University of Tsukuba Team at the TREC 2023 Deep Learning Track

KAIYU YANG, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan
LINGZHEN ZHENG, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan
HAITAO YU, Institute of Library, Information and Media Science, University of Tsukuba, Japan
SUMIO FUJITA, LY Research, LY Corporation, Japan
HIDEO JOHO, Institute of Library, Information and Media Science, University of Tsukuba, Japan

This paper describes the approaches used in three automatic submission runs for the TREC 2023 deep learning track specifically for the passage re-ranking task. We tested three different approaches using GPT-3.5-turbo, GPT-4, and a combination of multiple LLMs to explore effective methods for this task and demonstrated a variable performance of these methods, where none did better than the average results from the other participants in the track. These findings indicate a potential area for further exploration into how current LLMs re-rank search results, highlighting the need for careful prompt creation and model selection in information retrieval. Our work is an initial attempt to understand what LLMs can achieve and where they could be improved, offering some direction for future research in this area.

1 INTRODUCTION

This paper presents our participation in the Deep Learning Track. This year's Deep Learning Track continues its legacy in advancing passage and document ranking tasks. It introduces transformative elements that redefine the scope and potential of deep learning in information retrieval. The Deep Learning Track of TREC 2023 is specifically designed to address the challenges in Information Retrieval that substantially benefit from the availability of extensive training datasets. This context fosters an environment conducive to a detailed and comparative analysis of retrieval algorithms ranging from intricate, deep neural network architectures to robust non-neural methodologies. This consistent focus allows for a deeper and more nuanced understanding of those challenges and potential solutions.

A notable innovation in this year's track is the introduction of synthetic and human queries from MS MARCO. These synthetic queries, generated using a fine-tuned T5 model and GPT-4 prompts, represent a bold step towards diversifying the test set and exploring the feasibility of synthetic queries in test collection construction. This approach leads to a more challenging and varied test bed and probes the boundaries of current methodologies in information retrieval.

The Passage Ranking Task offers an opportunity to examine how Large Language Models (LLMs) perform in this context. In this year's track, we observed that runs involving LLM prompting, particularly

Authors' addresses: Kaiyu Yang, s2321730@u.tsukuba.ac.jp, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan; Lingzhen Zheng, s2221686@u.tsukuba.ac.jp, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan; Haitao Yu, yuhaitao@slis.tsukuba.ac.jp, Institute of Library, Information and Media Science, University of Tsukuba, Japan; Sumio Fujita, sufujita@lycorp.co.jp, LY Research, LY Corporation, Japan; Hideo Joho, hideo@slis.tsukuba.ac.jp, Institute of Library, Information and Media Science, University of Tsukuba, Japan.

using GPT-4, demonstrated some improvements compared to traditional retrieval algorithms. Thus, our investigation aims to understand the capabilities and limitations of LLMs in passage re-ranking.

Our exploration focused on the following research questions:

- **RQ1**: How do LLMs perform in passage re-ranking tasks compared to traditional retrieval algorithms and deep learning methods when evaluated on more challenging and diverse test query sets?
- **RQ2**: Can the capability of combining multiple small-scale local large language models in a ranking task compete with the effectiveness of mature commercial large language models?

2 METHODOLOGY

This section overviews the models and methods used in the Deep Learning Track and presents the details of the submitted runs. Three runs were submitted for evaluation, each adopting a different strategy and model for passage re-ranking. This approach facilitates investigating how different re-ranking methods perform when applied through LLMs.

2.1 Submission Runs Overview

- Run_1: Employs the GPT-3.5-turbo model used in ChatGPT [3]. The sliding window parameters are the ones in the RankGPT [6] paper, with a window size of 20, a step of 10, and a single pass (K=1).
- Run_2: This run is akin to **Run_1** but employs the GPT-4 model for the re-ranking process, leveraging its advanced capabilities compared to GPT-3.5-turbo.
- Run_3: This run implements a combination of multiple LLMs for re-ranking. It focuses on the top 10 passages retrieved in each iteration, applying pairwise ranking to refine the results.

The three runs can be categorized into two groups. The first group of RankGPT-based approaches uses commercial APIs, such as GPT-3.5-turbo and GPT-4 [4], and the second group utilized multiple offline LLMs to re-rank candidate passages. These runs provide the basis for evaluating the effectiveness of different models and re-ranking strategies in the TREC 2023 Deep Learning Track Passage Ranking Task.

2.2 RankGPT Based Re-ranking

This subsection describes the methodologies used in **Run_1** and **Run_2**, focusing on implementing Listwise Ranking Prompting and the Sliding Window Strategy within the RankGPT framework. These methods are employed to address the challenges and utilize the strengths of LLMs in processing and ranking textual data.

2.2.1 Listwise Ranking Prompting: The prompt input into the GPT model and the candidate passages are divided into the prefix and post prompt. The prefix prompt establishes the role of the GPT model, enabling it to understand the task it needs to handle correctly. The post prompt describes the specific task that GPT must process in each round of sorting. Table 1 reports the detailed form of the prompt. It should be noted that the proposed method ranks passages directly without producing an intermediate relevance score.

2.2.2 Sliding window strategy: Due to the token length limitation of the GPT model, it is unfeasible to submit all candidate paragraphs for ranking at once. Instead, a listwise ranking method is adopted, where each input includes a sublist containing 10 or 20 passages. This methodology employs two hyperparameters: the window size w and the step size s. Initially, LLMs are utilized to sequentially rank passages from the $(M - w)$-th position to the M-th position. Subsequently, the window is advanced in increments defined by step size s, and a re-ranking of passages is conducted for the new interval, extending from the $(M - w - s)$-th to the $(M - s)$-th passage. This iterative process of sliding and re-ranking is repeated until a comprehensive re-ranking of all passages is achieved.

Run_1 and **Run_2**: With the methodologies mentioned above as a foundation, **Run_1** and **Run_2** were conceived. Both runs were generated in alignment with the RankGPT paradigm. Initially, we used the Pyserini [2] toolkit to filter 100 candidate passages via the BM25 algorithm. Subsequently, these passages and their queries were input into the GPT model's API. The Listwise Ranking Prompting rule and the Sliding Window Strategy were instrumental in evaluating and ranking the returned paragraphs for relevance.

Prompt Category	**Content**
prefix_prompt	role: system, content: You are RankGPT, an intelligent assistant that can rank passages based on their relevance to the query. role: user, content: I will provide you with {num} passages, each indicated by number identifier []. Rank the passages based on their relevance to query: {query}."
post_prompt	Search Query: {query}. Rank the {num} passages above based on their relevance to the search query. The passages should be listed in descending order using identifiers. The most relevant passages should be listed first. The output format should be [] > [], e.g., [1] > [2]. Only response the ranking results, do not say any word or explain.

Table 1. Detailed Prompts Used in the Submission Run 1 and Run 2

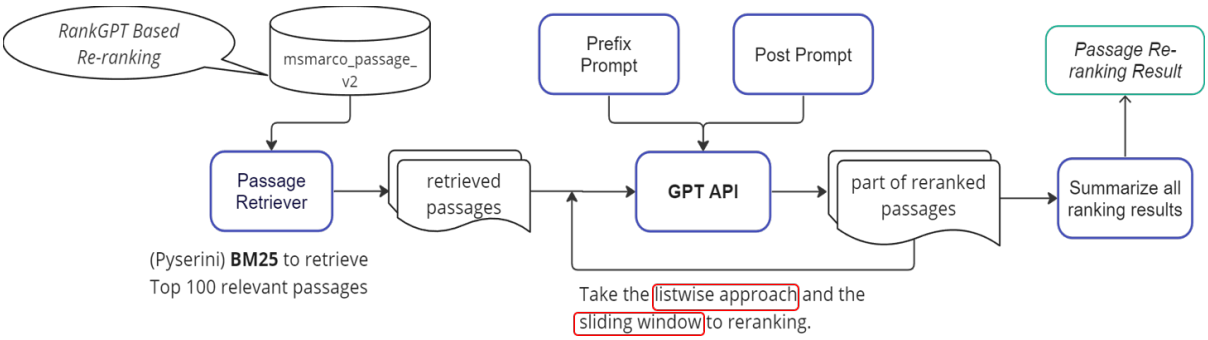

Fig. 1. Our Re-ranking Pipeline for the Submission Run 1 and Run 2

2.3 LLM Blender-Based Re-ranking

We adopt an innovative approach in **Run_3**, drawing on recent advancements in LLM research, particularly the LLM-Blender framework [1]. LLM-Blender, originally an ensembling framework for natural language generation tasks, was adapted in our work to enhance the re-ranking process. This adaptation leverages the combined strengths of multiple open-source LLMs, aiming for improved performance compared to single-model approaches.

We also incorporate ideas from recent literature on applying LLMs to ranking tasks and introduce a novel prompting technique, pairwise ranking prompting [5]. This method streamlines the ranking process by focusing on pairwise comparisons, offering an efficient alternative to traditional pointwise and listwise methods.

Run_3 utilizes a tailored LLM model within the LLM-Blender framework, specifically for re-ranking the top 10 documents. This model employs a pairwise comparison strategy designed to optimize the ranking accuracy in a more resource-efficient manner. The process is described in detail as follows:

(1) **Pairwise Document Comparison**:
   - The top 10 ranked documents are represented as a set $D = \{d_1, d_2, \ldots, d_{10}\}$.
   - These documents are paired in every possible combination without repetition, leading to $\binom{10}{2} = 45$ unique pairs. Each pair is denoted as (d_i, d_j), where $1 \leq i < j \leq 10$.

(2) **Prompt Generation for Each Pair**:
   - For each document pair (d_i, d_j), a specific prompt is crafted, yielding 45 prompts for evaluation.

(3) **LLM Processing**:
   - Five different large language models, represented by $LLM = \{llm_1, llm_2, \ldots, llm_5\}$, are employed to process these prompts.
   - Each model llm_k processes the prompts and provides a preference between the two documents in each pair.

(4) **Voting Mechanism**:
   - Each LLM llm_k processes the prompts and indicates a preference for one document over the other in each pair (d_i, d_j).
   - If llm_k prefers d_i over d_j, it is recorded as $V_{llm_k}(d_i) > V_{llm_k}(d_j)$, meaning a vote in favor of d_i against d_j from model llm_k.

(5) **Final Ranking Determination**:
   - The final document ranking in D is determined by aggregating the votes each document receives in all its pairings.
   - The document with the highest aggregate vote count is ranked first, establishing a final ranking order based on the collective insights from the LLM evaluations.

Table 2 reports a pairwise ranking prompt developed for Run 3. We anticipate two primary outputs from the LLM: ('Paragraph 1' and 'Paragraph 2'), streamlining the vote-counting process. This preliminary approach is an initial step in evaluating the potential benefits of such a method, as it provides initial insights into the performance of different LLMs in re-ranking scenarios. Nevertheless, further investigation is required to understand its impact. In **Run_3**, we explore a nascent perspective on LLM-based re-ranking strategies, aiming to understand how multiple LLMs might contribute to more effective passage re-ranking in a tentative and exploratory manner.

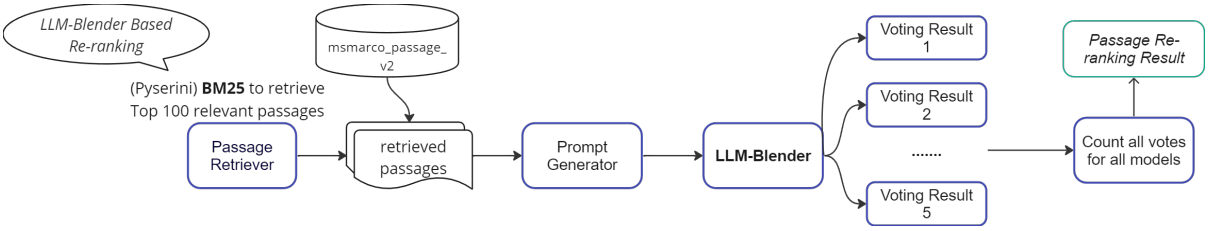

Fig. 2. Proposed Re-ranking Pipeline for the Submission Run 3

Prompt Category	**Content**
Instruction	For the query "Given a query ", at about what age do adults normally begin to lose bone mass?", which of the following two passages is more relevant?
Input	"Passage 1: 1. Losing bone density is normal in aging. We reach peak bone mass between ages 25 and 30, and then slowly lose and start losing bone mass at age 40. For women, reduced levels of estrogen after menopause accelerate bone density loss. Passage 2: Until about age 30, people normally build more bone than they lose. During aging, bone breakdown begins to outpace bone buildup, gradually losing bone mass. Once this loss of bone reaches a certain point, a person has osteoporosis.

Table 2. Table of Pairwise Ranking Prompt Used in the Submission Run 3

3 RESULTS

3.1 Evaluation Results

Run	NDCG@5	NDCG@10	P@10
Run_1	0.3925	0.3376	0.2098
Run_2	**0.4612**	**0.3927**	**0.2512**
Run_3	0.2822	0.2630	0.1683

Table 3. Evaluation Results of Our Submitted Runs on the TREC 2023 Deep Learning Track

Table 3 presents the performance of our three runs. Notably, **Run_2**, which employed the GPT-4 model, achieved the highest nDCG scores across all metrics, showcasing its strong performance in retrieval tasks. In contrast, **Run_1**, despite not reaching the top scores, still demonstrates reasonable effectiveness, particularly in terms of NDCG@5 and NDCG@10, indicating its potential in passage re-ranking tasks. However, **Run_3**, which explored an integrated approach using multiple LLMs, did not achieve as high scores as the other runs, indicating areas for further improvement and optimization in this method.

The results section evaluates the effectiveness of our methods, compares their performance, and identifies potential areas for improvement. Based on the initial expectations, we originally thought that the re-ranking method based on LLM Blender (**Run_3**) would achieve competitive nDCG scores because it adopted an integrated approach. However, the results indicated that **Run 2**, which utilized the GPT-4 model, outperformed in all aspects.

Run		NDCG@10
Run_1	Better than median	20
	Worse than median	62
Run_2	Better than median	26
	Worse than median	56
Run_3	Better than median	8
	Worse than median	74

Table 4. Statistics Results of Our Submitted Runs' Evaluation Results

Table 4 details each run's performance relative to the median, best, and worst scores. Notably, while **Run_1** and **Run_3** show balanced outcomes, **Run_2** dominates in terms of nDCG metrics.

3.2 Worst performing queries

Query 2006028 - what did colonial women wear: We found that the same query performed the worst in all three of our runs compared to the runs of the other groups. For this query, the nDCG@10 values for our three runs were close to 0 (0.0000 for **Run_1**, 0.0734 for **Run_2**, and 0.0734 for **Run_3**). We checked the first 10 documents initially filtered out by the BM25 algorithm and found zero number of paragraphs related to the query. Therefore, the performance of our method, which further re-ranks based on the BM25 algorithm, was naturally poor. This also serves as a reminder of the importance of improving the quality of our initial re-ranking.

3.3 Best performing queries

Query 2004980 - pokemon x how to delete profile: For **Run_1**, the query with the largest relative exceedance over the median nDCG@10 provided by TREC was query number 2004980 (with a value of 0.6758 for **Run_1**, compared to the median of 0.6392). This was a relatively difficult query, as the important 'x' in the query might be identified as a meaningless character. The effectiveness in handling this query likely stems from the model's ability to interpret the 'x' in its specific context correctly. Given the nuanced nature of the query, which combines a popular culture reference (Pokémon) with a technical task (deleting a profile), the model must understand both elements and their interplay. This requires a broad knowledge base and the capacity to discern the correct interpretation of 'x' as a part of the query rather than a meaningless character. The success in this case indicates a sophisticated level of natural language understanding and contextual analysis by the model used in Run 1. This is because it reflects an advanced capability in parsing and accurately responding to queries involving unique combinations of terms, often leading to ambiguous interpretations. This outcome highlights

the importance of deep learning models in effectively managing the intricacies of user queries in information retrieval tasks.

Query 2001575 - FDA definition of verification: The success of Query 2001575 (with a value of 0.4690 for **Run_2**, compared to the median of 0.3960) in Run 2 is because of the robust performance of the GPT-4 model used in this run. GPT-4's architecture is designed to handle large-scale data, which is crucial for disentangling ambiguous queries and providing contextually relevant rankings. For a query like "FDA definition of verification," which requires precise and technical information retrieval, GPT-4's capability to interact effectively with tailored prompts and extract precise relevance signals for each query-passage pair would have been instrumental. This suggests that the model's prompt responsiveness and error-handling capabilities played a significant role in identifying and prioritizing authoritative sources specific to FDA regulations and terminologies.

Query 3100922 - What is the meaning and origin of the name Corrin: For Query 3100922 (with a value of 0.7288 for **Run_3**, compared to the median of 0.7151) in Run 3, the approach involved a multi-model ensemble strategy. This strategy, although exploratory, faced practical challenges due to the complexity of integrating multiple Large Language Models (LLMs) and the extensive requirement of generating a high number of pairwise prompts. These complexities may have diluted the ranking accuracy. Regarding a query about the origin and meaning of a name, this approach might have struggled with efficiently synthesizing diverse linguistic and cultural information. The challenge would have been effectively combining insights from various models to provide a comprehensive and accurate answer to a query requiring a nuanced understanding of etymology and cultural context.

