

DoTA-RAG: Dynamic of Thought Aggregation RAG

Saksorn Ruangtanusak
SCBX
Bangkok, Thailand
saksorn.r@scbx.com

Natthapath Rungseesiripak
SCBX
Bangkok, Thailand
natthapath.r@scbx.com

Peerawat Rojratchadakorn
SCBX
Bangkok, Thailand
peerawat.r@scbx.com

Monthol Charattrakool
SCBX
Bangkok, Thailand
monthol.c@scbx.com

Natapong Nitarach
SCB 10X
Bangkok, Thailand
natapong@scb10x.com

Abstract

In this paper, we introduce **DoTA-RAG** (*Dynamic-of-Thought Aggregation RAG*), a Retrieval-Augmented Generation system optimized for high throughput, large-scale web knowledge indexes. Traditional RAG pipelines often suffer from high latency and limited accuracy over massive, diverse datasets. DoTA-RAG addresses these challenges with a three-stage pipeline: query rewriting, dynamic routing to specialized sub-indexes, and multi-stage retrieval and ranking. We further enhance retrieval by evaluating and selecting a superior embedding model, re-embedding the large FineWeb-10BT corpus. Moreover, we create a diverse Q&A dataset of 500 questions generated via the DataMorgana setup across a broad range of WebOrganizer topics and formats. DoTA-RAG improves the answer correctness score from 0.752 (baseline, using LiveRAG pre-built vector store) to 1.478 while maintaining low latency and achieves a 0.929 correctness score on the Live Challenge Day. These results highlight DoTA-RAG’s potential for practical deployment in domains requiring fast, reliable access to large and evolving knowledge sources.

Keywords

Retrieval-Augmented Generation; Dynamic routing; Hybrid retrieval; RAG benchmark synthesis

ACM Reference Format:

Saksorn Ruangtanusak, Natthapath Rungseesiripak, Peerawat Rojratchadakorn, Monthol Charattrakool, and Natapong Nitarach. 2025. DoTA-RAG: Dynamic of Thought Aggregation RAG. In *Proceedings of SIGIR2025 LiveRAG Challenge*. ACM, Padua, Italy, 7 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION AND RELATED WORK

Large language models (LLMs) have achieved strong results across NLP tasks, yet they often struggle with up-to-date or domain-specific knowledge, leading to hallucinations [7]. Retrieval-Augmented

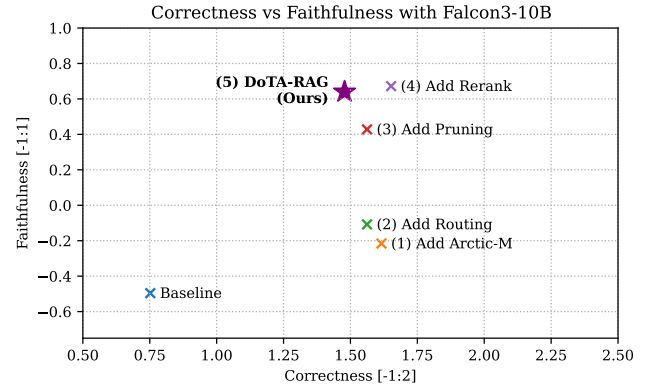


Figure 1: RAG Correctness and Faithfulness in the internal test set for different retrieval-augmented generation approaches. We use the Falcon3-10B-Instruct as the base LLM

Generation (RAG) mitigates this by allowing LLMs to access external documents: a retriever fetches relevant text, which the LLM then conditions on when generating answers [9]. Since its introduction by Lewis et al., RAG has become central to knowledge-intensive applications.

To improve retrieval robustness, *query rewriting* has emerged as a key technique [7], reformulating ambiguous or underspecified queries via prompt-engineered LLMs. For instance, HyDE prompts an LLM to generate a hypothetical answer document, enhancing retrieval by focusing on answer-answer similarity [5].

