RetPG: Query Decomposition for Complex Information Needs with Retriever Preference Guidance

Anonymous ACL submission

Abstract

Existing off-the-shelf generative systems, as strong as GPT-4, struggle to perform query decomposition in complex information retrieval scenarios, where the explicit user input does not directly reflect the implicit retrieval target. This paper proposes RetPG, which fine-tunes a small-scale LLM with retrieval preference guidance to perform query decomposition, generating subqueries to bridge the mismatch between user and retriever. Specifically, RetPG first uses both the original query and target documents to generate bridging "oracle" subqueries with high retrieval performance. The oracle data is used as ground truth to fine-tune a student query decomposition model via supervised fine-tuning and retrieval-guided preference optimization. Over two complex retrieval tasks, Researchy Questions retrieval and Tip-of-the-Tongue retrieval, RetPG outperforms existing non-neural and neural query expansion and decomposition baselines. Analysis and ablation studies demonstrate the necessity of constructing oracle subqueries as ground truth, and the effectiveness of applying retrieval preference signals for fine-tuning the query decomposer. [1]

1 Introduction

Understanding user instructions and information needs is a fundamental challenge in information retrieval. Both in open-web and task-specific retrieval scenarios, retrieval systems encounter diverse and complex user information needs, where the explicit user input does not directly reflect the implicit retrieval target. It is up to the retrieval system to bridge such mismatch between user and retriever in complex information retrieval scenarios.

A common approach to handle such search queries is to perform query **decomposition**, by generating new, focused *subqueries* for downstream retrieval. Applying pretrained generative LMs for question decomposition has led to improvement in tasks such as multi-hop QA (Wu et al., 2024a). However, query decomposition for complex retrieval presents new challenges for generative systems. Input-wise, the user input query in complex retrieval scenarios is often indirect and incomplete, lacking enough information to perform direct decomposition. The decomposer is required to determine "what to retrieve" and generate subqueries based on the retrieval task and its prediction of user intent. Objective-wise, the target of generation is to produce subqueries that improve *retrieval performance* of the system, which diverges from standard language modeling objectives.

This paper investigates the development of a generative system for query decomposition in complex retrieval scenarios. We first perform experiments over two retrieval tasks with complex information needs and find that existing off-the-shelf models as strong as GPT-4 cannot effectively perform decomposition solely from zero-shot or few-shot prompting, producing subqueries with low retrieval performance. This highlights the need to develop a query decomposition model with retrieval-centric alignment to overcome the previously discussed challenges. Specifically, the model should be fine-tuned on high-quality, task-specific subqueries that bridge complex queries and retrieval targets, and the training process should incorporate a retrieval performance-oriented objective, using guidance signals directly from the retriever.

Based on such discussions, this paper proposes a new framework, **Query Decomposition with Retriever Preference Guidance** (RetPG), which fine-tunes a small-scale LLM to perform effective query decomposition for complex user information needs with alignment to the retrieval task and retriever. Firstly, RetPG generates task-specific, high-quality training data for the query rewriting task through an oracle generation process, where the model is provided with both query and document

[1] Code and data will be publically released.

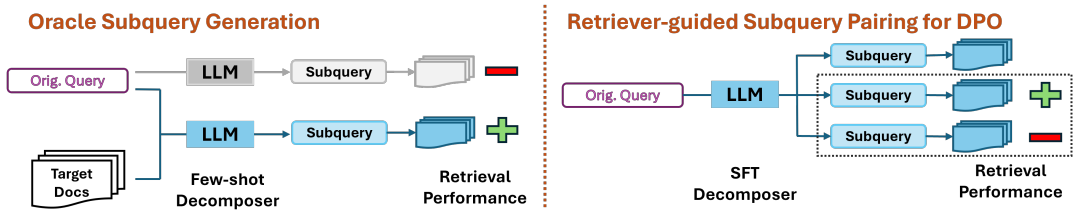


Figure 1: Illustration of the key steps in the RetPG framework: oracle subquery generation as ground-truth, and retrieval performance-based DPO preference pair sampling.

to predict bridging subqueries. After utilizing such oracle subqueries to fine-tune a student LLM via standard supervised fine-tuning, RetPG additionally perform offline preference optimization (Rafailov et al., 2023; Meng et al., 2024) over retrieval-based preference signals, where the decomposer is directly tuned to distinguish subqueries with superior retrieval performance.

Figure 1 introduces the oracle subquery generation and DPO preference pair construction processes in RetPG. We perform experiments over two complex retrieval settings: open-web Researchy Questions (Rosset et al., 2024) retrieval and entity-targeted Tip-of-the-Tongue (Borges et al., 2024) retrieval. On both tasks, we observe that RetPG is able to align with the retrieval setting and generate high-quality subqueries for the down-stream retriever, outperforming GPT-4 few-shot subquery generation and other query expansion techniques.

Further analysis of the model training procedure demonstrates the importance of oracle subquery generation and incorporating retrieval preference guidance in the RetPG framework. The oracle subquery generation step provides task-specific retrieval-effective training data for the student model, as subqueries generated by GPT-4 via few-shot prompting perform poorly when paired with the down-stream retriever. Moreover, selecting retrieval performance signals as the criteria to perform preference optimization is crucial to ensure the retrieval quality and stability of subqueries. We also investigate the effectiveness-efficiency tradeoff of RetPG's retrieval process and analyze its behavior after alignment over different retrieval tasks.

