# ReFormeR: Learning and Applying Explicit Query Reformulation Patterns

**Abstract.** We present `ReFormeR`, a pattern-guided approach for query reformulation. Instead of prompting a language model to generate reformulations of a query directly, `ReFormeR` first elicits short reformulation patterns from pairs of initial queries and empirically stronger reformulations, consolidates them into a compact library of transferable reformulation patterns, and then selects an appropriate reformulation pattern for a new query given its retrieval context. The selected pattern constrains query reformulation to controlled operations such as sense disambiguation, vocabulary grounding, or discriminative facet addition, to name a few. As such, our proposed approach makes the reformulation policy explicit through these reformulation patterns, guiding the LLM towards targeted and effective query reformulations. Our extensive experiments on TREC DL 2019, DL 2020, and DL Hard show consistent improvements over classical feedback methods and recent LLM-based query reformulation and expansion approaches.

## 1 Introduction

Recently, Large Language Models (LLMs) have been adopted to generate reformulations and expansions of search queries [22, 5, 20, 11, 24, 10, 18]. Early neural methods produced document side expansions and synthetic queries such as doc2query [16, 6, 17], followed by approaches that synthesize query side textual surrogates such as `query2doc` [22, 28, 27, 26, 8]. These methods use generative capacity to bridge vocabulary gaps and often improve first stage recall or reranking quality [7, 28]. Yet they largely treat the generator as a black box that produces useful text without articulating what transformation is being applied or why a particular reformulation to the query may be helpful.

Building on this foundation, a second wave of work aimed to add structure and control to LLM-based reformulation [5, 23]. Generative Query Reformulation approaches use prompt-guided editing to improve effectiveness over classical baselines, with ensemble variants such as `GenQREnsemble` increasing coverage and stability through diverse prompts and sampling strategies [5]. Methods that integrate semantic generation with retrieval, such as `MUGI`, couple dense retrieval with generated augmentations to improve matching beyond lexical overlap [28]. Question answer-aware expansion, such as `QA-Expand`, further condition generation on answer cues to inject discriminative terms [20]. Despite these advances, generation often remains weakly constrained and lacks an explicit account of which transformation should be applied for a given query and context. This gap makes it difficult to guarantee stability across queries and to transfer successful reformulations on queries across collections.

In this paper, we move beyond text generation as an end in itself by placing *reformulation patterns* at the center of the query reformulation process. The

premise is that reformulations are effective when they realign the query's distribution with that of the relevant set and reduce ambiguity; a system should therefore learn the patterns of useful reformulations that achieve this alignment and select among them based on observable evidence. Rather than prompting an LLM to produce query reformulations directly, we first elicit concise *reformulation patterns* from pairs of initial queries and empirically stronger reformulations, consolidate these *patterns* into a small library of reusable reformulation patterns, and then choose the most appropriate *pattern* for a new query in light of retrieval context. Only after a *pattern* is selected do we generate the reformulation under that *pattern*'s constraints. This design preserves the strengths of LLM generation while making the reformulation policy explicit, and transferable across queries and collections by consistently applying the patterns.

Reformulation patterns offer two concrete benefits. They constrain query reformulations to patterns that lead to meaningful revisions to the query, such as 'sense disambiguation', 'controlled vocabulary grounding', or 'discriminative facet addition', which may reduce query drift and improve interpretability. It also enables targeted use of patterns to justify why a particular reformulation should be applied, improving stability relative to prompt-only methods. In broad terms, our proposed approach induces a library of reformulation patterns extracted based on past successful reformulations, selects a pattern for an unseen query based on its context from the pattern library, and generates a controlled query reformulation based on that pattern.

The major contributions of our work can be enumerated as follows: (1) A pattern-guided framework for query reformulation that induces and reuses a compact library of reformulation patterns derived from past successful query reformulations. (2) A context-aware mechanism that selects an appropriate *pattern* for reformulation before generation, improving stability and reducing query drift. (3) A controlled reformulation procedure that operationalizes selected patterns and produces reformulated queries. (4) A comprehensive empirical study that situates the method against classical feedback baselines and recent LLM-based approaches, demonstrating consistent gains over various datasets.

