UFMG at TREC 2025: Retriever-Aligned Query Rewriting for Tip-of-the-Tongue Retrieval

Arthur Pontes Nader[1] Rodrygo L. T. Santos[1]
[1]Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
arthurnader@dcc.ufmg.br, rodrygo@dcc.ufmg.br

Abstract

Tip-of-the-Tongue queries are difficult to rewrite due to vague user descriptions and limited supervised training data. We address this by generating rewrite preference pairs automatically from dense and cross-encoder retrieval scores, enabling a reliable dataset for fine-tuning directly on ranker preferences. We compare prompt tuning, domain-specific DPO, and general DPO models within a Tree-of-Thoughts rewriting and retrieval pipeline. Results on the TREC Tip-of-the-Tongue track show steady gains from prompt tuning to DPO, with a GPT-5-nano ensemble of all runs achieving our best performance among our submissions (NDCG@1000 = 0.277, MRR@1000 = 0.199).

1 Introduction

Tip-of-the-Tongue (ToT) retrieval addresses queries in which users cannot recall the exact name or keywords of a target item, relying instead on partial, associative, or subjective descriptions [1]. Such queries exhibit weak lexical grounding and high semantic ambiguity, making it difficult for retrieval systems to identify the intended item.

Transformer-based models [2] have reshaped information retrieval by enabling semantic matching beyond exact term overlap. Pretrained encoders such as BERT [3] and Sentence-BERT [4] form the basis of cross-encoder and dense bi-encoder retrieval architectures. However, in ToT scenarios, the effectiveness of these models is constrained by the quality of the input query itself: vague or underspecified descriptions often fail to produce reliable retrieval signals.

Query rewriting has long been explored to address query ambiguity and vocabulary mismatch. Early approaches based on query expansion, term association, and relevance feedback [5, 6] are limited in ToT settings, where queries lack stable anchor terms and explicit user feedback is unavailable.

Large Language Models (LLMs) offer a promising alternative due to their ability to generate semantically rich reformulations through paraphrasing and associative reasoning [7, 8, 9]. However, adapting LLMs for ToT query rewriting remains challenging, as supervised rewrite annotations are costly and unreliable. Recent preference-based optimization methods, such as Direct Preference Optimization (DPO) [10], enable training from pairwise preferences rather than explicit targets. In retrieval settings, such preferences can be derived automatically from ranking outcomes, providing scalable supervision without manual annotation [11, 12].

The TREC 2025 Tip-of-the-Tongue track extends prior editions by moving beyond single-domain retrieval, introducing a general-domain evaluation over 53 entity types, and incorporating a mix of MS-ToT, human-elicited, and LLM-generated queries over a large Wikipedia corpus [13]. These changes amplify the importance of rewrite generalization and robustness across heterogeneous query styles.

In this work, we propose a retrieval-based preference optimization for query rewriting in the Tip-of-the-Tongue setting. We generate multiple rewrite candidates per query, derive pairwise preferences from dense retrieval rankings, and fine-tune a rewriting model using DPO to prefer more effective reformulations. We additionally investigate prompt tuning as a lightweight adaptation strategy for query rewriting, the use of a Tree-of-Thoughts retrieval pipeline to explore multiple reformulation paths during inference, and domain-specific pipelines for handling difficult scene-based movie queries. Experiments on the official TREC ToT benchmark show that our preference-based rewriting framework consistently improves retrieval effectiveness.

2 Methodology

Our system relies on four modern techniques for adapting large language models and integrating them into the retrieval pipeline: Low-Rank Adaptation (LoRA), Prompt Tuning, Direct Preference Optimization (DPO),

and Tree-of-Thoughts reasoning. Below we briefly describe each component.

2.1 Direct Preference Optimization (DPO)

DPO [10] is a technique for aligning an LLM with a target reward without requiring explicit reinforcement learning. The model is trained on preference pairs, where rewrite A is preferred over rewrite B because it yields better retrieval performance (i.e., it ranks the target item higher, or yields a higher cross-encoder logit). This objective encourages the model to produce rewrites that behave in a way the retriever prefers.

A key advantage of DPO in our setting is that it naturally addresses the challenge described before: the lack of supervised training data for rewriting noisy, incomplete queries. Instead of relying on hand-annotated examples, we generate multiple candidate rewrites per query and evaluate them with dense and cross-encoder retrievers. The differences in their retrieval scores form *implicit preference labels*, allowing us to construct large numbers of preference pairs automatically.

In practice, this means that every query can yield many training signals, effectively expanding the dataset far beyond the size of the original supervised set. This preference-based augmentation proved critical to achieving stable and meaningful DPO training.

2.1.1 Preference Pair Construction

To train DPO models, we automatically derive preference pairs from the retrieval pipeline. Given a query, we generate a pool of LLM rewrites and evaluate each of them using dense retrieval and a cross-encoder reranker. These signals allow us to establish relative preferences without requiring human annotations.

Dense-based Preferences. All rewrites are ranked according to their dense retrieval scores (using the `all-mpnet-base-v2` encoder). This ranking naturally partitions the rewrite pool into three groups:

- **Good rewrites:** rewrites that improve the rank of the target item with respect to the original query.
- **Original query:** treated as a neutral baseline.
- **Bad rewrites:** rewrites that worsen the rank relative to the original query.

Considering rank as a metric, this induces the ordering:

$$\text{Bad rewrites} \;>\; \text{Original query} \;>\; \text{Good rewrites}.$$

For every rewrite in the "good" set, we construct a preference pair against every rewrite in the "bad" set. This yields pairs of the form:

$$\text{good rewrite} \succ \text{bad rewrite}.$$

Additionally, the good rewrites themselves are strictly ranked by their retrieval scores. If the good set contains rewrites G_1, G_2, G_3, G_4 in decreasing quality, we generate intra-group preferences:

$$G_1 \succ G_2 \succ G_3 \succ G_4.$$

These intra-group comparisons enrich the dataset with fine-grained positive signals.

Cross-Encoder Preferences. We also derive preferences from the cross-encoder logits (`cross-encoder/ms-marco-MiniLM-L-12-v2`). Each rewrite is scored by computing its similarity with every paragraph of the candidate document, and we retain the maximum logit value.

This produces another ordering:

$$\text{Bad logit} \;<\; \text{Original logit} \;<\; \text{Good logit}.$$

As with dense retrieval, we generate preference pairs between all good–bad combinations, as well as within the ordered good set, based on the cross-encoder scores.

Training Data Expansion. Because each query typically produces dozens of rewrites, this process automatically converts a single query into a large set of meaningful preference pairs. In practice, every query yields:

- good–bad dense preference pairs,
- better-good dense ordering pairs,
- good–bad cross-encoder pairs, and
- better-good cross-encoder ordering pairs.

This dramatically expands the effective size of the training set and mitigates the limited availability of supervised Tip-of-the-Tongue examples.

2.2 Low-Rank Adaptation (LoRA)

LoRA [14] is a parameter-efficient fine-tuning method that injects small trainable matrices into selected layers of a pretrained LLM. Instead of updating all model weights, LoRA learns a low-rank decomposition that modifies the model's behavior with only a small number of additional parameters. This enables fast training on modest hardware and supports the creation of multiple domain-specialized adapters on top of the same

base model. In our setting, LoRA is essential for applying Direct Preference Optimization (DPO), as it allows preference-aligned fine-tuning without modifying the full set of model weights, making DPO training both feasible and computationally efficient.

2.3 Prompt Tuning

Prompt tuning [15] is a lightweight adaptation strategy in which a fixed pretrained model is conditioned through learned soft prompts. Instead of modifying the model's internal weights, the system learns a vector prefix that steers the model toward a desired behavior—in our case, rewriting vague or underspecified user queries into more precise formulations. This provides a low-cost alternative to full fine-tuning or DPO, enabling fast experimentation with domain-specific behaviors.