The rest of the paper is constructed as follows: Section 2 introduces related works and discusses the motivation of our approach. Section 3 describes the RetPG framework. Sections 4 and 5 report our implementation settings and experiment results. Section 6 concludes.

2 Related Work

Query decomposition and expansion have long been a fundamental technique to improve the performance of text retrieval systems (Rocchio Jr, 1971). It aims to inject new, meaningful information to better align the original query with the detailed retrieval task and information need. Classic query expansion methods often involve pseudo relevance feedback (Jaleel et al., 2004; Yu et al., 2021; Wang et al., 2023b), while more recent developments utilize the language understanding ability and parametric knowledge of large language models (Ouyang et al., 2022; OpenAI, 2023) to perform generative query expansion, such as keyword expansion (Jagerman et al., 2023; Li et al., 2024) and answer document prediction (Wang et al., 2023a; Gao et al., 2023). Query rewriting and reformation play a fundamental role in retrieval or question-answering scenarios with complex queries, such as multi-hop QA (Yang et al., 2018; Shao et al., 2023)), long-form QA (Fan et al., 2019; Stelmakh et al., 2022; Min et al., 2020), retrieval with specific tasks and instructions (Asai et al., 2022; Zhao et al., 2024), and *diversified* retrieval.(Clarke et al., 2012; Wu et al., 2022).

Instruction fine-tuning of LLMs on human completions or performing RLHF (Christiano et al., 2017; Ouyang et al., 2022) have been found to align LLMs on down-stream tasks or user intents effectively, but it is often costly to generate training data for such procedures. Recent work focuses on fine-tuning LLMs over human *preferences* or relative judgements (Schulman et al., 2017) with an explicit reward model (Bradley and Terry, 1952). The introduction of DPO (Rafailov et al., 2023) further demonstrated that it is possible to perform offline optimization over only the generative language model to implicitly maximize the RLHF reward. Recent research further introduces improvements to offline preference optimization fine-tuning, such as more robust loss functions and performing self-

play (e.g. Liu et al. (2024); Azar et al. (2024); Wu et al. (2024b)). In this work, RetPG perform fine-tuning with SimPO (Meng et al., 2024), an offline preference optimization variant based on the average log probability of target sequence generation.

Recent advances apply preference-based optimization to retrieval-augmented generation (RAG) tasks, including performing PPO for training modules for RAG question rewriting (Ma et al., 2023) or generative re-ranking (Wang et al., 2024b). In this paper, we directly focus on complex retrieval tasks and utilizing feedback directly from the retrieval performance for decomposition systems.

3 RetPG: Query Decomposition with Retriever Preference Guidance

In complex retrieval scenarios, there is often a gap between a search engine user's input q and the underlying retrieval targets $\{d^+\}$ within the corpus C, making the user query q insufficient as a query for the retriever. In this paper, RetPG aims to resolve this issue by performing **query decomposition**, where it generates subqueries based on the original query $q \Rightarrow \{q^s\} = \{q_1^s, \cdots, q_N^s\}$. Ideally, subqueries should complement the original user query in the retrieval process: they should cover important facets of the original query and be detailed, high-quality queries for the retriever. At retrieval time, the subqueries individually perform retrieval, and retrieved evidence is merged to produce a final retrieval ranking presented to the user.

In the following sections, we introduce the three steps of RetPG's training and inference: oracle subquery generation as ground truth, decomposer fine-tuning, and RetPG's retrieval process.

3.1 Oracle Subquery Generation

Fine-tuning a retrieval scenario-specific query decomposer or rewriter first requires respective supervision data consisting of (original query, subquery) training pairs $\{(q, q^s)\}$. However, standard retrieval datasets only contain training data pairs of (original query, relevant documents) $\{(q, \{d^+\})\}$. Given a set of training set queries with such relevance pairings, we must first construct candidate subqueries bridging the query and document.

A common approach for task-specific fine-tuning data augmentation is to utilize off-the-shelf large-scale LLMs as the teacher generator for reference (e.g. Wang et al. (2024a)), directly performing few-shot generation $\text{LLM}(q) \Rightarrow q^s$. However, such setup is potentially ineffective in our retrieval setting: with only the user query as input and no control of subquery content, the generated subqueries do not necessarily connect with the actual target documents. This directly limits the *retrieval performance* of the generated subqueries and may harm performance when used as ground truth data for query decomposer fine-tuning.

Instead, RetPG performs ground truth data generation in an **Oracle** scenario, where both the query *and its relevant documents* are presented to generate subqueries. The Oracle generation process ensures that the generated subquery bridges the query and relevant documents, which directly aligns with the overall objective of our query decomposer framework. While at retrieval time the oracle generation setting is not realistic, at model training time this produces (q, q^s) pairs with higher subquery retrieval quality. RetPG uses such subqueries to fine-tune the query decomposer model, which we denote as the **Student** model.

3.2 Query Decomposer Training

After acquiring oracle subqueries, RetPG uses the oracle subquery pairings $\mathcal{D}_t = \{(q, q^s)\}$ to fine-tune a small-scale LM via a two-stage approach: initial supervised fine-tuning (SFT) and additional fine-tuning with retrieval-guided direct preference optimization (DPO).

