

---

# HiFi-RAG: Hierarchical Content Filtering and Two-Pass Generation for Open-Domain RAG

---

Cattalyya Nuengsigkapian  
Google  
cattalyya@google.com

## Abstract

Retrieval-Augmented Generation (RAG) in open-domain settings faces significant challenges regarding irrelevant information in retrieved documents and the alignment of generated answers with user intent. We present **HiFi-RAG** (Hierarchical Filtering RAG), the winning closed-source system in the Text-to-Text static evaluation of the MMU-RAGent NeurIPS 2025 Competition. Our approach moves beyond standard embedding-based retrieval via a multi-stage pipeline. We leverage the speed and cost-efficiency of Gemini 2.5 Flash (4-6× cheaper than Pro) for query formulation, hierarchical content filtering, and citation attribution, while reserving the reasoning capabilities of Gemini 2.5 Pro for final answer generation. On the MMU-RAGent validation set, our system outperformed the baseline, improving ROUGE-L to 0.274 (+19.6%) and DeBERTaScore to 0.677 (+6.2%). On **Test2025**, our custom dataset evaluating questions that require post-cutoff knowledge (post January 2025), HiFi-RAG outperforms the parametric baseline by 57.4% in ROUGE-L and 14.9% in DeBERTaScore.

## 1 Introduction

The MMU-RAGent (Massive Multi-Modal User-Centric Retrieval Augmented Generation Benchmark) competition challenges participants to build systems that retrieve from web-scale corpora to answer diverse user queries. A common failure mode in such systems is the retrieval of irrelevant context which causes hallucination ("garbage-in, garbage-out").

Our solution, HiFi-RAG, prioritizes *precision* in the context window. We abandon standard vector-similarity search in favor of a hierarchical filtering approach. Similar to multi-stage ML model cascades that utilize low-power signals to gate high-power processing [8], we employ Gemini 2.5 Flash [2] as a lightweight gatekeeper to semantically filter hierarchically parsed web content before forwarding it to the more computationally intensive Gemini 2.5 Pro. This ensures the deep reasoning model receives only the most salient information, significantly reducing computational load.

## 2 Methodology

Our pipeline consists of five distinct stages: Query Planning, Retrieval, Hierarchical Filtering, Two-Pass Generation, and Citation Verification.

### 2.1 Query Formulation

User queries are often too verbose or conversational for effective retrieval (e.g., asking for an "ELI5" explanation). We utilize Gemini 2.5 Flash to analyze the user intent and propose optimized search queries. We explicitly instruct the model to "*Create an effective and concise Google search query*" (see full prompt in Appendix A.1). As shown in Table 1, using samples from the MMU-RAGent

validation set, this step extracts core intent and distinct search terms, improving recall for complex constraints.

Table 1: Examples of Query Formulation (Inputs from MMU-RAGent Validation Set)

User Input (Raw)	Generated Search Queries
You are a media technology professor with 20 years of experience. Explain to me like I am five, how a camera works.	[‘how a camera works explained for 5 year old’, ‘ELI5 how a camera works’]
how do i stack up 10 four-feet chairs vertically so that it remains stable?	[‘how to stack chairs safely’, ‘stable chair stacking techniques’]
Tell me the difference between “the office” and “modern family” about the actors looking at the camera	[“The Office” “Modern Family” looking at camera difference”, “The Office” “Modern Family” fourth wall comparison’]

## 2.2 Retrieval and URL Filtering

Upon receiving initial Google Search API results, we employ a pre-fetch filtering step. Instead of scraping every result, Gemini Flash analyzes the URL, title, and preview content from Search API to select only the most relevant sources (see Appendix A.2). This process reduces the URL count by 33.5% (averaging across 100 queries). By proactively discarding irrelevant domains (e.g., gaming vs. aerospace), outdated information, mismatched contexts, missing key constraints, and speculative discussions before expensive scraping, we improve both latency and context quality.

## 2.3 Hierarchical Content Parsing & Filtering

**Hierarchical Content Parsing:** We utilize the Scrapingdog API (web) and Reddit API (forums) to handle structural complexity.