In the LiveRAG competition, we leverage the FineWeb-10BT corpus [12]. To handle its diverse sources, *routing techniques* are employed to select optimal retrieval paths based on query semantics or metadata [10, 11]. *WebOrganizer* [16] enhances this by categorizing documents by topic and format, enabling precise sub-corpus selection.

To reduce retrieved document count, we use *rerankers*—cross-encoder models that reorder results based on deeper query-document interactions. Inspired by ColBERT [8], production systems like Cohere’s Rerank 3.5 [2] improve relevance and filter noise.

Benchmarking RAG systems remains difficult due to the lack of realistic, diverse test sets. *DataMorgana* [4] addresses this with configurable tools for generating synthetic Q&A datasets that reflect varied user intents.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR2025 LiveRAG Challenge, TBD

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 9781450312345

<https://doi.org/XXXXXXX.XXXXXXX>

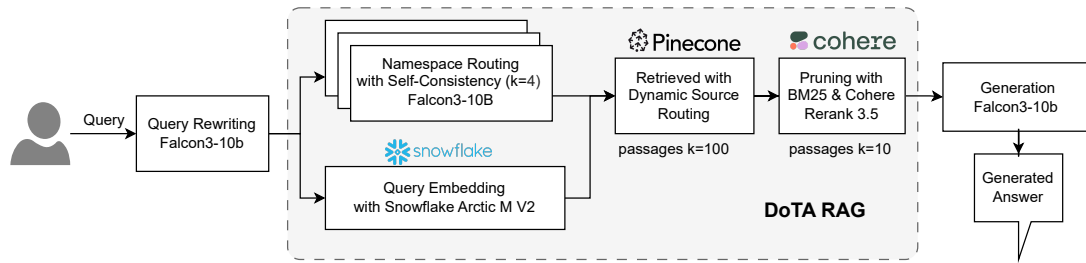


Figure 2: Diagram illustrating the components and workflow of DoTA-RAG.

The SIGIR 2025 LiveRAG Challenge mandates a fixed 15M-document corpus (FineWeb-10BT) and a standardized LLM (Falcon-3-10B-Instruct [13]), posing significant challenges for scalable and accurate retrieval.

We propose DoTA-RAG (Dynamic of Thought Aggregation RAG) to address these issues. It integrates *dynamic routing* and *hybrid retrieval* to enhance retrieval efficiency and accuracy at scale.

In summary, our pipeline is built around two key components:

Dynamic Routing: A framework that optimizes the retrieval path for large-scale corpora, reducing latency and enabling aggregation from diverse sources.

Hybrid Retrieval: A multi-stage strategy combining dense expansion, sparse filtering, and reranking for high-relevance document selection.

2 DoTA-RAG

This section presents the production **DoTA-RAG** stack deployed for the SIGIR 2025 LiveRAG Challenge. We provide a detailed breakdown of each component in the end-to-end RAG pipeline as shown in Figure 2, highlighting the enhancements implemented to raise performance and reduce inference latency. In particular, we demonstrate how the system efficiently retrieves relevant passages from a knowledge base containing 15 million documents, consistently delivering responses in under one minute per query.

2.1 Query Rewriting (Stage 1)

Low-Temp Rewrite Query. Initially, we excluded a query rewriting stage, since methods like HyDE [6] and Chain-of-Draft [17], along with crafted prompts, were heavily tested but decreased Correctness and Faithfulness. They frequently did not generalize well or address internal retrieval challenges.

On Live Challenge Day, we discovered new failure cases involving user queries with either highly specialized terms or significant misspellings. This led to few or irrelevant results (e.g., "wut iz ra-jun cajun crawfsh festivr", "wut r sum side affects of nicotine gum"), prompting us to re-evaluate query rewriting to tackle these issues. Appendix A details the query rewriting prompt.

2.2 Dynamic Namespace Routing (Stage 2)

Ensemble-Based Query Routing. To route queries efficiently across topical domains, we maintain a separate Pinecone namespace for each domain (see Appendix C). For every incoming question, we generate four independent classifications using Falcon3-10B-Instruct with self-consistency voting [15]. Each returns one or more namespace predictions, and we tally which namespaces appear most often.