Training Signal from the Best Rewrite. To train the prompt-tuned adapters, we generate a pool of candidate rewrites for each query and evaluate each of them using dense retrieval or cross-encoder reranking. Among the candidates, the rewrite that produces the *best* retrieval performance (i.e., highest dense rank or best cross-encoder score) is selected as the training target. This best rewrite is treated as the model-preferred reformulation, and the prompt-tuning objective encourages the model to reproduce or approximate this rewrite when given the original query.

Effect on the Rewriting Model. By repeatedly conditioning on these high-quality rewrites, the prompt-tuned model learns to move ambiguous Tip-of-the-Tongue queries closer to formulations that align well with the retriever's scoring behavior. Although less expressive than DPO, since it relies on a single chosen rewrite rather than a full preference structure, this approach offers a stable, efficient, and cost-effective method for improving rewrite quality without updating the full set of LLM parameters.

2.4 Tree-of-Thoughts

Tree-of-Thoughts [16] reasoning extends chain-of-thought prompting by iteratively generating and evaluating alternative query reformulations. Our system follows a greedy expansion strategy composed of the following steps:

1. **Generation:** Given the current best path, the LLM produces multiple candidate reformulations of the query.
2. **Evaluation:** Each reformulation is evaluated by performing dense retrieval to obtain candidate items, followed by reranking with a cross-encoder to assign final scores.
3. **Expansion:** The reformulation with the highest score is selected as the next node in the search tree, and the process repeats.

At each iteration, the system also tracks all candidate items retrieved across all reformulations and aggregates their scores. This iterative process enables the model to refine vague or incomplete queries over multiple reasoning steps, mimicking how a human might try several descriptions before reaching the intended target.

Tree-of-Thoughts is used purely as an inference-time search strategy; no additional supervision is applied to intermediate reasoning steps.

3 Submissions

All our runs share a common retrieval framework composed of:

(1) an LLM-based query rewriting module,

(2) dense retrieval using `all-mpnet-base-v2`,

(3) cross-encoder reranking using `cross-encoder/ms-marco-MiniLM-L-12-v2`,

(4) a Tree-of-Thoughts reasoning procedure for iterative refinement.

All rewrite models are built on top of `meta-llama/Llama-3.1-8B-Instruct` [17] and are trained using either prompt tuning or LoRA-based Direct Preference Optimization (DPO). Below we describe the aspects that differentiate each submitted run.

3.1 Prompt Tuning (Domain-Specific Adapters)

This run (`runid2`) uses domain-specialized LoRA adapters trained with prompt tuning to generate movie-specific and general-purpose query rewrites.

3.1.1 Rewrite Generation

We trained four adapters:

- **Movies:** `movies-dense` and `movies-cross`
- **General domain:** `all-dense` and `all-cross`

Training examples were constructed by selecting, for each query, the rewrite that achieved the best retrieval rank among a pool of LLM-generated candidates. A simple classifier routed each test query to the movie or general adapters. Each query produced two rewrites

(dense-oriented and cross-oriented), which were passed into the shared retrieval pipeline.

3.1.2 Retrieval with Tree-of-Thoughts

The Tree-of-Thoughts module expanded the query via iterative thoughts and candidate rewrites generated by the base LLM. Each newly generated rewrite was re-evaluated through dense retrieval and cross-encoder reranking, and the highest-scoring branch was expanded greedily until convergence or maximum depth. Final rankings were produced by selecting the highest-scoring candidates across all explored nodes.

3.2 Movies DPO (Domain-Specific Preference Optimization)

This run (`runid3`) aligns rewrite generation with dense and cross-encoder preferences using DPO, while still leveraging domain classification.

3.2.1 Preference Modeling

For each training query, multiple candidate rewrites were generated using a pretrained LLM. Dense and cross-encoder scores were used to derive preference pairs based on rank or score improvements. Separate movie and general LoRA adapters were then fine-tuned using the DPO objective.

