IISERK@ToT_2024: Query Reformulation and Layered Retrieval for Tip-of-Tongue Items

Subinay Adhikary*
sa21rs094@iiserkol.ac.in
Indian Institute of Science Education and Research
Kolkata, West Bengal, India

Shuvam Banerji Seal*
sbs22ms076@iiserkol.ac.in
Indian Institute of Science Education and Research
Kolkata, West Bengal, India

Soumyadeep Sar
ss19ms150@iiserkol.ac.in
Indian Institute of Science Education and Research
Kolkata, West Bengal, India

Dwaipayan Roy
dwaipayan.roy@iiserkol.ac.in
Indian Institute of Science Education and Research
Kolkata, West Bengal, India

Abstract

In this study, we explore various approaches for known-item retrieval, referred to as "Tip-of-the-Tongue" (ToT). The TREC 2024 ToT track involves retrieving previously encountered items, such as movie names or landmarks when the searcher struggles to recall their exact identifiers. In this paper, we (**ThinkIR**) focus on four different approaches to retrieve the correct item for each query, including BM25 with optimized parameters and leveraging Large Language Models (LLMs) to reformulate the queries. Subsequently, we utilize these reformulated queries during retrieval using the BM25 model for each method. The *four-step* query reformulation technique, combined with *two-layer* retrieval, has enhanced retrieval performance in terms of NDCG and Recall. Eventually, *two-layer* retrieval achieves the best performance among all the runs, with a Recall@1000 of **0.8067**.

Keywords

Known item search, Tip-of-Tongue search, Query reformulation, Large Language Model, BM25 retrieval

ACM Reference Format:
Subinay Adhikary, Shuvam Banerji Seal, Soumyadeep Sar, and Dwaipayan Roy. 2024. IISERK@ToT_2024: Query Reformulation and Layered Retrieval for Tip-of-Tongue Items. In *Proceedings of (TREC 2024).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The phenomenon of recalling an item whose characteristics are vaguely known but the user fails to remember the exact name of the item is known as Tip-of-tongue (ToT). This is a very prevalent issue, which resulted in multiple websites and forums on Reddit, trying to help users get their desired answer. *I Remember The Movies*[1] is an example of such a platform where users suffering from ToT, can share their experience and possibly get the movie name that is sought. Users facing ToT experiences frequently turn to such forums rather than attempting traditional searches, highlighting a gap in Information Retrieval (IR) methods designed for incomplete or imprecise information. Recently, Arguello et al. [2] highlighted this gap in research revealing that the current retrieval systems are inadequate for handling such queries. This led to creation of a new track at TREC, aimed at encouraging further investigation into effective IR techniques to deal with ToT queries. One of the primary challenges with these queries, is their minimal term overlap with the relevant documents (Wikipedia pages for our case). As a result, traditional retrieval techniques like BM25, which heavily relies on the bag-of-word approach, tend to underperform on such queries.

The TREC 2023 Tip of the Tongue (ToT) track [1] addressed the challenge of matching movies to naturally ambiguous or verbose queries for the first time. The primary goal was to retrieve a ranked list of 1000 movie names, positioning the correct movie as high as possible. Evaluation metrics such as discounted cumulative gain (DCG), reciprocal rank (RR), and success@k were used to assess system performance. In continuation, the TREC ToT 2024 shared task expands on this by incorporating queries from multiple domains. In addition to movie-related queries, the test set now includes queries related to landscapes and personalities. This has driven the development of more robust and generalized information retrieval techniques capable of handling a broader range of user queries.

In this work, we leverage query reformulation techniques to enhance ToT queries. Our approach is an extremely light-weight method involving BM25 as our only retriever working on reformulated queries. For reformulation, we employ a multi-layered *prompting* framework, which finally gives a compact and precise query shaped as a passage from Wikipedia article. Additionally, we also apply filtering layers to screen documents from corpus based various heuristics which are discussed in the subsequent sections. Our final result show that such techniques actually work very well in finding answers, especially for personalities and landscapes based queries.