We then query the top two namespaces in parallel, shrinking the average search space by 92% and reducing dense-retrieval latency from 100.84 s to 19.01 s per question. The prompt used for namespace routing can be found in Appendix A.

2.3 Hybrid Retrieval (Stage 3)

- (1) **Dense search.** Embed the refined query with Snowflake Arctic-embed-m-v2.0 [18] and fetch $k=100$ candidates using cosine similarity.
- (2) **BM25 pruning.** Compute lexical scores on-the-fly; retain the top 20 passages.
- (3) **Cohere's Rerank 3.5** Cross-encode and select the 10 highest-scoring passages.

2.4 Context Aggregation (Stage 4)

Concatenate the text of the top-10 passages, separated by blank lines. If the concatenation exceeds 8k tokens, truncate passages proportionally.

2.5 Answer Generation (Stage 5)

For answer generation, we use the prompt shown in Appendix A, which is prepended with the aggregated context and the rewritten input query. The answer is then generated by Falcon3-10B-Instruct [13].

Section Summary. DoTA-RAG is a RAG pipeline developed for the SIGIR 2025 LiveRAG Challenge, designed to deliver high-quality answers with significantly reduced inference latency over a 15M-document knowledge base. The system features five key stages: (1) Query Rewriting, which enhances retrieval for noisy or misspelled queries; (2) Dynamic Namespace Routing, using an ensemble classifier to dynamically select relevant sub-indexes and shrink the search space; (3) Hybrid Retrieval, combining dense search, BM25 pruning, and cross-encoder re-ranking; (4) Context Aggregation, which compiles and trims top-ranked passages; and (5) Answer Generation, where Falcon3-10B-Instruct [13] produces answers grounded in the retrieved context.

3 EXPERIMENTAL SETUP AND EVALUATION

This section details how we built our evaluation benchmark and which systems we compare against.

3.1 Internal test set Construction

To create our internal test set, we utilized the **DataMorgana API** [4] to generate an initial collection of 1,000 Q&A pairs. To ensure question diversity, we adopted DataMorgana's categorization framework (user types, phrasing styles, premises, and linguistic

variations) and enhanced it with our own question-formulation taxonomy (see Table 1). This new categorization introduces carefully defined question types with varying complexity levels, including questions requiring either single or multiple document sources for resolution.

For example, “Temporal-evolution” questions require the retrieval system to distinguish between documents discussing the same entity across different time periods, while challenging the generation module to synthesize information about chronological changes from these temporally distinct sources. While “Verification” questions test the system’s ability to evaluate mixed assertions containing both facts and falsehoods. This category directly challenges the generation module to distinguish truth from misinformation, correct user misconceptions, and maintain factual integrity even when prompted with partially incorrect premises. Appendix B provides descriptions and examples of the newly defined categories.

Table 1: The question-formulation categorization.

Category	Required Document(s)
Multi-aspect	Two documents
Comparison	Two documents
Temporal-evolution	Two documents
Problem-solution	Two documents
Procedural	Single document
Causal	Single document
Quantitative	Single document
Verification	Single document

Each Q&A pair’s document was auto-tagged using **WebOrganizer**’s TopicClassifier (24 topics) and FormatClassifier (24 formats) [16]. This tagging process was instrumental in creating **Morgana-MultiDocQA**—our 500-question benchmark. The classification ensured diverse document topics and formats throughout the test set, which is critical for robust evaluation of retrieval-augmented models. This diversity prevents benchmark bias toward specific domains or document structures, thereby offering a more comprehensive assessment of model generalization capabilities across the heterogeneous content types found on the internet. We then applied stratified sampling to maintain proportional representation across all topic-format combinations, preserving the natural distribution patterns found in broader information ecosystems:

$$n_c = \left\lceil \frac{N_c}{\sum_{c' \in C} N_{c'}} \times 500 \right\rceil, \quad (1)$$

where N_c is the number of candidates in category c and n_c the final sample size. This guarantees balanced coverage across the $|C| = 24 \times 24$ strata.

