2/21/2026 5:55:31 PM

Compare Results

Old File:		New File:
2024.emnlp-main.431.pdf	versus	**2024_emnlp-main_.431.pdf**
12 pages (341 KB)		**9 pages (253 KB)**
10/31/2024 10:32:48 PM		2/21/2026 7:34:12 AM

Total Changes

750

Content

105 Replacements

34 Insertions

308 Deletions

Styling and Annotations

101 Styling

202 Annotations

Go to First Change (page 2)

GENRA: Enhancing Zero-shot Retrieval with Rank Aggregation

Georgios Katsimpras
Georgios Paliouras NCSR Demokritos, Athens, Greece gkatsibras,paliourg@iit.demokritos.gr

Abstract

Large Language Models (LLMs) have recently been employed to enhance zero-shot information retrieval by generating synthetic passages that are subsequently used to retrieve relevant documents. However, the generated passages often exhibit weak correlation with the original query, which can negatively affect retrieval performance. In this work, we propose GENRA, a framework that improves zero-shot retrieval by applying rank aggregation over multiple retrieval results derived from LLM-generated passages. GENRA incorporates relevance assessment mechanisms, including LLM-based filtering and re-ranking, to mitigate the impact of weakly correlated passages. Through extensive experiments on multiple benchmark datasets, we demonstrate that our approach consistently improves retrieval effectiveness compared to baseline zero-shot retrieval methods. Our findings highlight the importance of aggregation strategies and relevance filtering in leveraging LLM-generated content for robust information retrieval.

1 Introduction

Recent studies in zero-shot retrieval have demonstrated remarkable advancements, significantly improving the effectiveness of retrievers with the use of encoders like BERT (Devlin et al., 2018) and Contriever (Izacard et al., 2022). With the emergence of Large Language Models (LLMs) (Brown et al., 2020; Scao et al., 2022; Touvron et al., 2023), research focused on how to leverage LLMs for information retrieval tasks, such as zero-shot retrieval. Early zero-shot ranking with LLMs relied on methods that score each query-document pair and select the top-scoring pairs (Liang et al., 2022). Researchers have attempted to boost these methods by enriching contextual information to help LLMs understand the relationships between queries and documents. This often involves using LLMs to generate additional queries, passages, or other relevant content (Mackie et al., 2023; Li et al., 2023a). These enhancements have significantly improved retrieval performance, especially for unseen (zero-shot) queries.

In a typical retrieval setting, as shown in Figure 1a, queries and documents are embedded in a shared representation space to enable efficient search. The success of the entire approach depends strongly on the quality of the results of the retrieval step. However, LLMs can generate potentially non-factual or nonsensical content (e.g. “hallucinations”), and their performance is susceptible to factors like prompt order and input length, which can hurt the performance of the retriever (Yu et al., 2022b).

To address this problem, some studies (Liang et al., 2022; Thomas et al., 2023) propose employing LLMs as relevance assessors, providing individual relevance judgments for each query-document pair. These approaches aim to enhance trustworthiness by leveraging the LLM’s strengths in understanding nuances and identifying potentially irrelevant content. Additionally, recent work (Sun et al., 2023; Pradeep et al., 2023) suggests incorporating re-ranking models into the retrieval process. Such models process a ranked list of documents and directly produce a reordered ranking.

However, most existing methods focus solely on retrieval without a separate relevance assessment step, which could be beneficial. To address this gap, our approach utilizes rank aggregation techniques to combine individual rankings generated by separate retrieval and relevance assessment sub-processes. This allows our method to combine the strengths of the two stages, leading to a more refined and accurate final ranking of documents.

While combining multiple rankings (rank aggregation) has proven highly effective in various domains, like bio-informatics (Wang et al., 2022) and recommendation systems (Bałchanowski and Boryczka, 2023), its use with LLMs in zero-shot retrieval has not been explored thus far.

Our method (Figure 1b), named GENRA, first utilizes the LLM to generate informative passages that capture the query's intent. These passages serve as query variants, guiding the search for similar documents. Next, we leverage the LLM's capabilities to further refine the initial retrieval. This can be achieved through either direct relevance assessment (generating 'yes' or 'no' judgments) or by employing a re-ranking model to optimize the document order and select the top-ranked ones. This step acts as a verification filter, ensuring the candidate documents can address the given query. Using each verified document as a query, we retrieve new documents from the corpus, generating document-specific rankings that capture diverse facets of the query. By combining these individual rankings through a rank aggregation method, we mitigate potential biases inherent in any single ranking and achieve a more accurate final ranking.