Supervised fine-tuning (SFT) To align a small-scale LM on the specific retrieval scenario, RetPG first performs standard supervised fine-tuning on the task of oracle subquery sequence generation $q \Rightarrow \{q^s\}$. The student decomposer model is optimized over a next-token prediction objective: hiven the original query and instruction as the prefix $\mathbf{p} = \text{Inst}(q)$ as input, the model is trained to generate the target text sequence $\mathbf{t} = q^s$, where $\mathbf{t}$ is an oracle subquery.

$$\mathcal{P}_\theta(\hat{\mathbf{t}} = \mathbf{t}; \mathbf{p}) = \prod_{k=1}^{L_\mathbf{t}} \mathcal{P}_\theta(\hat{\mathbf{t}}_k = \mathbf{t}_k; \mathbf{p}, \mathbf{t}_{<k})$$

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{\mathcal{D}_t}[-\log \mathcal{P}_\theta(\hat{\mathbf{t}} = \mathbf{t}; \mathbf{p})]$$

$$= \mathbb{E}_{\mathcal{D}_t}[-\log \sum_{k=1}^{L_\mathbf{t}} \mathcal{P}_\theta(\hat{\mathbf{t}}_k = \mathbf{t}_k; \mathbf{p}, \mathbf{t}_{<k})]$$

θ is the query decomposer language model, and $\mathcal{P}_\theta(\hat{\mathbf{t}} = \mathbf{t}; \mathbf{p})$ is the generation probability of the target sequence $\mathbf{t}$ given prefix $\mathbf{p}$. This provides the model with a basic understanding of the decomposition task and the subquery generation format.

Retrieval-guided direct preference optimization (DPO) The SFT training step aligns the student model over the retrieval task, but does not further include actual information or training signals of the actual retriever. To further align the subquery generation module to produce subqueries with high retrieval performance, RetPG additionally performs fine-tuning with direct preference optimization (Rafailov et al., 2023), which fine-tunes the model over text sequence pairs with labeled "chosen-rejected" **preference** selection. The model is fine-tuned to learn to increase the margin of generation probability between the preference sequence pairs. After the SFT process, for each original query q over a set of training queries, RetPG first generates multiple instances of subqueries from the SFT-tuned decomposer:

$$q_1^s, q_2^s, \cdots, q_n^s \sim \pi^{\text{SFT}}(q^s \mid q_o)$$

RetPG then tests the retrieval performance of each subquery $\{q_1^s, q_2^s, \cdots, q_n^s\}$, selecting the best-performing and worst-performing subqueries (q_w^s, q_l^s) to form a chosen-rejection training pair. Over a large set of model output samples with preference labeling $\mathcal{D} = \{q_o^{(i)}, q_w^{s(i)}, q_l^{s(i)}\}$, we apply the SimPO (Meng et al., 2024) objective which optimizes over the average generation likelihood of the text sequence:

$$\hat{r}_{\theta,w} = \frac{\beta}{|q_w^s|} \log \pi_\theta(q_w^s|q_o)$$
$$\hat{r}_{\theta,l} = \frac{\beta}{|q_l^s|} \log \pi_\theta(q_l^s|q_o)$$
$$\mathcal{L}_{\text{SimPO}} = -\mathbb{E}_{\mathcal{D}}[\log \sigma(\hat{r}_{\theta,w}^{(i)} - \hat{r}_{\theta,l}^{(i)} - \gamma)]$$

where π_θ is the decomposition model, $|q^s|$ is the length of the subquery sample, $\hat{r}_\theta$ is the implicit re-parameterized reward target for π_θ. β and γ are hyper-parameters controlling deviation of the learned policy from initial system and a target reward margin respectively. In practice, RetPG performs initialization with $\pi_\theta = \pi^{\text{SFT}}$.

The DPO training process maximizes the generation likelihood of the chosen sequence output $\pi_\theta(q_w^s|q_o)$ and minimizes the likelihood of the rejected sequence output $\pi_\theta(q_l^s|q_o)$. Overall, RetPG first obtain a retrieval task-specific student decomposer model via SFT from oracle subqueries, and then refine the student's subquery generation quality through DPO. The fine-tuning process enables RetPG to generate text sequences that better align with the preference criteria, which for RetPG is the downstream retrieval performance. We discuss specific sampling details and retrieval metric selection in Section 4.

3.3 Inference and Retrieval

At inference time, given the original query q_o, we sample multiple outputs of the finetuned RetPG model to produce a set of N subqueries $\{q^s\}$. All subqueries along with the original query $\{q, q^{s1}, ..., q^{sN}\}$ are used to individually perform retrieval, producing N+1 rankings. To produce the aggregated ranking, we further perform min-max normalized score fusion (Belkin et al., 1995) of the rankings, a simple non-supervised fusion method that aggregates normalized document scores. For each document d_i in a generated ranking with score $S(q, d_i)$,

$$S'(q, d_i) = \frac{\max(S(q, d_i) - S(q, d_K), 0)}{S(q, d_1) - S(q, d_K)}$$
$$\mathbf{S}(d) = w_o S'(q_o, d) + \frac{1 - w_o}{N} \sum_{i \in N} S'(q_i^s, d)$$

where $S'(q, d_i)$ is the document score after normalization with the maximum and minimum document scores within the ranking. $\mathbf{S}(d)$ is the aggregated document score after fusion which we use to produce the final ranking. A higher w_o represents less reliance of the subqueries.