*Both authors contributed equally to this research.

TREC 2024, November 18-22, 2024, NCCoE, Rockville, Maryland

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

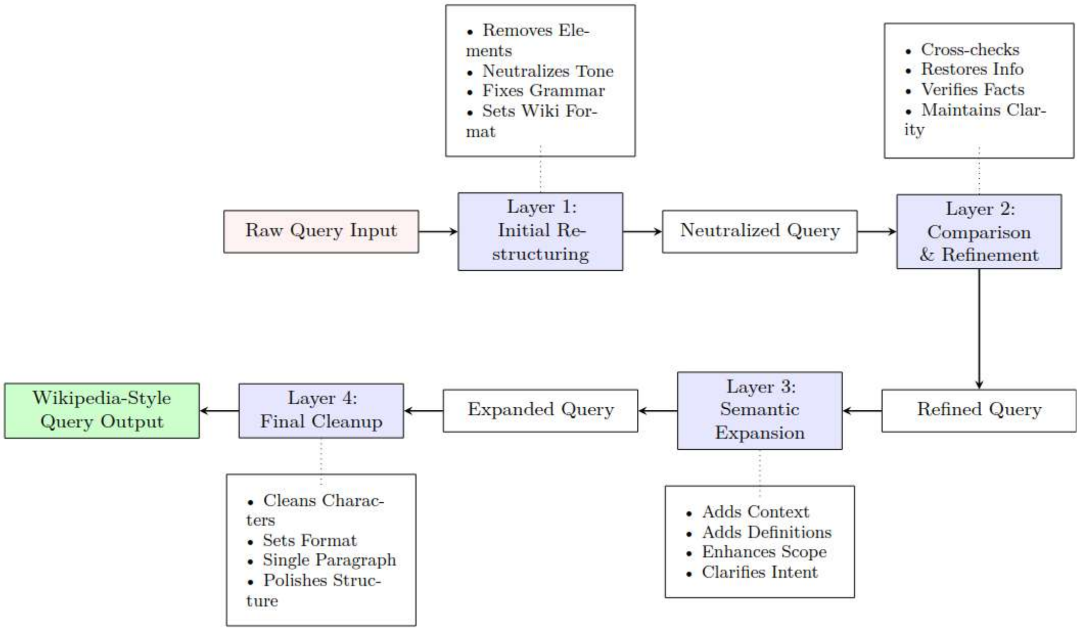

Figure 1: We follow a four-step query formulation strategy, as described in Section 4.2 . Later, this reformed query is utilized during retrieval.

2 Related Work

The ToT phenomenon sits at the intersection of natural language understanding (NLU) and information retrieval (IR), leading to a significant body of research aimed at addressing the verbose and complex nature of these queries in the field of NLP. ToT belongs to the class of studies based on known-item retrieval [11]. This prevalent field has been studied from many diverse perspectives, ranging from multi-modal [9] to mutli-domain [10] Among other approaches, Lin et al. [8] introduces a novel framework for decomposing complex ToT queries into smaller sub-queries. Each subquery focuses on specific clues about the item in question. These specialized queries are then fed into retrievers that are trained on the respective domains for each sub-query. The results from each sub-query are ensembled to produce the final ranked list of items. This approach led to an 8% improvement in *Recall@5*, offering a promising method for tackling such queries.

Research on ToT queries is still in its developmental stages, and with the creation of new datasets [6, 13], further insights can be gained, leading to a deeper understanding of these types of queries. In the 2023 edition of TREC ToT, Borges et al. [3] produced the best-performing run using a dense passage retriever (DPRs) fine-tuned on a curated dataset of ToT queries and relevant documents for semantic search. Their method, known as the *max-P* technique, first scores each small paragraph in a document based on the query. The maximum similarity score across all paragraphs is then assigned to that document. This is followed by re-ranking using GPT-4 in a round-robin fashion. Another facinating study by Fröbe et al. [5] demonstrates how query reduction using large language models (LLMs) can help mitigate the confusing nature of ToT queries. They used ChatGPT and DeepCT to effectively reduce long, verbose queries through various prompting techniques. For retrieval, they used BM25 and Mono-T5 on these simplified queries to generate the final ranked list. This study shows how effective query reformulation can be, resulting in a performance enhancement even with traditional retrievers, like BM25, which are proven to be ineffective against such queries.

3 Methodology

Our approach is centered around handling queries as efficiently as possible. The rationale behind this is straightforward: the given a corpus consists of documents in the scale of millions, which is already quite large. In the real world, however, there are billions of articles scattered across the internet. Any form of pre-processing applied to the main corpus would essentially mean pre-processing the entire internet, should our algorithm be applied at scale. This highlights the importance of designing solutions that are scalable and can efficiently handle vast amounts of data without relying on exhaustive pre-processing.

Considering that rationale, our query processing pipeline is designed to be simple and efficient, consisting of the following two main phases in the pre-processing step:

(1) Query Reformulation (QR),
(2) Focused Document Retrieval (FDR)

After the query passes through the aforementioned steps, a BM25 similarity [12] calculation with the corpus is performed to retrieve

[1] https://irememberthismovie.com/

the final results. The BM25 retrieval is done using optimized parameters for $k1$ and b. We discuss, each of the two steps in the following sections.

3.1 Query Reformulation (QR)

Query Reformulation (QR) is the process of restructuring a given query to improve its clarity and relevance [4]. The goal of this restructuring is to transform an initial query – often vague or lacking specificity – into a more precise and meaningful form.

> So, there's this movie I watched ages ago, and it's been bugging me because I can't remember the name. It was about this group of kids who spent their summer playing baseball in this dusty, rundown field. I think it was set in the '60s or something, and the main kid had just moved to a new neighborhood with his mom and stepdad. He was kind of a misfit and didn't know much about baseball, but he really wanted to fit in with these local boys who played every day. One of the boys, who was like the star player, took the new kid under his wing and taught him how to play. They had all these little adventures, and there was this one hilarious scene where they tried to retrieve a ball from this yard guarded by a monstrous dog. The dog was like a legend among the kids, and they had all these wild stories about it. The new kid accidentally hit a ball signed by some famous player into the dog's yard, and they spent a good chunk of the movie trying to get it back. There was also this part where the star player had a dream or something, and it inspired him to do something really brave. The movie had a lot of heart and was super nostalgic. I remember watching it with my cousins during a summer break, and we laughed so much. The ending was pretty touching too, with the kids growing up and going their separate ways, but the main kid and the star player stayed friends...

Aforementioned query can be termed as vague in the sense that there are hardly any detectable keyword terms for the movie like the genre, actors, location of the filming set, etc. Also the information that has been provided is mixed with the author's emotions while watching the said movie. Henceforth, we need to extract out the information that only related to the movie, while keeping all the crucial information about the main leads or scenes in the movie which could be used for the identification. This process of identification of the searchable information whilst removing linguistic vagueness can be done using any standard Large Language Model (LLM). The objective of QR, therefore, is to systematically refine queries to improve search accuracy and relevance, as described in Section 4.

3.2 Focused Document Retrieval (FDR)

Once we have reformulated queries, we focus on retrieving the correct answer for each query.

In the initial step of retrieval, we retrieve the $top - m$ documents for each query using the BM25 model. Subsequently, compute the word count of each retrieved document and determine the average word count. Documents with a word count below this average are filtered out, since short Wiki documents have less information. Eventually, we re-rank the remaining documents and select top-n (n«m) documents as the final list. This process is schematically illustrated in Figure 2, and we call this the retrieval approach as **1-Layer Retrieval or LR1** for short.

