GenCRF: Generative Clustering and Reformulation Framework for Enhanced Intent-Driven Information Retrieval

Wonduk Seo[1,2]* **Haojie Zhang**[1]* **Yueyang Zhang**[1]* **Changhao Zhang**[1,2]
Songyao Duan[1,2] **Lixin Su**[1] **Daiting Shi**[1] **Jiashu Zhao**[3] **Dawei Yin**[1]†

Baidu.inc[1] Peking University[2] Wilfrid Laurier University[3]
{seowonduk}@pku.edu.cn, {2301210522, duansy}@stu.pku.edu.cn
{zhanghaojie03, zhangyueyang, sulixin, shidaiting01, yindawei02}@baidu.com, {jzhao}@wlu.ca

Abstract

Query reformulation is a well-known problem in Information Retrieval (IR) aimed at enhancing single search successful completion rate by automatically modifying user's input query. Recent methods leverage Large Language Models (LLMs) to improve query reformulation, but often generate insufficient and redundant expansions, potentially constraining their effectiveness in capturing diverse intents. In this paper, we propose ***GenCRF: a Generative Clustering and Reformulation Framework*** to capture diverse intentions adaptively based on multiple differentiated, well-generated queries in the retrieval phase for the first time. GenCRF leverages LLMs to generate variable queries from the initial query using customized prompts, then clusters them into groups to distinctly represent diverse intents. Furthermore, the framework explores to combine diverse intents query with innovative weighted aggregation strategies to optimize retrieval performance and crucially integrates a novel Query Evaluation Rewarding Model (QERM) to refine the process through feedback loops. Empirical experiments on the BEIR benchmark demonstrate that GenCRF achieves state-of-the-art performance, surpassing previous query reformulation SOTAs by up to 12% on nDCG@10. These techniques can be adapted to various LLMs, significantly boosting retriever performance and advancing the field of Information Retrieval.

1 Introduction

Query reformulation is a well-known problem in Information Retrieval (IR) to enhance search effectiveness by automatically modifying the initial query into well-formed one(s)(Carpineto and Romano, 2012). Traditional Pseudo-Relevance Feedback (PRF) based methods, such as RM3, improve the initial query by selecting terms from relevant documents (Robertson, 1991; Lavrenko and Bruce, 2001). Similarly, researchers expand initial queries by incorporating semantically similar terms with pre-trained word embeddings (Kuzi et al., 2016; Roy et al., 2016; Zamani and Croft, 2016). With the advent of Large Language Models (LLMs), query reformulation has re-emerged as a prominent research area within the field of information retrieval (Zhao et al., 2023). In contrast to past methods that relied on using existing related terms in the retrieval system for expansion, the current approaches to query reformulation harness the exceptional generative understanding abilities of LLMs (Wang et al., 2023a; Li et al., 2023). They leverage foundational LLM techniques such as prompt engineering and Chain-of-Thought (CoT) to enhance initial queries by generating keywords and detailed descriptions (Wei et al., 2022; Jagerman et al., 2023). However, these methods often face limitations in enriching information capacity through single expansions.

More recently, ensemble approaches utilizing multiple prompts to generate various keywords have emerged, demonstrating improved performance compared to earlier single expansion methods (Li et al., 2023; Dhole and Agichtein, 2024; Dhole et al., 2024). Although these methods demonstrate the benefits of utilizing various expansions to enrich original queries and improve retrieval effectiveness, these methods face several challenges: ① The variations in their prompts tend to be simplistic and homogeneous prompt variations, lacking effective methods to capture the diverse user intents from multiple perspectives, ② These methods primarily lack of dynamic assessment of intent importance and query relevance, ③ There is a lack of effective mechanisms to detect generation quality, potentially introducing negative biases in query performance.

To overcome these limitations, we propose GenCRF: a Generative Clustering and Reformulation Framework. Unlike previous methods that gen-

*equal contribution.
†corresponding author.

erate keywords or documents, GenCRF directly leverages LLMs to generate multiple differentiated queries derived from the original input by utilizing various types of customized prompts. Through detailed analysis and observation, we identified several query expansion types and designed customized prompts: "contextual expansion," "detail specific," and "aspect specific". GenCRF then dynamically clusters these queries to capture diverse intents, minimizing information redundancy and maximizing the potential of query reformulation.

In order to efficiently integrate abundant and diversified multi-intent queries, GenCRF incorporates several weighted aggregation strategies, including similarity-based dynamic weighting *(GenCRF/SimDW)* and score-based dynamic weighting *(GenCRF/ScoreDW)*, to adjust the relative weights of reformulated queries based on various criteria and efficiently integrate diverse multi-intent queries. To further enhance performance, we introduce a fine-tuning step *(GenCRF/ScoreDW-FT)* that optimizes the model's ability to evaluate and score reformulated queries. Ultimately, we introduce the Query Evaluation Rewarding Model *(QERM)*, which evaluates clustering generation quality and guides query refinement through a feedback loop. QERM subsequently guides the LLMs to either continue refining the queries or conclude the process as appropriate, ensuring optimal refinement and high-quality query formulation.

Extensive experiments on the BEIR dataset (Thakur et al., 2021) through competitive LLMs demonstrate GenCRF 's consistent superiority over state-of-the-art query reformulation techniques across diverse domains and query types. Comprehensive analyses of initial query weight, prompt quantity, number of generated queries and QERM iteration count further validate GenCRF's effectiveness. Our investigations confirm GenCRF's robustness, capacity to generate highly diverse results, and ability to effectively cluster and retrieve a wide spectrum of intents. These findings not only validate our approach but also offer valuable insights for future information retrieval research.