4 Experiment Methodology

In this section we introduce the retrieval tasks used to evaluate RetPG's performance, Researchy Question retrieval and Tip-of-the-Tongue retrieval, as well as details of RetPG implementation.

Researchy Questions Retrieval A "researchy question" (Rosset et al., 2024) is defined as a non-factoid user question that requires multi-faceted answer sources to fulfill the underlying information need, therefore entailing extra effort to research. Such queries make up a substantial portion of real-user search engine queries and presents a larger challenge to search systems compared to simpler factoid-centric queries.

Experiments were performed on the recently released Researchy Questions (RQ) dataset (Rosset et al., 2024), a large-scale dataset containing real user researchy queries and the respective clicked web pages from the search logs of Bing [2]. The clicked document web pages are included in

[2] https://www.bing.com/

ClueWeb22 (Overwijk et al., 2022), a publicly available web corpus. [3] Retrieval experiments were performed over the English portion of the B category (most popular) ClueWeb22 documents, which consists of about 88 million web pages. The final RQ dataset used for experiments contains 82K training set queries and 5.9K test set queries with 3.1 relevant documents per query.

Tip-of-the-Tongue Retrieval Tip-of-the-Tongue (ToT) retrieval involves users searching for specific entities from pieces of their memory, such as searching for a movie based on an estimate of watched time, and memories of overall plot and some specific details. In such scenarios, the user query is a long paragraph in natural language often containing irrelevant or incorrect information.

We evaluated `RetPG`'s performance over the TREC-ToT benchmark, which contains 150 movie-targeted ToT queries with human-verified answers produced in the TREC 2023 ToT retrieval track (Jaime Arguello, 2023). For model fine-tuning, we mainly followed the experiment setting and used the movie category of the processed dataset by Borges et al. (2024), which contains "user query, target entity Wikipedia page" pairs curated from the Reddit ToT dataset as relevance signals (Bhargav et al., 2022; Fröbe et al., 2023).

Evaluation NDCG@30, Recall@10 and Recall@100 are used to evaluate retrieval performance. On both tasks, we compare `RetPG`'s retrieval performance using BM25 to the following types of baselines for query decomposition or expansion: 1) **RM3** (Jaleel et al., 2004), a non-neural query vocabulary expansion method based on documents retrieved by the original query; 2) **GPT-4 FewShot decomposition**, where GPT-4 was directly prompted to rewrite and decompose the original query into adequate subqueries. On Researchy Questions, we directly used the GPT-4 query decomposition output provided within the RQ dataset, which were generated via few-shot prompting. On ToT datasets we performed the GPT-4 expansion ourselves with a zero-shot prompt; 3) **GPT-4 Answer Prediction**, where GPT-4 was prompted to directly generate an answer document or retrieval target. On Researchy Questions, we used HyDE (Gao et al., 2023) expansion method, which utilizes the LLM to generate a "hypothetical document". For ToT queries, we apply the answer prediction baseline discussed by Borges et al. (2024), which first uses GPT-4 to predict 25 movie guesses from a ToT query and use BM25 to retrieve 4 documents per guess, resulting in 100 documents total.

Implementation GPT-4 (OpenAI, 2023) [4] was used as an off-the-shelf teacher model to generate "oracle" ground-truth subqueries. Oracle subqueries were generated for 5000 training queries on RQ retrieval and 3000 training queries on ToT retrieval. Prompts for `RetPG` and baselines are listed in the appendix (Section A).

Llama-3-8B (Meta-AI, 2023) was used to initialize all student subquery generation models. We first used GPT-4-generated subquery sets as ground truth to fine-tune the SFT student model with LoRA adaptation (Hu et al., 2022). The SFT model was trained for 2 epochs with lr=2e-4 and linear weight decay. For further fine-tuning, over a set of training queries, we sampled 10 subquery outputs from the SFT model and perform retrieval for all subquery samples. We used the individual subquery Recall@100 as the preference metric to generate preference pairs of best-performing and worst-performing subqueries, and used such data to further fine-tune the student model with DPO on the SimPO (Meng et al., 2024) preference loss with LoRA adaptation. The DPO model was trained for 2 epoch with lr=1e-5, β=2.0, γ=1.0 and linear weight decay. Both the SFT and DPO training process required less than 12 hours on a single NVIDIA RTX A6000 GPU.

At inference time, the subquery generation temperature was set to 0.9 for all models. We used BM25 (Robertson and Zaragoza, 2009), a popular lexicon-based retriever as the base ranker for all experiments. We performed min-max score fusion using rankings of the original query and 5 subqueries, with the default original query weight set to w_o=0.5 and w_o=0.3 for RQ and ToT respectively.