## 2 Proposed Approach

**Rationale for Proposed Approach.** Query reformulation is effective when reformulations move the query or its representation toward that of the relevant documents. Classical information retrieval explains this effect through models that reward alignment between query terms and the language of the relevant set, including probabilistic relevance modeling, BM25 scoring, and relevance models that estimate a term distribution for relevant items. Recent LLM-based approaches achieve this and improve effectiveness, yet they often generate query reformulations without specifying the transformation being applied or why it should help for a given query and context, which limits interpretability and can induce drift.

Our proposed `ReFormeR` approach addresses this gap by inducing a compact set of reusable reformulation patterns from pairs of initial queries and empir-

ically stronger reformulations. Each pattern captures a common reformulation approach such as "sense disambiguation", "controlled vocabulary grounding", or "discriminative facet addition", among others. At inference time, the system uses the retrieval context to select an appropriate pattern from the reformulation pattern library and then generates a pattern-guided reformulation. By making the reformulation policy explicit and reusable rather than implicit in prompts or term scoring heuristics, `ReFormeR` provides stable, interpretable, and context aware reformulations.

**Approach Formalization.** `ReFormeR` learns from pairs of queries and empirically stronger reformulations and turns these observations into a compact library of reusable reformulation patterns that encode common semantic edits such as replacing a colloquial variant with a controlled vocabulary term, adding a discriminative facet to resolve sense ambiguity, or expanding an acronym. Let $C$ be the document collection and let $R$ be a base retriever with scoring function $S(q, d)$ that ranks $d \in C$ for a query $q$. For any $q$, denote by $D_k(q)$ the top $k$ documents returned by $R$. Let $Q = \{(q_i, \tilde{q}_i)\}_{i=1}^{N}$ be training pairs where $\tilde{q}_i$ is a higher performing reformulation of $q_i$ on a disjoint development split. The objective is to induce a finite set $\mathcal{P} = \{p_1, \ldots, p_M\}$ of patterns that capture these recurrent reformulations. We obtain an induction mapping $g$ that assigns each pair to one pattern, $g(q_i, \tilde{q}_i) \in \mathcal{P}$, by eliciting a concise description of the reformulation with an LLM for each pair and clustering the resulting representations to produce non-overlapping groups. Writing $y_i = g(q_i, \tilde{q}_i)$ for the assigned label, we then train a context aware selector $s_\theta$ that, given a new query $q$ and its retrieval context $D_k(q)$, produces a distribution over patterns $\pi_\theta(p \mid q, D_k(q)) \in \Delta^{M-1}$. The selector uses the induced labels for supervision with loss as follows:

$$\mathcal{L}_{\text{sel}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log \pi_\theta\big(y_i \mid q_i, D_k(q_i)\big).$$

To generate a concrete reformulation, a conditional generator $G_\phi$ takes the query, its context, and a chosen pattern and returns a reformulated variant:

$$r = G_\phi\big(q, D_k(q), p\big).$$

The final retrieval query is the hybrid $q^\star = q \oplus r$, where $\oplus$ denotes concatenation that preserves the original phrasing while injecting the pattern-guided reformulation. Ranking proceeds with $S(q^\star, d)$ for $d \in C$. When graded relevance is available, let $\mathrm{Eff}(q)$ denote an effectiveness measure such as nDCG@10 for the ranking induced by $S(q, \cdot)$. Training seeks parameters that improve expected effectiveness under the selected pattern and the generated reformulation:

$$\max_{\theta,\phi} \; \mathbb{E}_q \; \mathrm{Eff}\big(q^\star\big) \quad \text{with} \quad p \sim \pi_\theta(\cdot \mid q, D_k(q)),\; r = G_\phi(q, D_k(q), p),\; q^\star = q \oplus r.$$

In practice, we optimize $\mathcal{L}_{\text{sel}}$ together with a supervised sequence loss for $G_\phi$ using $(q_i, D_k(q_i), y_i, \tilde{q}_i)$, which ties the rationale for reformulation to the concrete reformulation and yields a single process that explains *why* a reformulation should help and *how* it is applied.

**Prompt for Extracting Reformulation Patterns**

**System:** You are QueryReformulationLLM, an intelligent assistant that identifies and updates abstract patterns that describe how queries are reformulated to improve retrieval effectiveness.

**User:** Given a set of query reformulation pairs below and optional prior list of consolidated patterns, your objectives are:

1. Identify the transformation pattern(s) underlying each reformulation.
2. Consolidate the global pattern set by merging semantically similar strategies and refining their names and descriptions.

Query Reformulation Pairs: {query_pairs}
Consolidated Patterns: {existing_patterns}

Each extracted pattern should include a pattern name, an informative description, a generalized transformation rule, and representative examples. Return the results of consolidated patterns:

{"Consolidated Patterns": [...]}

Fig. 1: Overview of prompt used to extract/consolidate reformulation patterns.

## 3 Experiments and Results

All code, data, and prompts are publicly available on our GitHub Repository[1].

### 3.1 Experimental Setup

**Datasets and Metrics.** Our experiments evaluate the proposed framework on three benchmark datasets, namely TREC DL 2019 [4], TREC DL 2020 [3], and TREC DL Hard [14]. These datasets encompass queries with comprehensive graded relevance judgments designed to span diverse retrieval challenges. Retrieval performance is assessed using mAP@1000, nDCG@10, and Recall@1k, capturing the trade-off between early precision, global ranking quality, and coverage of relevant documents.

**Baselines.** We benchmark our framework against a set of SOTA query reformulation methods spanning traditional, feedback-based, and neural approaches. We include traditional pseudo-relevance feedback via RM3 [1, 12, 25] and Rocchio [19], as well as neural reformulation baselines comprising `GenQR` [23], an LLM-based keyword-oriented generative model, and `GenQREnsemble` [5], a prompt-ensemble variant designed to improve keyword coverage. In addition, we evaluate the integration of our pattern-guided framework, `ReFormeR`, with state-of-the-art context-based expansion and generation methods, including `QA-Expand` [20], which generates relevant questions and pseudo-passages to enrich query context; `Query2Doc` [22] in zero-shot (ZS), few-shot (FS), and chain-of-thought (CoT) configurations, which produces a pseudo-passage that capture the semantics of the query; and `MUGI` [28], a dense generation-augmented retriever.

**Implementation Details.** Retrieval was carried out using BM25 implemented by Pyserini [13], All query reformulation methods and experiments were conducted using the `Qwen2.5-7B-Instruct` model [21], running inference via the vLLM framework [9]. For `ReFormeR` implementation, the generation parameters were set to a maximum of 512 tokens and a decoding temperature of 1.0. Baseline results were reproduced following publicly available implementations and settings specified in prior work and where applicable, hyperparameters were adopted directly from published baselines.

[1] `https://github.com/queryreformulation/ReFormeR`

Table 1: Consolidated list of reformulation patterns extracted by `ReFormeR`.

| | | | |
|---|---|---|---|
| Clarify Intent | Clarify Subject | Conceptual Shift | Contextual Expansion |
| Contextual Restriction | Generalization | Location Specification | Purpose Specification |
| Semantic Clarification | Temporal Adjustment | | |

**Extracting Reformulation Patterns.** To derive effective reformulation patterns, we employ the *Diamond* subset of the *Matches Made in Heaven* collection released by [2]. This dataset comprises MS MARCO [15] train queries paired with their reformulated counterparts that achieve perfect retrieval effectiveness (MRR=1), thereby providing an ideal setting for analyzing the semantic transformations underlying successful query reformulations. For this purpose, we utilize `Qwen2.5-72B` as the underlying LLM for pattern induction. A total of 10,000 query–reformulation pairs were sampled from the dataset, and LLM was prompted to traverse these pairs, infer the underlying patterns that explains the improvement of each reformulation, and consolidate recurring rationales into a unified reformulation pattern library using the prompt shown in Figure 1. A summary of final list of consolidated patterns is available in Table 1. A more detailed version of this table is available on our GitHub.

### 3.2 Findings

Table 2 presents the retrieval effectiveness of `ReFormeR` compared with both keyword-based and context-aware reformulation approaches across TREC DL 2019, TREC DL 2020, and TREC DL Hard. For context-aware baselines, `ReFormeR` is integrated within their generation pipeline by embedding its pattern-guided reformulated queries into the original prompt context.

**Comparison with keyword-based reformulation.** textttReFormeR delivers consistent and meaningful improvements over both classical feedback and generative baselines. Against traditional methods such as `RM3` and `Rocchio`, `ReFormeR` achieves up to 16% higher nDCG@10 on TREC DL 2020 and about 26-34% higher mAP@1k on DL Hard, underscoring the advantage of pattern-guided reformulation over purely statistical expansion. While keyword-based feedback models tend to amplify frequent but noisy terms, `ReFormeR` introduces targeted semantic refinements that align better with the underlying retrieval intent based on the identified reformulation patterns. This advantage becomes more evident as query difficulty increases on DL Hard, where queries are sparse or ambiguous, pattern-guided reformulations provide the largest gains by improving precision without sacrificing recall.

Compared to keyword-based generative reformulation methods such as `GenQR` and `GenQREnsemble`, `ReFormeR` achieves consistent and substantial improvements, especially on the challenging TREC DL 2020 and DL Hard benchmarks. On TREC DL 2020, it improves nDCG@10 by 26% relative to `GenQREnsemble`, showing greater retrieval consistency through more targeted expansions. On DL Hard, `ReFormeR` yields 11% and 8% gains in mAP@1k and nDCG@10 over `GenQR`, demonstrating its strength in handling underspecified queries. Rather than generating unconstrained paraphrases or keyword additions, `ReFormeR` performs guided query reformulation resulting in more effective retrieval outcomes.

**Comparison with context-based reformulation.** When integrated with advanced context-aware reformulators, `ReFormeR` consistently boosts retrieval ef-

Table 2: Retrieval effectiveness in terms of mAP@1k, nDCG@10, and Recall@1k for keyword-based and context-based reformulation methods across three TREC datasets. ReFormeR-enhanced methods are highlighted.

| | DL 2019 | | | DL 2020 | | | DL Hard | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | mAP | nDCG | Recall | mAP | nDCG | Recall | mAP | nDCG | Recall |
| `BM25` | 0.290 | 0.497 | 0.745 | 0.288 | 0.488 | 0.803 | 0.164 | 0.290 | 0.678 |
| `BM25 + RM3` | 0.334 | 0.515 | 0.795 | 0.302 | 0.492 | 0.829 | 0.154 | 0.264 | 0.699 |
| `BM25 + Rocchio` | **0.347** | 0.528 | **0.801** | 0.312 | 0.491 | 0.816 | 0.164 | 0.277 | 0.704 |
| `GenQR` [23] | 0.341 | 0.517 | 0.797 | 0.336 | 0.551 | 0.840 | 0.186 | 0.313 | **0.719** |
| `GenQREnsemble` [5] | 0.278 | 0.442 | 0.792 | 0.287 | 0.453 | 0.795 | 0.122 | 0.197 | 0.651 |
| `ReFormeR` | 0.318 | **0.544** | 0.783 | **0.339** | **0.572** | **0.845** | **0.207** | **0.337** | 0.718 |

| | DL 2019 | | | DL 2020 | | | DL Hard | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | mAP | nDCG | Recall | mAP | nDCG | Recall | mAP | nDCG | Recall |
| `QA-Expand` [20] | 0.363 | 0.587 | 0.816 | 0.376 | 0.567 | 0.847 | 0.189 | 0.284 | 0.725 |
| + `ReFormeR` | **0.406** | **0.605** | **0.836** | **0.382** | **0.579** | **0.880** | **0.209** | **0.302** | **0.762** |
| `Query2Doc (ZS)` [22] | 0.426 | 0.632 | 0.861 | 0.379 | 0.579 | 0.881 | 0.219 | 0.348 | 0.783 |
| + `ReFormeR` | **0.438** | **0.651** | **0.865** | **0.410** | **0.618** | **0.883** | **0.227** | 0.342 | 0.756 |
| `Query2Doc (FS)` [22] | 0.409 | 0.596 | 0.841 | 0.394 | 0.604 | 0.876 | 0.215 | 0.334 | 0.782 |
| + `ReFormeR` | **0.449** | **0.677** | **0.886** | **0.413** | **0.626** | **0.890** | **0.233** | **0.365** | **0.799** |
| `Query2Doc (CoT)` [22] | 0.391 | 0.584 | 0.862 | 0.372 | 0.595 | 0.875 | 0.210 | 0.332 | 0.751 |
| + `ReFormeR` | **0.428** | **0.647** | 0.859 | **0.413** | **0.611** | **0.885** | **0.218** | **0.353** | **0.798** |
| `MUGI` [28] | 0.466 | 0.692 | 0.891 | 0.417 | 0.639 | 0.887 | 0.224 | 0.347 | 0.794 |
| + `ReFormeR` | 0.466 | 0.684 | 0.887 | **0.432** | **0.653** | **0.901** | **0.250** | **0.375** | **0.814** |

fectiveness across datasets. For instance, combining `QA-Expand` with `ReFormeR` improves mAP@1k by 12% on TREC DL 2019 and 11% on DL Hard, while enhancing recall from 0.847 to 0.880 on TREC DL 2020. Similarly, for `Query2Doc (FS)`, `ReFormeR` yields up to 13.6% higher nDCG@10 on TREC DL 2019 and 9% higher on DL Hard, reflecting stronger early ranking precision. The largest benefit emerges under more challenging retrieval settings, where coupling `MUGI` with `ReFormeR` achieves an mAP@1k of 0.250 and nDCG@10 of 0.375 on DL Hard. These correspond to relative gains of 11.6% and 8.1% over the baseline, establishing new best results among context-based models. These improvements reveal that pattern-guided reformulation is effective when contextual generation alone is insufficient. While context-aware models rely on document content, `ReFormeR` injects reformulation patterns that preserve query intent, prevent semantic drift, and enhance interpretability across diverse retrieval conditions.

## 4 Concluding Remarks

This paper presented `ReFormeR`, a pattern-guided approach that induces a compact, reusable library of query reformulation patterns and applies them to generate targeted query reformulations. By making the reformulation policy explicit through patterns, `ReFormeR` delivers consistent gains over feedback and LLM-based reformulation baselines on DL'19, DL'20, and DL-Hard. It also complements context-aware reformulation baseline methods, yielding further improvements when integrated with state of the art methods such as `QA-Expand`, `Query2Doc`, and `MUGI`.

## References


1. Abdul-Jaleel, N., Allan, J., Croft, W.B., Diaz, F., Larkey, L., Li, X., Smucker, M.D., Wade, C.: Umass at trec 2004: Novelty and hard (2004)
2. Arabzadeh, N., Bigdeli, A., Seyedsalehi, S., Zihayat, M., Bagheri, E.: Matches made in heaven: Toolkit and large-scale datasets for supervised query reformulation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 4417–4425 (2021)
3. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2020 deep learning track. CoRR **abs/2102.07662** (2021), `https://arxiv.org/abs/2102.07662`
4. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the trec 2019 deep learning track. arXiv preprint arXiv:2003.07820 (2020)
5. Dhole, K.D., Agichtein, E.: Genqrensemble: Zero-shot llm ensemble prompting for generative query reformulation. In: European Conference on Information Retrieval. pp. 326–335. Springer (2024)
6. Gospodinov, M., MacAvaney, S., Macdonald, C.: Doc2query–: when less is more. In: European Conference on Information Retrieval. pp. 414–422. Springer (2023)
7. Jagerman, R., Zhuang, H., Qin, Z., Wang, X., Bendersky, M.: Query expansion by prompting large language models. arXiv preprint arXiv:2305.03653 (2023)
8. Kassaie, B., Kane, A., Tompa, F.W.: Exploiting query reformulation and reciprocal rank fusion in math-aware search engines. In: Proceedings of the 2025 ACM Symposium on Document Engineering. pp. 1–10 (2025)
9. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J.E., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention. In: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles (2023)
10. Lai, Y., Wu, J., Wang, Z., Zhou, D.: Adarewriter: Unleashing the power of prompting-based conversational query reformulation via test-time adaptation. arXiv preprint arXiv:2506.01381 (2025)
11. Lai, Y., Wu, J., Zhang, C., Sun, H., Zhou, D.: Adacqr: Enhancing query reformulation for conversational search via sparse and dense retrieval alignment. arXiv preprint arXiv:2407.01965 (2024)
12. Lin, J.: The neural hype and comparisons against weak baselines. In: Acm sigir forum. vol. 52, pp. 40–51. ACM New York, NY, USA (2019)
13. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). pp. 2356–2362 (2021)
14. Mackie, I., Dalton, J., Yates, A.: How deep is your learning: The dl-hard annotated deep learning dataset. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2335–2341 (2021)
15. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: A human-generated machine reading comprehension dataset (2016)
16. Nogueira, R., Lin, J., Epistemic, A.: From doc2query to doctttttquery. Online preprint **6**(2) (2019)
17. Nogueira, R., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. arXiv preprint arXiv:1904.08375 (2019)

18. Ran, K., Alaofi, M., Sanderson, M., Spina, D.: Two heads are better than one: Improving search effectiveness through llm-generated query variants. In: Proceedings of the 2025 ACM SIGIR Conference on Human Information Interaction and Retrieval. pp. 333–341 (2025)
19. Rocchio Jr, J.J.: Relevance feedback in information retrieval. The SMART retrieval system: experiments in automatic document processing (1971)
20. Seo, W., Lee, S.: Qa-expand: Multi-question answer generation for enhanced query expansion in information retrieval. arXiv preprint arXiv:2502.08557 (2025)
21. Team, Q., et al.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 **2**(8) (2024)
22. Wang, L., Yang, N., Wei, F.: Query2doc: Query expansion with large language models. arXiv preprint arXiv:2303.07678 (2023)
23. Wang, X., MacAvaney, S., Macdonald, C., Ounis, I.: Generative query reformulation for effective adhoc search. arXiv preprint arXiv:2308.00415 (2023)
24. Wen, Q., Liu, Y., Zhang, J., Saad, G., Korikov, A., Sambale, Y., Sanner, S.: Elaborative subtopic query reformulation for broad and indirect queries in travel destination recommendation. arXiv preprint arXiv:2410.01598 (2024)
25. Yang, W., Lu, K., Yang, P., Lin, J.: Critically examining the" neural hype" weak baselines and the additivity of effectiveness gains from neural ranking models. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp. 1129–1132 (2019)
26. Yao, W., Wang, Y., Yu, Z., Xie, R., Zhang, S., Ye, W.: Pure: Aligning llm via pluggable query reformulation for enhanced helpfulness. In: Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 8721–8744 (2024)
27. Yu, W., Iter, D., Wang, S., Xu, Y., Ju, M., Sanyal, S., Zhu, C., Zeng, M., Jiang, M.: Generate rather than retrieve: Large language models are strong context generators. arXiv preprint arXiv:2209.10063 (2022)
28. Zhang, L., Wu, Y., Yang, Q., Nie, J.Y.: Exploring the best practices of query expansion with large language models. arXiv preprint arXiv:2401.06311 (2024)