RAGify@IRTM 2025: Chatting with Your Documents via Retrieval-Augmented Generation

YEO GUAN WEI*
Dept. of CSIE, NTU
Taipei, Taiwan
b11902091@csie.ntu.edu.tw

CHEN, PIN-HSIANG†
Dept. of IM, NTU
Taipei, Taiwan
b12705037@ntu.edu.tw

YU-WEN CHIANG†
Dept. of Accounting, NTU
Taipei, Taiwan
r14722040@ntu.edu.tw

LIN, LI-CHIEH†
Dept. of History, NTU
Taipei, Taiwan
b11103049@ntu.edu.tw

LUO, LI-CHEN†
Dept. of IM, NTU
Taipei, Taiwan
b11705061@ntu.edu.tw

Abstract

In the course *Introduction to Information Retrieval and Text Mining (Fall 2025)*, we undertake a final project to replicate and analyze the mechanisms behind popular document-chatting tools like NotebookLM. Our project consists of two main pillars: (1) An in-depth analysis of how different Query Rewriting and Retrieval components affect performance in a conversational search setting, and (2) The development of **RAGify**, a functional web application serving as a "NotebookLM-lite" system. Using the TREC CAsT 2022 dataset, we evaluate various configurations, including Human Rewrite, CHIQ-AD, and LLM4CS strategies combined with Binary Retrieval, TF-IDF, and BM25 models. Our experimental results demonstrate that the configuration of **BM25 + LLM4CS + `Qwen/Qwen3-4B-Thinking-2507`** achieves the best performance, reaching a MAP of 0.0759 and Recall@100 of 0.2903. This report details our methodology, experimental findings, and the system architecture of RAGify. The source code is available at the public Github Repository.

Keywords

Retrieval-Augmented Generation, Conversational Search, Query Rewriting, Information Retrieval, RAG Systems

1 Introduction

As modern college students, we increasingly rely on AI-powered assistants like NotebookLM to digest complex literature. These tools have democratized the ability to synthesize information from large document collections. However, despite their utility, current cloud-based solutions present significant limitations for academic workflows. First, relying on external servers raises privacy concerns, especially when dealing with sensitive materials such as unpublished drafts or proprietary data. Second, general-purpose models often draw on web-scale knowledge, leading to hallucinations or answers that drift away from the specific source text—a critical flaw for scholarly work requiring strict evidence grounding. Finally, the standard "chatbot" paradigm often fails to capture the iterative nature of research, which requires structured note-taking alongside conversational querying to organize thoughts and terminology.

Motivated by these real-world needs, we propose **RAGify**, a local, open-source alternative to NotebookLM designed specifically for the academic context. RAGify distinguishes itself by operating entirely offline to ensure data sovereignty, offering a notebook-oriented interface for structured inquiry, and implementing a **Document-Restricted Retrieval** mechanism to ensure all responses are strictly grounded in user-provided files.

To build such a system effectively, we must look inside the "black box" of Retrieval-Augmented Generation (RAG), specifically focusing on **Conversational Search**. Unlike traditional ad-hoc retrieval where queries are independent, conversational search requires systems to understand context, resolve ambiguities, and track evolving user intents across multiple turns. The TREC Conversational Assistance Track (CAsT) [11] has been instrumental in advancing this field, highlighting the critical role of **Query Rewriting**. Prior works like CHIQ [10] and LLM4CS [9] have demonstrated that prompt-based methods can significantly enhance query quality. Furthermore, studies like CFDA [2] have explored how different rewriting strategies impact retrieval performance.

In this paper, we extend the research of CFDA [2] by further investigating the impact of varying **Retrieval Methods** (Binary Retrieval, TF-IDF, BM25) and **LLM Model Levels** (`meta-llama/Llama-3.1-8B-Instruct` vs. `Qwen/Qwen3-4B-Thinking-2507`) on final retrieval performance.

Our contributions are twofold:

- **Component Analysis:** We rigorously analyze how different query rewriting strategies and retrieval models affect performance using the TREC CAsT 2022 benchmark.
- **System Implementation:** We build **RAGify**, a robust web application that integrates our best-performing configurations to provide a user-friendly conversational search experience.