5 Experimental results

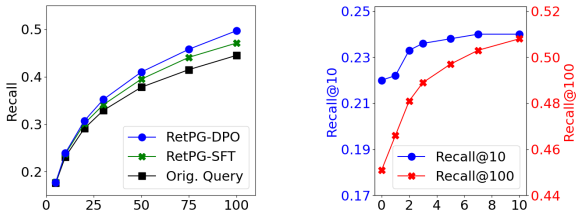
In this section, we first present `RetPG`'s performance on the RQ and ToT retrieval tasks and compare them to the respective baselines. We then discuss the effectiveness of the `RetPG` framework through training procedure analysis and output case study.

[3] The RQ dataset only provides per-query user click distribution (i.e., the percentage of user clicks on each document). It does not provide cross-query comparison of query frequency or click count. Our experiments used binary relevance, where all clicked document pages are considered equally relevant and all other documents are considered not relevant.

[4] gpt-4-turbo-2024-04-09

Table 1: Retrieval results on Researchy Questions retrieval and Tip-of-the-Tongue retrieval tasks. RetPG retrieval results produced via min-max ranking score fusion of 5 generated sub-queries' retrieval and the original query's retrieval. † and ‡ denote the statistical significance of RetPG's retrieval performance compared to the original query and GPT-4 FewShot Decomposition respectively (one-sided paired t-test, p<0.05).

	Researchy Questions				ToT TREC		
Retriever: BM25	**N@30**	**R@10**	**R@100**	**BLEU-3**	**N@30**	**R@10**	**R@100**
Original Query	0.222	0.228	0.452	N/A	0.102	0.120	0.240
RM3	0.194	0.198	0.452	N/A	0.091	0.120	0.220
GPT-4 FewShot Decomp.	0.219	0.228	0.431	0.257	0.154	0.180	0.340
GPT-4 Answer Prediction	0.223	0.224	0.451	0.038	–	0.213	0.260
RetPG-DPO	**0.229**†‡	**0.238**†‡	**0.497**†‡	**0.447**	**0.182**†‡	**0.227**†‡	**0.347**†



(a) Recall Depth (b) Subquery Count

Figure 2: Analysis of RetPG performance on Researchy Questions. Left: Recall for different RetPG variants at different depths. Right: Retrieval performance for RetPG-DPO with different number of subqueries.

5.1 Performance on Complex Retrieval Tasks

Table 1 reports RetPG's performance on the Researchy Questions and Tip-of-the-Tongue retrieval tasks. On the Researchy Questions retrieval task, all baseline non-neural and neural generative systems have small or no improvement over the retrieval performance of the search engine users' original input query. This demonstrates that query rewriting and decomposition for complex retrieval needs is a realistic, non-trivial problem for real-world search systems.

RetPG decomposition improves retrieval performance for BM25 compared to all baselines. We observe that there is a larger gain for RetPG over Recall@100 compared to shallow ranking depths. Figure 2a further plots the Recall at different depths of the original query and the RetPG model. We hypothesize this is due to the nature of the retriever and task: for RQ retrieval, the target of decomposition is to generate subqueries with new, focused topics that complement the original query. The user query already provides a general topic and may be already capable of retrieving a subset of relevant documents with lower difficulty.

To analyze subquery effectiveness from a lexical content perspective, we additionally introduce a subquery *BLEU score* metric, where we calculate the average BLEU-3 of decomposed subqueries when matched to its respective set of *oracle subqueries* as reference. subqueries generated by RetPG achieves much higher BLEU-3 compared to baseline methods, indicating higher similarity in lexical content with the target "oracle" subqueries.

At inference time, RetPG samples multiple subqueries from the student model and performs retrieval with each original query and subquery individually before ranking fusion. Figure 2b reports the performance of the final fused ranking with a different number of subquery samples. We observe a general trade-off in retrieval effectiveness and efficiency, as more subqueries lead to more robust retrieval at the cost of extra retrieval. For shallow-depth metrics, we only require a smaller amount of subqueries while a larger number of subqueries is needed to achieve higher Recall performance before convergence.

On the Tip-of-the-Tongue first-stage retrieval task, we also see that RetPG outperforms the original query performance by a wide margin, and also outperforms the GPT-4 decomposition and answer prediction baseline. As a much smaller-scale and efficient model with less world knowledge, this demonstrates RetPG's contribution to the retrieval system solely as a fine-tuned query decomposer. We further discuss the difference between RetPG's behavior across the two retrieval tasks through a case study in Section 5.3.

5.2 Analysis

In this section, we look deeper into RetPG's performance on the Researchy Questions dataset to analyze the source of RetPG's gain in retrieval performance, as well as the importance of performing oracle ground truth generation and applying retriever preference during decomposer fine-tuning.

Training framework analysis Table 2 reports the performance of different RetPG variants at dif-

Table 2: Analysis of RetPG student model's performance on the Researchy Questions dataset at different stages of training. We report performance under 2 settings: retrieval with only each sub-query and after fusion with the original query. *"Best of 10 samples"* represents the maximum sub-query performance out of 10 independent generation samples. We also report the BLEU-3 score of each sub-query type, with oracle sub-queries as reference.