3.2 Embedding Models (MTEB Leaderboard)

To select a suitable dense retriever for our pipeline, we considered embedding model performance using the MTEB English retrieval tasks leaderboard [3]. Among models with fewer than 1 billion parameters, Snowflake’s Arctic v2.0 embedding models demonstrated state-of-the-art results [18], achieving mean scores of 58.56 (large) and 58.41 (medium), significantly outperforming our initial model, E5-base-v2 [14], which scored 49.67. Based on these findings, we re-embedded and re-indexed the FineWeb-10BT corpus [12] using

Arctic-embed-m-v2.0. On our internal test set, this change led to an improvement in retrieval quality, with Recall@10 increasing from 0.469 to 0.518.

3.3 Evaluation Metrics

We assess system performance using two main metrics: *Correctness* and *Faithfulness*, each evaluated on a continuous scale.

- **Correctness** (-1 to 2): Assesses the relevance and coverage of the generated answer. A score of -1 indicates an incorrect answer, while 2 represents a fully correct and relevant response with no extraneous information.
- **Faithfulness** (-1 to 1): Measures whether the answer is grounded in the retrieved passages. A score of -1 indicates no grounding at all, whereas 1 means the entire answer is fully supported by retrieved content.

3.4 LLM-as-a-Judge Evaluation

We graded answers for correctness and faithfulness using two automatic judges. While Claude 3.5 Sonnet [1] served as a strong judge, we found Falcon3-10B-Instruct [13] to be a faster and cheaper alternative with comparable evaluation quality. During our experiments for the competition, we frequently used Falcon3-10B-Instruct to lower evaluation costs and accelerate the process. On the Morgana-MultiDocQA test set, Falcon3-10B-Instruct yielded slightly higher scores on both metrics (see Table 2).

Table 2: Judge agreement on the test set.

Metric	Claude 3.5 Sonnet [1]	Falcon3-10B Instruct [13]
Correctness [-1:2]	1.382	1.430
Faithfulness [-1:1]	0.520	0.580

4 RESULTS AND ANALYSIS

This section presents our RAG pipeline evaluation across internal and live benchmarks. We report the incremental effects of each ablation step on correctness and faithfulness, and analyze final performance on the LiveRAG Live Challenge Day leaderboard.

4.1 Internal test set Result

Our ablation path incrementally augments the baseline pipeline (E5-base-v2 embeddings [14] + Falcon3-10B-Instruct [13] generation) with various enhancements to isolate their individual and combined effects on performance:

- **BASELINE** Intfloat’s e5-base-v2 → 10 retrieved passages → Falcon3-10B-Instruct generation.
- **+ARCTIC-M** Swap in Snowflake’s arctic-embed-m-v2.0 [18].
- **+ROUTING** Dynamic routing for namespace selection.
- **+PRUNING** Retrieve 100 passages, then reduce to 10 passages using BM25.
- **+RERANK** Prune to 20 passages, then use Cohere’s Rerank 3.5 [2] to select the top 10.
- **+REWRITE** Rewrite query to make the RAG pipeline more robust to the live test set – this is **DoTA-RAG**.

Table 3: Pipeline’s performance and inference time on internal test set using Claude 3.5 Sonnet [1]. Metrics are shown for full text and truncated inputs.

Method	Correctness [-1:2]		Faithfulness [-1:1]		Sec/Question
	All words	300-word cap	All words	300-word cap	
Baseline	0.752	0.761	-0.496	-0.493	–
+ Arctic-M	1.616	1.626	-0.216	-0.225	100.84
+ Routing	1.562	1.577	-0.108	-0.090	19.01
+ Pruning	1.562	1.566	0.428	0.404	29.84
+ Rerank	1.652	1.686	0.672	0.662	35.20
+ Rewrite	1.478	1.484	0.640	0.620	35.63

The results in Figure 1 and Table 3 illustrate the incremental improvements from each ablation step on the internal test¹ set. Replacing the baseline embeddings with `arctic-embed-m-v2.0` [18] (+ARCTIC-M) yields a substantial increase in correctness, though faithfulness remains negative. The addition of dynamic routing (+ROUTING) and BM25-based pruning (+PRUNING) maintains high correctness and leads to a notable gain in faithfulness. Incorporating Cohere’s Rerank 3.5 [2] (+RERANK) further improves both correctness, representing the strongest overall performance. Although adding rewriting decreases performance, we believe including it as part of the DoTA-RAG would better align our model with the test set in the LiveRAG Live Challenge Day, as discussed in Section 2.1.

