UFMG at the TREC 2023 Tip of the Tongue Track

Rita Borges de Lima
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
ritarezborges@dcc.ufmg.br

Rodrygo L. T. Santos
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
rodrygo@dcc.ufmg.br

Abstract

In the TREC 2023 Tip of the Tongue (ToT) track, we address the challenge of movie retrieval from queries laden with imprecise or incorrect natural language. In particular, the Movie Identification Task aims to produce a well-ranked list of movies, identified by Wikipedia page IDs, in response to a set of queries in Tip of the Tongue (TOT) format. In our participation, we experiment with reranking techniques, leveraging both sparse and dense retrieval approaches to refine the returned results. Additionally, we incorporate term filtering heuristics for both queries and documents, enhancing the overall effectiveness of our approach.

1 Introduction

Within the domain of information retrieval, the TREC 2023 Tip of the Tongue (ToT) track presented an intriguing challenge centered around matching movies from queries characterized by imprecision and inaccuracies in natural language. Given a query topic, the primary objective of this task was to return a ranked list of a 1000 candidate movies, with the correct movie being positioned as prominently as possible. Evaluation metrics pertinent to information retrieval tasks with a single relevant document, such as discounted cumulative gain, reciprocal rank, and success@k, were employed to gauge the performance of the systems.

In the following sections, we describe the approaches developed in our participation. In particular, in Section 2, we detail how we prepared our data for both topics and documents, and describe our chosen reranking methods. In Section 3, we discuss the results attained by both our official and unofficial runs, while Section 4 offers our concluding remarks and insights.

2 Approaches

2.1 Data Preparation and Filtering Heuristics

TREC provided queries, sourced from internet forums [1, 3], and a set of Wikipedia articles for the corpus. Upon closer examination, it became evident that a predominant proportion of these topics were notably reliant on the movie's plot for their core information. Given the computational costs incurred by the contextual language modeling approaches we aimed to experiment with, we sought heuristics to identify the most pertinent sections within the corpus, to use those as input.

To achieve this, our approach involved an exploration of the articles within our corpus. This corpus was divided into sections which we leveraged to search for the plot of the movie by comparing the section titles with keywords such as "synopsis", "plot", "description", and others. We also appended the beginning of the text which generally contained important details such as the year of release, director, genre, and language, enriching the topics with valuable context. However, in cases where an explicit plot section was absent, we resorted to extracting information from the introductory segments of the text, truncating it to adhere to the prescribed constraints.

Topics provided by TREC were presented in sentence form, with each sentence denoting different features in Boolean form. To improve the quality and effectiveness of our reranking process, we removed all sentences tagged as "social", recognizing that these segments often contained superfluous content that had the potential to confuse the reranking system.

2.2 Ranking Strategies

In our participation, we implemented a cascade retrieval approach to enhance the effectiveness and

efficiency of our movie retrieval process. Firstly, we employed well-established techniques, such as BM25, for the initial ranking of movies within both the original, truncated corpus and the meticulously processed version. These methods served as a foundational layer for the retrieval process.

To further refine and optimize the ranking, we used techniques such as BERT-based models, including monoBERT [4], monoT5 [5], and duoT5 [5]. These models offered advanced natural language understanding capabilities and were integrated into the reranking process. Each model brought distinct strengths to the table, such as semantic understanding and context-awareness, enhancing the overall quality of the results. Moreover, we looked into how combining these BERT-based models [6] in a pipeline could enhance the effectiveness of reranking. This approach involved sequential reranking steps, each executed by a different model.

In addition to these strategies, we reranked the outputs generated by the DistilBERT [7] and GPT4 [2] baseline runs, using the modified corpus and queries. This approach integrated semantic matching capabilities into the first-pass retrieval stage of our cascading pipeline, in the hope of improving recall for the subsequent reranking stages.