	N@30		**R@10**		**R@100**		**BLEU-3**
	Subquery	**Fusion**	**Subquery**	**Fusion**	**Subquery**	**Fusion**	
Original Query	N/A	0.222	N/A	0.228	N/A	0.452	N/A
GPT-4 Oracle Decomp.	0.197	0.290	0.213	0.304	0.444	0.624	N/A
GPT-4 FewShot Decomp.	0.036	0.222	0.038	0.228	0.213	0.452	0.257
SFT [GPT-4 Oracle]	0.066	0.219	0.071	0.235	0.281	0.468	0.299
Best of 10 samples	*0.224*	*0.300*	*0.247*	*0.319*	*0.546*	*0.577*	*0.602*
+ DPO (BLEU)	0.104	0.213	0.111	0.228	0.344	0.476	**0.610**
+ DPO (R@100)	**0.148**	**0.229**	**0.156**	**0.238**	**0.397**	**0.497**	0.447

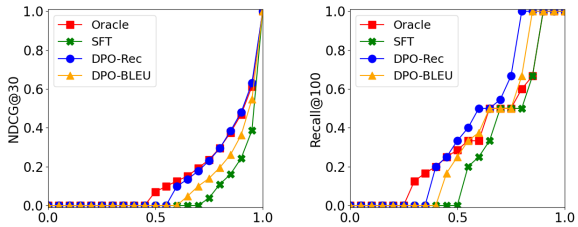


Figure 3: Distribution of retrieval performance of sampled sub-queries for the Oracle, SFT student and DPO student models. The X-axis denotes sorted sub-query performance.

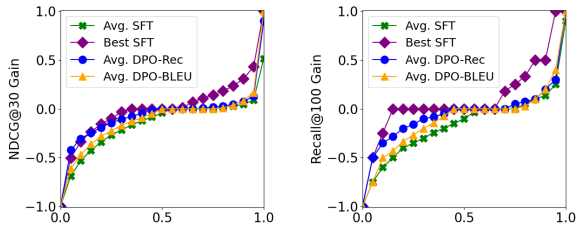


Figure 4: Distribution of sub-query retrieval performance gain over original query for RetPG variants: average SFT, best SFT (out of 10) and average SimPO. The X-axis denotes sorted sub-query performance.

ferent stages of the training pipeline. The subqueries generated by GPT-4 FewShot decomposition do not perform well individually, as the decomposition model (GPT-4) has no specific alignment over the retrieval task and no information on the target documents. We conducted preliminary experiments and find that using such subqueries as ground truth for SFT fine-tuning is also ineffective. It is crucial to provide the system with information on the relevant documents to generate bridging subqueries with high retrieval performance.

When using Oracle subquery data to fine-tune the student model in query-only generation settings, we observe a drop in subquery performance due to the increased task difficulty. The subqueries produced by the SFT student model introduce minor performance gain in complement of the original query. However, there is large variance in the subqueries' retrieval performance: across multiple generation samples for the same input query, the **best**-performing subquery consistently provide significant gain in retrieval performance, suggesting that there is headroom for further model alignment over the retrieval performance objective.

After DPO fine-tuning, RetPG learns to partially cover the headroom of the SFT model's upper bound performance, leading to improvement in subquery quality. Figure 3 reports the distribution of subquery performance samples for the original query and subqueries produced by the oracle teacher, SFT student, and DPO student models. Figure 4 further reports the distribution of performance gain over the original query by the RetPG SFT and DPO student. We see that the DPO student model improves the average performance of the subqueries, generating much fewer low-quality queries with zero or close to zero performance.

DPO with retriever preference The core idea in RetPG's training framework is to utilize the preference signals of the retriever to fine-tune the student decomposition model. To look into the necessity of applying *retrieval* preference signals, we conduct a separate experiment where we perform DPO over the subquery BLEU-3 metric discussed in Section 5.1. We observe a two-fold message from Table 2: there is a positive correlation between RetPG variants' retrieval performance and BLEU-3 scores; however, directly performing preference optimization over the BLEU-3 metric instead of a retrieval objective leads to lower retrieval performance. This again demonstrates the distinction between language modeling targets and retrieval targets, and the importance of RetPG's training with

Table 3: Case study of subqueries generated by RetPG for Researchy Question Retrieval and ToT retrieval.

Researchy Questions	**Tip-of-the-Tongue**
Query: how have virtual training methods and technology impacted hr's role in employee development?	**Query:** [TOMT][MOVIE] 1990s or early 2000s horror movie about awakened evil that snatches people through drains I've only heard about this movie through friends but none of us can figure out which it was! We know it was about a creature or force that would pull people through the drain; might have been tentacles or worms that would grab the victim by the face? Supposedly this monster was responsible for the extinction of Mayan or other Meso-American cultures.
RetPG-DPO (BLEU) Output: 1. How has virtual training technology influenced employee development in HR? 2. How has virtual training technology enhanced HR employee development? 3. How has virtual training technology improved HR employee development? 4. How has virtual training technology enhanced employee development in HR? 5. How has virtual training technology influenced HR in employee development?	
RetPG-DPO (R@100): 1. How has technology changed the role of HR in employee development and training? 2. How do virtual training methods and technology impact HR's role in employee development? 3. How does virtual training technology impact HR's role in employee development? 4. How has technology changed the role of HR in employee development? 5. What is the impact of technology on HR's role in training and development?	**RetPG-DPO (R@100) Output:** 1. 2000s horror film about a supernatural entity in drains 2. 1990s horror film about a Mayan creature in drains 3. 1998 horror film about a creature that causes extinction of Mesoamerican cultures 4. 2000s horror film about ancient evil awakened through a drain 5. 1990s horror film about an awakened evil through drains