4.2 LiveRAG Live Challenge Day Performance

On the official Live Challenge Day leaderboard, our final pipeline achieved a **correctness score of 0.929**, confirming that our retrieval-generation loop produced high-quality answers. However, because the evaluation enforced a strict *300-word output cap*—a constraint we overlooked during tuning, our faithfulness score sank to just **0.043**.

After the score was announced, it was surprising to see our faithfulness score so low. To understand what went wrong, we decided to conduct a deeper investigation. We re-ran the evaluation using Claude 3.5 Sonnet as the judge [1], with a prompt tailored to the faithfulness criteria. We reassessed the 500 answers, both uncut and capped at 300 words. The faithfulness score dropped significantly from 0.702 to 0.336.

5 LIMITATIONS AND FUTURE WORK

In future research endeavors, our goal is to explore strategies for multi-source routing that leverage graph-based knowledge bases, delving into the complexities of these systems. Furthermore, we are planning to investigate self-improvement methods applied after generating responses, aiming to refine this process. Another avenue of our research involves developing techniques for compacting context within windows larger than 8,000 tokens, as well as honing the procedures involved in reasoning retrieval, thereby enhancing overall performance.

¹Note that inference time (Sec/Question) is omitted for the Baseline, as it uses a pre-built index that is not directly comparable to our FineWeb-based Pinecone index.

6 CONCLUSION

We introduce **DoTA-RAG**, a live Retrieval-Augmented Generation pipeline that couples *Dynamic-of-Thought Aggregation* with dynamic routing and hybrid retrieval. By extracting on-the-fly meta-data and steering each query to the most appropriate sub-index, DoTA-RAG reconciles the seemingly contradictory requirements of *web-scale knowledge integration*, *precision*, and *low latency* that characterize the SIGIR 2025 LiveRAG Challenge.

Experiments on our 500-question MORGANAMULTIDOCQA benchmark demonstrate that successive upgrades—from changing the dense retriever to `Arctic-embed-m-v2.0` [18] to incorporating query rewriting—increase correctness from 0.752 to 1.478 and faithfulness from -0.496 to 0.640, while maintaining a median end-to-end latency of 35.63 seconds per question. On the Live Challenge Day dataset, the same configuration achieves 0.929 correctness, validating the generalization of our RAG beyond in-house data.

Our ablation studies highlight two actionable insights for the Gen-IR community:

- (1) **Metadata-guided routing delivers outsized gains.** Even lightweight routing cues significantly reduce irrelevant results and cut retrieval latency by more than half compared to static top-*k* search.
- (2) **Hybrid retrieval significantly enhances document quality.** By combining a fan-out dense retriever for broad semantic coverage and a sparse retriever for keyword-based filtering, the system improves the faithfulness score from -0.108 to 0.428 — a substantial gain in factual alignment.

Acknowledgments

We appreciate the SIGIR 2025 LiveRAG Challenge organizers for supplying the resources and tools needed for our participation and the evaluation set creation.