Thus, the main contributions of the paper are the following:

- We propose a new pipeline for zero-shot retrieval, which is based on the synergy between LLMs and rank aggregation.
- We confirm through experimentation on several benchmark datasets the effectiveness of the proposed approach.

GENRA can be combined with different LLMs and different rank aggregation methodologies. However, the computational cost associated with these models can be significant, requiring substantial resources for both training and inference. Furthermore, relying on black-box models poses significant challenges.

2 Related Work

Zero-shot retrieval has gained significant attention in recent years, particularly with the advent of large pre-trained language models. Early approaches relied on dense retrieval methods that embed queries and documents into a shared vector space, enabling similarity-based matching without task-specific training (Karpukhin et al., 2020; Izacard et al., 2022). These methods demonstrated strong generalization capabilities across domains and tasks.

With the emergence of LLMs, researchers explored their potential in retrieval settings. Liang et al. (2022) proposed Promptagator, which leverages LLMs to generate synthetic queries for training dense retrievers in a zero-shot manner. Mackie et al. (2023) introduced HyDE, where LLM-generated hypothetical documents are used as intermediaries to improve retrieval performance. Similarly, Li et al. (2023a) investigated the use of generated passages to enhance retrieval quality.

Beyond generation-based approaches, LLMs have also been used directly for ranking and relevance estimation. Thomas et al. (2023) employed LLMs as relevance assessors, scoring query-document pairs without additional training. Sun et al. (2023) and Pradeep et al. (2023) incorporated re-ranking mechanisms that refine initial retrieval results by leveraging cross-encoder architectures or LLM-based scoring strategies.

Rank aggregation techniques have been extensively studied in information retrieval and related domains. Classical methods such as Borda Count and Reciprocal Rank Fusion (Cormack et al., 2009) combine multiple ranked lists to produce a more robust final ranking. These methods have demonstrated improvements in scenarios where individual rankers capture complementary aspects of relevance.

Despite the growing body of work on LLM-enhanced retrieval, the integration of rank aggregation with LLM-generated content in zero-shot retrieval remains underexplored. Our work addresses this gap by systematically combining generation, relevance assessment, and aggregation within a unified framework.

3 Preliminaries

Let $Q = q_1, q_2, \ldots, q_m$ be a set of queries and $D = d_1, d_2, \ldots, d_n$ be a corpus of documents. The goal of a retrieval system is, for each query $q \in Q$, to produce a ranked list of documents from D ordered by their estimated relevance to q.

In dense retrieval, both queries and documents are encoded into a shared embedding space through an encoder function $f(\cdot)$, typically based on a Transformer architecture. The relevance score between a query q and a document d is computed as the similarity between their embeddings: [s(q,d) = sim(f(q), f(d)),] where sim$(\cdot, \cdot)$ is often the dot product or cosine similarity.

Given a query q, the retrieval model produces a ranking π_q over the document set D, where $\pi_q(d)$ denotes the rank position of document d for query q.

In the context of rank aggregation, suppose that for a given query q, we obtain k different ranked lists $\pi_q^{(1)}, \pi_q^{(2)}, \ldots, \pi_q^{(k)}$, each generated by a different retrieval or assessment process. The goal of rank aggregation is to combine these individual rankings into a single consensus ranking π_q^* that better reflects the underlying relevance signal.

Aggregation methods assign a combined score to each document based on its positions across the individual rankings. For example, in Reciprocal Rank Fusion (RRF), the aggregated score of a document d is defined as: [RRF(d) = $\sum_{i=1}^{k} \frac{1}{c+\pi_q^{(i)}(d)},]wherecisaconstantthatcontrolstheinfluenceoflower-rankeddocuments.Documentsarethensortedindescendingorderoftheiraggregatedscorestoproducethefinalranki$

These formulations provide the foundation for the methodology introduced in the next section.

4 Methodology

Our approach, named GENRA, is designed to enhance zero-shot retrieval by combining LLM-based generation, relevance assessment, and rank aggregation into a unified framework. Figure 1b illustrates the overall pipeline. The method consists of three main components: passage generation, relevance assessment, and rank aggregation.

4.1 Passage Generation

Given a query q, we prompt the LLM to generate a set of synthetic passages that reflect the potential content of relevant documents. Let $P_q = p_1, p_2, \ldots, p_k$ denote the generated passages for query q. Each passage p_i serves as a query variant and is used to retrieve documents from the corpus.