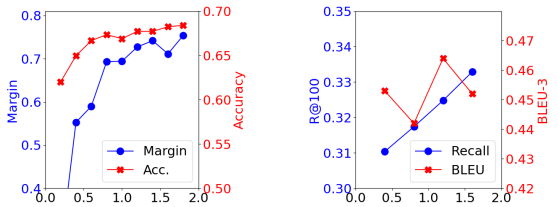


Figure 5: RetPG DPO fine-tuning (margin and classification accuracy) and retrieval performance on eval set during training process. X-axis denotes training epoch.

direct retriever guidance.

In Figure 5, we further analyze the training procedure during DPO fine-tuning. From the SFT model checkpoint initialization, as the training progresses, the model gradually improves its ability to distinguish the preference signals of the retriever, with gains in preference pair classification accuracy and fine-tuning loss, as well as downstream retrieval performance R@100.

5.3 Case Study

Table 3 presents an example query and output of RetPG over the RQ and ToT retrieval tasks. We observe the contrast in retrieval task and the consequent differences in RetPG's behavior: on the Tip-of-the-Tongue retrieval task, the original query is not an effective query for the retrieval system. Compared to the Researchy questions retrieval task, the RetPG trained over the ToT retrieval task is required to decompose the original query by removing verbose descriptions and irrelevant information and extracting useful details.

From the case study, we also see that across both tasks, over multiple individual subquery samples, there are query samples with very close lexical form and meaning. The RetPG variant trained over the BLEU objective has less variance and diversity within its contents. One hypothesis is that the BLEU reward may easily lead to the model overly focusing on specific lexical patterns of the subquery format instead of content, leading to subpar retrieval performance. As RetPG does not perform specific deduplication or diversification over the generated subqueries, this also suggests directions for future extensions to the RetPG framework, such as directly generating the *set* of subqueries instead of pointwise generation, specific post-hoc diversification procedures, and fact and content verification within decomposition subqueries.

6 Conclusion

In complex retrieval scenarios, the user query is often incomplete and imprecise in reflecting the implicit retrieval target. This paper introduces RetPG, which aims to bridge the user and retriever through query decomposition. RetPG is trained over specific retrieval tasks in a two-stage framework: It first generates high-quality subqueries in oracle generation settings, and then uses the oracle subqueries to fine-tune a student model to perform subquery generation through SFT and retriever-preference based DPO optimization.

Over two retrieval tasks, Researchy Questions retrieval and Tip-of-the-Tongue retrieval, RetPG outperforms current query expansion and decomposition systems, including direct few-shot prompting of off-the-shelf language models. This demonstrates the importance of aligning the generation framework with the detailed retrieval scenario and retriever. We hope this work encourages more exploration in effectively and efficiently applying generative systems to improve retrieval systems.

Limitations

In this paper we experiment under generation settings of LLaMa and GPT-4, and do not further test the performance of other off-the-shelf generative language models. We also see room for improvement in our retrieval experiment settings, where we do not further train a retriever or reranking system on the specific dataset.

Efficiency-wise, `RetPG` is a subquery generation model, and at retrieval time may use different numbers of subqueries per input query. Section 5.1 discusses the effectiveness-efficiency tradeoff of `RetPG` on the RQ dataset. Also as discussed in Section 5.3, the current `RetPG` does not perform extra diversification or filtering of generated subqueries, which may affect retrieval efficiency.

References

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen tau Yih. 2022. Task-aware retrieval with instructions. In *Annual Meeting of the Association for Computational Linguistics*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Nicholas J. Belkin, Paul B. Kantor, Edward A. Fox, and Joseph A. Shaw. 1995. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manag.*, 31(3):431–448.

Samarth Bhargav, Georgios Sidiropoulos, and Evangelos Kanoulas. 2022. 'it's on the tip of my tongue': A new dataset for known-item retrieval. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 48–56. ACM.

Luís Borges, Rohan Jha, Jamie Callan, and Bruno Martins. 2024. Generalizable tip-of-the-tongue retrieval with llm re-ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2437–2441, New York, NY, USA. Association for Computing Machinery.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307.

Charles L. A. Clarke, Nick Craswell, and Ellen M. Voorhees. 2012. Overview of the TREC 2012 web track. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*, volume 500-298 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: long form question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics.

Maik Fröbe, Eric Oliver Schmidt, and Matthias Hagen. 2023. A large-scale dataset for known-item question performance prediction. In *Proceedings of the The QPP++ 2023: Query Performance Prediction and Its Evaluation in New Tasks Workshop co-located with The 45th European Conference on Information Retrieval (ECIR), Dublin, Ireland, April 6th, 2023*, volume 3366 of *CEUR Workshop Proceedings*, pages 13–19. CEUR-WS.org.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1762–1777. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *CoRR*, abs/2305.03653.

Samarth Bhargav Jaime Arguello. 2023. Trec 2023 tip-of-the-tongue (tot) track.

Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. Umass at TREC 2004: Novelty and HARD. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC*

2004, Gaithersburg, Maryland, USA, November 16-19, 2004, volume 500-261 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Minghan Li, Honglei Zhuang, Kai Hui, Zhen Qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. 2024. Can query expansion improve generalization of strong cross-encoder rankers? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2321–2326. ACM.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *CoRR*, abs/2305.14283.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *CoRR*, abs/2405.14734.

Meta-AI. 2023. Meta llama 3.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5783–5797. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Arnold Overwijk, Chenyan Xiong, and Jamie Callan. 2022. Clueweb22: 10 billion web documents with rich information. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3360–3362. ACM.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*.

Corby Rosset, Ho-Lam Chung, Guanghui Qin, Ethan C. Chau, Zhuo Feng, Ahmed Awadallah, Jennifer Neville, and Nikhil Rao. 2024. Researchy questions: A dataset of multi-perspective, decompositional questions for LLM web agents. *CoRR*, abs/2402.17896.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9248–9274. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *ArXiv*, abs/2204.06092.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11897–11916. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9414–9423. Association for Computational Linguistics.

Shuting Wang, Xin Yu, Mang Wang, Weipeng Chen, Yutao Zhu, and Zhicheng Dou. 2024b. Richrag: Crafting rich responses for multi-faceted queries in retrieval-augmented generation. *CoRR*, abs/2406.12566.

Xiao Wang, Craig MacDonald, Nicola Tonellotto, and Iadh Ounis. 2023b. Colbert-prf: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Trans. Web*, 17(1):3:1–3:39.

Haolun Wu, Yansen Zhang, Chen Ma, Fuyuan Lyu, Bowei He, Bhaskar Mitra, and Xue Liu. 2022. Result diversification in search and recommendation: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Jian Wu, Linyi Yang, Yuliang Ji, Wenhao Huang, Börje F. Karlsson, and Manabu Okumura. 2024a. Gendec: A robust generative question-decomposition method for multi-hop reasoning. *CoRR*, abs/2402.11166.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024b. Self-play preference optimization for language model alignment. *CoRR*, abs/2405.00675.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*.

HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3592–3596. ACM.

Xinran Zhao, Tong Chen, Sihao Chen, Hongming Zhang, and Tongshuang Wu. 2024. Beyond relevance: Evaluate and improve retrievers on perspective awareness. *ArXiv*, abs/2405.02714.

A Prompts used for `RetPG`

We list prompts used for the RQ and ToT retrieval settings in Table 4 respectively. In the ToT setting, we found that part of the training data contained incorrect "ToT query-entity" pairings due to the construction process of the dataset. We additionally add an analysis step where we ask GPT-4 to evaluate the correctness of the pairing, and filter out the incorrect pairings during the SFT training stage. We find that there were about 30% incorrect training pairs, which is slightly higher than what Borges et al. (2024) reports.

Table 4: Prompt for GPT-4 Generation on the Researchy Questions dataset.

GPT-4 Researchy Questions Retrieval, Oracle Generation
You are an assistant for a commercial web search engine.

Instruction: Given a user question and the set of clicked documents that the search engine users found useful for the question. Please specify why and which aspect each document fulfills the query information need. Based on this analysis, generate one 'subquestion' for each clicked document. The 'subquestion' should be a short, concise search engine query based on the original question, that best match the information need fulfilled by the clicked document. The 'subquestion' should be a complete sentence.

Format: Please output a python dictionary in json format. The dictionary has only one key:
- Subquestions: A list of strings. Each string corresponds to the generated subquestion of a clicked document.

Make sure to output only the valid dictionary, without comments or other extraneous output and formatting. Output exactly one matching subquestion per input click document. Do not include the "json" prefix.

Question: *{Query}*
Documents: *{Document 1, Document 2, ...}*
Output:

GPT-4 Tip-of-the-Tongue Movie Retrieval, Oracle Generation
You are an assistant for a commercial web search engine.

Instruction: You are given a user question describing their memory of a given movie, and the Wikipedia description of the corresponding movie that the user was trying to remember. Please analyze and specify which details of the user query is useful towards retrieving the target movie. Based on this analysis, rewrite the user query into five 'sub-queries' that are specific search engine queries which do not directly reveal the name or extra information of the target movie, but can retrieve the description of the target movie.

Description: A 'sub-query' should be a short, concise search engine query that connects the original user query with the target. It should be based on the original question and accurately retrieve the target movie, but it should only include information given within the original query, and not leak any additional information given directly from the Wikipedia page. A 'sub-query' should be a complete sentence.

Format: Please output a python dictionary in json format. The dictionary has three keys:
- Summary: A string. A paragraph that discusses how the target movie is related to the original query, and how the user query matches the target.
- Subquestions: A list of strings. Each string corresponds to a generated subquestion that may retrieve the target movie. Generate 5 most important subquestions.
- Correctness: Check if the user query actually matches the given document movie. If there is a mismatch or mis-understanding, output 'Mismatch', else output 'Correct'.

Make sure to output only the valid dictionary, without comments or other extraneous output and formatting. Do not include the "json" prefix.

Question: {Query}
Target: {*Wikipedia document of target movie*}
Output:

LLaMa-3-8B Retrieval
Generate a subquestion for the following query: "*{Query}*"
Output: