that HyPE can improve retrieval context precision by up to 42 percentage points and claim recall by up to 45 percentage points, compared to standard approaches, while remaining compatible with re-ranking, multi-vector retrieval, query decomposition, and other RAG advancements.

INDEX TERMS Dense retrieval, embeddings, hypothetical prompt embeddings, large language models, retrieval-augmented generation.

I. INTRODUCTION

Retrieval-augmented generation (RAG) systems have emerged as a powerful paradigm in natural language processing, combining the strengths of retrieval-based approaches with the generative capabilities of large language models (LLMs) [1]. By leveraging external knowledge sources, RAG systems enhance the factual accuracy and relevance of generated responses, addressing the limitations of standalone generative models, such as outdated knowledge or limited access to restricted information. Despite their success, existing RAG implementations often struggle with aligning retrieval and generation in a way that efficiently bridges the gap between user queries and relevant document content.

To optimize retrieval performance, researchers have explored a variety of strategies, ranging from efficient chunking (splitting texts into coherent subunits) [1], [2] to re-ranking methods that refine initial retrieval results via cross-encoders or boosted similarity scoring [3]. Advanced frameworks such as GraphRAG exploit graph structures to capture cross-document relationships for more nuanced multi-hop or contextual queries [4]. Meanwhile, domain adaptation techniques focus on tailoring retrieval to specialized topics, ensuring that queries can be matched effectively, even in areas where language models might otherwise lack expertise.

Despite these efforts, a persistent hurdle in RAG remains the mismatch between user queries, which typically adopt an interrogative style, and corpus content, which is usually expository or declarative in nature. This style difference

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

hampers the alignment of query embeddings with document embeddings, occasionally allowing key information to go unretrieved. A notable solution to this problem is Hypothetical Document Embeddings (HyDE) [5], which prompts an LLM at query time to generate a synthetic answer, then uses that short text as the query for retrieval.

In this paper, we introduce Hypothetical Prompt Embeddings (HyPE), a new approach that tackles query-document style mismatch without adding overhead to every user request. Rather than generating synthetic answers at inference, HyPE precomputes multiple hypothetical questions for each corpus chunk at indexing time. These question-like prompts are embedded and stored, so that query matching effectively becomes a question-question retrieval problem. By shifting hypothetical generation offline, HyPE avoids additional runtime LLM calls.

To evaluate HyPE's effectiveness, we compare it against a naive RAG implementation and HyDE across multiple datasets and evaluation metrics, including precision, recall, and faithfulness. Our results demonstrate that HyPE offers substantial improvements in retrieval efficiency, reducing the computational burden while achieving comparable or better retrieval accuracy and contextual relevance.

The contributions of this paper are as follows:

- We introduce the concept of precomputed hypothetical prompt embeddings to optimize retrieval efficiency in RAG systems.
- We provide a comprehensive performance comparison between a Naive RAG implementation, HyDE, and HyPE.
- We present experimental results showcasing the trade-offs in retrieval quality and the effect of retrieval approach on generation across various datasets.

By shifting the hypothetical generation process from runtime to indexing, HyPE represents a scalable and efficient alternative for RAG systems, offering practical benefits for real-world applications requiring fast and reliable retrieval-augmented text generation. The rest of the paper is structured as follows: Section II presents the related works and surveys, Section III presents the used methodology. Follows the presentation of the experiment setting with the presentation of the datasets and the evaluation metrics. Section V presents the results with thorough analysis, Section VI presents the finals conclusions.

II. RELATED WORKS

Lewis et al. [1] introduced the original RAG framework, combining dense retrieval with generation to mitigate hallucination and provide grounding via external documents. While effective, early RAG pipelines [1], [6] often suffer from limitations in retrieval precision by simply embedding the user queries and retrieving text chunks via approximate nearest neighbour (ANN) search over a vector index. However, a persistent challenge remains: user queries are often phrased in question form, whereas documents or chunks are stored in an expository or statement-oriented style, creating a semantic or ''lexical-conceptual'' gap [7], [8]. This mismatch can degrade the retrieval's accuracy and ultimately weaken the generative model's faithfulness.

Cross-encoder reranking and multi-vector representations [9] enhance retrieval quality post hoc, but introduce additional inference-time latency.

One direction of research attempts to alleviate this mismatch by expanding documents with likely queries. For instance, Doc2Query [10] uses a sequence-to-sequence model (often T5) to generate synthetic questions for each document, appending them to the text so that a bag-of-words ranker like BM25 [11] can match real user queries more easily. While Doc2Query often boosts first-stage recall, subsequent studies noted that the generation process can hallucinate irrelevant expansions, leading to index bloat. Among them, Doc2Query– (Minus-Minus) [12] addresses this by filtering out low-quality expansions, improving accuracy and reducing index size for BM25-based retrieval. However, these methods predominantly operate in a lexical retrieval space rather than dense embeddings, and they store expansions as text appended to each document and, as such, act more like an enrichment of the chunks.

Another related direction focuses on generating synthetic training data to train or fine-tune dense retrievers in new domains. Ma et al. [13] propose using a question-generation model, trained on general-purpose Question-Answer(QA) pairs, to generate ''pseudo-queries'' for domain-specific passages. A dual-encoder retrieval model is then trained on these synthetic (question, passage) pairs—effectively learning domain adaptation in a zero-shot setting. While powerful for building domain-specialized retrievers, the method requires re-training or fine-tuning a dense model on large-scale synthetic data. By contrast, our approach bypasses model re-training at retrieval time and, instead, alters how we store the passages (i.e., their hypothetical question embeddings).

More recently, HyDE [5] addresses query-document mismatch by generating a hypothetical answer or short passage at query time. Instead of embedding the user's question directly, HyDE prompts an LLM to produce an approximate response, then embeds that synthetic text. This is used to retrieve relevant real documents from a vector index. While HyDE can improve retrieval accuracy for zero-shot question answering, it incurs an extra inference cost per user query. Additionally, the method may struggle, where the prompt queries for niche domain knowledge, where the model may not have sufficient knowledge to produce a representative sample. Building on this line of work, Eibich et al. [14] conducted an empirical study comparing RAG retrieval enhancements, including HyDE, reranking, multi-query expansion, and maximal marginal relevance (MMR). Their findings highlight HyDE's strong performance in both recall and faithfulness metrics, while also noting its trade-offs in runtime efficiency.