Search Domain Contraction (SDC). [7] reported that to improve the performance of the retrieval model, filtering out the irrelevant items is essential. That motivates us to reduce the search space during retrieval. Here, SD represents the full searchable domain, which contains all available documents for each query q ∈ Q. The goal of SDC is to iteratively construct a sequence of subsets $SD \supset SD_1 \supset SD_2 \supset \cdots \supset SD_n$, where SD_k at each stage k contains only documents increasingly relevant to q. The desired outcome of SDC is to reach an iteration SD_n that optimizes both the relevance and the size of the domain, effectively balancing domain contraction with the retention of highly relevant documents.

Optimization Problem Formulation. We can frame SDC as an optimization problem. Let $f : SD \longrightarrow \mathbb{R}$ represent a relevance measure that quantifies the similarity between each document and the query q. Our objective is to minimize the domain size $|SD_n|$ while retaining only documents in SD_n that exceed a relevance threshold α. Mathematically, we can express this as:

$$\min_{SD_n \subset SD} |SD_n| \quad \text{subject to} \quad f(d,q) \geq \alpha \quad \forall d \in SD_n$$

where α is chosen to ensure only highly relevant documents remain in SD_n.

In our specific case, the threshold α was based on the Mean Reciprocal Rank (MRR) scores obtained from BM25 retrieval. The MRR provides a measure of retrieval quality by evaluating the rank position of the first relevant document, with higher MRR scores indicating higher relevance. By setting α according to these MRR scores, we retained documents in SD_n that exhibited significant relevance as per the BM25 scoring metric.

Observed Convergence Behavior. Empirical observations with respect to the reformulated queries on the whole corpus showed that after two iterations, denoted as SD_2, the contracted domain was sufficiently optimal in terms of relevance. Further iterations beyond SD_2 did not enhance the quality of the subset and, in fact, led to diminishing relevance. This behavior indicates that SD_2 may represent a locally optimal solution in our SDC process, suggesting a stopping criterion for the sequence based on diminishing returns in relevance metrics. Also, it might have been the case that the criteria chosen for layer 2 was lead to too much contraction of the search domain that lead to relevant documents being eliminated.

With this motivation, we leverage LLMs to extract categories from each query q. Subsequently, we utilize those categories to retrieve top-k documents, which leads to contracting the search domain. Again, we utilize a reformulated query to retrieve top-m documents. Finally, we follow the aforementioned technique to filter out the small documents and select the top-n documents. In Figure 2, we demonstrate this method, and we termed it as **2-Layer Retrieval or LR2** for short.

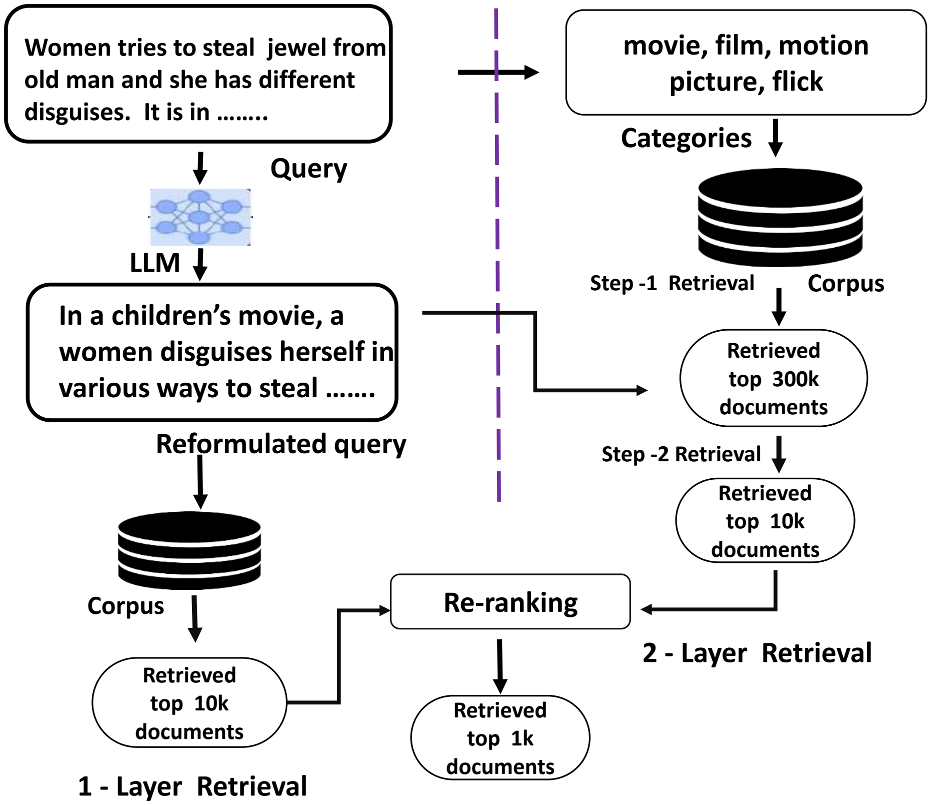
Women tries to steal jewel from old man and she has different disguises. It is in ....
Query
LLM
In a children's movie, a women disguises herself in various ways to steal ....
Reformulated query
Corpus
Retrieved top 10k documents
1 - Layer Retrieval
movie, film, motion picture, flick
Categories
Step -1 Retrieval
Corpus
Retrieved top 300k documents
Step -2 Retrieval
Retrieved top 10k documents
Re-ranking
2 - Layer Retrieval
Retrieved top 1k documents

