QueryAligner: Customizing User Query to Match LLMs Preferences for Better Intent Recognition

Yunlong Ma[1], Bo Wang[2*], Yihong Tang[3], Zifei Yu[4], Chenyun Xue[4], Gaoke Zhang[2], Yuexian Hou[2]

[1]School of New Media and Communication, Tianjin University, Tianjin, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin, China
[3]Harbin Institute of Technology, Shenzhen, China
[4]Tianjin Huizhixingyuan Information Technology Co., Ltd.
{mayunlong, bo_wang, toyhom, zhanggaoke, yxhou}@tju.edu.cn,
{pf_fish, xueyeguiren}@126.com

Abstract

The interpretative efficacy of large language models (LLMs) fundamentally hinges on the intricate alignment between user inputs and model-specific linguistic priors. Existing methodologies predominantly employ static input optimization strategies, failing to account for the empirically observed divergence in linguistic preference spaces across distinct LLM architectures, including variations in syntactic parsing heuristics, semantic grounding mechanisms, and knowledge retrieval pathways. We propose QueryAligner, an adaptive rewriting system implementing dynamic model-aware input transformation through architecture-specific preference modeling. Our framework introduces two pivotal innovations: 1) A dual-phase optimization engine integrating supervised learning on reverse-engineered cross-architectural training data with reinforcement learning driven by multi-objective reward signals, ensuring simultaneous preservation of semantic integrity and maximization of target model compatibility; 2) An architecture-informed rewriting protocol that automatically discovers latent alignment patterns encoded within distinct LLMs' parametric configurations. Experimental results demonstrate that our method achieves superior performance compared to conventional input optimization techniques.

Introduction

LLMs have demonstrated remarkable capabilities across a range of tasks (Naveed et al. 2023; Zhao et al. 2025; Kasneci et al. 2023), attracting an increasing number of non-expert users who primarily interact with these systems through zero-shot queries rather than carefully engineered prompts (Zhao et al. 2023). However, there are still many issues in practice. To achieve satisfactory results, it is not enough to solely focus on model optimization, human-machine interaction need also be refined. And the core of optimizing this interaction lies in improving LLMs intent recognition capabilities. Current approaches for enhancing these capabilities can be broadly categorized into three main methods: prompt engineering, multi-turn interactions and fine-tuning (Cross and Ramsey 2021). The features of each method are presented in Figure 1. Prompt engineering involves carefully crafting user inputs or placing them into predefined templates to guide LLMs (White et al. 2023; Lester, Al-Rfou, and Constant 2021; Tang et al. 2025), while multi-turn interactions refine model understanding through iterative dialogue (Wang et al. 2023; Yang et al. 2024), and fine-tuning adjusts its parameters to specific tasks or domains using labeled data (Han et al. 2024; Ding et al. 2023).

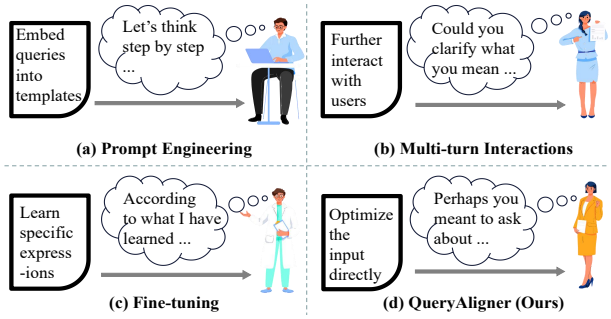


Figure 1: Schematic illustration of diverse enhancement approaches for intent recognition.

Fully harnessing the potential of LLMs to understand user intent remains a complex and ongoing challenge (Kaddour et al. 2023). The three main approaches currently encounter challenges, primarily due to two unresolved issues: (1) these methods force LLMs to understand user inputs, but sometimes user's low-quality questions, such as unclear wording, grammatical inconsistencies, or colloquial expressions, often hinder LLMs recognition. (2) More importantly, preference discrepancies often arise between users and LLMs, where even subtle differences in word order can impact performance (Berglund et al. 2023). For instance, an LLM may perform well with 'What is A' but struggle with 'A is what'. In summary, although these approaches have achieved notable progress, they rarely focus on direct input optimization and often entail high computational or expertise costs. Moreover, they overlook the crucial alignment between user input preferences and LLM requirements.

To address these issues, we propose the QueryAligner model, which directly optimizes user queries to align with LLMs expression preferences, enhancing intent recognition.

**Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The training of the model harnesses a two-phase paradigm. Supervised learning is initially supported by reverse engineering, followed by reinforcement learning using a modified TRPO algorithm (Schulman 2015) tailored for single-round optimization. Specifically, QueryAligner leverages the BART-based architecture (Lewis et al. 2019), wherein a bidirectional transformer-based encoder captures the semantic nuances and contextual dependencies of imprecise user queries, and an autoregressive decoder generates fluent, structurally aligned rewritings.