Several surveys have recently addressed retrieval-augmented systems. Gupta et al. [15] and Cheng et al. [16]

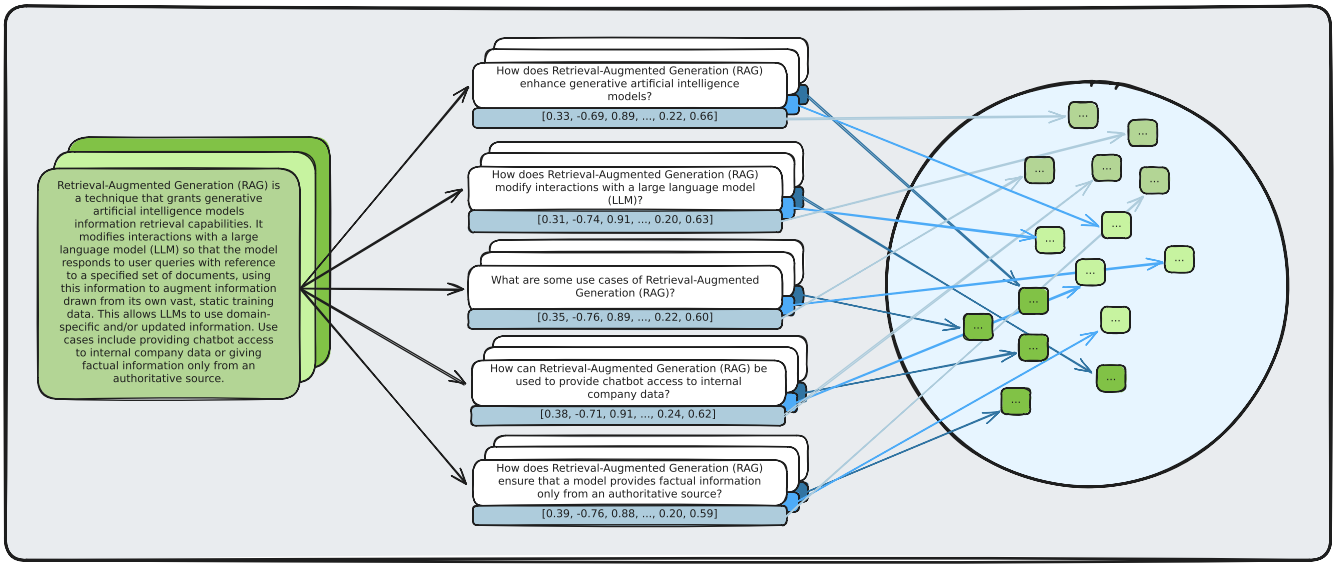

FIGURE 1. **Illustration of the Hypothetical Prompt Embeddings (HyPE) framework, showcasing the process of precomputing hypothetical questions during indexing to optimize retrieval efficiency in Retrieval-Augmented Generation (RAG) systems.**

offer extensive taxonomies of retrieval mechanisms, identifying open challenges in domain adaptation, embedding alignment, and inference efficiency. Notably, both surveys identify the need for methods that preserve retrieval quality while reducing reliance on runtime LLM calls.

In this context, our proposed Hypothetical Prompt Embeddings (HyPE) introduces a novel retrieval strategy that pre-computes hypothetical question-style prompts at indexing time. This shifts the burden of LLM generation to the offline phase.

III. METHODOLOGY

HyPE addresses the challenge of aligning user queries and relevant content by pre-computing hypothetical prompts at the indexing stage, contrasting with HyDE's runtime generation of synthetic answers. This shift avoids additional inference overhead per query and improves retrieval precision by ensuring that both user queries and stored embeddings share a question-like form.

The method begins by splitting the corpus D into coherent chunks $C_1, C_2, \ldots, C_n$, where each chunk provides a self-contained unit of information. For each chunk C_i, an LLM G generates multiple hypothetical prompts $Q_i = q_{i1}, q_{i2}, \ldots, q_{ik}$, simulating possible user queries that the chunk might answer. This offline step does not introduce any additional computational cost at query time, as no new prompts need to be generated for each user request.

Each hypothetical prompt q_{ij} is then mapped to an embedding $v_{ij} = f(q_{ij}) \in \mathbb{R}^d$ using a pre-trained dense retrieval model f. Rather than storing these prompt embeddings separately, we associate each v_{ij} with the original chunk C_i, thus building an index of vector-chunk pairs:

$$E = \{(v_{11}, C_1), (v_{12}, C_1), \ldots, (v_{nk}, C_n)\}$$

Each chunk is effectively represented multiple times, once for each hypothetical prompt. This extends the coverage of how queries may be phrased and matched.

Algorithm 1 HyPE Indexing Phase (offline)

Require: Corpus $D = \{d_1, \ldots, d_M\}$; chunker $\mathcal{C}$; generator LLM G; encoder f; prompts-per-chunk k
Ensure: Vector index E mapping embeddings to chunks

```
E ← ∅
for all document d ∈ D do
    C ← C(d)                                  ▷ split d into coherent chunks
    for all chunk c ∈ C do
        Q ← G(GenerateQuestions(c), k)
        for all question q ∈ Q do
            v ← f(q)                          ▷ embed question
            E ← E ∪ {(v, c)}
        end for
    end for
end for
return E
```

The retrieval process at runtime follows a standard approximate nearest-neighbor (ANN) search in the vector space. When a user query q arrives, it is embedded into $q = f(q)$. The system then locates the nearest v_{ij} vectors within E, and retrieves the associated chunks for final answer generation by an LLM. Although the pipeline remains structurally similar to a Naive RAG, the key difference is that HyPE matches questions against questions, rather than questions against chunk text.

At present HyPE treats every generated question in the set $Q_i = \{q_{i1}, \ldots, q_{ik}\}$ with equal importance: each prompt is embedded once and contributes a single vector to the index. We do not yet attempt to decide which

hypothetical questions are ''better'' or discard those that are less representative. Determining prompt quality is an open issue—and likely to be domain-dependent. In settings where domain knowledge is available (e.g. biomedical literature or legal texts) conditioning the LLM on that knowledge could produce more accurate or stylistically appropriate questions, which in turn should strengthen retrieval. Investigating prompt scoring and domain-specific generation therefore remains future work.

Algorithm 2 HyPE Retrieval Phase (online)

Require: User query q; encoder f; index E; top-k
Ensure: Relevant chunk set $\mathcal{R}$
1: $\mathbf{v}_q \leftarrow f(q)$
2: $\mathcal{R} \leftarrow \text{ANN\_Search}(E, \mathbf{v}_q, k)$ ▷ k nearest vectors **return** $\mathcal{R}$

This question-question alignment increases the probability of finding the correct chunks for two main reasons. First, many embedding models exhibit style-based clustering [17]. Texts of similar form (e.g., interrogative sentences) often lie closer in the vector space. As a result, a user's real-world query naturally aligns more closely with the hypothetical prompts that share its interrogative style. Second, generating multiple hypothetical queries per chunk broadens the ''semantic reach,'' covering a wider range of possible question formulations. Even if a user query is phrased in a slightly different way, there is a higher chance that at least one of the chunk's hypothetical questions closely corresponds to it.

Another advantage of HyPE lies in how it addresses the inherent chunking tradeoff in retrieval systems. If chunks are too large, their embeddings become less precise because they encode a mix of multiple concepts, making vector-based similarity less reliable [18]. Conversely, reducing chunk size improves embedding specificity but risks losing crucial surrounding context. HyPE mitigates this issue by ensuring that each stored vector represents a specific piece of information within a chunk, while retrieval still returns the entire chunk with its broader context. This allows the system to retain the benefits of detailed, fine-grained embeddings without sacrificing the context for accurate retrieval.

HyPE invokes the language model once per chunk, prompting it to return a set of m hypothetical questions in a single call. Consequently, a corpus consisting of n chunks requires n LLM calls during indexing, regardless of how many prompts are generated for each chunk. While this upfront cost can be substantial for very large datasets, it is paid only once and is strictly proportional to the dataset size. After the index is built, HyPE's online path involves nothing more than standard vector search, incurring no additional LLM calls at query time and keeping serving latency and operating cost flat even as query volume grows.

IV. EXPERIMENTS

We evaluated three RAG pipelines to assess retrieval and generation performance as presented in Table 1. While a naïve retriever establishes how much can be achieved without any style-bridging, HyDE is the only published RAG component that explicitly targets the same ''question-to-statement'' gap as HyPE, albeit at inference time via synthetic-answer generation. Including HyDE therefore yields a good comparison and isolates the effect of moving hypothetical content creation from the query stage (HyDE) to the indexing stage (HyPE).

TABLE 1. Key differences of compared pipelines.

Retriever Pipeline	Augmentation stage	Context Space
Naive RAG	/	prompt-to-document
HyDE	Inference	document-to-document
HyPE	Indexing	prompt-to-prompt

For evaluating the RAG pipelines, we used the RAGChecker framework [19], a comprehensive evaluation toolkit developed by Amazon Science for assessing both retrieval and generation performance in RAG systems. It provides structured metrics to analyse retrieval effectiveness through context precision, which measures how many retrieved passages are relevant, and claim recall, which assesses whether all necessary information is retrieved.

For generation, RAGChecker evaluates faithfulness, ensuring the generated text remains grounded in the retrieved passages, along with the hallucination rate, which identifies unsupported claims, and context utilization, which measures how effectively retrieved passages contribute to responses. Additionally, it assesses robustness through noise sensitivity, testing the system's response to query variations, and self-knowledge, which quantifies the model's ability to recognize when it lacks sufficient information. Additionally, it computes precision, recall, and F1 scores, providing an overall measure of retrieval and response accuracy [19].

A. DATASETS

We evaluated our approach on six datasets, chosen to test distinct aspects of RAG systems. *MS MARCO* [20], a large-scale question-answering benchmark, and *Ragas-WikiQA*[1] evaluate general-purpose retrieval in real-world scenarios. *RAG-dataset-12000*[2] and *MultiHopRAG* [21] emphasize multi-hop reasoning, requiring systems to synthesize information across multiple documents. *RAGBench* [22] tests hybrid tasks demanding both precise retrieval and coherent generation. Finally, *Single-Topic RAG* dataset[3] focuses on narrow domains, assessing precision in specialized contexts. A concise summary of their key statistics is given in Table 2.

[1]https://huggingface.co/datasets/explodinggradients/ragas-wikiqa
[2]https://huggingface.co/datasets/neural-bridge/rag-dataset-12000
[3]https://www.kaggle.com/datasets/samuelmatsuoharris/single-topic-rag-evaluation-dataset

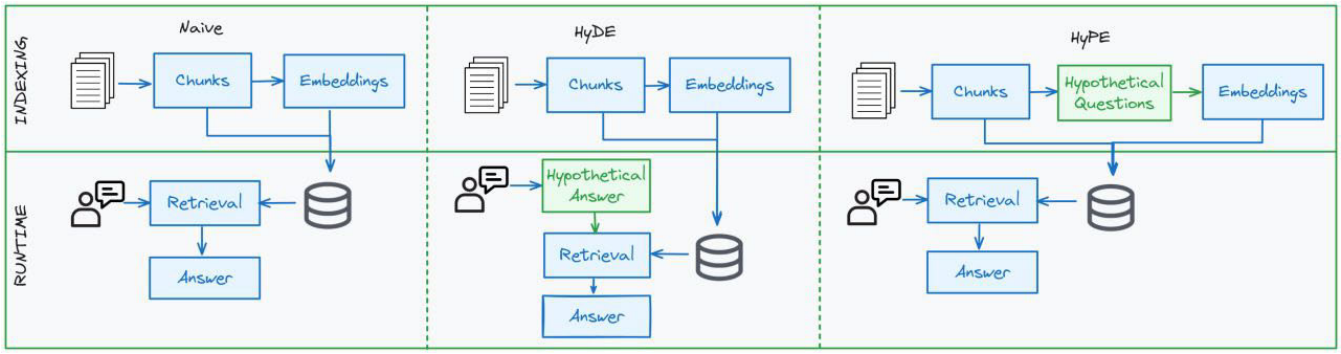

FIGURE 2. The image depicts the workflows of three retrieval-augmented generation (RAG) pipelines tested in the experiments: Naive RAG, HyDE, and HyPE. Blue components show parts of the pipeline that are the same across all setups and the green components show additional steps in the pipelines.

> Analyze the input text and generate essential questions that, when answered, capture the main points and core meaning of the text. The questions should be exhaustive and understandable without context. When possible, named entities should be referenced by their full name. Only answer with questions where each question should be written in its own line (separated by newline) with no prefix.

FIGURE 3. Prompt used to generate hypothetical prompts.

TABLE 2. Descriptive statistics of the six datasets used in our study. For each dataset we list its thematic focus, the number of gold question-answer pairs, the total number of text chunks created by our preprocessing, and the average chunk length in tokens.

Corpus	Domain / Focus	# Q&A pairs	# Chunks	Avg. chunk len. (tokens)
MS MARCO	Web search passages	82326	676193	82
RAGBench	Mixed downstream tasks	73286	317563	173
Ragas-WikiQA	Wikipedia factoid QA	232	460	688
RAG-dataset-12000	General knowledge QA	9600	18321	378
MultiHopRAG	Multi-hop reasoning	2556	3101	443
Single-Topic RAG	Narrow domain articles	80	1324	467

All datasets already come pre-segmented into chunks, except for *RAG-dataset-12000* that contains a single context block, and *MultiHopRAG*, which contains multiple larger documents. For these two cases, we manually split the source text into segments of maximum 500 tokens, overlapping each segment by 50 tokens to preserve cross-boundary coherence. This approach, while straightforward, may not be optimal as the choice of chunking strategy can significantly impact retrieval effectiveness and quality of generation. Accordingly, we apply the same chunking procedure across all three pipelines for consistency in our experiments, but we note that different pipelines might benefit from tailored chunking strategies. We leave an in-depth exploration of chunk size, overlap, and other segmentation heuristics as a direction for future research.

B. EVALUATION METRICS

For all pipelines, we tested retrieval depths $k \in \{1, 3, 5, 10\}$. At $k = 5$, we additionally compared cosine similarity and Euclidean distance functions to assess their impact on chunk relevance ranking. The embedding model chosen for all pipelines was *bge-m3* [23] for dense vector representations. The generator LLM used was *Mistral-NeMo*.[4]

We chose Mistral-NeMo because its openly released weights make the model easy for anyone to download and run, ensuring that every result in this paper can be reproduced without relying on a commercial API. During the time of testing, published benchmark scores place it in the top tier of 7-13 B open-source LLMs for instruction following and QA, so it provides competitive generation quality while remaining fully replicable.

V. RESULTS WITH ANALYSIS

In Table 3, we compare the three retrieval methods across six datasets and varying numbers of retrieved chunks (k). Each cell reports context precision (how many of the retrieved

[4]https://mistral.ai/news/mistral-nemo/

chunks directly match the query's needs) and claim recall (how many relevant pieces of information are captured). Overall, HyPE improves recall by about 16 percentage points and precision by about 20 percentage points compared to Naive RAG, on average. The difference can be even larger on specific datasets, such as *Single-Topic RAG* at $k = 1$ where HyPE surpasses Naive RAG by more than 40 percentage points in precision and at $k = 10$ surpasses by 44.6 percentage points in recall. Although HyPE performs slightly below Naive RAG on *MS MARCO* at $k = 1$, it catches up or exceeds Naive RAG at deeper retrieval. For *RAGBench* and *Ragas-WikiQA*, HyPE also achieves strong gains, especially at lower k values, indicating its ability to retrieve the correct chunks accurately.

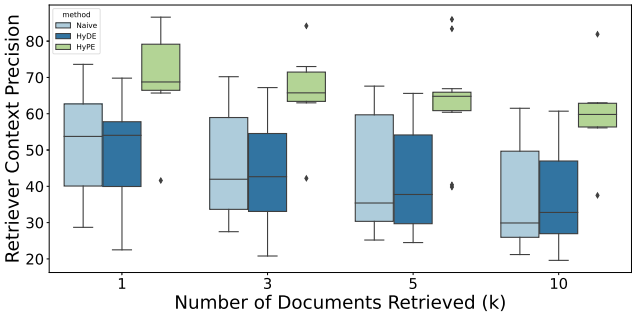

FIGURE 4. Box plot comparison of Retriever Context Precision across different numbers of documents retrieved (k) for three methods: Naive, HyDE, and HyPE. The plot illustrates the distribution and variability of precision scores for each method and retrieval depth.

Figure 4 compares Retriever Context Precision across different numbers of documents retrieved (k) for the Naive, HyDE, and HyPE methods revealing significant improvement in precision, using HyPE. As the number of retrieved documents increases from 1 to 10, HyPE consistently demonstrates higher precision. Its precision is notably superior to both Naive and HyDE methods. This suggests that HyPE's approach of precomputing hypothetical questions during the indexing phase effectively aligns retrieved content with user queries, reducing the semantic mismatch often encountered in traditional methods. The narrower interquartile ranges for HyPE further indicate its consistency in retrieving relevant information across varying retrieval depths.

Additionally, the figure 5 of claim recall complements these findings. Claim recall measures the proportion of relevant information successfully retrieved from the documents. The balanced performance in both metrics highlights HyPE's effectiveness in bridging the gap between user queries and relevant document content.

The comparison in Figure 6 shows that for all three methods the choice between Euclidean and Cosine distance metrics does not significantly impact their effectiveness.

In Figure 7 we report generator metrics: context utilization, noise sensitivity, hallucination, self-knowledge, and faithfulness. These scores are properties of the generator language model, not of the retriever itself. Because HyPE intervenes

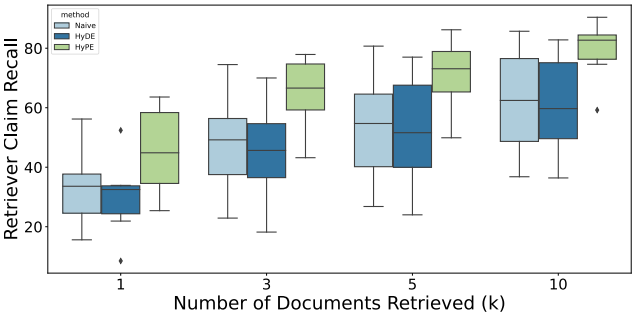

FIGURE 5. Box plot comparison of Retriever Claim Recall across different numbers of documents retrieved (k) for three methods: Naive, HyDE, and HyPE. The plot illustrates the distribution and variability of precision scores for each method and retrieval depth.

only in the retrieval stage, we keep the generator LLM fixed and any other foundation model could be dropped in without changing the retrieval logic. Accordingly, shifts in these metrics across pipelines reflect how the quality of the retrieved context influences a given LLM's behaviour, rather than inherent differences between language models.

HyPE consistently achieves higher context utilization and faithfulness, indicating that its retrieval strategy provides more relevant and coherent supporting text for generation. However, the performance on noise sensitivity metrics presents a more nuanced picture. For 'Noise sensitivity in relevant contexts', HyPE registers a higher score compared to Naive RAG and HyDE. Within the evaluation framework [19], this higher score signifies worse performance, suggesting the generator makes more errors when relevant retrieved documents are affected by noise. A potential explanation lies in HyPE's retrieval of multiple copies of the relevant chunks. Although beneficial for faithfulness through reinforcement, this very redundancy could increase the generator's susceptibility to errors since noise is repeated along with relevant information. Conversely, HyPE shows marginally better performance on 'Noise sensitivity in irrelevant contexts' (achieving a slightly lower score), indicating slightly improved handling of noise associated with irrelevant documents compared to the baseline methods.

HyPE also exhibits lower hallucination rates compared to Naive RAG and HyDE, reinforcing the idea that better-aligned retrieval reduces the likelihood of introducing incorrect or unsupported claims. Although these results are based on a single LLM, the trend is likely to generalize across different models, as improvements in retrieval typically translate to improved generation performance. However, the exact degree of impact may vary depending on the LLM's retrieval dependence and sensitivity to context quality. The combination of improved context grounding, reduced hallucinations, and stronger response alignment highlights HyPE's potential for enhancing the reliability of RAG systems.

Figure 8 compares the overall F1 scores for retrievers and generators of the pipelines through the datasets. With the

TABLE 3. Performance comparison of retrieval methods across datasets using context precision and claim recall metrics (bigger is better) with varying numbers of retrieved context chunks (k). Gradient intensity reflects metric strength (lighter to darker green indicates lower to higher values), with bold entries highlighting the best-performing method for each configuration.

Dataset@k	Naive		HyDE		HyPE	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
RAG-12000@1	55.8	34.7	55.1	33.3	**82.6**	**63.6**
RAG-12000@3	36.5	44.2	36.6	42.8	**73.0**	**76.2**
RAG-12000@5	31.3	49.0	32.0	47.9	**66.9**	**80.1**
RAG-12000@10	26.4	56.1	27.4	55.6	**56.1**	**84.6**
MS MARCO@1	**73.6**	**56.2**	69.8	52.4	68.7	50.2
MS MARCO@3	**70.2**	**74.5**	67.2	70.0	66.9	70.2
MS MARCO@5	**67.6**	**80.6**	65.5	76.8	65.6	77.3
MS MARCO@10	61.5	**85.7**	60.7	82.8	**62.5**	84.0
MultiHop@1	36.2	21.9	35.6	21.9	**43.7**	**27.2**
MultiHop@3	32.7	35.3	31.9	34.4	**42.2**	**43.2**
MultiHop@5	30.3	39.9	30.3	39.8	**40.1**	**50.6**
MultiHop@10	25.8	46.2	26.8	47.6	**37.5**	**59.2**
RAGBench@1	65.0	38.7	58.7	33.9	**65.7**	**39.5**
RAGBench@3	62.8	54.2	56.5	48.5	**63.0**	**58.0**
RAGBench@5	60.0	60.7	54.3	55.3	**60.7**	**65.6**
RAGBench@10	55.1	68.8	49.9	63.8	**57.1**	**74.6**
Single-Topic@1	28.7	15.6	22.5	8.5	**68.8**	**32.9**
Single-Topic@3	27.5	22.9	20.8	18.2	**64.6**	**63.0**
Single-Topic@5	25.2	26.8	24.5	28.6	**64.5**	**69.0**
Single-Topic@10	21.2	36.8	19.6	36.4	**63.0**	**81.4**
WikiQA@1	51.7	32.5	53.0	31.7	**86.6**	**61.1**
WikiQA@3	47.4	57.1	48.7	56.7	**84.2**	**77.9**
WikiQA@5	40.1	65.0	43.8	67.6	**84.7**	**85.6**
WikiQA@10	33.4	79.1	38.2	78.9	**81.9**	**90.4**

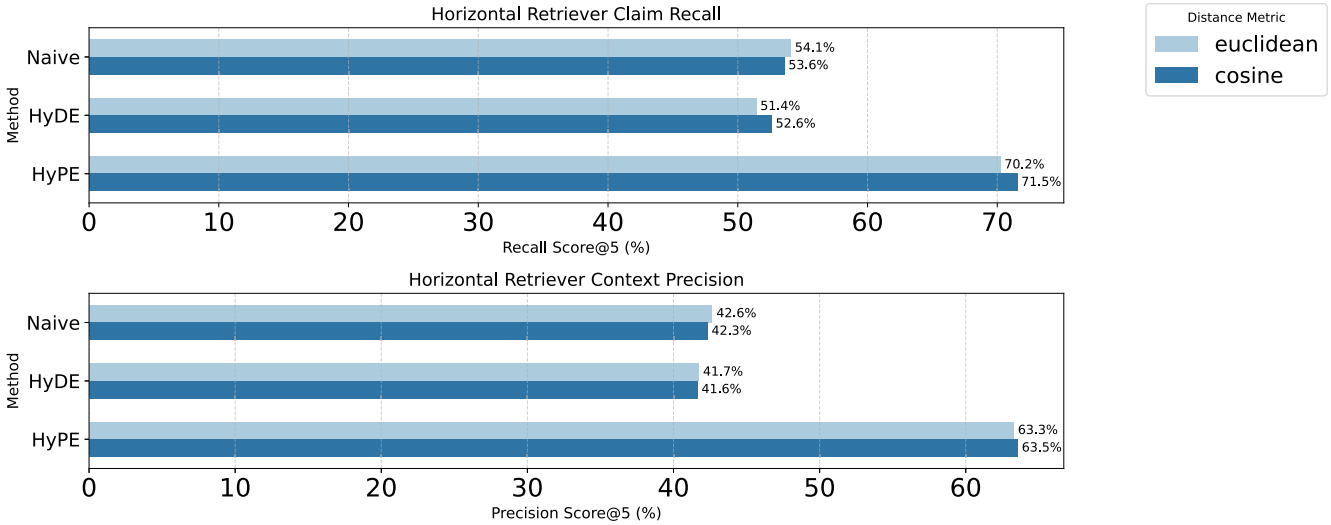

FIGURE 6. Comparison of Retriever Claim Recall and Context Precision across three retrieval methods using two distance metrics: Euclidean and Cosine. Performance measured at $k = 5$.

exception of the *MS MARCO* dataset where all three methods show comparable performance, the F1 scores show, that HyPE consistently outperforms the other two methods, particularly in datasets like *Single Topic RAG* and *RAG-dataset-12000*, where the complexity and specificity of queries demand precise retrieval.