2 RELATED WORK

Numerous methods have been applied for query reformulation, which has significantly evolved over the years, adapting to new methodologies in information retrieval (IR). Early approaches relied on classical retrieval models such as BM25 (Robertson et al., 1994), which focused on exact matching statistical features including term frequency and document length to assess relevance. These methods often utilize techniques such as RM3 and query logs for pseudo-relevance feedback (Robertson, 1991; Lavrenko and Bruce, 2001; Jones et al., 2006; Craswell and Szummer, 2007). Neural networks have provided a new perspective on developing more sophisticated methods for query reformulation. Grbovic et al. (2015) proposed a rewriting method based on a query embedding algorithm, and Nogueira et al. (2017) also explored reinforcement learning-based models. Dense neural networks further advanced the field of query reformulation, with pre-trained embeddings, capturing complex semantics and facilitating transfer learning in IR tasks (Devlin et al., 2019; Xiong et al., 2021).

More recently, Large Language Models (LLMs) have significantly transformed query reformulation strategies. Weller et al. (2024) emphasized the potential of LLMs to utilize their ability including query reformulation, and showed that LLMs outperform traditional methods for query expansion. Generating keywords and pseudo documents such as Query2Doc (Q2D) (Wang et al., 2023a) and Query2Expansion (Q2E) (Jagerman et al., 2023) have shown their effectiveness in improving retrieval quality (Nogueira et al., 2019; Claveau, 2022; Wang et al., 2023b). Although those methods have shown promise in query reformulation to some extent, they often rely on a single model or prompt. To address the limitations of previous query reformulation methods, recent studies found that applying multiple different prompts to generate various keywords or documents could further boost the overall quality of query reformulation, as they can provide a certain degree of information gain for retrieval queries more closely, thereby effectively capturing a broader range of user intent. (Li et al., 2023; Dhole and Agichtein, 2024).

Despite their improvements, above methods still face formidable challenges. They often tend to employ simplistic prompt variations that may not adequately capture the breadth of diverse user intents, resulting in redundant keyword generations that undermine the effectiveness of query reformulation. Moreover, their ensemble techniques frequently fall short in appropriately emphasizing the significance of various intents and in dynamically weighting the relevance between initial queries and reformulated ones. There is also a noticeable absence of robust mechanisms to evaluate the quality

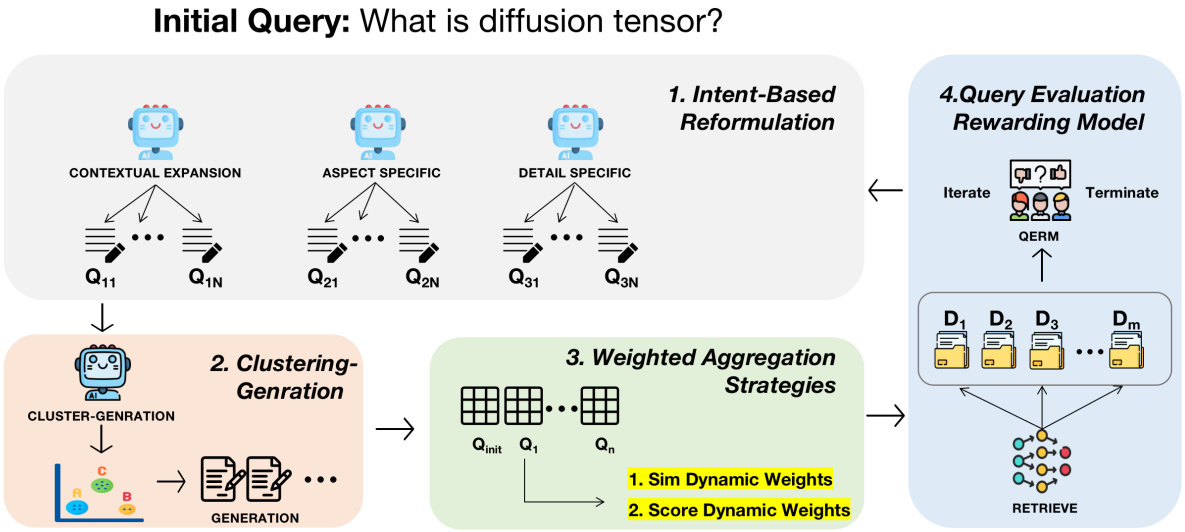

Figure 1: Overview of the GenCRF: Generative Clustering and Reformulation Framework

of the generated outputs, which can result in the inclusion of semantically ambiguous terms, ultimately detracting from the overall performance.

3 METHODOLOGY

In this section, we first provide a comprehensive overview of our innovative ***Generative Clustering and Reformulation Framework (GenCRF)*** (Section 3.1), followed by our specific Generation & Clustering settings and a comparative analysis with existing methods (Section 3.2). We then present weighted aggregation and fine-tuning strategies to optimize retrieval performance (Section 3.3). Finally, we introduce our Query Evaluation Rewarding Model (QERM), desgined to further enhance GenCRF's performance through intent-driven query capture and critical feedback for query re-generation and re-clustering (Section 3.4).

3.1 Overview of GenCRF

To construct the GenCRF, we first utilize LLMs to reformulate the initial query q_{init}, into a new form q_{new}. This process generates N queries for each of the 3 diverse customized prompts in set P:

$$\mathcal{Q}_{\text{gen}} = \bigcup_{\text{prompt} \in P} \{R(q_{\text{init}}, \text{prompt})\} \quad (1)$$

In this equation, $\mathcal{Q}_{\text{gen}}$ represents the set of all generated queries. The reformulation LLM, denoted as R, applies each prompt in P to the initial query. To reduce information redundancy and capture diverse intents, we introduce a clustering step in our framework. This step dynamically clusters generated queries into several intentional groups and produces a representative, comprehensive query for each cluster:

$$\mathcal{Q}_{\text{final}} = \text{G}(q_{\text{init}}, \mathcal{Q}_{\text{gen}}) \quad (2)$$

The function G clusters the set of generated queries $\mathcal{Q}_{\text{gen}}$ into 1 to 3 groups dynamically. It then generates a new representative query for each cluster, resulting in the set $\mathcal{Q}_{\text{final}}$. This procedure ensures comprehensive coverage of derived query intents beyond the original query, while eliminating similar or redundant queries. Following the clustering step, the framework proceeds with a retrieval process. This process combines various weighted aggregation strategies designed to effectively capture both q_{init} and $\mathcal{Q}_{\text{final}}$:

$$\mathcal{D}_{\text{retrieval}} = \text{Retrieve}(\mathcal{Q}_{\text{final}}, q_{\text{init}}, \mathcal{W}) \quad (3)$$

where $\mathcal{W}$ represents the weighting parameters used in the aggregation strategies and $\mathcal{D}_{\text{retrieval}}$ is the set of final retrieved documents. To further enhance its performance, the GenCRF framework incorporates a novel Query Expansion Rewarding Model (QERM), which detects the superiority of clustered intent-driven queries and provides effective feedback to LLMs, signaling when re-generation and re-clustering are necessary. The overall pipeline of the GenCRF is shown in Figure 1.

3.2 Generation and Clustering

Prompts used by current baselines, such as Query2Doc (Q2D) and Query2Expansion (Q2E), typically instruct models to produce relevant keywords or documents without considering the inherent intent and underlying value of the query.

Moreover, these prompts often exhibit simplicity and homogeneity, so that lack of the depth required to effectively capture diverse user intents.

Through comprehensive analysis and observation, we have identified several distinct query expansion intents: *Contextual Enrichment* broadens queries with relevant context; *Detail-Oriented Exploration* focuses on specific subtopics; *Aspect-Focused Expansion* concentrates on particular facets; *Clarification-Focused Refinement* clarify ambiguities; and *Exploratory Intent* investigates related but unexplored areas. From these observations, we devise three types of tailored and effective intents to diversify generated queries from multiple perspectives as follows:

1. ***Contextual Expansion:*** Expands the initial query's context while maintaining clarity, ensuring comprehensive understanding and generating more relevant, refined reformulations.
2. ***Detail Specific:*** Elicits specific details or subtopics within the query, providing focused insights and enhancing the granularity of retrieved information.
3. ***Aspect Specific:*** Concentrates on a specific aspect or dimension of the topic, broadening the query's scope while focusing on the target dimension to enrich result diversity.

To further enhance query diversity while maintaining focus on core intent, we propose a clustering generation prompt to guide the LLM to explore multi-type demands, as follows:

4. ***Clustering-Generation:*** Extracts up to three intent queries from differentiated queries in GenCRF, enriching the query reformulation process, improving overall query intent understanding and reformulation strategies.

3.3 Weighted Aggregation Strategies

In order to optimize retrieval performance by effectively capturing both q_{init} and reformulated queries from $\mathcal{Q}_{\text{final}}$, we introduce two distinct weighted aggregation strategies and a fine-tuning process.

Similarity Dynamic Weights (SimDW). This novel strategy dynamically adjusts the weights of reformulated queries based on their similarity to q_{init}, while incorporating a filtering mechanism to ensure relevance. After assigning a fixed weight to the initial query, the method considers only those reformulated queries exceeding a predefined similarity threshold in the dynamic weighted aggregation. The aggregation equation is given by:

$$q_{\text{agg}}^{\text{simDW}} = w_0 \cdot q_{\text{init}} + \sum_{\substack{i=1 \\ \text{sim} \geq \theta}}^{|Q_f|} \text{sim}(q_{\text{init}}, q_{f,i}) \cdot q_{f,i} \quad (4)$$

where w_0 is the fixed weight for the initial query; sim represents a dynamic weight estimating the relative magnitude of $q_{\text{f},i}$, calculated as the cosine similarity between the embeddings of the initial query and the i-th reformulated query $q_{\text{f},i}$ using a sentence embedding model; and θ is the similarity threshold for filtering irrelevant queries.

Score Dynamic Weights (ScoreDW). Building upon the SimDW approach, the ScoreDW strategy offers a more comprehensive evaluation of reformulated queries by employing a multidimensional scoring system to assess query quality, using these scores as dynamic weights in the aggregation process. The method retains the fixed weight for the initial query and the filtering mechanism from SimDW, but enhances the evaluation criteria. The aggregation equation for ScoreDW is expressed as:

$$q_{\text{agg}}^{\text{scoreDW}} = w_0 \cdot q_{\text{init}} + \sum_{\substack{i=1 \\ \text{score} \geq \theta}}^{|Q_f|} \text{score}(q_{\text{init}}, q_{f,i}) \cdot q_{f,i} \quad (5)$$

Specifically, score is a dynamic weight representing the estimated importance of each $q_{\text{f},i}$, derived from an LLM's evaluation of the reformulated query relative to the initial query. The evaluation considers five key dimensions: Relevance, Specificity, Clarity, Comprehensiveness, and Usefulness for retrieval. The threshold θ ensures that only high-scoring, pertinent reformulations contribute to the final aggregated query.

Fine-Tuning for ScoreDW. For the purpose of optimizing the ScoreDW strategy, we implement a fine-tuning process for the LLMs to enhance their precision in evaluating and scoring reformulated queries. The process begins with the generation of a diverse set of query pairs $(q_{\text{init}}, q_{\text{ref}})$ using each LLM, where q_{ref} is the reformulated query. These pairs are then evaluated by GPT-4o, serving as a high-quality benchmark, to produce reference scores. The fine-tuning objective is formulated as:

$$\phi^* = \arg\min_{\phi} \sum_{i=1}^{N} \mathcal{L}(\text{LLM}_\phi(q_{\text{init},i}, q_{\text{ref},i}), s_i) \quad (6)$$

where ϕ^* denotes the optimal LLM parameters, $(q_{\text{init},i}, q_{\text{ref},i})$ represents the i-th query pair, s_i is the

corresponding score generated by GPT-4o, and $\mathcal{L}$ is the loss function. Fine-Tuning process aims to enhance the LLM's ability to discriminate between high and low-quality reformulations, ensuring consistent and scalable query quality assessment.

3.4 Query Evaluation Rewarding Model

To further improve the performance of GenCRF, we also introduce a novel approach: the Query Evaluation Rewarding Model (QERM). This innovative model functions as a multi-intent gain detection model that assesses the quality and effectiveness of queries generated by GenCRF, focusing on their alignment with diverse, intent-driven clusters. QERM evaluates how well generated queries capture user intent and contribute to meaningful query clusters. It provides feedback to LLMs for re-generation and re-clustering if necessary, addresses limitations in initial scoring.

Algorithm 1 Query Evaluation Rewarding Model

Require: nDCG threshold τ, output logit threshold ε, training dataset with Queries $Q = \{q_1, q_2, \ldots, q_n\}$, Maximum Iteration M

// Construct training dataset for reward model
for each $q \in Q$ **do**
    Implement Generation, Clustering and Weighted Aggregation in the GenCRF for q
    Compute nDCG@10(q) from retrieval documents
    if nDCG@10(q) $< \tau$ **then**
        return label(q) $\Leftarrow 0$
    else
        return label(q) $\Leftarrow 1$
    end if
end for
// Training
Train Reward Model with labeled datasets to assess the superiority of clustered intent-driven queries
// Inferring
Initialize timestep $t \Leftarrow 0$
while $t < M$ **do**
    Provide feedback from the output logit produced by reward model
    if output logit $< \varepsilon$ **then**
        Implement re-Generation and re-Clustering in the GenCRF
    else
        return retrieval results
    end if
    $t \Leftarrow t + 1$
end while

QERM calculates nDCG@10 scores for each query, assigning labels based on a threshold (ε). Queries below the threshold are labeled as "0" for re-generation, while those above are labeled as "1", denoting satisfactory performance expected. A language model is then trained on these labeled queries to guide query refinement decisions. The trained reward model is subsequently used to infer the query quality in the test set, providing critical feedback for re-generation and re-clustering as described in Algorithm 1, thereby ensuring high query quality standards and improving retrieval performance.

4 EXPERIMENTS

4.1 Setup

We detail the experimental configuration, including datasets, baseline methods, and model specifications. We also detail the prompts used and specific parameters for each component of our framework.

4.1.1 Experimental Datasets

We conduct our main experiments on six datasets from the BEIR benchmark (Thakur et al., 2021) to evaluate retrieval performance: *SciFact*, *TREC-COVID*, *SciDOCS*, *NFCorpus*, *DBPedia-entity*, and *FiQA-2018*. For ablation studies and parameter analysis, we used two additional datasets: *ArguAna* and *CQADupStack-English*. The *Quora* dataset is utilized for both constructing scoring data in the Fine-Tuning process and training our Query Evaluation Rewarding Model (QERM).

4.1.2 Models Used

LLMs: We employ Mistral-7B-Instruct-v0.3 and Llama-3.1-8B-Instruct models (Jiang et al., 2023; Touvron et al., 2023), with temperature 0.8 and top_p 0.95 for diverse outputs. GPT-4o (OpenAI, 2024) is used to generate high-quality reference scores for fine-tuning. We apply full-parameter fine-tuning to both models, using a learning rate of $1e-5$ for 5 epochs, with a batch size of 16.

Similarity Model: SentenceBERT (all-mpnet-base-v2) (Reimers and Gurevych, 2019) is used to generate embeddings of initial query and generated queries. These embeddings are then used to calculate the cosine similarity between them within the GenCRF framework. We set a similarity threshold $\theta = 0.2$ to filter out irrelevant queries.[1]

Retrieval Model: MSMARCO-DistilBERT-base-TAS-B model is used for our retrieval step, which is specifically designed for Dense Passage Retrieval and trained on the MSMARCO passage dataset (Campos et al., 2016), featuring 6-layer DistilBERT architecture optimized for retrieval.

[1] Similarity Threshold analysis in Appendix C.1.

Model	Approach	Methods	scifact	trec-covid	scidocs	nfcorpus	dbpedia-entity	fiqa	Avg.
Mistral	non-fine-tuned	Q2D	0.5966	0.5669	0.1316	0.2964	0.3750	0.2645	0.3718
		Q2E	0.5895	0.5941	0.1267	0.2863	0.3200	0.2677	0.3641
		CoT	0.5986	0.5978	0.1314	0.2916	0.3740	0.2418	0.3725
		GenQRE	0.6306	0.6062	0.1490	0.2844	0.3688	0.2353	0.3791
		GenQRFusion	0.6370	0.5474	0.1406	0.3068	0.3662	0.2701	0.3780
		GenCRF/SimDW†	0.6629*	0.6447*	0.1529*	0.3268	0.3867*	0.3083*	0.4137*
		GenCRF/ScoreDW†	0.6574*	0.6451*	0.1521	0.3267*	0.3862	0.3068*	0.4124*
	fine-tuned	GenCRF/ScoreDW-FT†	0.6596*	0.6527*	0.1523	0.3270	0.3873*	0.3077*	0.4143*
		GenCRF/ScoreDW-FT-QREM†	**0.6637***	**0.6587***	**0.1534***	**0.3294***	**0.3905***	**0.3101***	**0.4176***
Llama	non-fine-tuned	Q2D	0.6082	0.6032	0.1459	0.2613	0.2833	0.2419	0.3573
		Q2E	0.6385	0.5987	0.1460	0.3001	0.3554	0.2807	0.3866
		CoT	0.5661	0.5847	0.1223	0.2552	0.2995	0.2168	0.3408
		GenQRE	0.5093	0.4786	0.0978	0.2459	0.2195	0.1545	0.2843
		GenQRFusion	0.6273	0.5474	0.1406	0.3109	0.3762	0.2889	0.3819
		GenCRF/SimDW†	0.6517*	0.6939*	0.1505	0.3274*	0.3897*	0.3126	0.4210*
		GenCRF/ScoreDW†	0.6561*	0.6824*	0.1539*	0.3286	0.3903*	0.3147*	0.4210*
	fine-tuned	GenCRF/ScoreDW-FT†	0.6566*	0.6889	0.1552*	0.3336*	0.3913	0.3164*	0.4237*
		GenCRF/ScoreDW-FT-QREM†	**0.6613***	**0.6952***	**0.1557***	**0.3357***	**0.3940***	**0.3199**	**0.4270***

Table 1: nDCG@10 scores for GenCRF compared with multiple baselines across six datasets from the BEIR benchmark. Bold text for the best performance, underlined text for the second best. * denotes significant improvements (paired t-test with Holm-Bonferroni correction, $p < 0.05$) over the indicated baseline model(s). † denotes our proposed methods.

Query Evaluation Rewarding Model: We use RoBERTa-Large Model (Liu et al., 2019) as QERM's backbone for its robustness in NLP tasks. The training uses a learning rate of $1e-5$ for 4 epochs, with a maximum of 2 iterations. The output logit threshold (ε) is set to the mean of the first iteration logits, ensuring an adaptive and contextually relevant baseline for query quality evaluation.

4.1.3 Baseline Methods

We compare our method against several established competitive baselines. For non-fusion methods, queries are structured as "initial query [SEP] generated query", where [SEP] is a separator token. The baseline methods include: Query2Doc (***Q2D***): Generate pseudo-documents for query expansion (Wang and andFuru Wei, 2023); Query2Expansion (***Q2E***): Expand queries with relevant keywords (Jagerman et al., 2023); Query2CoT (***Q2C***): Apply Chain of Thoughts for query reformulation (Wei et al., 2022); GenQREnsemble (***GenQRE***): Use multiple prompts to generate and concatenate keywords with initial query(Dhole and Agichtein, 2024); ***GenQRFusion***: Extend GenQREnsemble with keyword fusion method.

4.1.4 Prompts Used

Baseline methods utilize varying numbers of prompts: Q2D, Q2E, and Q2C each use four few-shot prompts, GenQRE uses ten (Dhole and Agichtein, 2024), and GenQR-Fusion randomly selects three prompts with a fusion strategy (Dhole et al., 2024). Our framework utilizes five types of prompts: three for diverse query generation (*Contextual Expansion*, *Detail Specific*, *Aspect Specific*), one for *Clustering-Generation*[2], and one for *Scoring*. The Scoring prompt evaluates generated queries based on Relevance, Specificity, Clarity, Comprehensiveness, and Usefulness, assigning scores from 1 to 100, with a threshold $\theta = 60$ to ensures only high-scoring reformulations contribute to the final aggregated query.[3] Detailed descriptions of all prompts are provided in the appendix.

4.2 Experimental Analysis

In our experiments, we evaluated the performance of our proposed GenCRF framework across six datasets from the BEIR benchmark, as shown in Table 1. Ensemble-based methods such as GenQRE and GenQRFusion outperforms single prompted methods on average, with GenQRFusion demonstrating particularly strong results. This indicates that ensemble based apporaches using multiple prompts to expand retrieval queries enhance information gaining and improve retrieval performance.

However, our proposed GenCRF methods, such as SimDW and ScoreDW, further improve upon these ensemble-based approachs. Our strategies consistently outperform GenQRE and GenQRFusion across all datasets. This result demonstrates

[2]Cluster analysis in Appendix D.1. and Appendix D.2.
[3]Score Threshold analysis in Appendix C.2.

the effectiveness of both our multi-intent query generation and dynamic weight aggregation techniques, offering an effective approach compared to static weighting strategies in advance. We provide a more detailed analysis and examination of these two components in Section 4.3.1. and Section 5.

Additionally, the fine-tuned method ScoreDW-FT demonstrates stronger performance across all datasets, indicating fine-tuning process enhances the LLM's consistent and scalable quality assessment. Moreover, ScoreDW-FT-QERM consistently achieves the best results among all methods. It effectively guides the LLMs in the query refinement process by iteratively assessing GenCRF based on nDCG@10 scores, thereby enhancing the overall adaptability of our GenCRF framework. The improvements are most pronounced in trec-covid-beir and dbpedia-entity, highlighting the robustness of our approach across various retrieval tasks.

4.3 Ablation Studies

To validate GenCRF's robustness, we conduct ablation studies on key parameters using *ArguAna* and *CQADupStack-English* datasets. For comparison with other methods, particularly those that do not use weighted aggregation, we introduce Direct Concatenation (DC) method:

$$q_{\text{agg}}^{\text{DC}} = q_{\text{init}} + [\text{SEP}] + \sum_{i=1}^{|\mathcal{Q}\text{final}|} (q_{\text{final},i} + [\text{SEP}]) \quad (7)$$

DC combines the initial query q_{init} with all reformulated queries using [SEP] tokens as separators. Also, we introduce Fixed Weights (FW) method for ablation study:

$$q_{\text{agg}}^{\text{FW}} = w_0 \cdot q_{\text{init}} + \frac{1 - w_0}{|\mathcal{Q}\text{final}|} \sum_{i=1}^{|\mathcal{Q}\text{final}|} q_{\text{f},i} \quad (8)$$

Here, w_0 is the fixed weight for the initial query, and $(1 - w0)/|\mathcal{Q}_{\text{final}}|$ is the equal weight applied to each reformulation query.

4.3.1. Initial Query Weight. To determine the optimal weight for Weighted Aggregation, we investigate on both Fixed Weights (FW) and ScoreDW/FT strategies. We vary w_0 from 0.3 to 0.9 in 0.1 increments, evaluating nDCG@10 for both strategies. As shown in Figure 2, $w_0 = 0.7$ achieves optimal performance across both strategies and datasets. For fairness in building baseline, we applied this optimal $w_0 = 0.7$ to GenQRFusion method as well.

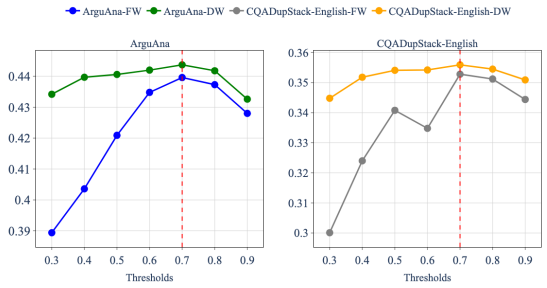

Figure 2: Initial Weight Comparison of FW and DW

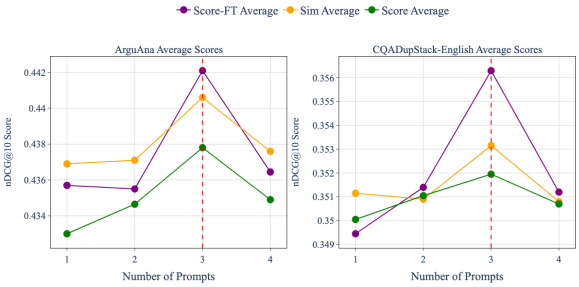

Figure 3: nDCG@10 scores for different prompt quantities

4.3.2. Impact of Prompt Quantity. We investigate the effect of varying the number of prompt types on retrieval performance, measured by nDCG@10 scores. The prompts included "contextual expansion," "detail specific," "aspect specific," and "clarity enhancement."[4] We evaluated all possible combinations of these prompts and calculated the average performance for each number of prompts used. As shown in Figure 3, performance typically improves when increasing from 1 to 3 prompts, but adding a fourth prompt does not lead to further enhancements and conversely decreases performance. Thus, we selected 3 prompt types as the optimal configuration for maximizing retrieval performance in this study.

Datasets	Gen(N)	SimDW	ScoreDW-FT
ArguAna	1	0.4361	0.4367
	2	**0.4395**	**0.4421**
	3	0.4366	0.4395
	4	0.4351	0.4369
CQADupStack-English	1	0.3251	0.3478
	2	**0.3543**	**0.3563**
	3	0.3482	0.3498
	4	0.3491	0.3497

Table 2: Comparison of nDCG@10 scores for different numbers of generated queries N using SimDW and ScoreDW-FT strategies.

[4]Clarity Prompt in Appendix B.3.

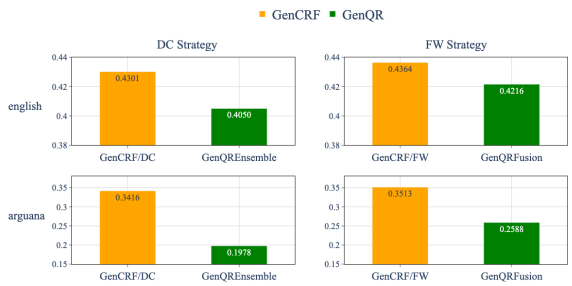

Figure 4: Performance comparison of GenCRF with GenQR using DC and FW strategies

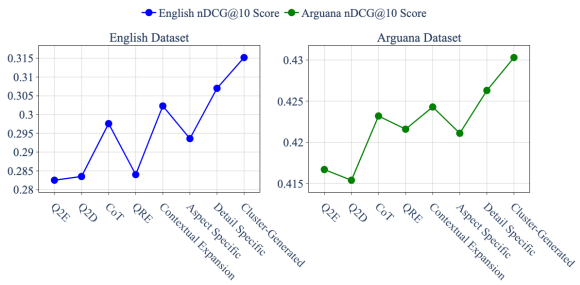

Figure 5: Comparsion of nDCG@10 scores between GenCRF's prompts and baseline methods

4.3.3. Impact of Generated Query Count.

We explore the effect of varying the number of generated queries N per prompt on retrieval performance, as described in Section 3.2. Experiment were conducted with N ranging from 1 to 4 for both SimDW and ScoreDW-FT strategies across *ArguAna* and *CQADupstack-English* dataasets. As shown in Table 2, generating 2 queries per prompt consistently yields the best performance across both datasets and strategies. The performance decline for N suggests that additional generation may introduce noise or redundancy, which may be attributed to the excessive length of single-prompt generated responses or their mutual interference.

4.3.4. Iterative Optimization with QERM.

We examine the impact of iterative optimization using the Query Evaluation Rewarding Model (QERM) on our ScoreDW-FT-QERM framework, with iteration counts ranging from 1 to 4, as shown in table 3. We observe that the best iteration count is 2, significantly surpassing the score where iteration count is 4. The result indicate that the integration of QERM with two iterations achieve an optimal result, allowing the ScoreDW-FT-QERM framework to adaptively optimize query generation and clustering, resulting in more precise and relevant retrieval outcomes across diverse datasets.

Datasets	Iteration	nDCG@10
ArguAna	1	0.4397
	2	**0.4421**
	3	0.4369
	4	0.4358
CQADupStack-English	1	0.3527
	2	**0.3563**
	3	0.3540
	4	0.3536

Table 3: nDCG@10 scores for different QERM iteration counts using ScoreDW-FT-QERM.

5 Generation study and Discussions

Our GenCRF has demonstrated superior performance in baseline comparisons, effectively capturing query intent compared to existing methods. Figure 4 reveals GenCRF outperforming GenQR methods, even with basic aggregation strategies such as Direct Concatenation (DC) and Fixed Weights (FW). While GenQRFusion relies on keyword-based methods that often fails to capture the underlying query intent, GenCRF's prompts explore various query facets, resulting in more comprehensive reformulations that capture nuances keyword-based methods neglect.

As shown in Figure 5, our individual prompts (*Contextual Expansion, Detail Specific and Aspect Specific*) outperform baseline methods such as Q2E, Q2D and CoT. Our prompts capture deeper query semantics, contrasting with conventional methods' focus on surface-level information. Notably, our Cluster-Generated method, which combines diverse insights from various prompts, achieves the best results, demonstrating the effectiveness of integrating multiple perspectives in query reformulation-an approach absent in single-prompt methods.

6 Conclusion

We present the Generative Clustering and Reformulation Framework (GenCRF), which demonstrates significant advancements over existing competitive baseline methods, achieving up to 12% increase on BEIR benchmark. Our approach combines diverse prompting strategies and clustering refinement to accurately capture and reformulate query intents. We introduced our optimization techniques including weighted aggregation methods: *SimDW*, *ScoreDW*, *ScoreDW-FT* and the evaluation rewarding model *QERM*, enhancing GenCRF's performance and offering a more precise, user-centric

information retrieval experience. Extensive ablation studies have confirmed the reasonableness and robustness of the GenCRF framework by exploring key parameters and settings across datasets from the BEIR benchmark. Future work could explore GenCRF's application to real-world search scenarios, potentially enhancing its effectiveness in practical information retrieval contexts.

References

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg an d Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, L. Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@NIPS. 4 November 2016*.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. In *ACM Computing Surveys (CSUR), Volume 44, Issue 1*.

Vincent Claveau. 2022. Neural text generation for query expansion in information retrieval. In *WI-IAT '21: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*.

Nick Craswell and Martin Szummer. 2007. Random walks on the click graph. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North Association for Computational Linguistics*.

Kaustubh D. Dhole and Eugene Agichtein. 2024. Genqrensemble: Zero-shot llm ensemble prompting for generative query reformulation. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part III*.

Kaustubh D. Dhole, Ramraj Chandradevan, and Eugene Agichtei. 2024. Generative query reformulation using ensemble prompting, document fusion, and relevance feedback. ArXiv preprint.

Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. 2015. Context- and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. ArXiv preprint.

Albert Q. Jiang, Alexandre Sablayrolles, Chris Bamford Arthur Mensch, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. ArXiv preprint.

Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*.

Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *CIKM '16: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*.

Victor Lavrenko and W. Bruce. 2001. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*.

Minghan Li, Honglei Zhuang, Kai Hui, Zhen qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. 2023. Can query expansion improve generalization of strong cross-encoder rankers? In *SIGIR '24: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. ArXiv preprint.

Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. In *arXiv preprint arXiv:1904.08375*.

OpenAI. 2024. Chatgpt-4o.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

S. E. Robertson. 1991. On term selection for query expansion. In *Journal of Documentation, Volume 46, Issue 4*.

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*.

Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. In *Neu-IR '16 SIGIR Workshop on Neural Information Retrieval July 21, 2016, Pisa, Italy*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Naman Goyal Baptiste Rozière, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. ArXiv preprint.

Liang Wang and Nan Yang andFuru Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

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*.

Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2023b. Generative query reformulation for effective adhoc search. In *The First Workshop on Generative Information Retrieval*.

Jason Wei, Xuezhi Wang, Maarten Bosma Dale Schuurmans, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2024. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Paul Bennett Jialin Liu, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *ICLR 2021 Poster*.

Hamed Zamani and W. Bruce Croft. 2016. Embedding-based query language models. In *ICTIR '16: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. ArXiv preprint.

A Appendix A. Overview

This appendix provides a comprehensive overview of the methodologies and experimental setups employed in our study. We detail the prompts used in our baseline models and our GenCRF framework, including those used for ablation studies. Additionally, we present our methods for finding optimal simlilarity and score thresholds, also conduct a cluster anaylsis within the GenCRF framework.

B Appendix B. Prompts

We share five prompts utilized in our experiments, including: Query2Doc (***Q2D***): Generate pseudo-documents and expands queries (Wang and andFuru Wei, 2023); Query2Expansion (***Q2E***): Expand queries with relevant keywords (Jagerman et al., 2023); Query2CoT (***Q2C***): Reformulate queries based on Chain of Thoughts prompting (Wei et al., 2022); GenQREnsemble (***GenQRE***): Applies multiple prompts to generate various keyword sets concatenated within the initial query(Dhole and Agichtein, 2024) and GenCRF (Ours).

B.1 Q2D, Q2E, Q2C

Query2Doc
Write a passage that answers the given query: Query: {query 1} Passage: {doc 1} Query: {query 2} Passage: {doc 2} Query: {query 3} Passage: {doc 3} Query: {query 4} Passage: {doc 4} Query: {query} Passage:

Table 4: Prompt for Q2D

Query2Expansion
Write a list of keywords for the given query: Query: {query 1} Keywords: {expansion 1} Query: {query 2} Keywords: {expansion 2} Query: {query 3} Keywords: {expansion 3} Query: {query 4} Keywords: {expansion 4} Query: {query} Keywords:

Table 5: Prompt for Q2E

Query2CoT
Let's think step by step. Answer the following query, and give the rationale before answering. Below is the query: {query}

Table 6: Prompt for Q2C

B.2 GenQRE

GenQREnsemble
1 Improve the search effectiveness by suggesting expansion terms for the query. 2 Recommend expansion terms for the query to improve search results. 3 Improve the search effectiveness by suggesting useful expansion terms for the query. 4 Maximize search utility by suggesting relevant expansion phrases for the query. 5 Enhance search efficiency by proposing valuable terms to expand the query. 6 Elevate search performance by recommending relevant expansion phrases for the query. 7 Boost the search accuracy by providing helpful expansion terms to enrich the query. 8 Increase the search efficacy by offering beneficial expansion keywords for the query. 9 Optimize search results by suggesting meaningful expansion terms to enhance the query. 10 Enhance search outcomes by recommending beneficial expansion terms to supplement the query.

Table 7: Prompt for GenQRE

B.3 GenCRF

Contextual Expansion
You are a contextual expansion expert. Your task is to understand the core intent of the original query and provide a refined, contextually expanded answer. Provide a clear and concise response based on the original query. Below is the query: {...}

Table 8: Prompt for Contextual Expansion

Detail Specific
You are a detail-specific expert. Your task is to understand the core intent of the original query and provide a refined, detailed answer focusing on particular details or subtopics directly related to the query. Provide a clear and concise response based on the original query. Below is the query: {...}

Table 9: Prompt for Detail Specific

Aspect Specific
You are an aspect-specific inquiry expert. Your task is to understand the core intent of the original query and provide a refined answer focusing on a specific aspect or dimension within the topic. Provide a clear and concise response based on the original query. Below is the query: {...}

Table 10: Prompt for Aspect Specific

Clarity Enhancement
You are a clarity-enhancement expert. Your task is to understand the core intent of the original query and reformulate it to enhance clarity and specificity. Focus on eliminating ambiguity and ensuring the query is straightforward, which aids in retrieving the most relevant contexts. Provide a clear and concise response based on the original query. Below is the query: {...}

Table 11: Prompt for Clarity Enhancement

Clustering Refinement
You are an expert in clustering and query refinement. Your task is to review the original query alongside the generated queries, and then cluster them into 1 to 3 groups based on their similarity and relevance. The number of clusters should be determined dynamically. Focus primarily on the relationship of the generated queries to the original query. For each identified cluster, provide only one refined query that incorporates elements from the original and generated queries within that cluster with useful information for document retrieval. The output should be presented in JSON format, structured as follows: {'cluster1': 'refined_query_1', 'cluster2': 'refined_query_2', 'cluster3': 'refined_query_3'} The output must be restricted to 1 to 3 groups. Below is the query: {...}

Table 12: Prompt for Clustering Refinement

LLM Scoring
You are an expert in scoring cluster queries. Evaluate the clustering of queries using the following criteria for each cluster: Relevance, Specificity, Clarity, Comprehensiveness, and Usefulness for retrieval. Assign a score from 1 to 100, where 1 is the lowest and 100 is the highest performance in relation to the original query. Avoid defaulting to high scores unless they are clearly justified. Carefully consider both the strengths and weaknesses of each cluster. For instance, a cluster with relevant but not highly specific results might score between 40 and 60, while a cluster that is both highly relevant and specific might score between 70 and 100. Conversely, a cluster lacking clarity or comprehensiveness should score lower, between 10 and 30. Provide scores that accurately reflect the variation in quality across clusters. List your scores for each cluster in the following format: [score_cluster1, score_cluster2, score_cluster3]. Return your scores in a list format only, without additional commentary. Initial Query: {...} Cluster-Generated Queries : {...}

Table 13: Prompt for Scoring

C Appendix C. Similarity and Score Threshold for Dynamic Weights

We conducted comprehensive experiments to determine the optimal similarity and score thresholds.

C.1 Similarity Threshold

We experimented with similarity ranging from 0.1 to 0.3 to identify the optimal threshold. As illustrated in Figure 6, a threshold of 0.2 yielded the highest nDCG@10 score across all the ablation datasets. This finding demonstrates that an appropriate threshold can effectively enhance the retrieval performance and that the threshold is generalizable across different testsets.

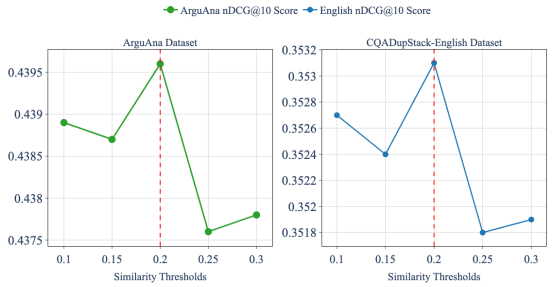

Figure 6: Impact of similarity thresholds on nDCG@10 scores

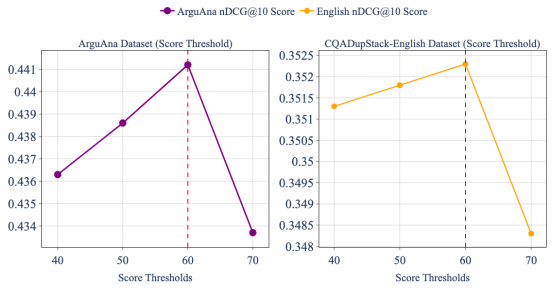

Figure 7: Impact of score thresholds on nDCG@10 scores

C.2 Score Threshold

We also investigated the impact of various score thresholds on the performance of our model. Figure 7 shows the results of experiments with score thresholds ranging from 40 to 70. For both datasets, a score threshold of 60 resulted in the highest nDCG@10 scores, which suggests that our threshold effectively filters our lower-quality generated queries while retaining those that contribute most to improved retrieval performance.

D Appendix D. Cluster Analysis

We analyze how the GenCRF framework clusters data across our main datasets, focusing on the distribution of cluster counts and the similarity between clusters.

D.1 Cluster Counts

As shown in Figure 8, the GenCRF framework predominantly form three clusters across all datasets, followed by two clusters, with a small proportion of single clusters. The result indicates that our framework often identifies multiple distinct aspects of query intents. This multi-faceted clustering approach likely contributes to the framework's ability

to generate diverse and comprehensive query reformulations.

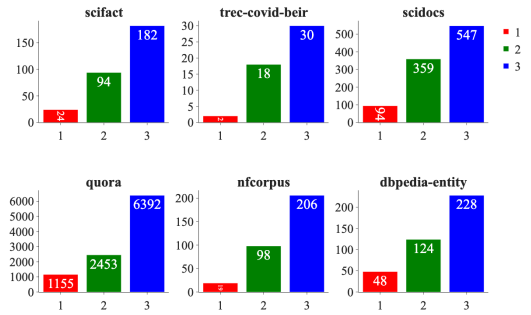

Figure 8: Distribution of Cluster Counts Across Datasets

D.2 Similarity between Clusters

Figure 9 illustrates the similarity between clusters when two or three clusters are formed. We observe that 2 cluster formation consistently show higher similarity scores compared to three cluster formations, and the similarity scores for both 2 cluster and 3 cluster formations are relatively high, indicating greater diversity, and making these clusters more effective at capturing different aspects of query intents.

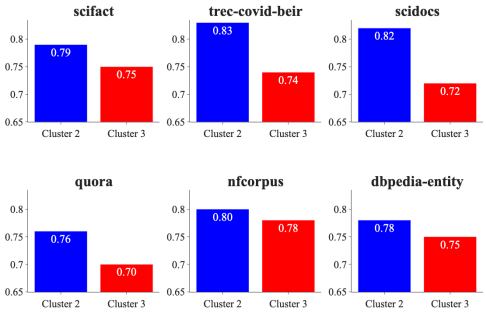

Figure 9: Similarity between Clusters