Our contributions are summarized as follows:

(1) We identify a key insight experimentally: discrepancies between human and LLMs preferences in vocabulary, grammar, and expressive style, highlighting the need for better alignment in human-machine interaction.

(2) Based on this insight, we propose the QueryAligner model for input optimization,which employs a BART-based model architecture and two-phase joint training strategies optimized for this task.

(3) Experimental results validate the effectiveness of our approach, demonstrating that the observed improvements stem from aligning user inputs with LLMs preferences.

Related Work

Understanding user intent Enhancing LLMs understanding capabilities is a crucial area of research for optimizing their effectiveness and usability (Xi et al. 2025; Chang et al. 2024). To improve their performance in this aspect, researchers employ three primary approaches: (1) Prompt engineering involves users designing intricate prompts or inserting their inputs into predefined structures to guide LLMs (Giray 2023; Chen et al. 2023), which may neglect that ordinary users often input poorly structured or ambiguous queries. (2) Multi-turn interactions allow users to iteratively refine their queries (Yi et al. 2024), but such methods are constrained by latency and user engagement requirements, limiting their scalability in time-sensitive applications. (3) Fine-tuning is effective for understanding tasks within specific domains (Fu et al. 2023), but the process is often resource intensive and requires substantial labeled data, which limits its applicability in rapidly changing or diverse environments (Kumar et al. 2022). Furthermore, these approaches often overlook the alignment between user preferences and model expectations, as well as the challenges posed by their low-quality inputs. To address this, our proposed QueryAligner model directly optimizes user inputs, enhancing their quality and ensuring a more coherent alignment with LLMs.

Query Optimization Directly Research in this area is still in its early stages, primarily encompassing rule-based and deep learning-based approaches. Rule-based methods are limited to simple modifications and expansions of user inputs, such as correcting spelling errors, removing stop words or low-relevance terms, and substituting words in the query with synonyms or related terms (Jurafsky 2000; Joulin et al. 2016). In contrast, recent LLM-based methods have progressed beyond rule-based techniques through automatic prompt and query rewriting. PromptBreeder (Fernando et al. 2023) explores evolutionary strategies to iteratively refine prompts, progressively adapting them to latent reasoning patterns of LLMs, while RaR (Deng et al. 2023) prompts LLMs to first rephrase and expand a given question and then provide a response. Although these methods have demonstrated measurable gains in model comprehension by refining the phrasing of inputs, they primarily emphasize output-side adaptation and do not explicitly target the systematic preference gap between naturally formulated user queries and the intrinsic linguistic priors of LLMs. Building upon this line of research, our work advances query optimization by directly modeling and bridging these underlying preference discrepancies.

Methodology

Overview

As illustrated in Figure 2, the proposed QueryAligner framework operates through a multi-stage pipeline designed to bridge the preference gap between user queries and LLMs. **(1) Reverse Engineering** for Preference Alignment initiates the process by constructing a synthetic dataset that reflects LLMs' implicit preferences. This is achieved by leveraging LLMs to generate candidate reformulations of original user queries from retrieved document contexts, followed by semantic similarity filtering to retain optimal query pairs. **(2) Supervised Alignment** with BART-based Architecture which inherently combines bidirectional contextual encoding and autoregressive decoding capabilities. The BART encoder captures nuanced semantic patterns and syntactic irregularities in raw user inputs, while its decoder, guided by cross-attention mechanisms, generates refined queries that adhere to LLMs' lexical and structural preferences. **(3) Reinforcement Learning** with Integrated Reward Signals further optimizes the system through policy gradient updates driven by a composite reward function. By integrating these signals into a modified TRPO framework, the model learns to produce query reformulations that balance human intent preservation with LLM-friendly expression styles, ultimately enhancing intent recognition robustness across diverse interaction scenarios.

Reverse Engineering

This engineering is designed to generate training data for the QueryAligner. It begins with the preparation of a dataset consisting of questions Q and related documents D. For each Q, we employ a retriever to extract the top-k most relevant document chunks from D, which are likely to contain the answer. These top-k chunks are then fed into an LLM that generates five potential questions $\{Q_i^*\}_{i=1}^5$, representing the way of formulating queries that aligns with the LLMs preferences. Then we calculate the cosine similarity between the original query Q and each generated question Q_i^* in the embedding space. The question Q^* with the highest similarity to Q is selected as the optimized question. Finally, the original query Q is paired with its optimized counterpart Q^*, providing the training data for subsequent processes.

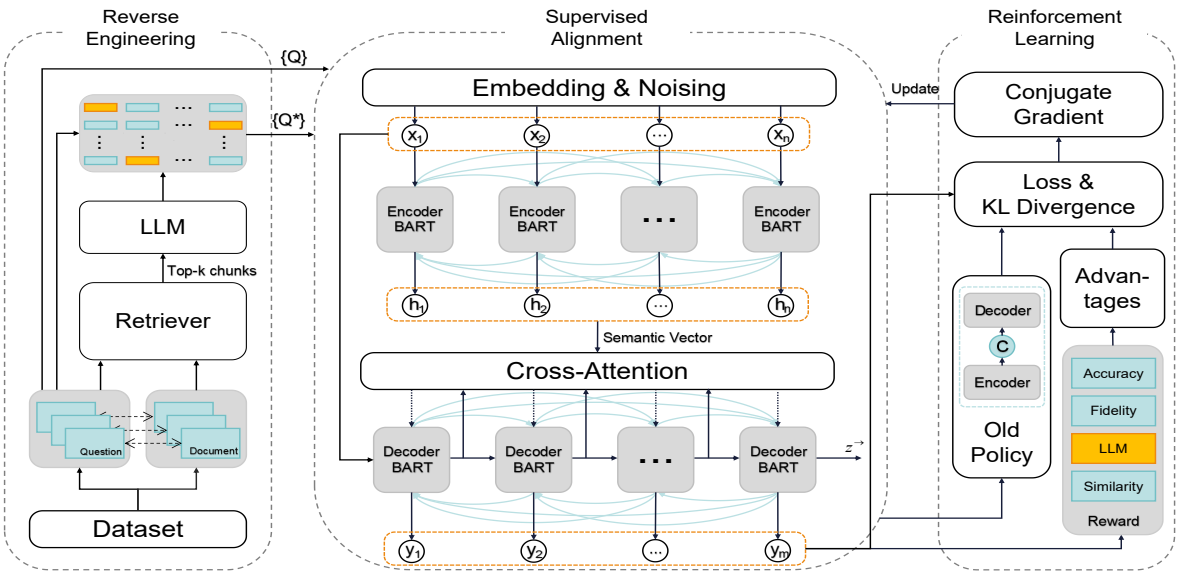


Figure 2: The overview structure of the proposed model and joint training methods. It begins with the left section which depicts the reverse engineering of supervised learning data from an LLM. The central section highlights the model's Seq2Seq structure with BART-based encoder-decoder mechanisms. Finally, the right section outlines the refinement process, where the model undergoes further optimization using reinforcement learning.

Supervised Alignment

In the supervised alignment phase, we propose a hybrid training framework that synergizes the bidirectional auto-regressive architecture of BART with our task-specific objectives. Given the parallel corpus $\{(Q_i, Q_i^*)\}_{i=1}^N$ generated through reverse engineering, the model learns to reconstruct LLM-preferred queries while preserving original semantic intent. The BART encoder processes the input sequence $Q = \{q_1, \ldots, q_n\}$ through bidirectional self-attention layers to capture contextual representations:

$$\mathbf{H}^e = \text{Encoder}\left(Q; \theta_e\right) \in \mathbb{R}^{n \times d}, \tag{1}$$

where d denotes the hidden dimension. The decoder then generates the aligned query Q^* through conditional auto-regressive prediction with cross-attention mechanisms:

$$P\left(Q^* \mid Q\right) = \prod_{t=1}^{m} P\left(q_t^* \mid \mathbf{H}^e, q_{<t}^*; \theta_d\right), \tag{2}$$

To address the dual objectives of semantic preservation and preference alignment, we implement a composite loss function:

$$\mathcal{L}_{\text{align}} = \lambda_1 \mathcal{L}_{\text{CE}} + \lambda_2 \mathcal{L}_{\text{sim}}\,, \tag{3}$$

The cross-entropy loss $\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{m} \log P\left(q_t^* \mid q_{<t}^*, Q\right)$ ensures basic sequence reconstruction, while the semantic similarity loss $\mathcal{L}_{\text{sim}} = 1 - \cos\left(\mathbf{v}_Q, \mathbf{v}_{Q^*}\right)$ employs contrastive learning in the embedding space, where $\mathbf{v}$ denotes the mean-pooled sentence embeddings from a pretrained language model. This dual-objective optimization enables the model to balance grammatical conformity with LLM preferences against strict semantic equivalence.

In addition, rather than employing the standard BART architecture directly, we introduce two critical modifications tailored to the characteristics of our task: sparse expert adaptation in the feed-forward network layers, and an attention-constrained decoding mechanism for preserving query-specific entities. To enhance the model's ability to perform diverse query optimizations—such as syntactic refinement, factual completion, and stylistic normalization—we replace the original fully-connected feed-forward network (FFN) layers within BART's transformer blocks with a sparsely-activated Mixture of Experts (MoE) architecture. Let the hidden representation at layer l be denoted as $h^{(l)}$. The modified FFN layer computes the output as:

$$\text{MoE}\left(\mathbf{h}^{(l)}\right) = \sum_{i \in \mathcal{S}} g_i\left(\mathbf{h}^{(l)}\right) \cdot \text{FFN}_i\left(\mathbf{h}^{(l)}\right), \tag{4}$$

where $\mathcal{S} \subset \{1, \ldots, N\}$ is a sparsely selected subset of experts, typically with $\mathcal{S} = 2$, and $g_i\left(\cdot\right)$ is a gating function that assigns weights to the selected experts. Each expert FFN_i is specialized—either implicitly through data-driven training or via task-specific initialization—to handle a particular transformation subtask. This sparse expert routing mechanism allows the model to dynamically adapt its generation behavior based on the latent characteristics of the input, facilitating fine-grained control over query rewriting without excessive parameter growth.

In parallel, we observe that in many real-world scenarios, user queries contain domain-specific entities or key expressions that must be preserved in the rewritten form. To this end, we incorporate a pointer-generator mechanism into the decoder of BART to constrain its attention behavior and support entity copying. Let $P_{\text{gen}} \in [0, 1]$ be the generation probability at each decoding step t, computed via a learned

sigmoid gate over the decoder state s_t. The final output probability over the vocabulary $V \cup \{w_j\}_{j=1}^{|x|}$ is given by:

$$P(w) = P_{\text{gen}} \cdot P_{\text{vocab}}(w) + (1 - P_{\text{gen}}) \cdot \sum_{j:x_j=w} \alpha_{tj}, \quad (5)$$

where $P_{\text{vocab}}(w)$ is the softmax probability over the base vocabulary, and α_{tj} is the attention weight assigned to input token x_j at decoding step t. This hybrid formulation enables the model to flexibly combine generative fluency with precise copy behavior, effectively grounding the rewritten queries in the original input while avoiding semantic drift.

During this phase, MoE routing and pointer-generator parameters are jointly optimized with the base BART layers, allowing the model to learn both high-level rewriting patterns and token-level alignment constraints. This supervised stage thus provides a strong initialization for subsequent reinforcement fine-tuning, equipping the model with both linguistic flexibility and factual fidelity.

Reinforcement Learning

While supervised alignment equips the BART-based model with a foundational ability to generate fluent and semantically preserved query rewrites, it remains limited in its capacity to internalize and respond to more abstract, task-specific objectives—such as stylistic alignment with LLMs, subtle preference matching, and effectiveness in downstream comprehension. To address these limitations, we introduce a reinforcement learning phase wherein the model policy is further refined through interaction with a composite reward function. This function operationalizes multiple alignment desiderata that are otherwise difficult to encode within maximum likelihood estimation.

In this formulation, the previously fine-tuned BART model serves as the initial policy π_θ, generating an optimized query $\hat{y}$ conditioned on the original user input x. The generated sequence is then evaluated via a reward function $r(\hat{y}, x)$ that aggregates four distinct signals: semantic similarity to the original input, conformity to LLM stylistic preferences, fidelity of informational content and accuracy as judged by downstream accuracy. Each component is mapped to a scalar score and weighted to reflect its relative importance. Formally, the composite reward is expressed as:

$$r(\hat{y}, x) = w_1 r_{\text{sim}} + w_2 r_{\text{llm}} + w_3 r_{\text{fid}} + w_4 r_{\text{acc}}, \quad (6)$$

where w_n are the respective weights for each reward component, r_{sim} is derived from cosine similarity in embedding space, r_{llm} reflects LLM preference feedback, r_{fid} penalizes omissions of critical entities or relations, and r_{acc} captures the impact of the rewritten query on answer correctness. This multidimensional reward structure provides a more nuanced learning signal than any single metric, aligning the optimization objective with both linguistic and functional quality.

To stabilize learning and preserve the fluency inherited from the supervised phase, we adopt a constrained policy optimization approach. Specifically, policy updates are regularized via a KL-divergence penalty that limits deviation from the initial supervised distribution. This ensures that the learned policy remains within a trust region, preventing collapse into low-reward or degenerate regimes. The final objective becomes:

$$\max_\theta \mathbb{E}_{\hat{y}\sim\pi_\theta}[r(\hat{y}, x)] \quad \text{s.t.} \quad \text{KL}(\pi_{\theta_{\text{old}}} \| \pi_\theta) \leq \delta, \quad (7)$$

where δ governs the allowed divergence. Empirically, this constraint encourages stable convergence while still enabling the model to exploit reward gradients. Through this reinforcement learning stage, the alignment model moves beyond surface-level similarity, acquiring a reward-aware query transformation capability that generalizes across query types, domains, and latent user intents.

Experiments

Datasets

The selection of datasets for this study is guided by several key considerations. Firstly, they should cover both general and specific domains, including both simple and complex questions. Secondly, due to the need for reverse engineering, the data must include relevant documents. Thirdly, the questions should be primarily from ordinary users, rather than being expert-annotated. Based on these considerations, the following three datasets were selected:

Natural Questions (NQ) (Kwiatkowski et al. 2019) is a large-scale dataset designed for open-domain question answering research. It consists of real user queries submitted to the Google search engine, paired with corresponding Wikipedia pages.

HotpotQA (Yang et al. 2018) is a comprehensive question answering dataset designed to evaluate multi-hop reasoning. It contains 113k question-answer pairs based on Wikipedia, where each question requires reasoning across multiple documents.

PubMedQA (Jin et al. 2019) is a biomedical question answering dataset, where each instance consists of a research question, the context derived from the abstract, and a long-form answer from the abstract's conclusion.

Our selected datasets span diverse domains, comprising general-domain queries from NQ and HotpotQA, with NQ representing zero-shot inquiries from ordinary users and HotpotQA emphasizing intricate multi-hop reasoning. In contrast, PubMedQA is dedicated to expert-crafted domain-specific questions.

Baselines

We compare the proposed model with six baselines, classified into two groups. The first group focuses on methods that utilize powerful proprietary LLMs with the aim of improving intent recognition by designing specialized pipelines or optimizing prompts. The second group includes methods based on community LLMs, where the outputs are improved through fine-tuning or further pre-training on task-specific datasets.

Proprietary Model Methods This category leverages gpt-4o-mini, which is pre-trained on extensive datasets and known for their exceptional performance across various natural language processing tasks. We first selected the **Naive-RAG** (Lewis et al. 2020) framework, which serves as

		Natural Questions			HotpotQA			PubMedQA	
		Accuracy	BLEU-1	Recall	Accuracy	EM	Recall	Accuracy	EM
Proprietary Models	Naive RAG	50.69	31.37	42.29	77.17	46.00	44.05	93.82	91.33
	CoT-SC	53.24	36.01	42.38	79.66	51.97	44.04	94.98	93.31
	RaR	53.69	36.27	44.68	81.44	53.28	46.73	95.31	92.59
	Direct Optimization	52.90	33.45	43.27	79.60	51.75	45.20	94.02	92.24
	QueryAligner *w/o* CoT	53.11	35.43	44.35	80.13	51.87	45.69	93.60	92.29
	QueryAligner (Ours)	**54.83**	**38.26**	**44.97**	**82.57**	**54.81**	**46.75**	**95.44**	**93.81**
Community Models	Naive RAG	43.95	30.98	42.43	67.90	38.00	44.05	90.16	88.71
	RAFT	47.91	32.89	42.49	71.25	45.83	44.16	91.64	89.38
	Self-RAG	47.03	32.27	44.52	73.33	47.47	47.06	92.48	90.32
	Direct Optimization	46.72	32.21	43.77	71.24	45.94	45.29	91.21	89.10
	QueryAligner *w/o* CoT	47.67	32.85	44.52	73.75	47.82	47.17	91.74	89.56
	QueryAligner (Ours)	**48.56**	**33.95**	**44.59**	**74.31**	**48.05**	**47.21**	**92.83**	**90.44**

Table 1: Experimental results on three datasets (Natural Questions, HotpotQA, and PubMedQA), comparing Proprietary Models and Community Models across multiple evaluation metrics, including Accuracy, BLEU-1, Recall, and EM. The best-performing results for each metric are highlighted in **bold**.

our simplest baseline by integrating retrieval and generation without any additional prompts. **CoT-SC** (Wang et al. 2022) guides LLMs through a search process to generate intermediate reasoning steps, which are then completed to form coherent solutions, enhancing structured and transparent decision-making. **RaR** enables LLMs to enhance response quality by rephrasing and elaborating on the input question before generating an answer, all within a single prompt.

Community Model Methods For the community baselines, we utilize approaches built on Llama-2-13b (Touvron et al. 2023). Similarly, **Naive-RAG** is also adopted as the most fundamental reference. **RAFT** (Zhang et al. 2024) optimizes domain-specific LLMs performance through retrieval-augmented fine-tuning, enhancing question-answering accuracy by distinguishing between relevant and distracting documents. **self-RAG** (Asai et al. 2023) integrates on-demand retrieval and self-reflection, empowering models to adaptively select when to leverage external knowledge and critique their own generations for improved performance.

Evaluation Metrics

To comprehensively evaluate the performance of the proposed QueryAligner model, we employ four widely recognized metrics: **Accuracy**, **Exact Match (EM)**, **BLEU-1** and **Recall**, which assess the semantic correctness of optimized queries, the precision of query generation, lexical overlap with reference queries, and the retrieval of relevant documents, respectively.

Experimental Results

Baseline Comparison The results in Table 1 demonstrate the consistent advantage of QueryAligner across all three datasets. (1) On Natural Questions, QueryAligner achieves the highest Accuracy (54.83% with proprietary models, 48.56% with community models) and BLEU-1 scores (38.26% and 33.95%, respectively), reflecting its effectiveness in rewriting informal, underspecified queries into forms that better align with LLM expectations. This result suggests that our model effectively enhances the understanding of ambiguous, informal queries typically posed by everyday users, thus improving task performance and retrieval accuracy. (2) In HotpotQA, which requires multihop reasoning, QueryAligner attains the best Accuracy (82.57% / 74.31%), Recall (46.75% / 47.21%), and EM (54.81% / 48.05%) in both settings, suggesting that its rewriting enhances the model's ability to identify and integrate relevant evidence. (3) On PubMedQA, where questions are more structured and domain-specific, QueryAligner still achieves the highest Accuracy under both setups (95.44% and 92.83%), confirming its robustness even when the input is already relatively well-formed. These results underscore QueryAligner's generalizability and its strength in aligning user inputs with model-internal preferences across varied query types and domains.

Model (with ours)	Accuarcy	Improvement	Recall	Improvement
CoT-SC	54.66	1.41	44.25	1.87
RaR	54.79	1.10	45.00	0.32
RAFT	50.18	2.27	43.71	1.22
Self-RAG	48.42	1.39	45.68	1.09
Naive RAG				
using gpt-4o-mini	53.47	2.78	44.45	2.16
using llama-2-13b	47.72	3.77	44.62	2.19
Average	51.54	2.12	44.61	1.47

Table 2: Performance results on the NQ dataset with various baseline models combined with our approach.

Integrated Evaluation A distinctive feature of our model is its ability to directly optimize user queries, setting it apart from traditional baselines that primarily focus on improving query outputs through techniques such as fine-tuning, prompt engineering, or refining RAG methods. Building on this unique capability, we explored the potential bene-

fits of integrating QueryAligner with these existing models. The results of this evaluation, presented in Table 2, indicate that combining QueryAligner with various baselines leads to consistent improvements across both accuracy and recall. On average, the integrated models demonstrated a notable increase of 2.12 in accuracy and 1.47 in recall. These findings suggest that QueryAligner's direct approach to optimizing user queries complements the strengths of conventional models, yielding a synergistic effect that enhances the overall performance of the query processing pipeline. The observed improvements across all baseline models highlight the versatility and utility of QueryAligner in advancing query understanding and retrieval accuracy.

Further Analysis

To fully understand the capabilities and limitations of QueryAligner, we perform an extensive set of analyses that delve deeper into its performance and behavior. These analyses are designed to demonstrate that: 1) the semantics remain unchanged after optimization, 2) the expression style has been modified, with these modifications serving to retain essential information and introduce new content, and 3) these changes enhance the LLM's comprehension of user intent. By exploring various dimensions of query optimization, this section aims to uncover the underlying mechanisms that contribute to the model's success and identify potential areas for improvement.

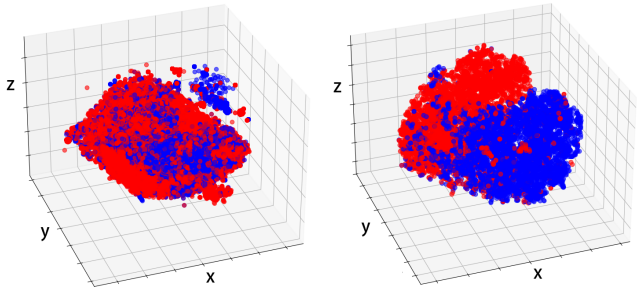
Optimization Evaluation

Semantic Consistency To ensure the optimized questions align semantically with the original input, we first evaluate the semantic similarity between them by using BERTScore and a cosine similarity metric calculated over sentence embeddings. The optimized questions achieve a Precision of 93.31%, a Recall of 92.31%, and an F1-score of 92.80%, which indicate that the optimization process effectively balances retaining the original semantic content (Recall) and refining the query expression (Precision). Additionally, the separate semantic similarity metric computed over sentence embeddings achieves a score of 95.19%, further confirming that optimized queries align closely with the original intent.

Information Integrity While a high degree of semantic similarity is essential, it does not necessarily ensure that the optimized questions capture all the critical information contained in the original queries. To evaluate this, we first measured the average question length (AQL) and vocabulary increment between the original and optimized questions. As shown in Table 3, the results reveal a notable increase in both AQL and vocabulary diversity, indicating the potential to preserve all the critical information of original queries while enriching their content. To further validate this matter, we introduce human evaluation. Evaluators scored the optimized questions on a scale from 0 to 5, focusing solely on information preservation. The average score of 4.8 indicates that the optimized questions effectively capture the key content of the original queries.

	AQL	Vocabulary Growth	Human Evaluation	Cross Entropy	Perplexity
Original	87.69	12.93%	4.8	5.162	328.14
Optimized	106.17			4.248	113.91

Table 3: Results of AQL, vocabulary growth, human evaluaiton, information entropy, and perplexity for original and optimized questions.



(a) Vocabulary t-SNE (b) Sentence t-SNE

Figure 3: t-SNE visualizations of original and optimized queries on the NQ dataset. (a) shows the vocabulary distribution and (b) displays the sentence embeddings. Blue spheres indicate original questions and red spheres represent optimized questions.

Redundancy To assess whether the increased length of the optimized questions leads to redundant information that may confuse the model, we evaluated the cross-entropy (Krizhevsky, Sutskever, and Hinton 2012) and perplexity (Jelinek et al. 1977) between the original and optimized questions using GPT-2 (Radford et al. 2019) to measure how well the probability distribution of the optimized questions matches LLMs expectations. The experimental results are summarized in Table 3. A lower cross-entropy indicates better alignment with LLMs distribution, and a lower perplexity implies that the optimized queries are more predictable and coherent. These results demonstrate that the additional information introduced during optimization does not lead to redundancy or confusion, but rather enhances the clarity and predictability of the queries.

In conclusion, the combined analysis of semantic similarity, information preservation, and language coherence confirms the effectiveness of our optimization process. the optimized questions not only retain the critical information and the original intent but also enhance their clarity and expression, ultimately enabling LLMs to better understand and align with user intent in subsequent tasks.

Expression Preference Gap Analysis

To investigate the differences in vocabulary, expression, and preferences between LLMs and users, we conducted a series of experiments comparing original queries with their optimized counterparts. We evaluated these queries across several metrics, including BLEU, ROUGE-1, Inclusion Rate, and Dependency Core Coverage Ratio (DCCR), which offer insights into both lexical and syntactic alignment between the two query sets. Inclusion Rate quantifies the proportion

BLEU-N				ROUGE-1			Inclusion Rate	DCCR	KL Divergence	LLM Evaluation
N=1	N=2	N=3	N=4	Precision	Recall	F1				
56.62%	44.03%	34.31%	25.66%	61.28%	75.59%	66.57%	58.06%	31.14%	13.51	4.423

Table 4: Comparison of optimized and original queries on the NQ dataset.

Base Dataset	**NQ**	**HotpotQA**	**PubMedQA**
Corresponding	+3.30	+4.44	+1.49
NQ	/	+2.97	+0.49
HotpotQA	+2.18	/	-0.27
PubMedQA	+0.36	+1.42	/

Table 5: Generalization performance across datasets, compared against a Naive RAG framework.

Model	Accuracy	BLEU-1	Recall
QueryAligner	**54.83**	**38.26**	**44.97**
w/o approximation matching	51.01	32.64	40.85
w/o supervised learning	33.52	27.71	37.10
w/o reinforcement learning	51.52	33.27	41.18

Table 6: Ablation experiments results on NQ.

of terms in the optimized query that are also found in the original query. This metric serves as an indicator of how much of the original vocabulary is retained in the optimized version. DCCR, on the other hand, measures the preservation of key syntactic dependencies in the query. Specifically, it evaluates whether core syntactic components, such as subjects, verbs, and objects, are maintained after optimization.

The experimental results, presented in Table 4, reveal significant discrepancies between the preferences for vocabulary and the expression style of LLMs and those of human users. To further investigate these differences, we computed the KL divergence between the original and optimized queries, which measures the discrepancy between two probability distributions. The result, 13.51, indicates a significant divergence between the two query sets.

To visually capture the differences between the original and optimized queries, we used t-SNE visualizations, as shown in Figure 3. (a) illustrates that the optimized queries have a larger and more diverse vocabulary, indicating a shift in lexical preferences, while (b) reveals a more pronounced separation between the original and optimized queries at the sentence level, suggesting significant changes in sentence structure and syntax.

Finally, to examine whether the model genuinely prefers the optimized queries, we asked the LLM to evaluate both original and optimized versions, yielding a 4.423/5 score. This result indicates that the optimized queries better match the model's preferences, effectively bridging the preference gap and enhancing its understanding, interpretability, and responsiveness to user intent.

Generalization Test

To assess the framework's generalization across datasets, we trained GPT-4o-mini on one dataset and fine-tuned it on 10% of a target dataset before testing. This lightweight adaptation mitigates vocabulary and domain discrepancies. As shown in Table 5, the framework demonstrates strong cross-domain generalization, especially between related domains, though performance declines in highly specialized ones. These results underscore the potential for broad applicability of the model, while highlighting the challenges posed by domain-specific linguistic divergence.

Ablation Study

We conducted an ablation study on the NQ dataset to assess the contribution of each QueryAligner component—approximation matching, supervised learning, and reinforcement learning. The results quantify each module's impact and reveal their complementary roles. Removing approximation matching simulates direct LLM optimization, underscoring the need for this refinement step. Performance was evaluated using Accuracy, BLEU-1, and Recall to measure semantic alignment, linguistic quality, and information retention. The results are summarized in Table 6, highlighting the indispensable contributions of each module within the QueryAligner framework. Approximation matching facilitates semantic alignment between user queries and model preferences, ensuring refined and interpretable outputs, while supervised learning provides a crucial foundation that enables the model to effectively balance semantic fidelity and linguistic quality. Reinforcement learning further enhances query optimization by fine-tuning outputs based on task-specific feedback, complementing the strengths of the other components.

Conclusion

In this work, we propose the QueryAligner model to optimize user query directly, bridging the preference gap between users and LLMs and enhancing the ability of LLMs to recognize user intention. The model utilizes a BART-based architecture, balancing the trade-offs between cost, comprehension, and generation capabilities. Experimental results confirm the preference gap between users and LLMs, and show that QueryAligner not only bridges this gap effectively but also outperforms existing input optimization methods. The results also suggest that the strengths of QueryAligner are more pronounced in handling open-domain, varied user queries, where the need for optimization is greater. Our work paves the way for more intuitive and human-centric AI interactions, fostering systems that adapt seamlessly to users' natural expressions and bridge the gap between human intent and machine understanding.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62376188, 62272340, 62276187, 62376192), the Intelligent Assistant Management System for Electronic Case Files of the Public Security Department of Inner Mongolia Autonomous Region (Legal Corps of the Public Security Department of Inner Mongolia Autonomous Region), and the Key Technology Research and Industrial Application Demonstration of General Large Model with Autonomous Intelligent Computing Power, No.24ZGZNGX00020.

References

Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; and Hajishirzi, H. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Berglund, L.; Tong, M.; Kaufmann, M.; Balesni, M.; Stickland, A. C.; Korbak, T.; and Evans, O. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3): 1–45.

Chen, B.; Zhang, Z.; Langrené, N.; and Zhu, S. 2023. Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review. *arXiv preprint arXiv:2310.14735*.

Cross, E. S.; and Ramsey, R. 2021. Mind meets machine: Towards a cognitive science of human–machine interactions. *Trends in cognitive sciences*, 25(3): 200–212.

Deng, Y.; Zhang, W.; Chen, Z.; and Gu, Q. 2023. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3): 220–235.

Fernando, C.; Banarse, D.; Michalewski, H.; Osindero, S.; and Rocktäschel, T. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.

Fu, Z.; Yang, H.; So, A. M.-C.; Lam, W.; Bing, L.; and Collier, N. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 12799–12807.

Giray, L. 2023. Prompt engineering with ChatGPT: a guide for academic writers. *Annals of biomedical engineering*, 51(12): 2629–2633.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Jelinek, F.; Mercer, R. L.; Bahl, L. R.; and Baker, J. K. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1): S63–S63.

Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W. W.; and Lu, X. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jurafsky, D. 2000. Speech and language processing.

Kaddour, J.; Harris, J.; Mozes, M.; Bradley, H.; Raileanu, R.; and McHardy, R. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Kasneci, E.; Seßler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günnemann, S.; Hüllermeier, E.; et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, 103: 102274.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Kumar, A.; Raghunathan, A.; Jones, R.; Ma, T.; and Liang, P. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Naveed, H.; Khan, A. U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; and Mian, A. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Schulman, J. 2015. Trust Region Policy Optimization. *arXiv preprint arXiv:1502.05477*.

Tang, Y.; Wang, B.; Wang, X.; Zhao, D.; Liu, J.; He, R.; and Hou, Y. 2025. Rolebreak: Character hallucination as a jailbreak attack in role-playing systems. In *Proceedings of the 31st International Conference on Computational Linguistics*, 7386–7402.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, X.; Wang, Z.; Liu, J.; Chen, Y.; Yuan, L.; Peng, H.; and Ji, H. 2023. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; and Schmidt, D. C. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2): 121101.

Yang, S.; Zhao, H.; Zhu, S.; Zhou, G.; Xu, H.; Jia, Y.; and Zan, H. 2024. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19368–19376.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yi, Z.; Ouyang, J.; Liu, Y.; Liao, T.; Xu, Z.; and Shen, Y. 2024. A Survey on Recent Advances in LLM-Based Multi-turn Dialogue Systems. *arXiv preprint arXiv:2402.18013*.

Zhang, T.; Patil, S. G.; Jain, N.; Shen, S.; Zaharia, M.; Stoica, I.; and Gonzalez, J. E. 2024. Raft: Adapting language model to domain specific rag, 2024. *URL https://arxiv.org/abs/2403.10131*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhao, Y.; Wang, B.; Wang, Y.; Zhao, D.; He, R.; and Hou, Y. 2025. Explicit vs. implicit: Investigating social bias in large language models through self-reflection. *arXiv preprint arXiv:2501.02295*.