The MS MARCO benchmark stands out as an outlier in our results, exhibiting comparable retrieval efficacy across the Naive RAG, HyDE, and HyPE pipelines. This convergence can be attributed to several factors inherent to the dataset that limit the differential impact of HyPE's enhancements. Firstly, MS MARCO utilizes short, answer-centric passages (typically around 60 tokens), which facilitates strong baseline performance via direct query-passage matching, leaving less room for augmentation techniques to provide significant added value. Secondly, the high lexical overlap inherent

TABLE 4. Aggregate performance across the six evaluation datasets (mean ± sd). For metrics marked ↑, higher is better; for those marked ↓, lower is better. Best value in each row is marked in bold.

Metric	Naive	HyDE	HyPE
Retriever claim recall ↑	53.6 ± 19.0	52.6 ± 17.8	**71.5 ± 12.5**
Retriever context precision ↑	42.3 ± 17.4	41.6 ± 15.9	**63.5 ± 13.8**
Generator context utilisation ↑	40.0 ± 13.1	39.0 ± 15.7	**53.5 ± 7.8**
Generator faithfulness ↑	52.2 ± 15.0	51.4 ± 14.8	**69.3 ± 6.0**
Generator hallucination ↓	26.0 ± 11.9	25.1 ± 11.4	**19.9 ± 8.2**
Noise sensitivity (irrelevant) ↓	7.5 ± 3.8	**7.2 ± 2.7**	**7.2 ± 4.0**
Noise sensitivity (relevant) ↓	**13.8 ± 7.8**	14.2 ± 6.6	21.0 ± 4.4
Self-knowledge ↓	6.0 ± 3.4	5.4 ± 2.6	**3.3 ± 1.4**
Overall F1 ↑	27.9 ± 9.7	27.2 ± 9.6	**37.6 ± 7.7**
Overall precision ↑	35.2 ± 8.4	33.8 ± 7.5	**42.6 ± 7.6**
Overall recall ↑	37.9 ± 14.1	38.5 ± 13.9	**50.4 ± 6.8**

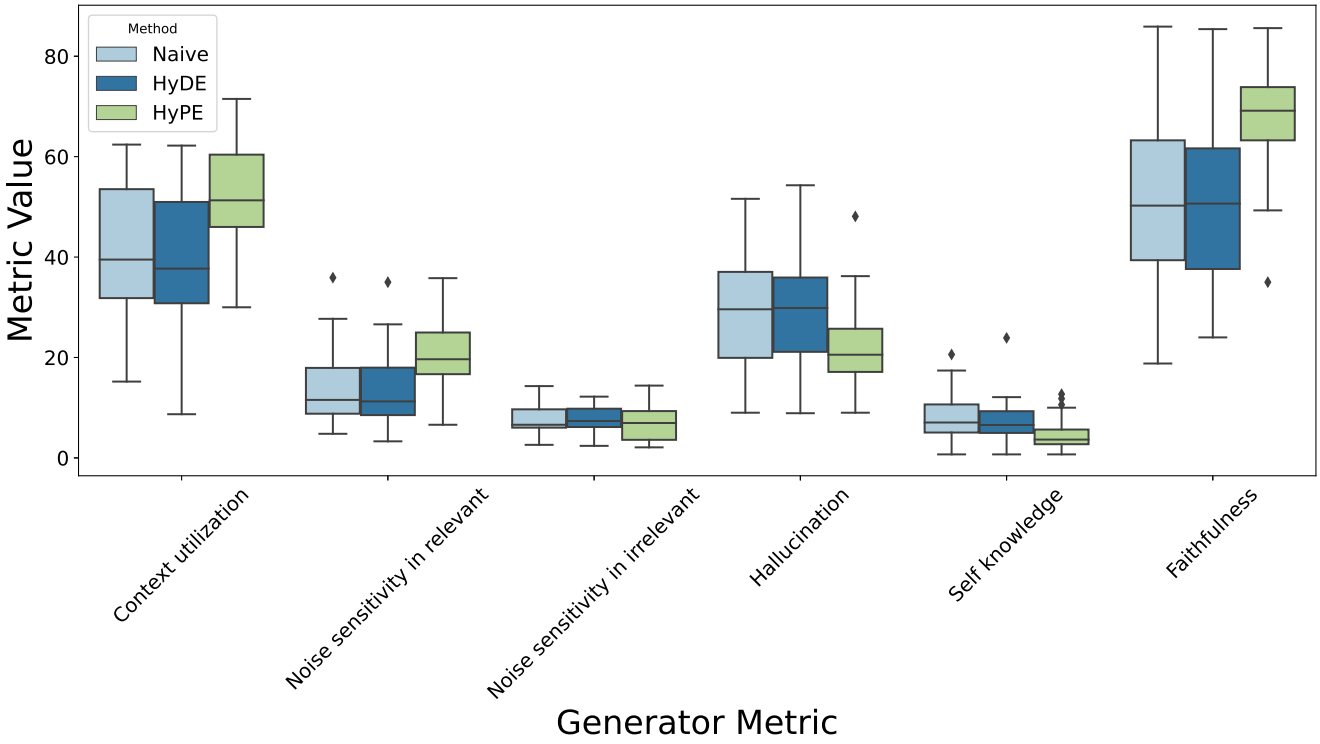

FIGURE 7. Comparative box plot analysis of various generator metrics when using one of the three retrieval methods, highlighting the performance differences among the methods in terms of their effectiveness and reliability in generating accurate and contextually relevant responses.

between its web search-derived queries and corresponding passages substantially reduces the query-document style gap that HyPE aims to mitigate. Lastly, the high baseline retrieval scores observed in our experiments on MS MARCO, achieved even without augmentation using a powerful dense embedding model, suggest a performance saturation effect, where further architectural refinements struggle to yield significant marginal improvements. Consequently, while HyPE offers substantial gains on datasets characterized by longer documents and greater stylistic divergence between queries and context, its advantages are less pronounced given the specific properties of the MS MARCO passage retrieval task.

Given the small paired sample size and the absence of any firm evidence for normality, we adopt the distribution-free Wilcoxon signed-rank test with Holm-Bonferroni adjustment instead of a paired t-test.