References

- [1] Anthropic. 2024. Claude 3.5 Sonnet Model Card Addendum. <https://paperswithcode.com/paper/claude-3-5-sonnet-model-card-addendum>. Accessed: 2025-05-18.
- [2] Cohere. 2024. Introducing Rerank 3.5: More Relevant Results with Less Compute. <https://cohere.com/blog/rerank-3pt5>. Accessed: 2025-05-18.
- [3] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrom, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri Krishnakumar, Anna Maksimova, Silvan Wehrli, Maria Tikhonova, Henil Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Miranda, Alena Fenogenova, Guangyu Song, Ruqiyi Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Hongjin Su, Jimmy Lin, Howard Yen, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. 2025. MTEB: Massive Multilingual Text Embedding Benchmark. *arXiv preprint arXiv:2502.13595* (2025). doi:10.48550/arXiv.2502.13595
- [4] Simone Filice, Guy Horowitz, David Carmel, Zohar Karnin, Liane Lewin-Eytan, and Yoelle Maarek. 2025. Generating Diverse Q&A Benchmarks for RAG Evaluation with DataMorgana. *arXiv:2501.12789* [cs.CL] <https://arxiv.org/abs/2501.12789>
- [5] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. *arXiv:2212.10496* [cs.IR] <https://arxiv.org/abs/2212.10496>
- [6] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1762–1777. doi:10.18653/v1/2023.acl-long.99
- [7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv:2312.10997* [cs.CL] <https://arxiv.org/abs/2312.10997>
- [8] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *arXiv:2004.12832* [cs.IR] <https://arxiv.org/abs/2004.12832>
- [9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv:2005.11401* [cs.CL] <https://arxiv.org/abs/2005.11401>
- [10] Xiaoqian Li, Ercong Nie, and Sheng Liang. 2023. From Classification to Generation: Insights into Crosslingual Retrieval Augmented ICL. *arXiv:2311.06595* [cs.CL] <https://arxiv.org/abs/2311.06595>
- [11] Darshil Modi. 2024. AutoMeta RAG: Enhancing Data Retrieval with Dynamic Metadata-Driven RAG Framework. Accessed: 2025-05-18.
- [12] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://openreview.net/forum?id=n6Sckn2QaG>
- [13] TII Team. 2024. The Falcon 3 family of Open Models.
- [14] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533* (2022).
- [15] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv:2203.11171* [cs.CL] <https://arxiv.org/abs/2203.11171>
- [16] Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. 2025. Organize the Web: Constructing Domains Enhances Pre-Training Data Curation. *arXiv:2502.10341* [cs.CL] <https://arxiv.org/abs/2502.10341>
- [17] Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of Draft: Thinking Faster by Writing Less. *arXiv:2502.18600* [cs.CL] <https://arxiv.org/abs/2502.18600>
- [18] Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. 2024. Arctic-Embed 2.0: Multilingual Retrieval Without Compromise. *arXiv:2412.04506* [cs.CL] <https://arxiv.org/abs/2412.04506>

A Prompt

Effective prompt engineering is critical for leveraging large language models (LLMs) in a variety of tasks, from rewriting text to classifying queries and generating informative responses. In this section, we outline carefully designed prompt templates that guide the behavior of LLM-powered assistants. These prompts are constructed to ensure clarity, precision, and relevance in model outputs, enhancing both usability and accuracy across diverse scenarios, as illustrated by the examples in Figure 3 (Query Rewriting), Figure 4 (Routing Namespace Classify), and Figure 5 (Generation Prompt).