2 Related Works

2.1 LLM4CS Framework

A widely adopted approach in Conversational Query Rewriting (CQR) is the Rewriting-And-Response (RAR) prompting method, exemplified by the LLM4CS framework [9]. LLM4CS leverages Large Language Models (LLMs) with Chain-of-Thought (CoT) prompting to generate more accurate reformulations of conversational queries. This approach handles coreference resolution and implicit intents

*Group leader of the RAGify project and responsible for the *lab-group* tasks.
†Members of the *webapp-group*, jointly responsible for building the RAGify system.

more effectively than direct LLM rewrites by explicitly incorporating reasoning signals before generating the final query.

2.2 CHIQ Framework

The CHIQ framework [10] introduces a two-step process for CQR. First, the dialogue history is enhanced through multiple LLM-based strategies (e.g., question disambiguation, response expansion, pseudo response, topic switch detection) to reduce ambiguity. Based on this refined history, three variants are proposed:

- **CHIQ-AD:** Adopts a prompt-based rewriting strategy leveraging enhanced history to reduce redundancy.
- **CHIQ-FT:** Fine-tunes a lightweight model for efficiency but suffers from limited context windows.
- **CHIQ-Fusion:** Combines the ranked lists retrieved from CHIQ-AD and CHIQ-FT using result-level fusion.

In our work, we focus on adapting **CHIQ-AD** for our comparison baselines.

2.3 Benchmark Datasets

Progress in conversational IR has been driven by benchmark datasets. The **TREC CAsT** series [3–5, 11] established standard protocols for multi-turn passage retrieval. Recently, the **TREC iKAT** track [1] advanced the field by incorporating personalization through user personas and a Personal Text Knowledge Base (PTKB).

3 Methodology

We model our system based on the TREC iKAT track [1] submission tasks and the CFDA framework [2]. For our experiments, we focus primarily on **Retrieval Performance**, excluding the response generation evaluation metric which is harder to quantify automatically, although response generation is fully implemented in our RAGify web application.

Formally, at each turn, given dialogue history H, current user utterance u, and a Personal Text Knowledge Base (PTKB) K, the pipeline is defined as:

$$f_{\text{pipeline}} : (H, u, K) \mapsto (r, D, P) \tag{1}$$

where r is the generated response, D is the set of retrieved passages, and P is the set of PTKB statements predicted as relevant. Our pipeline consists of three major stages.

3.1 Query Rewriting

The first stage reformulates the user's utterance u conditioned on dialogue history H into a self-contained query q'.

3.1.1 LLM4CS Adaptation. We reformulate queries using the logic:

$$q' = f_{\text{RAR}}(H, u, K) \tag{2}$$

where f_{RAR} is a Rewriting-And-Response prompting process with Chain-of-Thought.

Modification: Compared to the original work [9], we omit the *Pseudo Response (PR)* component in this specific configuration, as our preliminary tests suggested it often introduced noise that drifted the rewritten query away from the user's true intent.

3.1.2 CHIQ Adaptation (CHIQ-AD). We adopt CHIQ-AD using the process:

$$q' = f_{\text{rewrite}}\big(f_{\text{enhance}}(H, u), K\big) \tag{3}$$

Here, f_{enhance} employs strategies including Topic Switch (TS), Question Disambiguation (QD), and Response Expansion (RE).

Modification: We remove the *History Summary (HS)* step to preserve complete dialogue context, avoiding the loss of fine-grained details in long conversations.

3.2 Passage Retrieval

Given the rewritten query q', the system retrieves a set of passages D from the collection C. We define:

$$D = f_{\text{retrieve}}(q', C) \tag{4}$$

We implement and compare three retrieval models:

3.2.1 TF-IDF Retrieval. TF-IDF calculates the relevance of a query q to a document d based on Term Frequency (TF) and Inverse Document Frequency (IDF). The standard formula used is:

$$\text{score}(q, d) = \sum_{t \in q} tf(t, d) \cdot idf(t) \tag{5}$$

where $idf(t) = \log \frac{N}{df(t)}$, N is total documents, and $df(t)$ is the number of documents containing term t.

3.2.2 BM25 Retrieval. Okapi BM25 is a probabilistic retrieval model that improves upon TF-IDF by incorporating term saturation and document length normalization:

$$\text{score}(q, d) = \sum_{t \in q} IDF(t) \cdot \frac{tf(t, d) \cdot (k_1 + 1)}{tf(t, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{6}$$

In our experiments, we use standard parameters (e.g., $k_1 = 1.2$, $b = 0.75$).