Across all eleven metrics, the Wilcoxon results in Table 5 indicate a consistent advantage for HyPE over both baselines. Nine metrics achieve an adjusted $p < 0.10$ in both comparisons, and five of those also satisfy the stricter $p < 0.065$ step that corresponds to the minimum attainable exact level. Effect sizes are uniformly medium to large: Cliff's $|\delta|$ ranges from 0.44 to 0.72, implying that a random sample from HyPE exceeds its baseline counterpart in roughly 70-85% of the paired datasets. The only metric without a discernible difference is *noise-sensitivity in irrelevant context* ($p_{\text{adj}} = 1.0$, $|\delta| < 0.11$), indicating that all methods degrade equally when faced with distractor passages. Taken together with the statistics in Table 4, these findings show that HyPE's

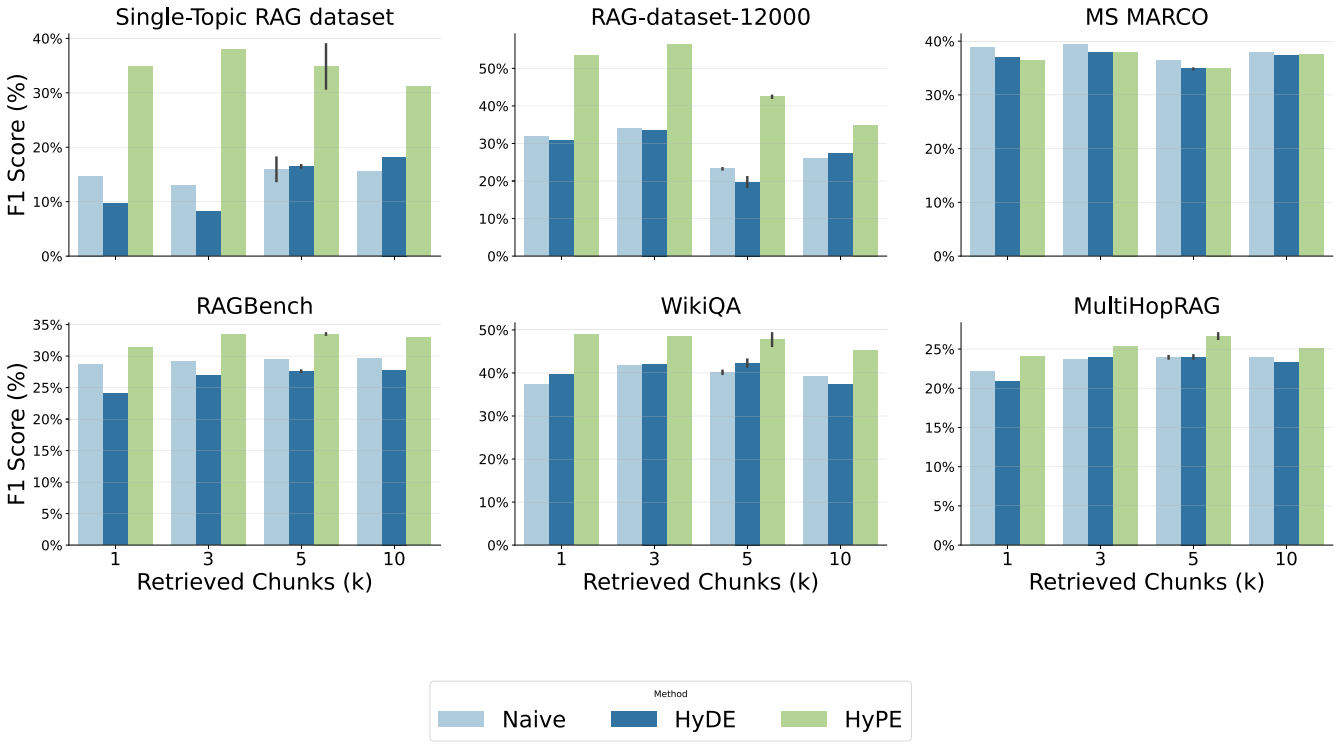

FIGURE 8. Bar chart comparison of F1 scores across six different datasets for three retrieval methods. Each subplot represents a dataset and shows the F1 scores for varying numbers of retrieved chunks (k = 1, 3, 5, 10).

TABLE 5. Wilcoxon signed-rank tests (HyPE vs. each baseline). Holm-adjusted *p*-values below 0.065 and Cliff's $|\delta| \geq 0.56$ are bold.

Metric	HyPE vs Naive	HyPE vs HyDE
Overall precision	**0.094 / 0.56**	**0.063 / 0.72**
Overall recall	0.086 / 0.44	**0.086 / 0.58**
Overall F1	**0.063 / 0.56**	**0.063 / 0.61**
Retriever claim recall	**0.063 / 0.61**	**0.063 / 0.67**
Retriever context precision	**0.094 / 0.67**	**0.063 / 0.72**
Generator context utilisation	**0.063 / 0.67**	**0.063 / 0.61**
Noise sens. (relevant)	**0.063 / 0.61**	**0.063 / 0.67**
Self-knowledge	0.086 / 0.64	0.086 / 0.64
Hallucination	0.086 / 0.42	0.156 / 0.28
Noise sens. (irrelevant)	1.000 / 0.03	1.000 / 0.11
Faithfulness	**0.063 / 0.67**	**0.063 / 0.72**

Numbers are "adjusted p / $|\delta|$".

improvements are both statistically reliable and practically meaningful.

Although HyPE is evaluated here as a stand-alone retriever, its design is orthogonal to other retrieval-side optimization approaches and therefore combinable with them. For example, the pre-computed question vectors can feed into query-decomposition modules (which break multi-hop questions into simpler sub-queries), query-expansion or rewriting steps (such as Doc2Query or RePlug), and even re-ranking or multi-vector fusion frameworks like BM25 or ColBERT. In these composite pipelines HyPE simply replaces the original passage vectors while leaving the higher-level orchestration intact, making it a drop-in upgrade rather than a competing subsystem.

VI. CONCLUSION

The paper presents Hypothetical Prompt Embeddings (HyPE), a framework that pre-computes hypothetical prompts at indexing time to reshape retrieval in RAG pipelines into a prompt-to-prompt matching process. Our experimental findings show that HyPE surpasses both Naive RAG and HyDE on multiple datasets and metrics, with notable gains in precision and recall. By eliminating the need for query-time synthetic answer generation and instead relying on strategically generated questions offline, HyPE improves efficiency and provides stronger alignment between user queries and relevant content.

Although HyPE may not outperform every specialized RAG variant in all domains, it offers a flexible and modular upgrade to existing pipelines. Swapping in pre-computed question embeddings remains compatible with advances in chunking, re-ranking, multi-vector retrieval, and fine-tuning large language models. HyPE also integrates smoothly into agent-based systems, where prompt-level alignment can help specialized retrieval sub-agents handle distinct query types more effectively.

Looking ahead, combining HyPE with GraphRAG, which maps information from documents or chunks as graph nodes, may further enhance multi-hop reasoning and retrieval accuracy in complex scenarios. Such a hybrid approach could be especially valuable for building robust RAG systems.

Another direction for future research involves investigating the chunking tradeoff in the context of expanding LLM

context windows. As language models evolve to accommodate larger input lengths, it becomes increasingly feasible to supply bigger chunks as prompts. However, larger chunks can dilute semantic specificity in their embeddings, resulting in less precise vector matching. This tension between maintaining detailed embeddings and preserving broader context may become more pronounced as context windows expand. Further testing of chunk size and indexing depth with HyPE would help clarify how embedding precision balances with retrieval breadth.

Finally, we plan to validate HyPE on multilingual RAG benchmarks to confirm that the question-question alignment holds across languages and writing systems.

Overall, HyPE demonstrates that shifting from question-to-document to question-to-question alignment leads to tangible gains in retrieval accuracy and cost-effectiveness. As RAG solutions continue to evolve, offline prompt generation strategies like HyPE can serve as a foundation for more efficient generation.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9459–9474.

[2] A. Jimeno Yepes, Y. You, J. Milczek, S. Laverde, and R. Li, "Financial report chunking for effective retrieval augmented generation," 2024, *arXiv:2402.05131*.

[3] P. Mishra, A. Mahakali, and P. S. Venkataraman, "SEARCHD–Advanced retrieval with text generation using large language models and cross encoding re-ranking," in *Proc. IEEE 20th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2024, pp. 975–980.

[4] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "QA-GNN: Reasoning with language models and knowledge graphs for question answering," 2021, *arXiv:2104.06378*.

[5] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," 2022, *arXiv:2212.10496*.

[6] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, "Atlas: Few-shot learning with retrieval augmented language models," 2022, *arXiv:2208.03299*.

[7] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "The vocabulary problem in human-system communication," *Commun. ACM*, vol. 30, no. 11, pp. 964–971, Nov. 1987.

[8] R. Nogueira, W. Yang, J. Lin, and K. Cho, "Document expansion by query prediction," 2019, *arXiv:1904.08375*.

[9] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. N. Bennett, J. Ahmed, and A. Overwijk, "Approximate nearest neighbor negative contrastive learning for dense text retrieval," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–12.

[10] R. Nogueira, J. Lin, and A. Epistemic, "From doc2query to doctttttquery," *Online preprint*, vol. 6, no. 2, pp. 1–3, 2019.

[11] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.

[12] M. Gospodinov, S. MacAvaney, and C. Macdonald, "Doc2Query-: When less is more," in *Proc. Eur. Conf. Inf. Retr.*, 2023, pp. 414–422.

[13] J. Ma, I. Korotkov, Y. Yang, K. Hall, and R. McDonald, "Zero-shot neural passage retrieval via domain-targeted synthetic question generation," 2020, *arXiv:2004.14503*.

[14] M. Eibich, S. Nagpal, and A. Fred-Ojala, "ARAGOG: Advanced RAG output grading," 2024, *arXiv:2404.01037*.

[15] S. Gupta, R. Ranjan, and S. Narayan Singh, "A comprehensive survey of retrieval-augmented generation (RAG): Evolution, current landscape and future directions," 2024, *arXiv:2410.12837*.

[16] M. Cheng, Y. Luo, J. Ouyang, Q. Liu, H. Liu, L. Li, S. Yu, B. Zhang, J. Cao, J. Ma, D. Wang, and E. Chen, "A survey on knowledge-oriented retrieval-augmented generation," 2025, *arXiv:2503.10677*.

[17] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.

[18] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," 2020, *arXiv:2004.04906*.

[19] D. Ru, L. Qiu, X. Hu, T. Zhang, P. Shi, S. Chang, C. Jiayang, C. Wang, S. Sun, H. Li, Z. Zhang, B. Wang, J. Jiang, T. He, Z. Wang, P. Liu, Y. Zhang, and Z. Zhang, "RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation," 2024, *arXiv:2408.08067*.

[20] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "MS MARCO: A human generated MAchine reading comprehension dataset," 2016, *arXiv:1611.09268*.

[21] Y. Tang and Y. Yang, "MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries," 2024, *arXiv.2401.15391*.

[22] R. Friel, M. Belyi, and A. Sanyal, "RAGBench: Explainable benchmark for retrieval-augmented generation systems," 2024, *arXiv:2407.11005*.

[23] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," 2024, *arXiv:2402.03216*.

DOMEN VAKE received the B.S. and M.S. degrees in computer science from the University of Primorska (UP), Slovenia. He is currently pursuing the Ph.D. degree in computer science with the Faculty of Mathematics, Natural Sciences and Information Technologies (FAMNIT), Department of Information Sciences and Technologies (DIST). Previously, he was a Programmer with InnoRenew CoE. He is an Assistant Researcher with InnoRenew CoE and UP FAMNIT. His research interests include artificial intelligence with a larger focus on large language models and their usage, and blockchain technology.

JERNEJ VIČIČ is currently an Associate Professor and a Research Associate with the University of Primorska and the Research Centre of the Slovenian Academy of Sciences and Arts. He is also the Head of the Laboratory DLTLT, UP FAMNIT. His research interests include artificial intelligence, natural language processing, computational linguistics, and distributed systems. In 2023, he received the Golden Plaque of the University of Primorska.

ALEKSANDAR TOŠIĆ received the B.S., M.S., and Ph.D. degrees in computer science from the University of Primorska, Slovenia, in 2011, 2016, and 2022, respectively. He is currently an Assistant Professor with the University of Primorska and a Researcher with the InnoRenew CoE, Izola, Slovenia. Previously, he was a Young Researcher with InnoRenew CoE and a Teaching Assistant with the University of Primorska. His research interests include distributed systems, privacy and security, sensors, and distributed ledger technologies. He was a recipient of the Solemn Charter of the University of Primorska (2023) and the University Recognition for Academic and Research Achievements (2021).