- **Hierarchical Parsing:** Rather than treating content as flat text, we parse HTML into hierarchical sections (chunks). Text blocks are explicitly grouped under their parent headers (e.g., `<h1>–<h4>`) as a markdown plain-text.
- **Reddit Tree Reconstruction:** Preserves discussion flow by retrieving top- $k$  comments with top- $m$  nested replies across two layers ( $k = 5, m_1 = 3, m_2 = 2$ ).

**Section filtering and ranking (LLM-as-a-Reranker):** Instead of embedding-based filtering, we deploy Gemini 2.5 Flash to evaluate each parsed section against the user query, using only its title and a small snippet (the first 200 characters of content) to make the evaluation as lightweight as possible. It ranks sections by relevance using a custom prompt (see Appendix A.3) and discards noise. This removes 60.5% of chunks (averaged across 100 queries), resulting in a context window dense with high-quality signals.

## 2.4 Two-Pass Generation (Gemini Pro)

We utilize **Gemini 2.5 Pro** in a two-turn conversation to separate factuality from style (see Appendix A.4):

1. **Turn 1 (Drafting):** The model generates a comprehensive answer from filtered content, where each site contains title, url, preview, and filtered sections sorted by LLM relevance.
2. **Turn 2 (Refinement):** The model is prompted to revise its answer to match the style and length of three hand-picked question-answering examples from the validation set, distinct from the first 100 pairs used for evaluation (e.g., step-by-step guides for “how-to” questions).

## 2.5 Post-Hoc Citation Verification

To ensure attribution accuracy, we employ a dedicated verification step using Gemini 2.5 Flash. We decouple citation from generation to prevent performance degradation in both answers and citations

caused by long context windows. This allows the verification step to focus exclusively on source attribution and prioritizing high-quality sources when duplicates exist (see Appendix A.5) to provide source context indices that *directly support* the claims.

## 3 Experiments and Results

### 3.1 Experimental Setup

The competition provided an official validation set (with ground truth) but no training set, alongside a final blind test set without answers. Consequently, we adopted a few-shot approach and utilized the MMU-RAGent Text-to-Text validation set as our primary development benchmark. Due to cost and time constraints during the ablation phase, we evaluated on the first 100 (out of 300) samples. Evaluation was performed using ROUGE-L [6] and DeBERTaScore [3]. For DeBERTaScore, we utilized the ‘microsoft/deberta-xlarge-mnli’ model [7] to ensure robust semantic matching.

Additionally, we evaluated on a custom **Test2025** dataset (100 samples). Test2025 was synthesized to evaluate retrieval performance on events occurring after February 2025, strictly enforcing a scenario where the model must rely on retrieved context rather than parametric memory.

### 3.2 Main Results (Standard Validation)

We performed an ablation study to quantify the impact of each pipeline component. Table 2 summarizes the performance progression.

For the **Baseline Q** (Raw Query) configuration, we queried Gemini Pro with the user’s query directly, but also provide word limit to ensure fair comparison with the ground truth: “*Please limit your answer to under 200 words. [USER\_QUERY]*”.

Table 2: Ablation on MMU-RAGent Standard Validation Set

System Configuration	ROUGE-L (F1)	DeBERTaScore (F1)
Baseline Q (Raw Query + Length Constraint)	0.2291	0.6375
Baseline Prompt (No Search)	0.2591	0.6667
RAG (Search enabled)	0.2664	0.6677
RAG w/ Filters (URL + Chunk)	0.2695	0.6712
<b>Final (RAG w/ Filters + Rephrase + 2-Turn)</b>	<b>0.2739</b>	<b>0.6772</b>

The final configuration provided a 19.6% and 6.2% improvement over the baseline on ROUGE-L and DeBERTaScore respectively. Notably, prompt engineering alone (Baseline Prompt) provided a significant 13% improvement on ROUGE-L, highlighting the importance of instructional clarity. The addition of Search (RAG) and active Filtering (Filters) added another 4%, with the full two-turn refinement system achieving the highest scores across all metrics.