3.2.3 Binary Retrieval. As a baseline, we implement Binary Retrieval. This is achieved by setting BM25 parameters to $k_1 = 0$ and $b = 0$. This effectively removes term frequency saturation and length normalization, treating document relevance purely based on the binary presence or absence of query terms.

3.3 Response Generation

Although not evaluated in the retrieval experiments, this component is core to our **RAGify** system. Response generation is divided into two stages: (i) **passage summarization** and (ii) **final response generation**. At each turn, the PTKB is dynamically updated by detecting whether the current user utterance u introduces new personal information not yet stored in K_{t-1}, yielding an updated K_t. A subset of relevant statements P is then selected from K_t to guide personalized response generation.

The retrieved passage list D is filtered by retaining the top-NUM_PASSAGES passages whose relevance scores exceed the predefined SCORE_THRESHOLD. The top-NUM_DIRECT_PASSAGES passages are then preserved as direct inputs, while the remaining passages are grouped into chunks of size SUMMARY_CHUNK_SIZE and summarized by a large language model into a set of concise passages D'. Finally, these processed inputs are used to generate the final response. Formally:

$$r = f_{\text{resp}}(H, u, D', P) \tag{7}$$

4 Experiments

4.1 Experimental Setup

This section outlines the experimental framework for evaluating our conversational retrieval system. We describe the dataset, computational environment, evaluation metrics, and the specific configurations tested in our experiments.

Dataset. We use the **TREC CAsT 2022** dataset [11] because the iKAT dataset is not publicly accessible. CAsT 2022 is the most comprehensive benchmark in the CAsT series, providing a standardized evaluation protocol for multi-turn conversational search.

Environment. Candidate generation was performed via vLLM serve [7] on `meow1.csie.ntu.edu.tw`, and experiments were run on the `ws5.csie.ntu.edu.tw` workstation.

Evaluation Metrics. We report the following standard information retrieval metrics:

- **MAP (Mean Average Precision):** Measures the average precision across all queries, accounting for the ranking quality of relevant documents.
- **nDCG@10 (Normalized Discounted Cumulative Gain at 10):** Evaluates ranking quality by considering both relevance and position of documents in the top-10 results.
- **Recall@100:** Measures the proportion of relevant documents retrieved within the top-100 results.
- **P@5 (Precision at 5):** Computes the proportion of relevant documents in the top-5 results.

Configurations. Overall, we compared the following combinations: (i) **Three retrieval methods:** Binary Retrieval, TF-IDF, and BM25; (ii) **Four query rewriting strategies:** Human Rewrite (gold standard), CHIQ-AD with `meta-llama/Llama-3.1-8B-Instruct` (3 runs averaged), LLM4CS with `meta-llama/Llama-3.1-8B-Instruct` (3 runs averaged), and LLM4CS with `Qwen/Qwen3-4B-Thinking-2507`.

4.2 Main Results

Table 1 presents the performance of our systems alongside baselines. We compare Human Rewrite (gold standard), CHIQ-AD (with Llama), and LLM4CS (with Llama and Qwen).

4.3 Analysis

Table 1 reveals several important patterns regarding the performance of different query rewriting strategies and retrieval models. In this section, we analyze these findings and provide insights into the underlying mechanisms that drive the observed performance differences.

4.3.1 Key Finding 1: BM25 > TF-IDF ≫ Binary Retrieval. BM25 consistently outperforms TF-IDF, which is much better than Binary Retrieval across all query rewriting methods.

Reason: Binary Retrieval ignores term frequency, treating every term occurrence equally. In contrast, BM25 applies term frequency saturation (diminishing returns for repeated terms) and document length normalization (penalizing longer documents). This makes BM25 more effective than simple TF-IDF. The probabilistic foundations of BM25 allow it to better capture document relevance in conversational settings where queries can be verbose or contain repeated terms.

4.3.2 Key Finding 2: LLM4CS Outperforms CHIQ-AD. LLM4CS slightly outperforms CHIQ-AD across all retrieval methods.

Reason: LLM4CS generates queries that work well with traditional keyword-based systems like BM25 and TF-IDF. The Chain-of-Thought prompting in LLM4CS helps the model better understand user intent and produce concise, focused queries. In contrast, CHIQ-AD's queries sometimes include extra context that can dilute keyword matching. While this additional context might be semantically useful, it introduces noise when matched against keyword-based retrieval models that rely on precise term matching.

4.3.3 Key Finding 3: "Thinking" Models Perform Better. The reasoning model `Qwen/Qwen3-4B-Thinking-2507` achieves the best answer quality across all metrics.

Reason: Reasoning models excel at advanced query rewriting because they can: (i) interpret context and infer the user's intent even when queries are ambiguous or incomplete, (ii) generate precise queries that focus on the most relevant concepts, and (iii) handle multi-step logic by combining information across conversation turns to produce coherent, retrieval-friendly queries. The explicit reasoning capability allows the model to better resolve coreferences, identify topic shifts, and maintain conversational coherence.

Based on our results, the optimal setup is **BM25 + LLM4CS + Qwen Thinking Model**, achieving a MAP of 0.0759, nDCG@10 of 0.0872, Recall@100 of 0.2903, and P@5 of 0.0253.

5 RAGify System Implementation

We implemented **RAGify**, a web application that mimics NotebookLM, using our best configuration (BM25 + LLM4CS) and the response generation pipeline described in Section 3.3. A demonstration video demonstrating the usage of the RAGify system is available here.

5.1 System Architecture

Frontend. Built with **Next.js**, utilizing Tailwind CSS for a responsive UI. It features a chat interface, document upload/management panel, and a markdown editor for notes.

Backend. Powered by **FastAPI (Python)**. It handles the core logic: parsing uploaded documents (PDF/Text), indexing them using BM25, maintaining the conversation history/PTKB, and interfacing with the LLM (Gemini/local models) for generation.

5.2 Document Processing & Chunking

To handle PDF documents effectively, we implemented a document processing pipeline that addresses common text extraction challenges. The system utilizes `pypdf` for raw text extraction and implements regex-based cleaning to merge fragmented words caused by excessive character spacing. We employ a **sliding window** algorithm with a chunk size of **500 characters** and an overlap of **100 characters** to maintain semantic continuity. Additionally, we designed a punctuation detection algorithm that searches backward (within 50 characters) for sentence boundaries to ensure chunks represent complete semantic units.

Table 1: Overall retrieval performance comparison on TREC CAsT 2022. Bold indicates best performance, underline indicates second-best within Human Rewrite baseline.

Rewrite Method	Retrieval Model	MAP	nDCG@10	Recall@100	P@5
Human Rewrite	Binary Retrieval	0.0205	0.0204	0.1799	0.0066
	TF-IDF	<u>0.0781</u>	<u>0.1029</u>	<u>0.3562</u>	<u>0.0275</u>
	BM25	**0.0827**	**0.1098**	**0.3700**	**0.0363**
CHIQ-AD[†,*]	Binary Retrieval	0.0166	0.0196	0.1507	0.0066
	TF-IDF	0.0482	0.0612	0.2577	0.0191
	BM25	0.0535	0.0657	0.2671	0.0213
LLM4CS[†,*]	Binary Retrieval	0.0144	0.0197	0.1252	0.0066
	TF-IDF	0.0558	0.0681	0.2462	0.0220
	BM25	0.0606	0.0795	0.2650	0.0227
LLM4CS[‡]	Binary Retrieval	0.0245	0.0314	0.1255	0.0099
	TF-IDF	0.0679	0.0801	0.2875	0.0220
	BM25	0.0759	0.0872	0.2903	0.0253

[†] Based on `meta-llama/Llama-3.1-8B-Instruct`; [‡] Based on `Qwen/Qwen3-4B-Thinking-2507`; [*] Results averaged over 3 runs

Table 2: Performance comparison on the TREC iKAT 2025 Offline Generation-Only Track. Rows are sorted by Nugget Recall, serving as the primary ranking metric. Bold values indicate the best performance.

Group	Run ID	LLMeval		Nugget Recall	BEM	F1	ROUGE-1
		SOLAR	GPT-4.1				
cfda	gen-only_npsg13_thru0_d4c5	**0.9333**	0.8000	**0.1195**	0.1641	**0.3136**	**0.2689**
uva	nuggets-noptkb	0.9111	**0.8222**	0.1070	0.1721	0.3026	0.2537
guidance	genonly_claritop10	0.7778	0.6889	0.1041	0.1650	0.2799	0.2306
uva	nuggets-ptkb	0.8667	0.7778	0.1030	**0.1923**	0.3052	0.2524
genaius	genaius-genonly-full-gpt4o	0.8889	0.7556	0.0999	0.1672	0.2827	0.2485
cfda	gen-only_npsg20_thru03_d3c5	0.9111	0.7778	0.0978	0.1524	0.3065	0.2552
genaius	genaius-genonly-summary-gpt4o	0.8667	0.7556	0.0811	0.1407	0.2750	0.2500
usiir	usiir_run2	0.4000	0.2222	0.0510	0.1267	0.1877	0.1708
usiir	usiir_run1	0.6000	0.3111	0.0508	0.1186	0.1916	0.1740

5.3 Information Retrieval System

Our IR system converts processed documents into a Lucene index for BM25 retrieval. To handle multilingual documents, the backend implements automatic language detection and dynamically switches Lucene analyzers based on the query language (e.g., Chinese `zh`, English `en`). To support the "only reference user-selected files" functionality, we implemented `doc_id` filtering at the retrieval level, excluding document chunks from unselected files.

5.4 Generation Configuration

For the response generation module, we adopted the specific parameter settings that won **1st place** in the TREC iKAT 2025 Offline Generation-Only Track (achieved by the CFDA team [2]):

- NUM_PASSAGES = 13
- SCORE_THRESHOLD = 0
- NUM_DIRECT_PASSAGES = 4
- SUMMARY_CHUNK_SIZE = 5

As shown in Table 2, this configuration demonstrates superior performance across key metrics, ensuring the generation quality of our RAGify system. The system also supports full Retrieval-Augmented Generation, citing sources from uploaded documents directly in the chat. Additionally, we implemented an exponential backoff retry mechanism for the Gemini API to handle quota limitations (429 errors).

6 Conclusion and Limitations

In this project, we successfully demystified the "black box" of tools like NotebookLM. We analyzed the critical components of a Conversational RAG system and built **RAGify**, a functional implementation. Our experiments on TREC CAsT 2022 revealed that **LLM4CS with a "Thinking" model (Qwen)** combined with **BM25** offers a strong balance of performance and efficiency.

6.1 Limitations

Despite the promising results, our work has several limitations:

(1) **Retrieval Methods:** Due to limited computational resources, we could not evaluate dense retrieval methods like **SPLADE** [6],

which might offer better semantic matching than BM25. Future work should explore the integration of dense and sparse retrieval methods through hybrid approaches.

(2) **Rewriting Diversity:** We focused on prompt-based rewriting. Future work could compare fine-tuning approaches like AdaRewriter [8] or other generative variations to understand the trade-offs between prompt-based and fine-tuned models.

(3) **Dataset Scope:** We evaluated only on TREC CAsT 2022 due to the unavailability of the newer iKAT dataset. Evaluation on more diverse conversational search benchmarks would strengthen our findings.

References

[1] Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffery Dalton, and Leif Azzopardi. 2024. Trec ikat 2023: The interactive knowledge assistance track overview. *arXiv preprint arXiv:2401.01330* (2024).

[2] Yu-Cheng Chang, Guan-Wei Yeo, Quah Eugene, Fan-Jie Shih, Yuan-Ching Kuo, Tsung-En Yu, Hung-Chun Hsu, Ming-Feng Tsai, and Chuan-Ju Wang. 2025. CFDA & CLIP at TREC iKAT 2025: Enhancing Personalized Conversational Search via Query Reformulation and Rank Fusion. *arXiv preprint arXiv:2509.15588* (2025).

[3] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The conversational assistance track overview. *arXiv preprint arXiv:2003.13624* (2020).

[4] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. Cast 2020: The conversational assistance track overview. In *In Proceedings of TREC*.

[5] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. TREC CAsT 2021: The Conversational Assistance Track Overview. In *Text Retrieval Conference*. https://api.semanticscholar.org/CorpusID:261241621

[6] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2292.

[7] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention.

[8] Yilong Lai, Jialong Wu, Zhenglin Wang, and Deyu Zhou. 2025. AdaRewriter: Unleashing the Power of Prompting-based Conversational Query Reformulation via Test-Time Adaptation.

[9] Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large Language Models Know Your Contextual Search Intent: A Prompting Framework for Conversational Search. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 1211–1225.

[10] Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. CHIQ: Contextual History Enhancement for Improving Query Rewriting in Conversational Search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2253–2268.

[11] Paul Owoicho, Jeff Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R Trippas, and Svitlana Vakulenko. 2022. TREC CAsT 2022: Going Beyond User Ask and System Retrieve with Initiative and Response Generation.. In *TREC*.