For each generated passage p_i, we compute a ranking $\pi_q^{(i)}$ over the document set D using a dense retriever. This process produces multiple candidate lists that capture different semantic aspects of the query.

4.2 Relevance Assessment

To mitigate the impact of weakly correlated or noisy generated passages, we incorporate a relevance assessment stage. This stage operates on the top-ranked documents retrieved using the generated passages and aims to filter or reorder them based on their actual relevance to the original query q.

4.2.1 LLM-based filtering

In the LLM-based filtering approach, we prompt the LLM to provide a binary relevance judgment (e.g., "yes" or "no") for each query-document pair. Only documents judged as relevant are retained for subsequent processing. This step reduces the influence of irrelevant documents introduced by potentially hallucinated or weakly aligned passages.

4.2.2 Re-ranking

Alternatively, we employ a re-ranking model that takes the original query q and a set of candidate documents as input and produces a refined ranking. The re-ranker assigns relevance scores to each document, allowing us to reorder the list and select the most promising candidates. This mechanism acts as a verification stage, improving the reliability of the retrieved results.

4.3 Rank Aggregation

After relevance assessment, each verified document can be used as a new query to retrieve additional documents from the corpus. This process generates multiple document-specific rankings that capture complementary evidence regarding relevance.

Let $\pi_q^{(1)}, \pi_q^{(2)}, \ldots, \pi_q^{(k)}$ denote the set of rankings obtained from the various retrieval and assessment steps. We apply a rank aggregation method to combine these rankings into a final consensus ranking π_q^*. Aggregation mitigates biases inherent in any single ranking and enhances robustness by integrating multiple relevance signals.

The final output of GENRA is the aggregated ranking π_q^*, which is returned as the retrieval result for query q.

5 Results and Analysis

5.1 Setup

We evaluate GENRA on multiple benchmark datasets commonly used in zero-shot retrieval. The datasets span different domains and vary in size and difficulty, enabling a comprehensive assessment of the proposed framework. For all experiments, we use a pre-trained dense retriever as the base retrieval model and a large language model for passage generation and relevance assessment.

Retrieval effectiveness is measured using standard metrics, including Mean Reciprocal Rank (MRR) and Recall at various cutoffs. All models are evaluated in a zero-shot setting, without task-specific fine-tuning on the target datasets. Hyperparameters related to passage generation, number of retrieved documents, and aggregation methods are selected based on preliminary experiments.

5.2 Ablation Study

We conduct a series of ablation experiments to analyze the impact of each component of GENRA. Specifically, we examine the influence of the number of generated passages, the number of relevant documents used for expansion, and different aggregation strategies.

5.2.1 Number of Passages Generated

We vary the number of synthetic passages generated per query and observe the resulting retrieval performance. Increasing the number of generated passages generally improves performance up to a certain point, as it allows the system to capture diverse aspects of the query. However, beyond a threshold, additional passages may introduce noise, leading to diminishing returns.

5.2.2 Number of Relevant Documents

We also investigate the effect of the number of verified documents used for secondary retrieval. Using more relevant documents as expansion queries enhances diversity in the retrieved results, which in turn benefits rank aggregation. Nevertheless, selecting too many documents may incorporate marginally relevant or noisy signals.

5.2.3 Different Aggregations

We compare several rank aggregation methods, including Reciprocal Rank Fusion and Borda Count. The results indicate that aggregation consistently outperforms single-ranking baselines. Among the tested methods, Reciprocal Rank Fusion demonstrates robust performance across datasets.

5.2.4 Model Efficiency

We assess the computational cost of GENRA in terms of inference time and resource usage. The incorporation of LLM-based generation and relevance assessment increases computational overhead compared to standard dense retrieval. However, the performance gains justify the additional cost in scenarios where retrieval quality is critical.

5.3 Passage Ranking

To further understand the behavior of generated passages, we analyze their individual ranking effectiveness. We observe that while some passages yield high-quality rankings, others exhibit weaker correlation with the original query. This variability motivates the use of relevance filtering and aggregation to stabilize performance.

5.4 Summarizing Crisis Events

We apply GENRA to a dataset focused on crisis event summarization to evaluate its performance in a realistic downstream scenario. The results demonstrate that the aggregated rankings provide more comprehensive and accurate document sets for summarization compared to baseline retrieval approaches.

6 Conclusion

In this work, we introduced GENRA, a framework that enhances zero-shot retrieval by integrating LLM-based passage generation, relevance assessment, and rank aggregation. By combining multiple retrieval signals and mitigating the effects of weakly correlated generated passages, GENRA achieves consistent improvements over standard zero-shot retrieval baselines across several benchmark datasets.

