Two Heads Are Better Than One: Improving Search Effectiveness Through LLM-Generated Query Variants

Kun Ran
RMIT University
Melbourne, Australia
kun.ran@student.rmit.edu.au

Marwah Alaofi
RMIT University
Melbourne, Australia
Marwah.alaofi@student.rmit.edu.au

Mark Sanderson
RMIT University
Melbourne, Australia
mark.sanderson@rmit.edu.au

Damiano Spina
RMIT University
Melbourne, Australia
damiano.spina@rmit.edu.au

Abstract

User query quality impacts retrieval effectiveness. This study categorizes thousands of user queries spanning one hundred topics into three groups – low, medium, and high quality – based on NDCG@10 scores. The study investigates the impact of fusing search results of Large Language Model (LLM)-generated query variants with results retrieved from user queries drawn from the three groups, similar to a collaborative search approach where users with diverse queries collaborate in locating relevant information. The findings indicate that a traditional search system can be significantly improved by fusing results for low-quality queries, offering a promising solution for users who struggle to find relevant information, particularly in contexts where advanced search systems are impractical due to technical or resource constraints, or where access to query logs are unavailable.

CCS Concepts

• **Information systems** → **Users and interactive retrieval**; *Information retrieval query processing*; Collaborative search; Combination, fusion and federated search.

Keywords

Query Variants, Large Language Models, Ranking Fusion

ACM Reference Format:
Kun Ran, Marwah Alaofi, Mark Sanderson, and Damiano Spina. 2025. Two Heads Are Better Than One: Improving Search Effectiveness Through LLM-Generated Query Variants. In *2025 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '25), March 24–28, 2025, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3698204.3716468

1 Introduction

Previous research in collaborative search – where users work together towards a shared information goal – has demonstrated the potential for different users to achieve search outcomes that surpass individual efforts [25]. One such approach involves users contributing different queries for the same information need, which we refer to as *query variants*. Combining these query variants was first empirically investigated by Belkin et al. [6], who found that using different representations of the information need improved retrieval. Further evidence suggests that this collaborative approach can enhance retrieval effectiveness, yielding results superior to at least the median system's performance on difficult topics [14]. However, this concept has not been extensively explored, likely due to challenges in accessing suitable query variants.

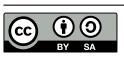
With advancements in LLMs capable of generating diverse query variants [2], this paper revisits collaborative search: rather than relying on multiple users to generate query variants, we use LLMs to generate variants. We examine whether this 'collaborative approach' with LLMs can enhance retrieval effectiveness.

While search engines are designed to meet user information needs, their effectiveness has traditionally been measured by averaging performance across topics, each represented by a single query. This one-query-per-topic model limits our understanding of how well different users' needs are met. Many challenge this abstraction [1, 4, 9, 26]. Most recently, Diaz [11] critiques this traditional focus on average utility in information access evaluation, advocating instead for a pessimistic approach that emphasizes worst-case scenarios. This alternative perspective better aligns with ethical standards and supports principles of social good and equal information access [12].

Given the potential of collaborative search using query variants to improve retrieval and in light of pessimistic evaluation where we go beyond the average utility of the system, this study distinguishes itself by examining the impact of incorporating query variants into a retrieval pipeline and evaluating across three groups of query quality: low, medium, and high. The groups represent distinct user categories, allowing us to explore the effectiveness of collaborative efforts for each group.

The study explores the effectiveness of 'aggregating' search results obtained from LLM-generated Variants (LLMVs) with those obtained from queries of different groups through *search ranking fusion* in an ad hoc document retrieval context. To achieve this, we investigate the following research questions:

RQ1 Does ranking fusion of search results of human queries with search results of *human-generated query variants* improve retrieval effectiveness?



CHIIR '25, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-1290-6/25/03
https://doi.org/10.1145/3698204.3716468

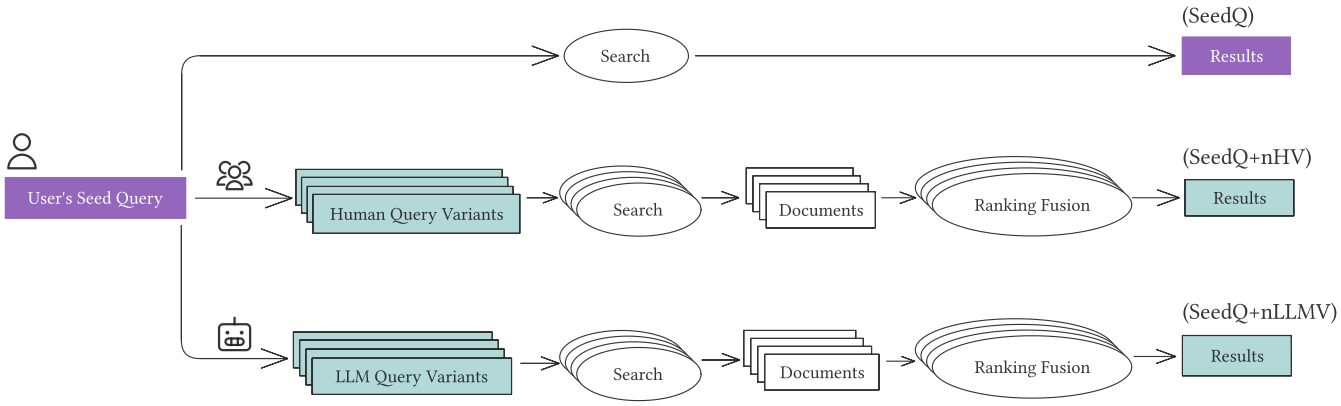


Figure 1: The three retrieval pipelines: using the seed query, and ranking fusion with human and LLM variants.

RQ2 Does ranking fusion of search results of human queries with search results of *LLM-generated query variants* improve retrieval effectiveness?

We address **RQ1** by using human-generated query variants as the gold standard for collaborative search, assuming this would represent the optimal setting for human collaboration. **RQ2** investigates LLM-generated variants as an automated method for producing these query variations.

2 Related Work

Improving retrieval effectiveness has been the subject of extensive research in the field of Information Retrieval (IR). In this section, we review the most relevant studies related to improving retrieval effectiveness through the use of query variants, fusion methods, generative query expansion, and the integration of LLMs in IR.

Query variants, which are different formulations of the same information need, have been studied from various perspectives, with particular emphasis on their role in system evaluation [2, 4, 5, 9, 15, 20]. In 2016, Bailey et al. [4] introduced the UQV100 dataset, which consists of 100 topics with query variants obtained from crowdworkers to reflect diverse query formulations. Building on this, Bailey et al. [5] examined the retrieval consistency of different systems when responding to these variants, demonstrating the impact of variants on performance. More recently, Alaofi et al. [2] explored the use of LLMs to generate query variants and assessed their similarity to human-generated versions, showcasing the potential of LLMs to create query variants for test collections.

The use of 'variants' as a means to improve retrieval effectiveness, rather than solely as a tool to evaluate system consistency, has been relatively limited, with most related work emerging from collaborative search studies (e.g., [6, 14]). Recently, Breuer [7] introduced the approach of ranking fusion using LLM-generated query variants from query descriptions and narratives.

Recently, LLMs have been widely used in IR online and offline tasks. Online tasks include generating pseudo-relevance feedback [10, 30], query reformulations [15, 24, 30], and expansions [17, 29]. Offline tasks involve assessing document relevance [13, 27], generating query variants [2], creating synthetic documents [3], and even constructing full synthetic test collections [21]. While [2] demonstrate that LLM can produce human-like query variants, those variants were not tested for their potential to improve effectiveness.

Using query variants requires a method to aggregate the search results retrieved for each variant; one such approach could be *search ranking fusion*, where results from different variants are combined into a single result list. Fusion can be achieved by either using only the ranking positions (e.g., Reciprocal Rank Fusion (RRF), Borda-fuse, Condorcet-fuse, RBC) [5, 8] or by also integrating the relevance score returned from the retrieval systems (e.g., CombMNZ, CombANZ, CombSUM) [28]. The RRF algorithm proposed in Cormack et al. [8] merges multiple ranked lists by assigning higher scores to documents that appear higher in the respective ranked list, regardless of their relevance score in the original ranked list.

This study investigates the use of LLMs to generate query variants aimed at bringing together different representations of the information need to assist in finding relevant information.

3 Experimental Setup

We describe the dataset, the approach used to group query variants, our retrieval pipelines, the LLMs used to generate query variants and the evaluation measures.

3.1 Dataset and Query Variants Grouping

The study uses the UQV100 test collection by Bailey et al. [4], which collected query variants by prompting crowdworkers to generate queries in response to one hundred backstories. Backstories were manually written to represent information needs selected from the TREC 2013 and 2014 Web Tracks. Documents were pooled from the ClueWeb12 corpus [1] in response to the collected variants and by using five retrieval systems. The ClueWeb12 corpus contains 733,019,372 English web pages, collected between February 10, 2012 and May 10, 2012. Document relevance was judged on a scale of 0–4, given the backstories. There are 55,587 relevance judgments in total. On average, a topic contains 57.65 unique query variants, 269.31 judged documents and 76.27 relevant documents. The number of query variants, judged documents and relevant documents in

[1] https://lemurproject.org/clueweb12/

Table 1: An example topic (UQV100.053) with three seed queries representing each query quality group. Seed query, three human variants and three LLM variants. The topic is 'tooth abscess', and the backstory provided to crowdworkers is 'A tooth at the back of your jaw is giving you a lot of pain - you think it might be an abscess. What treatments are available for it?'

Quality Group	SeedQ	Human Variants	LLM Variants
High	treatment for tooth abscess	• treatment for the tooth pain • treatments for a abscess tooth • tooth pain medicine	• How to treat a tooth abscess? • Remedies for dental abscess pain • Best cure for tooth infection
Medium	abscess pain treatment	• treatments available for abscess • symptoms of tooth abscess • treatment for jaw pain	• How to relieve abscess pain? • Best treatment for abscess • Abscess pain relief methods
Low	treatments for tooth pain	• tooth abscess • treatments of tooth abscess • abscess pain treatment	• Remedies for dental pain • How to relieve toothache • Tooth pain treatment at home

Table 2: Mean, minimum, and maximum number of query variants and documents per topic.

	Mean	Min	Max
Query Variants	57.65	19	101
Judged Documents	269.31	35	760
Relevant Documents	76.27	7	253

Table 3: Mean, minimum, and maximum number of query variants for each query quality group across topics.

Query Quality Group	Mean	Min	Max
Low	22.06	2	82
Medium	19.66	0	54
High	15.93	1	52

the dataset varies substantially across topics, as shown in Table 2. Consistent with the UQV100 test collection's methodology, we also use ClueWeb12 as the retrieval corpus for all query pipelines in this work.

The query variants for each topic were created by different users and demonstrate substantial variation, which is reflected in their retrieval effectiveness scores. Within each topic, we split all the query variants into three equal-width groups based on their query performance as measured with NDCG@10 score: (1) Low-Quality Group, (2) Medium-Quality Group, and (3) High-Quality Group as shown in Table 3. For example, if a topic has a number of variants with performance scores ranging from 0.1 to 0.7, we assign all variants ranging from 0.1 to 0.3 to the Low-Quality Group, 0.3 to 0.5 to the Medium-Quality Group and 0.5 to 0.7 to the High-Quality Group.

Within each quality group, we select each query as the seed query and examine the retrieval effectiveness and coverage using different search pipelines (as described in Section 3.2). This analysis is conducted for all query variants across all topics. In this context, a *seed query* refers to a query variant used as a starting point for a search pipeline, while *human variants* refers to query variants selected as human variants for the seed query. Details of the retrieval pipelines are provided in the following section.

3.2 Retrieval Pipelines

Figure 1 provides an overview of the three types of search pipelines: the seed query alone, fusion with human-generated variants from the UQV100 as our ideal collaboration, and fusion with LLM-generated variants.

All pipelines begin with the same seed query selected to represent the three query groups previously described. In the seed query alone pipeline (SeedQ), we search using only the seed query and retrieve documents directly. In the ranking fusion pipelines, we use the seed query along with multiple query variants, searching each variant individually. Once documents are retrieved for each variant, we fuse the rankings into a single search results ranking. We then evaluate retrieval effectiveness against the query relevance judgments (Qrels) provided in the UQV100 test collection. For human-generated variants (SeedQ+nHV), we randomly select variants from the entire topic, not limited to the group of the seed query. For LLM-generated variants (SeedQ+nLLMV), we use LLMs to generate a list of query variants in response to the seed query. In particular, we experiment with nine different pipelines:

(1) **Upper and lower bound performance**
   - **Worst Query (WorstQ):** Lowest-performing query averaged across topics, serving as a lower bound for seed query effectiveness.
   - **Best Query (BestQ):** Highest-performing query averaged across topics, serving as an upper bound for seed query effectiveness.

(2) **Baseline**
   - **Seed Query (SeedQ):** A seed query selected from each quality group for all topics.

(3) **Search ranking fusion**

- **SeedQ + n Human-generated variants ($n = 1, 5, 15$):** Fusing results of the seed query with results of human-generated variants to address **RQ1**.
- **SeedQ + n LLM-generated variants ($n = 1, 5, 15$):** Fusing results of the seed query with results of LLM-generated variants to address **RQ2**.

We use Okapi's BM25 [22] as the search retrieval model across all pipelines as implemented in Pyserini [18], with default parameters of $k1 = 0.9$ and $b = 0.4$. In the ranking fusion pipelines, we use Reciprocal Rank Fusion [8] as implemented in TrecTools [19] with the default parameter of $k = 60$ to combine the multiple document lists into a single ranking.

3.3 LLM Variants Generation

We employ three major LLMs to generate query variants: GPT-4o (2024-08-06) from OpenAI,[2] which is a closed source commercial model, and Llama 3.1 Instruct (both 8b and 70b versions) from Meta,[3] which are open source LLMs. The prompt is adapted from the one used by Alaofi et al. [2] with slight modification to control the output format. Based on their results, we also run all LLMs at a temperature = 0.5. However, since the LLMs used in this study are different, the prompt and temperature settings are not directly comparable, and there may be ways to improve the generation of query variants, which we leave for future work. The prompts we used in our experiments are provided in Appendix A.

For each seed query, we generate a list of 20 LLM query variants. When selecting query variants for a pipeline, we randomly sample n variants from this list. To illustrate that, Table 1 provides examples of seed queries, randomly selected human query variants, and LLM query variants for an example topic (UQV100.053) in the UQV100 test collection.

3.4 Evaluation

To address our research questions, we measure search effectiveness and coverage for each pipeline. Since we evaluate pipelines at the individual seed query level and the number of query variants per topic varies substantially (Table 2), we apply a two-stage macro-averaging approach: first averaging measures across all query variants within the query quality group, then averaging across topics within each quality group.

For effectiveness measures, we use NDCG@k ($k = 5, 10, 20, 30, 100, 200, 500, 1000$), measured at multiple cutoffs k to evaluate pipeline performance across different user engagement levels. Some tasks specifically prioritize coverage over ranking quality, such as pooling candidate documents for a downstream re-ranking model. For these tasks, we use Recall@10 to measure coverage. We chose Tukey's HSD test to determine statistical significance for its suitability to reject Type I errors when performing multiple comparisons across retrieval pipelines. The level of significance is set to $\alpha = 0.05$ throughout our tests [23].

We observe that the LLM query variants retrieve a large portion of documents not present in the UQV100 Qrels. Consequently, we also report the *residuals* [24] (i.e., the effectiveness score obtained when the unjudged retrieved documents are considered maximally relevant) for Discounted Cumulative Gain [16] at cutoff $k = 10$ (DCG@10) and Rank-Biased Precision (RBP) effectiveness measures.[4] For RBP, we set the patience parameter to $p = 0.9$ which is the equivalent to cutoff $k = 10$.

4 Results and Discussion

We ran the retrieval pipelines on all 5,765 queries across 100 topics in the UQV100 test collection and report the results in this section.

4.1 Ranking Fusion with Human-Generated Query Variants

RQ1 Does ranking fusion of search results of human queries with search results of *human-generated query variants* improve retrieval effectiveness?

Figure 2 illustrates the effect of ranking fusion using human variants compared to seed queries. For all queries, using a seed query with multiple human variants fusion consistently shows better retrieval effectiveness compared to the seed query alone. As the number of variants used in fusion increases, the retrieval effectiveness also increases. Specifically, using the seed query with five human variants achieves statistically significantly higher effectiveness w.r.t. NDCG than using it with one human variant or the seed query alone. However, fusing with 15 human variants yields similar results to just using five, suggesting more than five human variants provide only marginal improvement. As shown in Figure 3, the residual observed when using 15 human variants is lower than the one with five human variants. This indicates that 15 human variants did not retrieve more unjudged documents, the improvement is indeed marginal and not statistically significant.

The effectiveness of the best query initially starts high and gradually decreases. Beyond a depth of 100, however, it begins to increase again, likely due to the limited number of relevant documents judged in the UQV100 dataset (Table 2). Therefore, when interpreting NDCG@k, it is important to consider the limitations of the UQV100 dataset and focus on the depth where $k < 100$. The single best query achieves higher effectiveness than all other configurations, representing the upper bound of what an optimal query can achieve.

For the low-quality query group, we see statistically significant improvements in using fusion on multiple variants compared to using seed query alone. For the medium-quality group, we see statistically significant improvement using seed query with five human variants fusion compared to seed query only. Again, fusion with 15 human variants did not obtain significant improvement compared to five human variants. For the high-quality group, we see an effectiveness drop if we use fusion for shallow depth ($k < 30$).

These results show that, given a seed query, generating high-quality and human-like query variants can improve retrieval effectiveness, particularly for low- and medium-quality seed queries. However, for high-quality seed queries, fusion with query variants tends to decrease effectiveness. These findings establish the basis against which the usefulness of ranking fusion with LLM variants can be assessed.

[2] https://platform.openai.com/docs/models/gpt-4o
[3] https://ai.meta.com/blog/meta-llama-3-1/

[4] We opt for DCG@10 over NDCG@10 to avoid normalization issues, allowing direct observation of absolute gains; this is because the normalization factor *IDCG* becomes non-indicative after treating unjudged documents as maximally relevant in residuals.

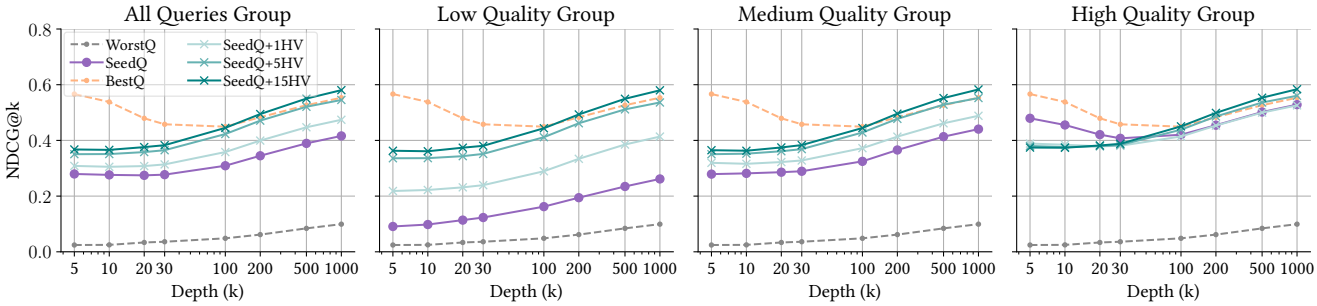


Figure 2: Effectiveness of combining human variants across query quality groups. Effectiveness is measured using NDCG@k at different cutoffs k. The best query (BestQ) line shows the top-performing single query for the topic, while the worst query (WorstQ) represents the lowest-performing one (both are shown in all plots). Within each query quality group, lines for $n = 1, 5, 15$ queries illustrate the performance when fusing results from the selected seed query from the group (SeedQ) and multiple human query variants from the topic (SeedQ+nHV).

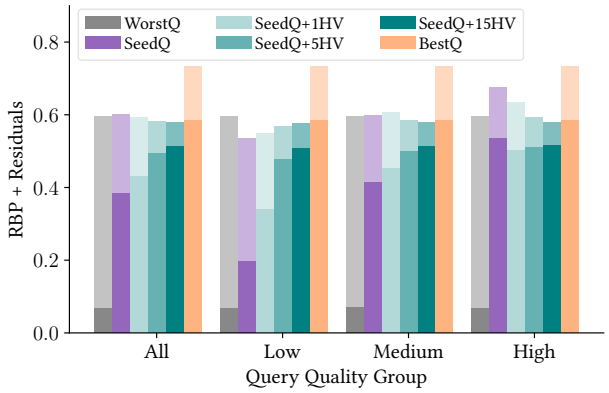


Figure 3: RBP and residuals scores for all queries and different query quality groups when using human query variants for fusion.

4.2 Fusion with LLM-Generated Query Variants

RQ2 Does ranking fusion of search results from human queries with search results of *LLM-generated query variants* improve retrieval effectiveness?

Figure 4 shows the effectiveness of fusion using seed query and LLM (Llama 3.1 Instruct 70b) generated query variants compared to other queries from the human set. The effectiveness of query variants from other LLMs is detailed in Section 4.3. For all queries, there is no statistically significant improvement for fusion using query variants generated by LLM compared to seed query at the depth of $k = 10$.

However, for seed queries from the low-quality group, we see **statistically significant improvement with fusion pipelines** using Seed Query + 5 LLM Variants (SeedQ+5LLMV) compared to SeedQ, specifically, at depth $k = 10$, Seed Query + 15 LLM Variants (SeedQ+15LLMV) achieves an effectiveness score of 0.20 compared to 0.10 for the seed query. For the medium-quality group, the improvement is not statistically significant, and for the high-quality group, the effectiveness is reduced when using fusion. This indicates that using LLM variants significantly improves the low-quality queries; improvement, however, has not reached the level observed when using human variants.

We plot the residuals graph for ranking fusion using seed query and LLM variants in Figure 5. Due to the way UQV100 document pooling works, it only covers part of the relevant documents [4]. With LLM variants, we are retrieving more unjudged documents compared to human queries. We can also see a larger amount of residuals from the SeedQ+nLLMV pipelines across all query quality groups, higher than the SeedQ from the corresponding group. For example, in the All Queries Group, the SeedQ+15LLMV shows almost equal RBP compared to SeedQ, however, it has more residuals, if we consider the unjudged documents as relevant for both SeedQ and SeedQ+15LLMV, the fusion will have a larger improvement.

4.3 Different Models and Parameters

Table 4 reports the average retrieval metrics measured for different query quality groups and fusion configurations. Similar effectiveness is achieved by different LLMs, regardless of their size or whether they are closed- or open-source. Within the All Queries Group, only Llama 3.1 70b demonstrates a statistically significant improvement over the seed query, and even this improvement is marginal and was not consistently reproduced across other models. Overall, the results suggest that the LLM variants generally do not outperform the seed query.

The human variants fusion is much more effective compared to seed query and all LLM variants, but as shown in Figure 3 and Figure 5, the SeedQ+15LLMV fusion has more residuals than the Seed Query + 15 Human Variants (SeedQ+15HV) fusion. It is uncertain what effectiveness we will have for SeedQ+15LLMV if we have all those unjudged documents judged, we can only ascertain that if all unjudged documents are deemed relevant, the SeedQ+15 LLM variants (Llama 3.1 70b) fusion will achieve a higher RBP than the SeedQ+15HV in the all queries group (Table 4).

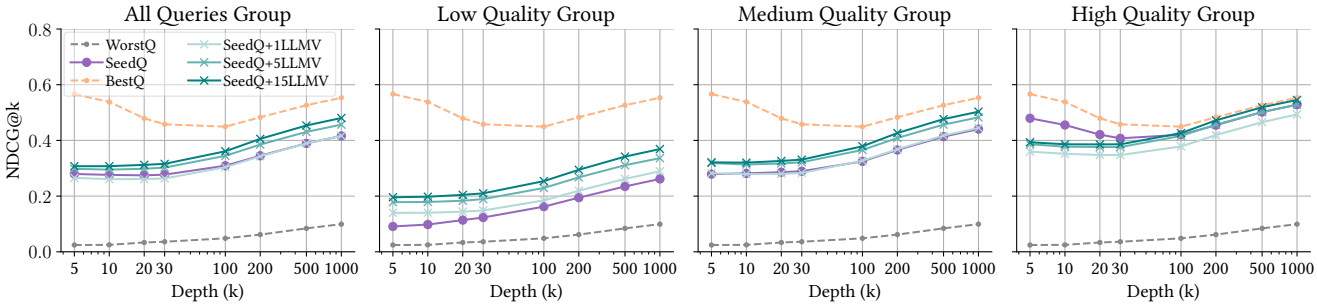


Figure 4: Effectiveness of fusing LLM variants for all queries and the three query quality groups. Effectiveness is measured using NDCG@k at different cutoffs k. The best query (BestQ) line shows the top-performing single query for the topic, while the worst query (WorstQ) represents the lowest-performing one (both are shown in all plots). Within each query quality group, lines for $n = 1, 5, 15$ queries illustrate the performance when fusing results from the selected seed query from the group (SeedQ) and multiple query variants generated by the LLM (SeedQ+nLLMV).

Table 4: Average retrieval measures. For each group, we compare the measures between SeedQ and fusion methods and use Tukey's HSD ($p < 0.05$) to indicate statistical significance. We annotate with a indicating statistically significant difference on seed query, b against variants generated by Llama 3.1 8b, c against variants generated by Llama 3.1 70b, d against variants generated by GPT-4o. For DCG@10 and RBP, we include the values before and after applying the residuals.

Retrieval Configuration	NDCG@10	Recall@10	DCG@10		RBP	
Worst Query	0.02	0.01	0.32	+7.94	0.07	+0.53
Best Query	0.54	0.13	6.89	+0.96	0.58	+0.15
All Queries Group						
SeedQ	0.28	0.08	3.57	+2.51	0.38	+0.22
SeedQ+15 LLM Variants (Llama 3.1 8b)	0.29	0.08	3.69	$+3.76^{a}$	0.39	$+0.28^{a}$
SeedQ+15 LLM Variants (Llama 3.1 70b)	0.31^{a}	0.09	3.90	$+3.24^{a}$	0.42^{a}	$+0.24^{a}$
SeedQ+15 LLM Variants (GPT-4o)	0.30	0.09	3.83	$+3.24^{a}$	0.41	$+0.25^{a}$
SeedQ+15 Human Variants	0.37^{abcd}	0.10^{abcd}	4.75^{abcd}	$+0.83^{bc}$	0.51^{abcd}	$+0.07^{bcd}$
Low Quality Group						
SeedQ	0.10	0.04	1.27	+4.59	0.20	+0.34
SeedQ+15 LLM Variants (Llama 3.1 8b)	0.18^{a}	0.06^{a}	2.36^{a}	$+5.90^{a}$	0.28^{a}	$+0.39^{a}$
SeedQ+15 LLM Variants (Llama 3.1 70b)	0.20^{a}	0.06^{a}	2.54^{a}	$+5.21^{a}$	0.30^{a}	$+0.35^{a}$
SeedQ+15 LLM Variants (GPT-4o)	0.19^{a}	0.06^{a}	2.43^{a}	$+5.36^{a}$	0.28^{a}	$+0.36^{a}$
SeedQ+15 Human Variants	0.36^{abcd}	0.13^{abcd}	4.69^{abcd}	$+0.83^{a}$	0.51^{abcd}	$+0.07^{bcd}$
Medium Quality Group						
SeedQ	0.28	0.09	3.62	+1.90	0.41	+0.18
SeedQ+15 LLM Variants (Llama 3.1 8b)	0.31	0.09	3.96	$+2.82^{a}$	0.42	$+0.23^{a}$
SeedQ+15 LLM Variants (Llama 3.1 70b)	0.32	0.09	4.13	$+2.58^{a}$	0.44	$+0.21^{a}$
SeedQ+15 LLM Variants (GPT-4o)	0.32	0.09	4.16	$+2.32^{a}$	0.44	+0.20
SeedQ+15 Human Variants	0.36^{ab}	0.10	4.71^{ab}	+0.79	0.51^{abcd}	$+0.07^{bcd}$
High Quality Group						
SeedQ	0.46	0.12	5.83	+1.10	0.54	+0.14
SeedQ+15 LLM Variants (Llama 3.1 8b)	0.36^{a}	0.10	4.67^{a}	+2.78	0.47^{a}	+0.22
SeedQ+15 LLM Variants (Llama 3.1 70b)	0.39^{a}	0.11	4.95^{a}	+2.12	0.50	+0.18
SeedQ+15 LLM Variants (GPT-4o)	0.38^{a}	0.11	4.85^{a}	+2.23	0.49	+0.19
SeedQ+15 Human Variants	0.37^{a}	0.10	4.84^{a}	$+0.86^{abc}$	0.51	$+0.06^{abcd}$

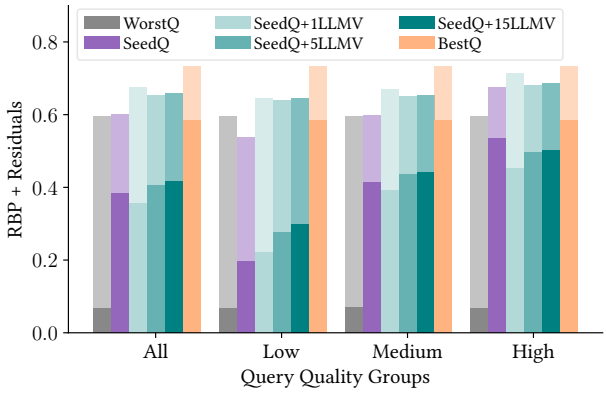


Figure 5: RBP and residuals scores for all queries and different query quality groups when using LLM variants for fusion.

For the low-quality query group, we see substantial effectiveness improvements with fusion using LLM-generated variants compared to seed query. Recall seems to be improving as well, but not statistically significant. Among all the variants configurations in this group, human query variants achieve far superior effectiveness and is statistically significantly more effective than SeedQ and all LLM variants configurations. For the medium-quality group, no statistically significant improvements are observed with LLM variants compared to SeedQ, and for the high-quality group, the effectiveness of LLM variants decreases compared to SeedQ.

5 Conclusion and Future Work

This study experimented with generating query variants using LLMs and applied ranking fusion to combine multiple document lists into a single ranked list, simulating a collaborative search among humans sharing the same information need. We examined the retrieval effectiveness across different query quality levels and found that this approach of fusion using LLM-generated query variants is most effective for low-quality queries, with limited improvement for medium- and high-quality queries. This finding provides valuable insight for search engine applications seeking to improve search effectiveness for under-performing queries. We used the UQV100 test collection for our experiments – to our knowledge, the most comprehensive dataset of queries and query variants available to date.

Regarding **RQ1**, our results indicate that ranking fusion of search results from human queries with those from human-generated query variants can significantly improve retrieval effectiveness, particularly for low- and medium-quality queries. For **RQ2**, we observed that incorporating search results from LLM-generated query variants enhances effectiveness, although the degree of improvement varies based on the quality of the initial user query.

Our findings align with those of Breuer [7], who demonstrated the effects of ranking fusion on LLM-generated query variants. They observed a decline in retrieval effectiveness for certain topics, which we hypothesize correspond to topics with medium- or high-quality human queries, where improvements are inherently more challenging. However, due to the absence of human query variants in their datasets, this hypothesis could not be verified. Additionally, we explored smaller models, such as Llama 3.1 Instruct 70b, and observed performance comparable to GPT-4o. This approach enables faster query generation and significantly reduces computational costs, making it a more efficient and cost-effective alternative.

One limitation of our approach is that the human query variants were created with a full backstory prompt, while the LLM variants were derived solely from the seed query, simulating a practical scenario where access to the backstory (or user intent) is not available (e.g., query logs). This, of course, makes the human-generated query variants more effective and more challenging for LLMs to replicate, leaving room for improvement in future work.

One question that arises is where the observed improvements for low-quality queries using LLM-generated query variants stem from. Preliminary analysis suggests that LLM-generated query variants expand the original search query through the inclusion of synonyms, related terms, and the correction of typos.

To create more effective query variants that can enhance retrieval, further research is needed to understand users' underlying information needs. Additionally, exploring and comparing different methods for integrating query variants into the pipeline – such as through query expansion and reformulation – could improve effectiveness. However, we leave such exploration for future work.

Acknowledgments

This research was conducted by the ARC Centre of Excellence for Automated Decision-Making and Society (ADM+S, CE200100005), and funded fully by the Australian Government through the Australian Research Council. The authors acknowledge the Woi wurrung and Boon wurrung language groups of the eastern Kulin Nation on whose unceded lands ACM SIGIR CHIIR 2025 was hosted. We pay our respect to their Elders past and present and extend that respect to all Aboriginal and Torres Strait Islander peoples today and their connections to land, sea, sky, and community. The authors thank RMIT AWS Cloud Supercomputing Hub (RACE) for their support in accessing LLMs. The authors also thank anonymous reviewers for their valuable feedback on this work.

References

[1] Marwah Alaofi, Luke Gallagher, Dana Mckay, Lauren L. Saling, Mark Sanderson, Falk Scholer, Damiano Spina, and Ryen W. White. 2022. Where Do Queries Come From?. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2850–2862. doi:10.1145/3477495.3531711

[2] Marwah Alaofi, Luke Gallagher, Mark Sanderson, Falk Scholer, and Paul Thomas. 2023. Can Generative LLMs Create Query Variants for Test Collections? An Exploratory Study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 1869–1873. doi:10.1145/3539618.3591960

[3] Arian Askari, Mohammad Aliannejadi, Evangelos Kanoulas, and Suzan Verberne. 2023. A Test Collection of Synthetic Documents for Training Rankers: ChatGPT vs. Human Experts. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 5311–5315. doi:10.1145/3583780.3615111

[4] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proceedings of the 39th International*

ACM SIGIR Conference on Research and Development in Information Retrieval (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 725–728. doi:10.1145/2911451.2914671

[5] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 395–404. doi:10.1145/3077136.3080839

[6] Nicholas J. Belkin, C. Cool, W. Bruce Croft, and James P. Callan. 1993. The effect multiple query representations on information retrieval system performance. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pittsburgh, Pennsylvania, USA) *(SIGIR '93)*. Association for Computing Machinery, New York, NY, USA, 339–346. doi:10.1145/160688.160760

[7] Timo Breuer. 2024. Data Fusion of Synthetic Query Variants With Generative Large Language Models. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region* (Tokyo, Japan) *(SIGIR-AP 2024)*. Association for Computing Machinery, New York, NY, USA, 274–279. doi:10.1145/3673791.3698423

[8] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Boston, MA, USA) *(SIGIR '09)*. Association for Computing Machinery, New York, NY, USA, 758–759. doi:10.1145/1571941.1572114

[9] J. Shane Culpepper, Guglielmo Faggioli, Nicola Ferro, and Oren Kurland. 2021. Topic Difficulty: Collection and Query Formulation Effects. *ACM Transactions on Information Systems* 40, 1, Article 19 (2021), 36 pages. doi:10.1145/3470563

[10] Kaustubh D. Dhole and Eugene Agichtein. 2024. GenQREnsemble: Zero-Shot LLM Ensemble Prompting for Generative Query Reformulation. In *Advances in Information Retrieval* (Cham), Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer Nature Switzerland, 326–335. doi:10.1007/978-3-031-56063-7_24

[11] Fernando Diaz. 2024. Pessimistic Evaluation. In *Proceedings of the SIGIR Asia Pacific (SIGIR-AP '24)*. Association for Computing Machinery, Tokyo, Japan. https://arxiv.org/abs/2410.13680

[12] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness in Information Access Systems. *Foundations and Trends® in Information Retrieval* 16, 1-2 (2022), 1–177. doi:10.1561/1500000079

[13] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval* (Taipei, Taiwan) *(ICTIR '23)*. Association for Computing Machinery, New York, NY, USA, 39–50. doi:10.1145/3578337.3605136

[14] Xin Fu, Diane Kelly, and Chirag Shah. 2007. Using collaborative queries to improve retrieval for difficult topics. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands) *(SIGIR '07)*. Association for Computing Machinery, New York, NY, USA, 879–880. doi:10.1145/1277741.1277955

[15] Seyed Mohammad Hosseini, Negar Arabzadeh, Morteza Zihayat, and Ebrahim Bagheri. 2024. Enhanced Retrieval Effectiveness through Selective Query Generation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 3792–3796. doi:10.1145/3627673.3679912

[16] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446. doi:10.1145/582415.582418

[17] Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, Louisiana, USA) *(SIGIR '01)*. Association for Computing Machinery, New York, NY, USA, 120–127. doi:10.1145/383952.383972

[18] 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 International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2356–2362. doi:10.1145/3404835.3463238

[19] João Palotti, Harrisen Scells, and Guido Zuccon. 2019. TrecTools: an Open-source Python Library for Information Retrieval Practitioners Involved in TREC-like Campaigns. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1325–1328. doi:10.1145/3331184.3331399

[20] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In *Advances in Information Retrieval*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer International Publishing, Cham, 397–412. doi:10.1007/978-3-030-99736-6_27

[21] Hossein A. Rahmani, Nick Craswell, Emine Yilmaz, Bhaskar Mitra, and Daniel Campos. 2024. Synthetic Test Collections for Retrieval Evaluation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2647–2651. doi:10.1145/3626772.3657942

[22] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994 (NIST Special Publication, Vol. 500-225)*, Donna K. Harman (Ed.). National Institute of Standards and Technology (NIST), 109–126. http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

[23] Tetsuya Sakai. 2018. Laboratory Experiments in Information Retrieval. *The Information Retrieval Series* 40 (2018), 4. doi:10.1007/978-981-13-1199-4

[24] Bahar Salehi, Damiano Spina, Alistair Moffat, Sargol Sadeghi, Falk Scholer, Timothy Baldwin, Lawrence Cavedon, Mark Sanderson, Wilson Wong, and Justin Zobel. 2018. A Living Lab Study of Query Amendment in Job Search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 905–908. doi:10.1145/3209978.3210082

[25] Chirag Shah. 2010. *Collaborative Information Seeking: A Literature Review*. Emerald Group Publishing Limited, 3–33. doi:10.1108/S0065-2830(2010)0000032004

[26] Damiano Spina, Mark Sanderson, Daniel Angus, Gianluca Demartini, Dana Mckay, Lauren L. Saling, and Ryen W. White. 2023. Human-AI Cooperation to Tackle Misinformation and Polarization. *Commun. ACM* 66, 7 (June 2023), 40–45. doi:10.1145/3588431

[27] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large Language Models can Accurately Predict Searcher Preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 1930–1940. doi:10.1145/3626772.3657707

[28] Liron Tyomkin and Oren Kurland. 2024. Analyzing Fusion Methods Using the Condorcet Rule. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2281–2285.

[29] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, Singapore, 9414–9423. doi:10.18653/v1/2023.emnlp-main.585

[30] Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2023. Generative Query Reformulation for Effective Adhoc Search. In *Proceedings of the First Workshop on Generative Information Retrieval (Gen-IR@SIGIR2023)*. doi:10.48550/arXiv.2308.00415

A Prompts

Here we list the prompts used to generate query variants. GPT-4o and Llama 3.1 70b are reletively larger models and are better at following prompt. However Llama 3.1 8b is much smaller and requires extra output control to produce stable, easy to parse results.

A.1 Prompt for GPT-4o and Llama 3.1 70b

```
Please create a list of unique search queries made by a
diverse group of users seeking answers for a given search
query. The queries should reflect the users' diverse
backgrounds and word choices. Queries can be expressed in
natural language, keywords, or abbreviations. Each list
should contain 20 queries. The length of queries may vary,
but they should average 5 words.
{query}
```

A.2 Prompt for Llama 3.1 8b

```
Please create a list of unique search queries made by a
diverse group of users seeking answers for a given search
query. The queries should reflect the users' diverse
backgrounds and word choices. Queries can be expressed in
natural language, keywords, or abbreviations. Each list
should contain 20 queries. The length of queries may vary,
but they should average 5 words.
Response format rule: after analyse steps, put the query
variants in a numbered list, enclosed in a pair of HTML tag
like this:
<list>
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
</list>
The user search query is: {query}
```