### 3.3 Retrieval Evaluation on Future Events (Test2025)

To strictly evaluate the system’s retrieval capabilities rather than parametric memory, we prompted Gemini 3.0 (Thinking mode with web search) to create the Test2025 dataset, containing question-answering pairs for knowledge that became available after February 2025. Note that as of November 2025, Gemini 2.5 Pro and Gemini 2.5 Flash, used in our system (as well as Gemini 3.0), all currently share a knowledge cutoff of January 2025.

On this dataset, Baseline Q performance drops significantly (0.2022). Consequently, the performance gap between the configuration with web corpora (RAG) and the one without (Baseline Q) widens dramatically from **16.3%** on the MMU-RAGent validation set to **44.16%** on Test2025, confirming the knowledge cutoff limitation.

Our RAG configurations restore performance, effectively bridging the gap to the present day and increasing ROUGE-L and DeBERTaScore by **57.4%** and **14.9%**, respectively, compared to the baseline. For the final test, we replaced the three one-shot examples (originally drawn from the validation set) with three new pairs generated by Gemini 3.0 to ensure style alignment with the test set distribution.

Table 3: Performance on Test2025 (Future Events &gt; Feb 2025)

System Configuration	ROUGE-L (F1)	DeBERTaScore (F1)
Baseline Q (Raw Query)	0.2022	0.6173
Baseline Prompt	0.2766	0.6574
RAG (Search enabled)	0.2915	0.6776
RAG w/ URL Filter Only	0.2966	0.6829
RAG w/ Filters (URL + Chunk)	0.3031	0.6840
RAG w/ Filters + Rephrase	0.2898	0.6832
<b>Final (RAG w/ Filters + Rephrase + 2-Turn)</b>	<b>0.3182</b>	<b>0.7092</b>

Furthermore, we observed that query rephrasing without URL filtering (RAG w/ Filters + Rephrase) resulted in lower scores, likely due to the removal of context from verbose queries which can introduce ambiguity. Because URL filtering mitigates these errors, query refinement proves most effective when implemented in conjunction with filtering.

### 3.4 Negative Results

We explored several alternative strategies that were discarded due to poor performance or high cost.

- **Embeddings vs. LLM Filtering:** We attempted to filter content using sentence embeddings (Voyage AI [9]). This performed worse than LLM-based filtering, likely because embeddings struggled to distinguish between topically related but factually irrelevant noise.
- **Agentic Workflows:** We implemented a full Gemini Agent with search tools. This approach was 10× more expensive and significantly slower, often timing out without improving ROUGE scores compared to our deterministic pipeline.
- **DSPy Optimization:** We used DSPy [4] with the GEPA evolutionary optimizer [1] to tune prompts. The optimizer tended to overfit to the validation set, producing brittle prompts that failed to generalize.
- **LLM-as-a-Judge Refinement:** A "Checker" module critiqued answers to catch hallucinations. While qualitatively better, it degraded automated metrics (ROUGE/DeBERTaScore), likely by altering the phrasing too aggressively away from reference text.

## 4 Conclusion

HiFi-RAG demonstrates that a structured, multi-stage pipeline using LLMs for filtering and generation outperforms traditional agentic approaches for open-domain RAG. By leveraging Gemini 2.5 Flash for high-throughput filtering and Gemini 2.5 Pro for reasoning, we achieved a balance of cost, latency, and accuracy suitable for the MMU-RAGent benchmark.

## Acknowledgments

We thank the MMU-RAGent organizers for providing the benchmark and putting this competition together.

## References

- [1] Lakshay A Agrawal, Eamon Keane, and Tianjun Zhang. GEPA: Reflective prompt evolution can outperform reinforcement learning. arXiv preprint arXiv:2507.19457, 2025.
- [2] Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261, 2025.
- [3] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In International Conference on Learning Representations, 2021.

