UWaterlooMDS at TREC 2023: Deep Learning Track and Tip-of-the-Tongue Track

Dake Zhang

School of Computer Science, University of Waterloo
Waterloo, Ontario, Canada

Abstract

Our submissions to the TREC 2023 Deep Learning Track and the Tip-of-the-Tongue Track utilized the power of language models. For the Deep Learning track, we prompted a Large Language Model (LLM) to generate more queries for BM25 retrieval, which did not yield better performance than the BM25 baseline. We also tried to prompt the model to perform passage assessments similar to human assessors, which effectively improved the ranking of the baseline. For the Tip-of-the-Tongue track, we used a general-purpose text embedding model to perform dense retrieval, achieving better performance than the dense retrieval baseline with a high recall. When we instructed an LLM to assess whether a Wikipedia page matches a user's description, the model did not seem to produce accurate assessments.

1 Introduction

TREC 2023 Deep Learning Track [2], in its fifth and last year, had a similar design to its fourth year, with two tasks: passage ranking and document ranking. Given a set of queries from users and synthetic queries from language models, the task was to return a ranked list of candidate passages or documents based on the likelihood of having an answer to the query.

TREC 2023 Tip-of-the-Tongue Track [1], a first-year track, aimed to foster the development of search algorithms to help people find previously watched movies based on their tip-of-the-tongue descriptions, which could be verbose, inaccurate, incomplete, and complex. Given a set of tip-of-the-tongue descriptions of movies and a subset of Wikipedia pages, the task was to return a ranked list of 1,000 candidate pages in terms of their possibilities to match each given description.

In the rest of this report, we describe our methods for submitted runs, which leveraged the power of Large Language Models (LLMs), and provide some preliminary analyses of the evaluation results.

2 Methods

2.1 Dense Retrieval

We used dense retrieval methods for the Tip-of-the-Tongue track, where the relevance score of each query-document pair was computed as the cosine similarity between the text embedding vectors of each. We experimented with several text embedding models and finally chose the General Text Embeddings (GTE) model GTE-LARGE [3] based on its best zero-shot performance in retrieving relevant documents, i.e., without fine-tuning, on the development set. We prepared two dense retrieval runs for the Tip-of-the-Tongue track: `WatS-DR` where GTE-LARGE was directly applied on the test set, and `WatS-TDR` where GTE-LARGE was first fine-tuned on the training set and then applied on the test set.

2.2 Large Language Models

With billions of parameters, LLMs exhibit superior performance of understanding text and are able to perform many downstream tasks in zero-shot or few-shot settings with proper prompts. In our experiment, we attempted to utilize the power of LLMs to expand queries and perform document-level assessments. To ensure the reproducibility of our experiment and endorse the spirit of open-sourcing, we chose LLAMA 2 [4] as the LLM component in our methods, which has multiple sizes: 7B, 13B, and 70B, each with two variant: the foundation version that was only trained on a massive text corpus, and the chat version that was further fine-tuned on a curated instruction set. At the time when we prepared our runs (from July to August in 2023), LLAMA 2 70B CHAT was the best-performing open-source LLM on

```
[INST] <<SYS>>
You are a search engine assistant. Your job is
    to help users create a list of candidate
    queries based on their information needs (
    questions). Your queries should be suitable
    for term-based retrieval algorithms, such as
     BM25. You may replace the original question
     with relevant words or synonyms. Your
    queries should be diverse and aim to find
    answers to the original question. You need
    to replace abbreviation words with their
    full forms. Put your candidate queries in
    quotation marks.
<</SYS>>
Original question: {query}
Please create a list of diverse candidate
    queries for search engines. [/INST]
```

Figure 1: Prompt template for expanding queries. Underscored words in curly brackets are placeholders, which will be filled with corresponding values in each model call.

```
[INST] <<SYS>>
You are a document labeler. Your task is to read
     a passage and a query from the user and
    judge whether the passage is relevant to the
     query. Your answer should be "yes" or "no".
     You don't need to explain your answer.
<</SYS>>
Query: {query}
Passage: {passage}
Question: {question regarding one aspect}
Please answer "yes" or "no". [/INST]
```

Figure 2: Prompt template for assessing passages from the Deep Learning track. Underscored words in curly brackets are placeholders, which will be filled with corresponding values in each model call.

the HuggingFace Open LLM Leaderboard[1].

2.3 Prompt Engineering

Due to time and hardware constraints, we were only able to run inference of Llama 2 models rather than fine-tuning them, i.e., writing suitable prompts to instruct them to perform different downstream tasks. In other words, we depended on the models' knowledge and reasoning capabilities acquired during pretraining and/or instruction fine-tuning. In the rest of this section, we describe how we prompted Llama 2 to perform several tasks and how we incorporated their outputs into producing runs.

2.3.1 Query Expansion

The intuition is that having a set of relevant queries for the same information need may increase the chance of finding more relevant documents, i.e., higher recall, for term-based retrieval algorithms such as BM25, which are poor at understanding synonyms and matching relevant terms. Figure 1 shows the prompt we used to instruct Llama 2 70B Chat to expand each query.

We used this query expansion method to produce a passage retrieval run for the Deep Learning track. For each generated query, we ran BM25 to get the top 100 results. Then we merged those ranked lists in a round-robin way, incorporating one relevant passage from each list at each round. When a retrieved passage was included into the merged list, its score would be its BM25 score. If the passage was already in the merged list, its score would be updated by adding the BM25 score of the duplicate passage to be merged. In this way, a document that is ranked highly for multiple queries will also be ranked highly in the final list. The final run `WatS-Augmented-BM25` was created by sorting their scores and taking the top 100 results for each query.

[1] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

2.3.2 Document Assessment

We tried to use LLMs to perform document assessments, based on which we further reranked documents from the previous retrieval stage. We had different prompts to assess several aspects of a candidate document regarding a given query. For the passage retrieval task of the Deep Learning track, we asked Llama 2 70B Chat to assess the following three aspects of a candidate passage regarding the query:

1. Is the above passage relevant to the query?
2. Does the above passage answer the query?
3. Does the above passage provide a direct answer to the query?

Figure 2 shows the prompt we used to assess each candidate passage regarding the query in terms of one aspect. For each query-passage pair, the model was called three times to generate three responses regarding the three aspects mentioned above respectively. If the model gave *yes* to the third aspect, the passage to be assessed would have a boost score of 3. If it was not the case but the model gave

```
[INST] <<SYS>>
You are a movie expert. You need to read the
    user's description of a movie and one
    Wikipedia page of a movie. Then you need to
    evaluate whether the Wikipedia page matches
    the user's description. You need first to
    find relationships between the description
    and the Wikipedia page, including
    consistencies and inconsistencies, and then
    give your overall confidence (perfect match,
     mostly match, partial match, no match) of
    whether the Wikipedia page matches the user'
    s description.
<</SYS>>
User description:
{query_title}. {query_text}
Wikipedia page:
{wiki_page[:10000]}
Now, please find relationships between the
    description and the Wikipedia page,
    including consistencies and inconsistencies,
     and then give your overall confidence (
    perfect match, mostly match, half match,
    less match, no match) of whether the
    Wikipedia page matches the user's
    description. [/INST]
```

Figure 3: Prompt template for assessing Wikipedia pages from the Tip-of-the-Tongue track. Underscored words in curly brackets are placeholders, which will be filled with corresponding values in each model call.

yes to the second aspect, the passage would have a boost score of 2. If it was still not the case not but the model gave *yes* to the first aspect, the passage would have a boost score of 1. Otherwise, the passage would not have a boost score, i.e., 0. To rerank results from the previous stage, we set the final score to be `10` $\times$ `boost` + `previous_score`. We produced two runs: `WatS-LLM-Rerank-Base` and `WatS-LLM-Rerank` by reranking `BM25-Baseline`, which was provided as a baseline run by the track organizers, and `WatS-Augmented-BM25` respectively. Unfortunately, we accidentally missed `WatS-LLM-Rerank-Base` in our submissions to the track, so we calculated its evaluation results on our own using the qrels released by the track organizers.

Note that running inference of Llama 2 70B Chat to assess documents was very time-consuming and computationally expensive, especially since we had 700 queries and 100 documents for each query. It took about two hours to finish assessing 1,000 documents with three aspects on a computing node with four A100 (40G RAM) GPUs.