Figure 2: We employ LLM to reshape the given query, resulting in a reformulated query. In 1-layer retrieval, we only utilize reformulated query. Conversely, 2-layer retrieval incorporates both categorical information and reformulated query in two different steps. Finally, we witness that 2-layer retrieval outperforms 1-layer retrieval.

4 Experimental Setup

4.1 Dataset

In this paper, we use a corpus of Wikipedia documents for our empirical study. The corpus contains approximately 3.1 million documents covering a wide range of topics, including movies, celebrities, popular games, and more. Additionally, the TOT organizers provide 150 training queries, all related to movies of various genres such as comedy and thriller. For the testing phase, they give a set of 600 queries, of which 302 queries are specifically to movies, 124 queries are related to landmarks, and the remaining are related to celebrities, etc.

4.2 Query Reformulation

The layer structure and ordering in our query processing workflow were designed with a specific intent: to maximize the clarity, accuracy, and depth of the query output without introducing noise or unnecessary artifacts. The sequential flow ensures each layer builds on the strengths of the previous, refining the query while maintaining fidelity to the original content. Below is a detailed explanation of the layer structure and the rationale behind the chosen order.

Query Restructuring. Each query from the input JSONL file is restructured by employing the model llama3-70b[2] to transform the text into an encyclopedic style, emphasizing clarity and factual tone, where temperature value is 0 to get the deterministic response. Finally, we have a coherent, wikipedia-style rephrased query, after the completion of **Phase 1** processing.

Accuracy and detail preservation. This layer validates the restructured query, comparing it against the original to identify and correct any lost details or inaccuracies. Finally after the finishing of **Phase 2**, a refined query that maintains almost all of the details of the original query.

Query expansion. The refined query undergoes expansion to incorporate additional context or definitions, enriching its descriptive depth. This was done to incorporate similar words into the query text, so that during BM25 word overlap the probability of matching with relevant documents which might not incorporate the exact same words from the previous layer increases as similar words are added. In **Phase 3**, a query with all the details of the main query text and similar words with the same intrinsic meaning added to it.

[2] https://huggingface.co/meta-llama/Meta-Llama-3-70B

Table 1: We observe that, *Query Formulation* affects the retrieval performance. More specifically, the Phase 4 reformulation increases the likelihood of retrieving the correct answer for a query.