- [4] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, et al. DSPy: Compiling declarative language model calls into self-improving pipelines. arXiv preprint arXiv:2310.03714, 2023.
- [5] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33:9459–9474, 2020.
- [6] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics, 2004.
- [7] Microsoft. Deberta-xlarge-mnli model card. <https://huggingface.co/microsoft/deberta-xlarge-mnli>, 2021.
- [8] Cattalyya Nuengsigkapian and Stephanie Renee Debats. Cascaded models for context awareness with wearable devices. U.S. Patent Application US 2025/0024131 A1, 2025.
- [9] Voyage AI. Voyage AI embeddings. <https://www.voyageai.com/>, 2024.

## A Prompts

### A.1 Query Refinement Prompt

We prompt Gemini 2.5 Flash with the following instruction to transform user queries into search-engine-friendly keywords:

```
Create an effective and concise Google search query for this question:  
[USER_QUESTION]  
Return a json list of strings for the best 1-2 search queries.
```

### A.2 URL Filtering Prompt

Before scraping, we filter the search results using Gemini 2.5 Flash with the following prompt to identify high-value targets:

```
What URLs from the list below would be helpful to read further to answer  
"[USER_QUESTION]"?
```

```
Please return a JSON list of URL strings. Here are the urls with their  
preview content:
```

```
[SEARCH_RESULT]
```

### A.3 Chunk Filtering & Ranking Prompt

The following prompt is used to select relevant sections based on their title and a preview of their content:

```
Given a webpage preview and its section titles and an opening snippet,  
help determine what sections are helpful for us to read further to  
help answer [USER_QUESTION] without having to search/research further.  
Return a JSON list of the useful section indices, sorted by most usefulness first.
```

Example output: [3, 2, 6, 7]

```
-----  
Webpage overview: [WEB_PREVIEW_CONTENT]  
-----  
Section previews in the page: [SECTION_PREVIEWS]  
-----  
Useful chunks:
```

### A.4 Two-Turn Generation Prompts

#### Turn 1 (Drafting):

You are a helpful and knowledgeable assistant.  
Answer the user question in a plain text in one paragraph (1-4 sentences).  
Include only the answer without any introductory phrases, conversational filler, or preamble.

```
User question: [USER_QUESTION]  
-----  
Here're extra information from Web Search, you might find helpful:  
[WEB_CONTENT]  
-----  
[USER_QUESTION]
```

#### Turn 2 (Refinement):

Revise your answer to have a style and length similar to the "answer" in the following examples:  
[VAL\_EXAMPLES]

### A.5 Citation Verification Prompt

We use a separate LLM call to extract citations, ensuring they strictly support the generated answer:

Read the ANSWER and identify which SOURCES (by [number]) directly support the information it contains (for citations purpose).  
Only list indices of the sources that directly support the answer.  
If no sources match, return [].  
If multiple sources support the same fact, prioritize the source that is the most specific and direct match.

Your output MUST be a single, valid JSON array of source indices.  
Example Output: [1, 4, 6]

-----  
ANSWER: [AI\_ANSWER]  
-----

SOURCES:  
[WEB\_CONTENT]

### A.6 Test2025 Dataset Generation Prompt

We prompted Gemini 3.0 Thinking (via the web interface with active Google Search access) to generate 100 question-answer pairs of knowledge that became public after their knowledge cutoff:

I want question and answer pairs in ".jsonl" format, similar to the style/length below, that focus on questions that test RAG-based LLM systems for knowledge that became public after February 2025.  
Give me 100 question-answering pairs

----- Example .jsonl -----  
[FIRST\_10\_MMURAG\_VALSET]

## B Qualitative Analysis and Examples

### B.1 Query Formulation Outputs

We provide additional examples of how Gemini 2.5 Flash rephrases user input (from the MMU-RAGent Validation Set) into search-optimized queries. This step is critical for handling conversational nuances ("I live on my own") or specific formatting requests ("step by step").

- **Input:** "I live on my own. I need to tie my shoelaces. How do I do that?"  
**Queries:** ['how to tie shoelaces', 'shoelace tying guide']
- **Input:** "i have just got a new puppy. how should i train it to walk on a leash? give me step by step instructions"  
**Queries:** ['puppy leash training step by step', 'how to train puppy to walk on leash']
- **Input:** "how to remove unwanted people from a photo in photoshop"  
**Queries:** ['photoshop remove person from photo', 'photoshop remove unwanted objects']
- **Input:** "Tell me how to dump my own bought games on switch and run them in an emulator"  
**Queries:** ['dump nintendo switch games for emulation guide', 'how to dump switch games yuzu ryujinx']

## B.2 Test2025 Analysis: The Knowledge Cutoff Gap

Test2025 highlights errors caused by the model's January 2025 knowledge cutoff. These examples show how HiFi-RAG bridges this gap, alongside instances where new knowledge is unnecessary and the baseline remains competitive.

**Case 1: RAG Success (New Events)** In these examples, the Baseline correctly identifies that it lacks the information or hallucinates a refusal, whereas HiFi-RAG retrieves the correct answer aligned with the ground truth.

- **Query:** *Who is the halftime show performer for Super Bowl LIX in 2025?*
  - **Reference:** Kendrick Lamar headlined the Apple Music Super Bowl LIX Halftime Show on February 9, 2025, at the Caesars Superdome in New Orleans, Louisiana.
  - **Baseline:** "The performer... has not yet been announced. Any names currently circulating are purely speculation." [Incorrect/Refusal]
  - **HiFi-RAG:** "Kendrick Lamar is the headliner for the Super Bowl LIX halftime show in 2025, joined by special guest SZA..." [Correct]
- **Query:** *What are the new features of the iPhone 17 Air released in September 2025?*
  - **Reference:** Apple released the iPhone 17 'Air' in September 2025... Key features include a significantly thinner aluminum chassis, a single 48MP rear camera, the A19 chip...
  - **Baseline:** "Apple has not announced or released an iPhone 17 Air... Any details currently available would be based purely on unconfirmed rumors." [Incorrect/Refusal]
  - **HiFi-RAG:** "The iPhone Air, released in September 2025, features the thinnest-ever iPhone design... and is powered by the A19 Pro chip..." [Correct]

**Case 2: RAG Correction (Dynamic Updates)** Here, the Baseline relies on outdated "future plans" known before the cutoff, while HiFi-RAG retrieves the updated reality (e.g., a delay).

- **Query:** *Did the Artemis II mission launch in 2025?*
  - **Reference:** No, the Artemis II mission did not launch in 2025. While originally targeted for late 2025, NASA delayed the mission to no earlier than April 2026...
  - **Baseline:** "The Artemis II mission has not yet launched. NASA is currently targeting a launch date of no earlier than September 2025." [Outdated Info]
  - **HiFi-RAG:** "No, the Artemis II mission did not launch in 2025... delayed to a target date of no later than April 2026." [Correct Update]

**Case 3: Baseline Competence (Long-term Planning)** Events planned years in advance (e.g., World Expos, End-of-Life cycles) are often correctly answered by the Baseline, proving that "future" questions do not always require RAG if the schedule is static.

- **Query:** *Where is the 2025 World Expo held and what is its theme?*
  - **Reference:** Expo 2025 is held in Osaka, Kansai, Japan... The official theme of the expo is 'Designing Future Society for Our Lives'.
  - **Baseline:** "The 2025 World Expo will be held in Osaka, Kansai, Japan. Its theme is 'Designing Future Society for Our Lives'." [Correct]
  - **HiFi-RAG:** "The 2025 World Expo will be held in Osaka, Japan... centered on the theme 'Designing Future Society for Our Lives'..." [Correct]
- **Query:** *When did Windows 10 officially reach its end of support?*
  - **Reference:** Microsoft officially ended support for Windows 10 Home and Pro editions on October 14, 2025.
  - **Baseline:** "Windows 10 is scheduled to officially reach its end of support on October 14, 2025." [Correct]
  - **HiFi-RAG:** "Windows 10 officially reaches its end of support on October 14, 2025..." [Correct]