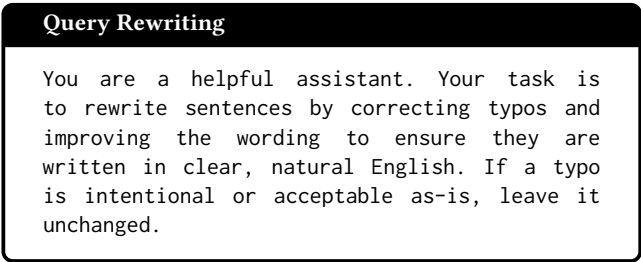


Figure 3: Query Rewriting

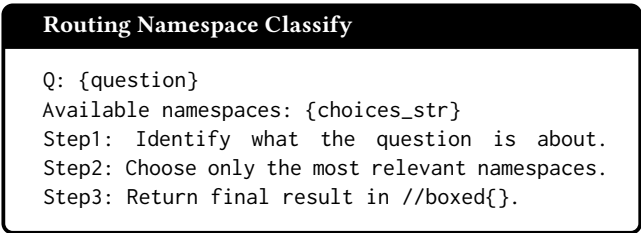


Figure 4: Routing Namespace Classify

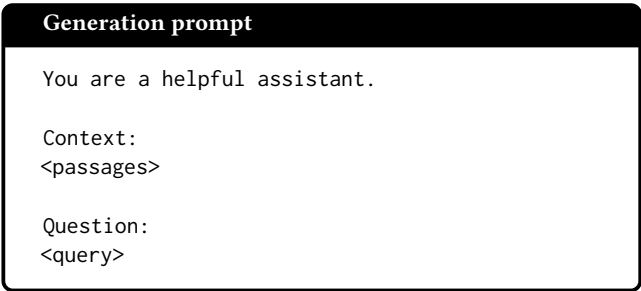


Figure 5: Generation Prompt

B Categorization and Description of Question Formulations with Examples

Multi-aspect. A question about two different aspects of the same entity/concept. For example: “What are the advantages of AI-powered diagnostics, and what are the associated risks of bias in medical decision-making?”, “How do cryptocurrencies enable financial inclusion, and what are the security risks associated with them?”.

The information required to answer the question needs to come from two documents; specifically, the first document must provide information about the first aspect, while the second must provide information about the second aspect.

Comparison. A comparison question that requires comparing two related concepts or entities. The comparison must be natural and reasonable, i.e., comparing two entities by a common attribute that is meaningful and relevant to both entities. For example: “Who is older, Glenn Hughes or Ross Lynch?”, “Are Pizhou and Jiujiang in the same province?”, “Pyotr Ilyich Tchaikovsky and Giuseppe Verdi have this profession in common”. The information required to answer the question needs to come from two documents; specifically, the first document must provide information about the first entity/concept, while the second must provide information about the second entity/concept.

Temporal-evolution. A question that explores how something has changed, progressed, or developed over time. The first document covers the earlier historical period or initial state, while the second document covers the later historical period or final state. For example: “How has smartphone technology evolved over the past two decades?”, “What changes have occurred in climate policy since the Paris Agreement?”, “How has the portrayal of women in film changed from the 1950s to today?”. The answer should describe trends, shifts, or stages of development across a timeline.

Problem-solution. Questions that ask about both a problem and potential solutions. The first document details the problem and its implications, while the second document explores possible solutions or mitigation strategies. For example: “What are the main challenges facing global food security, and what innovative agricultural technologies offer the most promising solutions?”, “What are the causes and consequences of urban air pollution, and what policies have proven effective in reducing it?”, “What factors contribute to the rise in mental health issues among teenagers, and what interventions can schools implement to support student well-being?”, “Why is plastic waste a growing environmental concern, and what strategies can be used to reduce its impact?”, “What are the limitations of current cybersecurity measures, and how can emerging technologies address them?”.

Procedural. A question that asks how to do something or requests step-by-step instructions. For example: “How do you train a neural network using TensorFlow?”, “What are the steps to apply for a student visa to the United States?”, “How can I set up a secure home Wi-Fi network?”. The answer should provide a clear, ordered list of steps or a detailed process to follow.

Causal. A question that seeks to understand why something happens or explores the relationship between cause and effect. For example: “Why does increasing carbon dioxide in the atmosphere lead to global warming?”, “What causes inflation to rise during economic booms?”, “Why do some people develop allergies while others do not?”. The answer should explain the underlying reasons or mechanisms behind the phenomenon.