Subject	Query text
Phase 1 (Rank: 40)	Woman tries to steal jewel from old man and she has different disguises. It is in color, and is a children's movie. She finally confronts him with a gun at a ski trip, and he wears a fake tattoo to hide from her. But it washes off in a jacuzzi.
Phase 2 (Rank: 15)	This children's movie features a woman who attempts to steal a jewel from an elderly man. She employs various disguises throughout the film. During a ski trip, she confronts the man with a gun. To evade her, he conceals his identity with a fake tattoo. However, the tattoo is washed off in a jacuzzi, revealing his true appearance.
Phase 3 (Rank: 145)	In a children's movie, a woman disguises herself in various ways to steal a jewel from an old man. The woman confronts the old man with a gun during a ski trip, and he uses a fake tattoo to hide from her. However, the tattoo washes off in a jacuzzi.
Phase 4 (Rank: 1)	In a children's movie, a woman disguises herself in various ways to steal a jewel from an old man. The woman confronts the old man with a gun during a ski trip, and he uses a fake tattoo to hide from her. However, the tattoo washes off in a jacuzzi.

Table 2: Final result of all test runs. Best performance per metric is highlighted in bold.

Method	NDCG@1000	NDCG@10	Recall@5	Recall@1000
Semantic Search	0.0234	0.0018	0.0033	0.1700
LR1	0.4142	0.3679	0.4200	0.7483
LR2	0.4056	0.3555	0.4117	0.7517
LRS2	**0.4239**	**0.3719**	**0.4250**	**0.8067**

Final Cleanup. In **Phase 4**, we perform a final cleanup prompt, that ensures the expanded query reads as a single, organized paragraph, free from extraneous characters or formatting issues. Output: A polished, consistent result that is ready for output with enhanced readability.

4.3 Focused Document Retrieval

In the main searching phase of our workflow, BM25 model was utilized. We ran tests to find out the most optimized parameters for the searcher in terms of k_1 and b, as shown in Figure 3. The test was brute force computing the best possible MRR score using the BM25 searcher using the original queries on the fully indexed corpus.

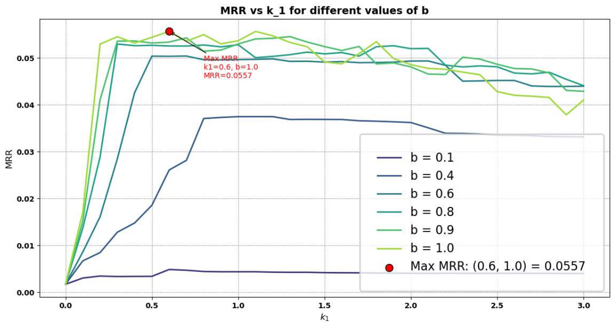

Figure 3: Utilizing the training data, we observed that the BM25 model achieved the highest MRR score with k1=0.6 and b=1.0. We then applied these k1 and b values for all subsequent retrieval tasks.

2-Layer Retreival (LR2). The initial layer performs broad-spectrum filtering based on categorical metadata. In Figure 2, we show that categorical information like movies, films, and pictures have been used to retrieve the top 300k documents from the whole corpus. Next, the final layer performs a query-specific search on the highly filtered document space. In this stage, we mainly utilize phase 4 reformulated query on previously retrieved documents. After completing this stage, we consider the top-10k documents relevant and

Table 3: Success@k metrics across different methods and categories. The best performance per metric is highlighted in bold.

Model	Category	Success@1	Success@3	Success@5	Success@10
Semantic Search	Overall	0.00%	0.00%	0.33%	0.50%
	Celebrity	0.00%	0.00%	0.67%	0.67%
	Landmark	0.00%	0.00%	0.67%	1.34%
	Movie	0.00%	0.00%	0.00%	0.00%
LR1	Overall	26.83%	36.50%	42.83%	44.50%
	Celebrity	28.86%	38.26%	45.64%	48.99%
	Landmark	41.61%	51.01%	61.07%	63.09%
	Movie	18.54%	28.48%	32.45%	33.11%
LR2	Overall	27.17%	37.33%	43.83%	48.17%
	Celebrity	29.53%	39.60%	47.65%	52.35%
	Landmark	42.95%	55.70%	**64.43%**	**73.15%**
	Movie	18.21%	27.15%	31.79%	33.77%
LRS2	Overall	**28.00%**	**38.00%**	**43.00%**	46.17%
	Celebrity	27.52%	40.27%	46.98%	50.34%
	Landmark	**43.62%**	53.02%	59.06%	65.77%
	Movie	**20.53%**	**29.47%**	**33.11%**	**34.44%**

re-rank them again. During the re-ranking, we follow two techniques: (1) we remove small documents as we discussed above and select top-1k as the final list, and (2) without removing small documents, we only consider top-1k as the final list, this method is termed as **2-Layer Retrieval with small documents or LRS2** for short.

4.4 Ablation Study

Extracting dense representation. To examine the effect of dense representation, we used pre-trained BERT embeddings, taking the [CLS] embedding for both queries and documents. We calculated the cosine similarity between each query and document, selecting the top 1,000 documents based on their cosine scores as the final list. This approach is referred to as **Semantic Search**, as shown in Table 2.

5 Result and Analysis

Table 2 and Table 3 summarize the overall test result. A semantic search shows significantly lower performance than other methods. Specifically, only 102 out of 600 queries are correctly identified. Among these, 48 are related to landmarks, 28 to movies, and 26 to celebrities. In Table 4, we show that the Mean Reciprocal Rank (MRR) is just 0.0021 for *semantic search*, indicating that correct answers are ranked extremely low in the list of results, reflecting the model's difficulty in retrieving relevant results.

Next, we utilized a reformulated query to retrieve the correct answer. In this LR1 method, BM25 model correctly identified 449 number queries out of 600. Among these, 202 are related to movies, 124 to landmarks, and 123 to celebrities. The value of MRR has also increased, now it becomes 0.3438, which implies that on average correct answers appear in around the 3rd position in the ranked list, as shown in Table 2.

Nevertheless, we explored the 2-Layer Retrieval (LR2) method, to enhance the model performance. In Table 2, we showcase that LR2 performs slightly better than LR1 in respect of Recall@1000. In this method, 451 out of 600 queries are correctly identified, and the MRR value is 0.3332. In Table 4, we present that, 204 are related to movies, 124 are related to landmarks and 123 are related to celebrities.

However, we demonstrate that LRS2 achieves the best performance among all four runs, as shown in Table 2. Using this method, the model correctly identifies 484 queries, with an MRR score of 0.3435. Specifically, among the correctly identified queries, 202 are related to movies, 144 to landmarks, and 138 to celebrities. This is a favorable outcome.

6 Conclusion and Future work

Our study displayed how traditional retrievers like BM25 can still be put to use when augmented with the generative capabilities of LLMs. A new way of query reformulation using a novel stratified prompting strategy gives us highly effective queries, producing better results. This study is an indication that more sophisticated prompting techniques need to be researched in order to produce the most optimized version of such queries.

Furthermore, the relation between further added layers of prompting and enhancement of performance needs to be studied in greater detail. The temperature, which was non-zero in the third and fourth layers, raises a very interesting question: does temperature have

Table 4: The table shows the percentage of correctly identified queries and Mean Reciprocal Rank (MRR) for each category across different methods on test data.

Model	Category	# Correct answer	Rate (%)	MRR	# Queries
Semantic Search	Landmark	48	32.21	0.005	149
	Movie	28	9.27	0.001	302
	Celebrity	26	17.45	0.002	149
	Overall	**102**	**17.00**	**0.002**	**600**
LR1	Landmark	124	83.22	0.501	149
	Movie	202	66.89	0.260	302
	Celebrity	123	82.55	0.355	149
	Overall	**449**	**74.83**	**0.344**	**600**
LR2	Landmark	124	83.22	0.487	149
	Movie	204	67.55	0.246	302
	Celebrity	123	82.55	0.356	149
	Overall	**451**	**75.17**	**0.333**	**600**
LRS2	Landmark	144	96.64	0.524	149
	Movie	202	66.89	0.240	302
	Celebrity	138	92.62	0.372	149
	Overall	**484**	**80.67**	**0.343**	**600**

a correlation with performance? Even though temperature introduces randomness in the response, challenging its reproducibility, yet it also gives greater creativity freedom to our LLM. This in turn can lead to better queries, which may reveal important ideas leading to better research in this field.

References

[1] Jaime Arguello, Samarth Bhargav, Fernando Diaz, Evangelos Kanoulas, and Bhaskar Mitra. 2023. Overview of the TREC 2023 Tip-of-the-Tongue Track. In *Proceedings of the 32nd Text REtrieval Conference (TREC 2023)* (Gaithersburg, MD, USA). National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA. https://trec-tot.github.io

[2] Jaime Arguello, Adam Ferguson, Emery Fine, Bhaskar Mitra, Hamed Zamani, and Fernando Diaz. 2021. Tip of the Tongue Known-Item Retrieval: A Case Study in Movie Identification. In *Proceedings of the 2021 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '21)* (Canberra, ACT, Australia). Association for Computing Machinery, New York, NY, USA, 10 pages. https://doi.org/10.1145/3406522.3446021

[3] Luís Borges, Jamie Callan, and Bruno Martins. 2023. Team CMU-LTI at TREC 2023 Tip-of-the-Tongue Track. In *Proceedings of the 32nd Text REtrieval Conference (TREC 2023)* (Gaithersburg, MD, USA). National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA.

[4] J Shane Culpepper, Guglielmo Faggioli, Nicola Ferro, and Oren Kurland. 2021. Topic difficulty: Collection and query formulation effects. *ACM Transactions on Information Systems (TOIS)* 40, 1 (2021), 1–36.

[5] Maik Fröbe, Christine Brychcy, Elisa Kluge, Eric Oliver Schmidt, and Matthias Hagen. 2023. Webis at TREC 2023: Tip-of-the-Tongue track. (2023).

[6] Maik Fröbe, Eric Oliver Schmidt, and Matthias Hagen. [n. d.]. A Large-Scale Dataset for Known-Item Question Performance Prediction. 15–19. http://ceur-ws.org/Vol-3366/#paper-03

[7] Junjie Huang, Jizheng Chen, Jianghao Lin, Jiarui Qin, Ziming Feng, Weinan Zhang, and Yong Yu. 2024. A Comprehensive Survey on Retrieval Methods in Recommender Systems. *arXiv preprint arXiv:2407.21022* (2024).

[8] Kevin Lin, Kyle Lo, Joseph Gonzalez, and Dan Klein. 2023. Decomposing Complex Queries for Tip-of-the-tongue Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5521–5533. https://doi.org/10.18653/v1/2023.findings-emnlp.367

[9] Jakub Lokoč, Gregor Kovalčík, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. 2019. VIRET: A Video Retrieval Tool for Interactive Known-item Search. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval* (Ottawa ON, Canada) *(ICMR '19)*. Association for Computing Machinery, New York, NY, USA, 177–181. https://doi.org/10.1145/3323873.3325034

[10] Jakub Lokoč, František Mejzlík, Patrik Veselý, and Tomáš Souček. 2021. Enhanced SOMHunter for Known-item Search in Lifelog Data. In *Proceedings of the 4th Annual on Lifelog Search Challenge* (Taipei, Taiwan) *(LSC '21)*. Association for Computing Machinery, New York, NY, USA, 71–73. https://doi.org/10.1145/3463948.3469074

[11] Paul Ogilvie and Jamie Callan. 2003. Combining document representations for known-item search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (Toronto, Canada) *(SIGIR '03)*. Association for Computing Machinery, New York, NY, USA, 143–150. https://doi.org/10.1145/860435.860463

[12] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. https://doi.org/10.1561/1500000019

[13] Zongyi Song, Vaibhav Tiwari, Jaap Kamps, and Arjen P. de Vries. 2024. A Benchmark for Known-Item Retrieval from Wikipedia. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management* (Glasgow, United Kingdom) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 2345–2352. https://doi.org/10.1145/3488560.3498421