3.2.2 Rewrite Inference

A domain classifier routed each query to the appropriate DPO adapters. The model generated multiple rewrites (dense-oriented and cross-oriented), with prompts encouraging factual, concise, and context-preserving transformations. The resulting rewrites were processed by the shared Tree-of-Thoughts based retrieval module.

3.3 General DPO (No Domain Classification)

This run (`runid1`) uses a single DPO-trained rewrite model for all queries.

3.3.1 Rewrite Model

Candidate rewrites were generated for each training query and scored using the dense and cross-encoder modules. Preference pairs were extracted based on retrieval improvements. Two general-purpose LoRA adapters (dense-aligned and cross-aligned) were trained using DPO on the full dataset.

3.3.2 Inference and Retrieval

All queries followed the same rewriting pipeline, generating:

- one dense rewrite, and
- one compact factual rewrite for reranking.

These rewrites were fed into the shared Tree-of-Thoughts retrieval framework.

3.4 Ensemble

The ensemble (`runid4`) combines the top-ranked results from the three previous runs using a relevance classifier.

3.4.1 Relevance Classification

For each query, the top-1 result from:

- Prompt Tuning,
- Movies DPO,
- General DPO

was evaluated using a lightweight `gpt-5-nano` relevance classifier, which assigned one of three labels: *relevant*, *maybe relevant*, or *not relevant*. The results of the run yielding the highest relevance label were selected.

3.4.2 Final Output

The ensemble output corresponds to the top-1 result from the run predicted to be most relevant. Ties were broken according to the priority: **General DPO** $\rightarrow$ **Movies DPO** $\rightarrow$ **Prompt Tuning**.

4 Results

Following the TREC 2025 evaluation protocol, we report NDCG@1000 and MRR@1000 as official metrics.

Table 1 summarizes the performance of our four submitted runs.

4.1 Discussion

Across all runs, we observe a consistent progression in performance as the rewriting models move from lightweight adaptation methods toward preference-aligned optimization.

The **Prompt Tuning + Tree-of-Thoughts** run yields the weakest results. Although soft prompts provide an inexpensive way to condition the model, prompt tuning

Table 1: Comparison of All Submitted Runs

Run	NDCG@1000	MRR@1000	Recall@1000	Success@5
Prompt Tuning + Tree-of-Thoughts	0.164	0.096	0.553	0.124
Movies Agents + Tree-of-Thoughts	0.222	0.141	0.650	0.190
DPO + Tree-of-Thoughts	0.246	0.162	0.675	0.206
GPT-5-nano Classifier Ensemble	0.277	0.199	0.675	0.238

alone is not sufficient for the highly variable and underspecified nature of Tip-of-the-Tongue queries. Without explicit preference signals, the model struggles to generate rewrites that generalize beyond the training distribution.

The **Movies Agents + Tree-of-Thoughts** run achieves stronger performance, largely due to the use of movie-specific DPO adapters. However, its gains appear limited by the narrow domain on which the adapters were trained. This suggests a degree of overfitting: rewrites improve for movie queries, but the model does not generalize as well across the broader Tip-of-the-Tongue dataset.

The **General DPO + Tree-of-Thoughts** model delivers further improvements across all metrics. By training preference-aligned adapters on the full dataset rather than a restricted subset, the model learns rewrite behaviors that generalize more broadly. This indicates that preference signals collected across diverse queries are more effective than domain-specific ones for Tip-of-the-Tongue rewriting.

Finally, the **GPT-5-nano ensemble** achieves the best overall performance. By selecting among the outputs of all rewrite pipelines, the ensemble leverages the complementary strengths of prompt tuning (broad coverage), Movies DPO (domain expertise), and general DPO (strong generalization). This results in the highest NDCG@1000, MRR@1000, and Success@5 scores.

Overall, the results show that:

- prompt tuning alone is insufficient for Tip-of-the-Tongue generalization;
- domain-specific DPO helps but risks overfitting;
- general DPO provides the best standalone rewrite quality; and
- combining diverse rewrite strategies yields the best overall retrieval performance.

4.2 Comparison with Track Submissions

Among the 32 runs submitted to the TREC 2025 Tip-of-the-Tongue track, our best system (runid4) ranked 10th overall according to the official NDCG@1000 metric. Additionally, our best submission relying exclusively on open-source language models throughout the entire pipeline (runid1) achieved the 12th position. This result is notable given that many higher-ranked systems employed proprietary models or heavier learning-to-rank pipelines.

We additionally compare our submissions against the official TREC 2025 baselines, which include lexical BM25 runs and a dense retrieval baseline. As reported in the track overview, the best-performing BM25 baseline outperforms several neural systems but remains substantially below our DPO-based and ensemble submissions. This highlights that, while strong lexical matching remains competitive for Tip-of-the-Tongue retrieval, preference-aligned query rewriting provides consistent gains beyond both lexical and vanilla dense baselines.

5 Conclusion

This work explored the use of preference-based query rewriting for the Tip-of-the-Tongue retrieval task. By automatically generating preference pairs from dense and cross-encoder scores, we were able to construct a large and reliable training signal without relying on human supervision, an important property given the scarcity of annotated Tip-of-the-Tongue data. Our results show that prompt tuning alone is insufficient for handling the wide variability of vague user descriptions, while domain-specific DPO adapters provide improvements but risk overfitting to narrow data.

General DPO, trained on preference signals across the entire dataset, demonstrated better performance, indicating that broad preference coverage leads to better generalization. Finally, our lightweight GPT-5-nano ensemble achieved the best overall retrieval effectiveness by leveraging complementary strengths across all runs.

Overall, these findings underscore the value of preference-based optimization as both an effective training data expansion method and a promising direction for retrieval-based rewriting. Integrating richer ranker feedback and exploring more structured preference signals represent compelling avenues for future work.

References

[1] Jaime Arguello, Samarth Bhargav, Fernando Diaz, To Eun Kim, Yifan He, Evangelos Kanoulas, and Bhaskar Mitra. Overview of the trec 2024 tip-of-the-tongue track. In *Proceedings of the Thirty-Third Text REtrieval Conference (TREC 2024)*, volume 1329 of *NIST Special Publication*, Gaithersburg, MD, USA, 2024. National Institute of Standards and Technology (NIST). Track overview for the TREC 2024 Tip-of-the-Tongue (ToT) known-item retrieval task.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, pages 5998–6008, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[4] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) and the 9th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. arXiv:1908.10084 [cs.CL].

[5] Bruno M. Fonseca, Paulo B. Golgher, Edleno S. De Moura, Bruno Pôssas, and Nivio Ziviani. Discovering search engine related queries using association rules. *Journal of Web Engineering*, 2(4):215–227, 2003.

[6] Xuanhui Wang and ChengXiang Zhai. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, pages 479–488, Napa Valley, CA, USA, 2008. Association for Computing Machinery.

[7] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. OpenAI Technical Report, 2018.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[10] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 53728–53741. Curran Associates, Inc., 2023.

[11] Chanwoong Yoon, Gangwoo Kim, Byeongguk Jeon, Sungdong Kim, Yohan Jo, and Jaewoo Kang. Ask optimal questions: Aligning large language models with retriever's preference in conversation. In *Findings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Albuquerque, NM, USA, 2025. Association for Computational Linguistics. arXiv:2402.11827 [cs.IR, cs.CL].

[12] Sungguk Cha, DongWook Kim, Taeseung Hahn, Mintae Kim, Youngsub Han, and Byoung-Ki Jeon. Annotation-free reinforcement learning query rewriting via verifiable search reward. arXiv preprint, 2025. arXiv:2507.23242 [cs.CV, cs.CL, cs.LG].

[13] Jaime Arguello, Fernando Diaz, Maik Fröbe, To Eun Kim, and Bhaskar Mitra. Overview of the trec 2025 tip-of-the-tongue track, 2026.

[14] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[15] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 3045–3059, 2021.

[16] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11809–11822, 2023.

[17] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.