4 DISCUSSION

Analyzing the results reveals nuanced insights into the efficacy of the employed models and strategies. **Run_2**, leveraging the GPT-4 model, exhibited superior performance across all metrics, underscoring the advancements in model architecture and training regimes. Specifically, GPT-4's larger parameter space and more refined training data likely contribute to a better understanding complex query contexts and relevance cues. This is reflected in the consistently higher nDCG scores, indicating a more accurate ranking of passages.

4.1 Analysis of **Run_2**'s Performance

GPT-4's robust performance suggests that its prompt responsiveness and error-handling capabilities are crucial for re-ranking tasks. The model's architecture, designed to process large-scale data, may enhance the ability to disentangle ambiguous queries and provide more contextually relevant rankings. Furthermore, the tailored prompt design for GPT-4 could have facilitated more effective interaction with the model, extracting precise relevance signals for each query-passage pair.

4.2 Reflections on **Run_1**

Despite its methodological similarity to **Run_2**, **Run_1**, which utilizes GPT-3.5-turbo, fell short in performance. This could be attributed to the model's constraints in capturing the full breadth of context within the sliding window parameters. The potential limitations of GPT-3.5-turbo in understanding deeper contextual nuances compared to its successor might have led to less accurate rankings.

4.3 Challenges in **Run_3**'s Approach

Run 3's multi-model ensemble strategy is an exploratory initiative that faces several practical challenges. The complexity of integrating multiple LLMs and the additional overhead from generating many pairwise prompts may have diluted the ranking accuracy. Variability in the foundational training and capabilities

4.4 Further Considerations for Model Improvement

To refine the models' performance, several considerations are proposed:

- Improving prompt design to align more closely with each model's capabilities, particularly for Run 3's LLM ensemble.
- Enhancing data representation and pre-processing to ensure that the models receive inputs in a format that maximizes their ranking abilities.
- Conducting a thorough error analysis to identify specific areas where models fail to correctly interpret the relevance, leading to informed adjustments in the models or strategies.
- Exploring the impact of query and passage complexity on model performance to tailor strategies for different information needs.

In conclusion, the results from **Run_2** indicate that the GPT-4 model holds a substantial advantage in passage re-ranking tasks. However, the balanced yet suboptimal outcomes of **Run_1** and **Run_3** highlight the need for ongoing refinement in model selection, prompt engineering, and ensemble strategies. Future work will dissect individual model performances, develop more sophisticated integration methods for LLM ensembles, and further customize prompt structures to leverage the unique strengths of each model.

5 CONCLUSION

This paper delineated our approach within the TREC 2023 Deep Learning Track, focusing on the intricate task of passage re-ranking by utilizing Large Language Models (LLMs). Our engagement has provided invaluable insights into the capabilities and limitations of cutting-edge LLMs in information retrieval. Our experimentation with various methodologies, ranging from single-model approaches such as GPT-3.5-turbo and GPT-4 to the multi-model ensembles in **Run_3**, has yielded rich outcomes. Notably, our **Run_2**, which leveraged the GPT-4 model, performed commendably across several metrics, achieving scores that surpassed the median of submitted runs. This success highlights the robust potential of LLMs in comprehending and ranking passages in response to complex queries. However, while these results are promising, they represent a single point in the vast landscape of information retrieval challenges.

The performance contrast between our LLM implementations suggests that factors such as each model's prompt design and intrinsic capabilities significantly influence retrieval effectiveness. Specifically, **Run_2**'s success indicates that the nuanced prompt interactions and the advanced architecture of GPT-4 have a marked impact on the quality of passage ranking. However, this does not imply a one-size-fits-all superiority of the latest LLM iteration.

Conversely, while innovative, our ensemble approach in **Run_3** did not yield the expected outcomes. This has prompted us to consider the complexities involved in effectively synthesizing the strengths of multiple models. Indeed, the influence of diverse training paradigms, prompt strategies, and model interoperability are areas that warrant further investigation.

Future work will focus on delving deeper into the interplay between LLMs and the dynamic requirements of multi-turn conversational information retrieval. We aim to refine our prompt engineering techniques and explore few-shot learning to bolster our models' performance. Specifically, by adjusting and fine-tuning these factors, we aim to elevate the precision and relevance of our retrieval systems, ensuring that they not only meet but exceed the standards set by the ever-evolving benchmarks of the TREC initiatives.

REFERENCES

[1] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. arXiv:2306.02561 [cs.CL]

[2] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[3] OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt Accessed: 2023-09-22.

[4] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[5] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. arXiv:2306.17563 [cs.IR]

[6] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *ArXiv* abs/2304.09542 (2023).