For the Tip-of-the-Tongue track, the number of documents to be assessed was even more, with 150 queries and 1,000 candidate documents retrieved by dense retrieval for each query. Meanwhile, the candidate documents are Wikipedia pages, which are much longer than passages in the Deep Learning track. However, Llama 2 can only handle at most 4,096 tokens (prompt + generation). Longer sequence length also results in exponentially higher GPU memory consumption. To the lower computation requirement, we chose a smaller model Llama 2 13B Chat to perform document assessments and prompted the model to return assessments out of several given options in one call. Figure 3 shows the prompt we used, where we truncated the Wikipedia page at its 10000th character to ensure the total sequence would not exceed the limit of Llama 2 13B Chat.

We mapped each option to a level score: *perfect match* - 4, *mostly match* - 3, *half match* - 2, *less match* - 1, *no match* - 0. Those level scores were used to rerank our dense retrieval run `WatS-TDR`, where the merged score was computed as `level` + `cosine_similarity` (`cosine_similarity` $\in [-1, 1]$). In some cases, the model failed to generate the exact words of the given options, which would require humans to read the response and determine what option the model chose, e.g., *no match* is equivalent to *completely different*. Thus, we did not consider this run as a purely automatic run. Perhaps more powerful models would adhere more strictly to the instructions. Unfortunately, we produced duplicate documents for some queries when we merged results when running distributed inference on a compute cluster and did not notice this issue when submitting this run `WatS-TDR-RR`. But we still got the evaluation result of this run from the track organizers.

3 Results and Discussion

3.1 Deep Learning Track

Table 1 shows the evaluation results of our runs and the BM25 baseline run provided by the track organizers. Note that our methods did not utilize the qrels from previous years.

3.1.1 Query Expansion

Table 2 shows an example list of generated queries based on the original query "*Why are the feathers of birds bright?*". We can see that `LLAMA 2 70B Chat` is good at paraphrasing sentences, especially replacing words with synonyms. However, replacing frequently used words with less frequently used ones seems to

Run	AP	NDCG@10	Precision@10	MRR
BM25-Baseline	0.092	0.262	0.381	0.627
WatS-Augmented-BM25	0.076	0.193	0.171	0.422
WatS-LLM-Rerank	0.107	0.290	0.267	0.522
WatS-LLM-Rerank-Base	**0.115**	**0.371**	**0.501**	**0.723**

Table 1: Evaluation results of some runs submitted to the Deep Learning track. BM25-Baseline was provided by the track organizers. Note that we accidentally missed WatS-LLM-Rerank-Base in our submissions to the track. We calculated its evaluation results ourselves after obtaining the qrels from the track organizers.

Original Query:
Why are the feathers of birds bright?
Expanded Candidate Queries:
1. Why do birds have vibrant feathers?
2. What is the purpose of brightly colored feathers in birds?
3. Birds' feather coloration: why are some species so bright?
4. The role of pigmentation in bird feathers: a search for brightness
5. Bright feathers, big birds: is there a connection?
6. Are brightly colored birds more attractive to mates?
7. Do bright feathers serve as a warning signal for predators?
...

Table 2: Some candidate queries generated by LLAMA 2 70B Chat based on the original query (query_id: 3004033).

have little help in improving BM25 retrieval, e.g., replacing *bright* with *vibrant* in the first generated question. Meanwhile, we observe that LLAMA 2 70B Chat tried to incorporate potential answers into the query, e.g., the sixth generated question "*Are brightly colored birds more attractive to mates?*". Those generated questions seemed to be reasonable and relevant to the user's information needs.

However, in terms of the overall evaluation results, from the comparison between BM25-Baseline and WatS-Augmented-BM25 in Table 1, we can see that our query expansion pipeline was not able to improve the retrieval performance of the BM25 algorithm under any of the four metrics. We need to further study the performance of generated queries from LLAMA 2 70B Chat with our prompt in terms of its ability to produce suitable queries for term-based retrieval algorithms. We may also need to investigate other ways to effectively combine the expanded queries with the BM25 algorithm.

Run	nDCG	Success@1000	MRR
BM25	0.139	0.447	0.084
Dense-Retrieval	0.143	0.547	0.068
GPT-4	**0.264**	0.376	**0.233**
WatS-DR	0.204	0.707	0.120
WatS-TDR	0.248	**0.753**	0.152
WatS-TDR-RR	0.124	**0.753**	0.034

Table 3: Evaluation results of some runs submitted to the Tip-of-the-Tongue track. The top three runs are baseline runs provided by the track organizers, and the bottom three runs are ours.

3.1.2 LLM Reranking

From the comparison between BM25-Baseline and WatS-LLM-Rerank-Base, and the comparison between WatS-Augmented-BM25 and WatS-LLM-Rerank, we can see that our reranking with LLM assessments was able to improve the overall retrieval performance under all four metrics. The relatively poor performance of WatS-LLM-Rerank was caused by the poor performance of WatS-Augmented-BM25 that it was based on. Meanwhile, WatS-LLM-Rerank-Base achieved the best performance among our runs, especially in terms of NDCG@10, precision@10 and mean reciprocal rank. This indicates that LLAMA 2 70B Chat with our prompt can help assess passages in terms of user queries, identify useful passages, and move them to higher positions in the final ranked list.

3.2 Tip-of-the-Tongue Track

Table 3 shows the evaluation results for some runs submitted to the Tip-of-the-Tongue track.

3.2.1 Dense Retrieval

We can see that all our runs that were based on dense retrieval achieved better performance than the BM25 baseline run, which indicates that term-based retrieval algorithms are not suitable for this task, i.e.,

determining whether a Wikipedia page matches the user's tip-of-the-tongue description.

Comparing our `WatS-DR` and the provided dense retrieval baseline `Dense-Retrieval`, we can see that even without being fine-tuned on the provided training set, the text embedding model GTE-LARGE achieved much better performance than the one used in the dense retrieval baseline under all three metrics. `WatS-TDR` achieved even better performance when we fine-tuned GTE-LARGE on the provided training set.

The comparison between our `WatS-TDR` and the provided `GPT-4` is interesting. `WatS-TDR` was able to get the correct answer, i.e., finding the matched Wikipedia page, within the top 1,000 ranked results for about 75% of the time, as observed from the Success@1000 metric. Furthermore, when it got the correct answer within the top 1,000, most of the time the correct answer was among the top ten results, as observed from the MRR metric. For `GPT-4`, it could only get the correct answer for 37.6% of the time, but when it got the answer, the answer seemed to be ranked extremely high (top one or two), as observed by considering the Success@1000 metric and the MRR metric together.

3.2.2 LLM Reranking

When we incorporated the LLM assessment described in Section 2.3.2, `WatS-TDR-RR` decreased the performance of a good run `WatS-TDR` to be even worse than the `BM25` run in terms of nDCG and MRR, which means `LLAMA 2 13B Chat` with our prompt was probably not able to make accurate assessments of whether a Wikipedia page matches the user's tip-of-the-tongue description and therefore messed up the initially good ordering of results in `WatS-TDR`.

4 Conclusion

We experimented with `LLAMA 2` variants during our participation in the TREC 2023 Deep Learning Track and the Tip-of-the-Tongue Track. For the Deep Learning track, with our crafted prompt, `LLAMA 2 70B Chat` was able to perform passage assessments, based on which we improved the baseline ranking. However, our attempt to prompt the model to generate more queries to augment the BM25 retrieval did not result in better performance than the BM25 baseline. For the Tip-of-the-Tongue track, using a more recently released and more powerful text embedding model, we achieved better dense retrieval performance than the provided dense retrieval baseline, with a high recall. But `LLAMA 2 13B Chat` did not seem to produce accurate assessments, when we instructed it to assess whether a Wikipedia page matches a user's description. Our work shows the potential of LLMs in improving search algorithms, with limitations such as poor performance for tasks requiring higher levels of reasoning and expensive computation costs.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (RGPIN-2020-04665, RGPAS-2020-00080), and in part by the Digital Research Alliance of Canada.

References

[1] Jaime Arguello, Samarth Bhargav, Fernando Diaz, Evangelos Kanoulas, and Bhaskar Mitra. 2023. Overview of the TREC 2023 Tip-of-the-Tougue Track. In *TREC*.

[2] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2023. Overview of the TREC 2023 Deep Learning Track. In *TREC*.

[3] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. arXiv:2308.03281 [cs.CL]

[4] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen

Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]