Our experimental results highlight the importance of aggregation strategies and verification mechanisms when leveraging LLM-generated content for retrieval tasks. The proposed framework is flexible and can be adapted to different LLMs, retrievers, and aggregation methods.

Future work may explore more efficient relevance assessment techniques and investigate alternative aggregation schemes to further improve scalability and effectiveness in large-scale retrieval settings.

Acknowledgements

This research was co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: [ILLEGIBLE]).

Limitations

The proposed framework relies heavily on large language models for passage generation and relevance assessment. As a result, its performance is influenced by the quality, biases, and limitations of the underlying LLMs. Generated passages may contain hallucinated or misleading information, which can negatively affect retrieval despite the incorporation of filtering and aggregation mechanisms.

Moreover, GENRA introduces additional computational overhead compared to standard dense retrieval pipelines. The generation of multiple passages, the relevance assessment step, and the aggregation of several rankings increase inference time and resource requirements. This may limit the applicability of the approach in real-time or large-scale production environments.

Finally, the reliance on black-box LLMs restricts transparency and interpretability. Access to model internals is often limited, and reproducibility may depend on specific model versions or API settings. These factors should be considered when deploying the framework in practical scenarios.

References

References

[1] Bałchanowski, K. and Boryczka, U. 2023. Rank aggregation in recommendation systems. [ILLEGIBLE].

[2] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. 2020. Language models are few-shot learners. Advances in Neural Information Processing Systems, 33:1877–1901.

[3] Cormack, G. V., Clarke, C. L. A., and Büttcher, S. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval.

[4] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT.

[5] Izacard, G., Caron, M., Hosseini, L., Riedel, S., and Grave, E. 2022. Unsupervised dense information retrieval with contrastive learning. Transactions of the Association for Computational Linguistics.

[6] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., et al. 2020. Dense passage retrieval for open-domain question answering. Proceedings of EMNLP.

[7] Li, X., Li, P., and Gaussier, É. 2023a. Enhancing zero-shot retrieval with generated passages. [ILLEGIBLE].

[8] Liang, P., Zhao, T., Yu, M., and Chen, D. 2022. Promptagator: Few-shot dense retrieval from 8 examples. Proceedings of ACL.

[9] Mackie, I., Huang, Y., and Nogueira, R. 2023. Hypothetical document embeddings for zero-shot retrieval. Proceedings of ACL.

[10] Pradeep, R., Nogueira, R., and Lin, J. 2023. MonoT5: Re-ranking with a T5-based model. Proceedings of EMNLP.

[11] Scao, T. L., et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. [ILLEGIBLE].

[12] Sun, Y., Yu, Z., and Liu, Y. 2023. Improving retrieval with LLM-based re-ranking. [ILLEGIBLE].

[13] Thomas, P., Gupta, S., and Xiong, C. 2023. Large language models as relevance assessors. Proceedings of SIGIR.

[14] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. 2023. LLaMA: Open and efficient foundation language models. [ILLEGIBLE].

[15] Wang, L., Zhang, X., and Li, J. 2022. Rank aggregation in bioinformatics applications. [ILLEGIBLE].

[16] Yu, W., et al. 2022b. Hallucinations in large language models. [ILLEGIBLE].

A Appendix

GENRA Algorithm

Algorithm 1 presents the detailed steps of the proposed GENRA framework. The algorithm describes the process of passage generation, relevance assessment, secondary retrieval, and rank aggregation for a given query.

[h] GENRA Framework [1] Query q, document corpus D, retriever R, LLM M Generate k synthetic passages $P_q = p_1, \ldots, p_k$ using M each passage $p_i \in P_q$ Retrieve top-n documents $\pi_q^{(i)}$ from D using R Apply relevance assessment on retrieved documents Select verified documents $\hat{D}_q$ each document $d \in \hat{D}_q$ Retrieve additional documents using d as query Aggregate all rankings using rank aggregation method **return** final ranking π_q

Datasets Statistics

Table **??** reports statistics of the datasets used in our experiments, including the number of queries and documents.

Dataset	Queries	Documents	Domain	height[ILLEGIB
[ILLEGIBLE]	[ILLEGIBLE]	[ILLEGIBLE] [ILLEGIBLE]		[ILLEGIBLE]
[ILLEGIBLE]	[ILLEGIBLE] [ILLEGIBLE]	[ILLEGIBLE]		[ILLEGIBLE]
[ILLEGIBLE] height				

Table 1: Dataset statistics.