Quantitative. A question that seeks numerical data, statistics, or measurements. For example: “What is the average life expectancy in Japan?”, “How many people use public transportation in New

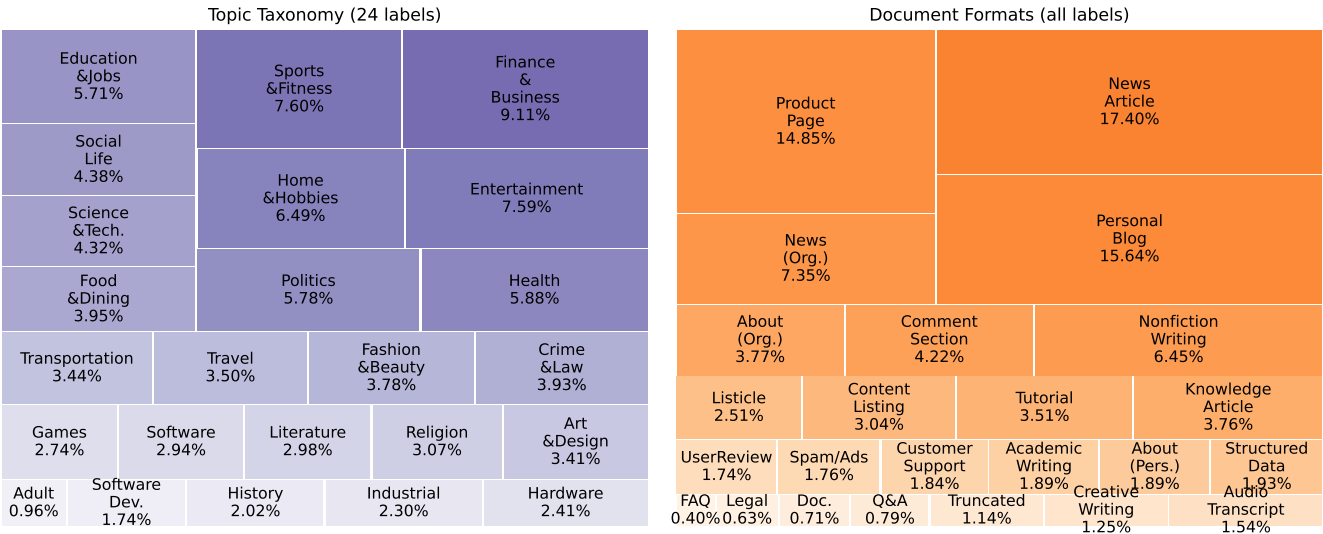


Figure 6: Distribution of topics (left) and document formats (right) in Fineweb-10BT, based on WebOrganizer classifiers. The area of each block reflects the number of documents per domain in the corpus.

York City each day?”, “What was the global GDP growth rate in 2023?”. The answer should include specific numbers, percentages, or quantitative comparisons supported by data.

Verification. A question that asks to confirm or deny the truth of a particular claim or statement. These questions involve evaluating one or more statements, where at least one is true and at least one is false. For example: “Is it true that vitamin C cures the common cold and that antibiotics are effective against viruses?”, “Did Einstein win the Nobel Prize for his theory of relativity and was he born in Austria?”. The answer should clearly indicate which statements are correct and which are incorrect, with justification.

C Fineweb-10BT Characteristic

Fineweb-10BT is a 15-million-document web corpus that we embed with *Snowflake Arctic-embed-m-v2.0* in the Pinecone vector database to use in our RAG pipeline. Using WebOrganizer’s [16] 24-label **TopicClassifier** and 24-label **FormatClassifier**, we tagged every document along two orthogonal axes—*topic* and *document format*. As the treemap in Figure 6 shows, the largest topical slices are *Finance & Business* (9.1%), *Sports & Fitness* (7.6%), and *Entertainment* (7.6%), while the dominant formats are *News Article* (17.4%), *Personal Blog* (15.6%), and *Product Page* (14.9%). Crucially, long-tail domains (*Software Dev.*, *History*, *Adult*, *FAQ*, *Legal*) remain present, giving the corpus the heterogeneity that retrieval-augmented models need for robust generalization.

Internal benchmark. We constructed a diverse 500-question benchmark by sampling across the full 24×24 topic–format grid, ensuring wide coverage of content types and styles.

Query routing. To improve retrieval speed, we map each topic to a dedicated Pinecone namespace. For each query, we use Falcon3-10B-Instruct with self-consistency voting to predict the relevant topics, querying only the top two namespaces.