3 Experiments

3.1 Runs Overview

We generated a total of 8 runs for our participation in the TREC ToT track, with five of them officially submitted:

- ufmgBMmBQ (unofficial): we reranked the BM25 baseline run using monoBERT, leveraging the processed queries.
- ufmgBMmBQD (unofficial): we reranked the BM25 baseline run using monoBERT, leveraging the processed queries and corpus.
- ufmgDBmBQ (official): we reranked the DistilBERT baseline run using monoBERT, leveraging the processed queries.
- ufmgDBmBQD (official): we reranked the DistilBERT baseline run using monoBERT, leveraging the processed corpus and queries.
- ufmgDBdTQD (unofficial): we reranked the DistilBERT baseline using duoT5, with both the processed corpus and queries.
- ufmgDBmBdTQD (unofficial): we reranked the DistilBERT baseline run first with monoBERT, and then with duoT5, using the processed corpus and queries.
- ufmgG4mBQD (official): we reranked the GPT4 baseline run using monoBERT, with both the processed corpus and queries.
- ufmgG4dTQD (official): we reranked the GPT4 baseline run using duoT5, with both the processed corpus and queries.

3.2 Results

Table 1 and Table 2 present the results of both our unofficial and officially submitted runs for the task, as evaluated by the official metrics on the dev and test sets, respectively. In the dev set results (Table 1), we initially conducted unofficial runs to assess our reranking techniques. Notably, reranking the DistilBERT baseline run using monoBERT with processed queries and corpus (ufmgDBmBQD) yielded promising results. Moving to the official results in the test set (Table 2), our reranking approach using monoBERT continued to yield strong results. Notably, ufmgDBmBQ outperformed ufmgDBmBQD, showing a different behavior from the dev set.

Moreover, in the official run ufmgG4mBQD, we observed an impressive performance surpassing other approaches that reranked the DistilBERT baseline in the test set, a notable contrast to the outcomes in the dev set. Overall, reranking strategies utilizing monoBERT consistently delivered superior results compared to monoT5, duoT5, and combined strategies. These findings underscore the overall effectiveness of our approach within the context of the TREC ToT track, demonstrating the trade-offs of combining sparse and dense rankers in a cascading pipeline.

Run	Submitted	nDCG	Success	MRR
ufmgBMmBQF	✗	0.1203	0.3033	0.0892
ufmgBMmBQDF	✗	0.1187	0.3033	0.0873
ufmgBMmBQ	✗	0.1148	0.2533	0.0860
ufmgBMmBQD	✗	0.1148	0.2533	0.0860
ufmgBMmBQD	✗	0.1297	0.3139	0.0872
ufmgDBmBQ	✓	0.1984	0.4133	0.1450
ufmgDBmBQD	✓	0.2045	0.4267	0.1499
ufmgDBmBdTQD	✓	0.1412	0.3990	0.7870
ufmgDBdTQD	✗	0.1812	0.4089	0.1424
ufmgG4mBQD	✓	0.1979	0.3200	0.1638
ufmgG4mTQD	✗	0.1780	0.3200	0.1484
ufmgG4dTQD	✓	0.1872	0.3200	0.1511

Table 1: Results on the dev set.

Run	Submitted	nDCG	Success	MRR
ufmgDBmBQ	✓	0.2090	0.3933	0.1636
ufmgDBmBQD	✓	0.1998	0.4067	0.1507
ufmgDBmBdTQD	✓	0.1108	0.4067	0.0505
ufmgG4mBQD	✓	0.2404	0.3733	0.2002
ufmgG4dTQD	✓	0.1668	0.3733	0.1189

Table 2: Official results on the test set.

4 Conclusions

In TREC 2023, we took part in the Tip of the Tongue (ToT) track. Our participation involved exploring the structure of the corpus and queries and harnessing specific aspects of the task to optimize our results. In addition, we experimented with a cascading retrieval pipeline combining both sparse and dense retrieval approaches.

References

[1] Samarth Bhargav, Georgios Sidiropoulos, and Evangelos Kanoulas. "'It's on the Tip of My Tongue': A New Dataset for Known-Item Retrieval". In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. WSDM '22. Virtual Event, AZ, USA: Association for Computing Machinery, 2022, pp. 48–56. ISBN: 9781450391320. DOI: 10.1145/3488560.3498421. URL: https://doi.org/10.1145/3488560.3498421.

[2] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[3] *I Remember This Movie, howpublished = https://irememberthismovie.com*.

[4] Rodrigo Nogueira et al. *Multi-Stage Document Ranking with BERT*. 2019. arXiv: 1910.14424 [cs.IR].

[5] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. *The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models*. 2021. arXiv: 2101.05667 [cs.IR].

[6] 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*. Association for Computational Linguistics, Nov. 2019. URL: https://arxiv.org/abs/1908.10084.

[7] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL].