Not All Terms Matter: Recall-Oriented Adaptive Learning for PLM-aided Query Expansion in Open-Domain Question Answering

Xinran Chen[1,2], **Ben He**[1,2], **Xuanang Chen**[2,*], **Le Sun**[2]
[1]School of Computer Science and Technology, University of Chinese Academy of Sciences
[2]Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences
chenxinran22@mails.ucas.ac.cn, benhe@ucas.ac.cn
chenxuanang@iscas.ac.cn, sunle@iscas.ac.cn

Abstract

The effectiveness of open-domain question answering (ODQA), particularly those employing a retriever-reader architecture, depends on the ability to recall relevant documents - a critical step that enables the reader to accurately extract answers. To enhance this retrieval phase, current query expansion (QE) techniques leverage pre-trained language models (PLM) to mitigate word mismatches and improve the recall of relevant documents. Despite their advancements, these techniques often treat all expanded terms uniformly, which can lead to less-than-optimal retrieval outcomes. In response, we propose a novel **Re**call-oriented **A**daptive **L**earning (ReAL) method, which iteratively adjusts the importance weights of QE terms based on their relevance, thereby refining term distinction and enhancing the separation of relevant terms. Specifically, ReAL employs a similarity-based model to classify documents into pseudo-relevant and pseudo-irrelevant sets, and then optimizes term weights via two tailored loss functions to maximize the scoring gap between them. Experiments on four ODQA datasets and five QE methods show that ReAL consistently enhances retrieval accuracy and overall end-to-end QA performance, providing a robust and efficient solution for improving QE strategies in ODQA scenarios.

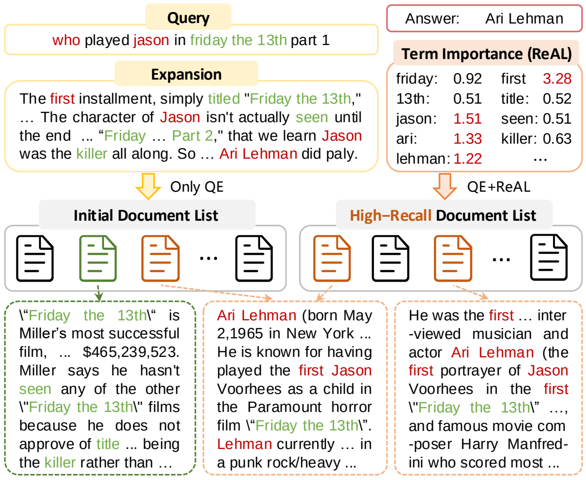

Figure 1: Illustration of query expansion with ReAL. Traditional QE retrieves documents predominantly containing weakly relevant terms, such as *"Friday"*, *"13th"*, and *"killer"*. ReAL enhances retrieval by assigning higher importance to key terms, such as *"Jason"*, *"first"* and *"Ari"*, resulting in improved recall and accuracy of relevant documents.

1 Introduction

Open-Domain Question Answering (ODQA) is a pivotal task in Natural Language Processing (NLP) that focuses on producing accurate answers to a broad range of factual questions across diverse domains (Kwiatkowski et al., 2019). ODQA systems typically adopt a retriever-reader architecture, where the retriever finds relevant documents from the corpus, and the reader extracts or synthesizes answers (Chen et al., 2017). Although more advanced retrieval and re-ranking models, such as dual-encoders (Karpukhin et al., 2020; Chen et al., 2022; Wen et al., 2023), cross-encoders (Chen et al., 2023a,b) and pairwise ranking prompting (Luo et al., 2024; Zhuang et al., 2024) are effective, sparse retrieval models (Salton et al., 1975; Robertson and Zaragoza, 2009) are still widely used for their speed and lack of training requirements, making them well-suited for large-scale applications (Thakur et al., 2021; Chen et al., 2021). However, sparse retrievers often struggle with word mismatches, leading to suboptimal recall of relevant documents (Mitra and Craswell, 2017) and undermining ODQA performance, especially given the reader's context length limitations (Lewis et al., 2020). To address this challenge, Query Expansion (QE) techniques augment the original query with additional terms (Rocchio Jr, 1971; Lavrenko and Croft, 2001), bridging the semantic gap. With the rapid advancement of large pre-trained language

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22139–22151
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

models (PLMs), their strong generative capabilities have been increasingly utilized in various information retrieval (Li et al., 2023b; Xiong et al., 2024) and ODQA tasks (Xin et al., 2025; Li et al., 2023d, 2025). In particular, PLM-based QE techniques utilize these models to enrich the original queries with semantically relevant terms, thereby enhancing document recall (Mao et al., 2021; Chuang et al., 2023; Chen et al., 2024). However, these methods often generate a broad set of potentially relevant terms to enrich the original queries without considering that not all expansion terms are equally important (Lavrenko and Croft, 2017).

Nevertheless, in the context of using sparse retrievers, accurately weighting query terms is of critical importance because they assign relevance scores for each term individually. In practice, PLM-aided query expansions often include many common terms alongside relevant ones, which intuitively should not be weighted the same as more critical terms. As illustrated by the examples in Figure 1, the top retrieved document fails to provide an accurate answer because several expanded terms, such as "Friday" and "13th", are only weakly relevant and deviate from the original query's intent, which is to find information about "Ari Lehman". Therefore, the shortcoming of these QE approaches is particularly evident in the inadequate importance weighting of expanded terms, which can lead to imbalances where certain terms are either underemphasized or overemphasized, ultimately resulting in suboptimal retrieval outcomes. Although traditional QE methods like relevance models (Rocchio Jr, 1971; Lavrenko and Croft, 2001) and SPLADE (Formal et al., 2021a,b), assign term weights as part of the query expansion process, they are not well-suited for modern PLM-aided approaches. These methods lack the scalability and flexibility to capture more nuanced relationships between terms that PLMs can model effectively.

To address the challenges of inadequate term weighting and limited retrieval performance in existing PLM-aided QE methods, we propose **Re**call-oriented **A**daptive **L**earning (ReAL), which enhances QE by adaptively optimizing a term importance vector for ODQA tasks. ReAL assigns a one-dimensional weight vector corresponding to the query terms, which is integrated into the retrieval model and iteratively refined using relevance signals from a classifier alongside original term frequency data. First, ReAL employs a relevance classifier to evaluate the relationship between expanded queries and initial retrieved documents, categorizing them into pseudo-relevant and pseudo-irrelevant sets. Next, ReAL optimizes the weight vector to consistently maximize the score disparity between the pseudo-relevant documents and the pseudo-irrelevant ones through two designed loss functions. Extensive experiments on four widely-used ODQA datasets and five popular QE methods demonstrate that ReAL not only improves retrieval recall but also enhances the overall performance of end-to-end QA systems.

Our contributions are three-fold: 1) We introduce a recall-oriented adaptive learning method ReAL[1], which accounts for the varying importance of expansion terms, leading to more accurate retrieval. 2) Extensive experiments show that ReAL improves both retrieval quality and end-to-end QA performance across diverse datasets, highlighting its utility in practical applications. 3) Compared to previous PLM-aided QE methods, ReAL assigns importance level to the expanded query terms, aiding in the analysis of their role in retrieval.

2 Related Work

2.1 Query Expansion for ODQA

Query expansion (QE) has long been a central technique in information retrieval for enhancing retrieval by enriching queries with related terms (Croft et al., 2009; Carpineto and Romano, 2012). Especially, with the development of pre-trained language models (PLM) in various natural language processing tasks (Li et al., 2023a,c), current QE methods have shifted towards using these models to generate contextually relevant expansions (Zheng et al., 2020; Brown et al., 2020; Naseri et al., 2021). Researches like GAR (Mao et al., 2021) and EAR (Chuang et al., 2023) have leveraged sequence-to-sequence models to improve the retrieval accuracy in Open-Domain Question Answering (ODQA) tasks. Building on this foundation, large language models (LLMs) have further advanced QE for ODQA. Methods like Query2Doc (Wang et al., 2023) and AGR (Chen et al., 2024) utilize LLMs to generate more semantically enriched expansions that resolve word mismatch issues to QDQA tasks.

However, these PLM-aided QE methods often struggle with the static selection and weighting of expanded terms, leading to suboptimal retrieval per-

[1]Our code and data are publicly available at `https://github.com/process-cxr/ReAL`.

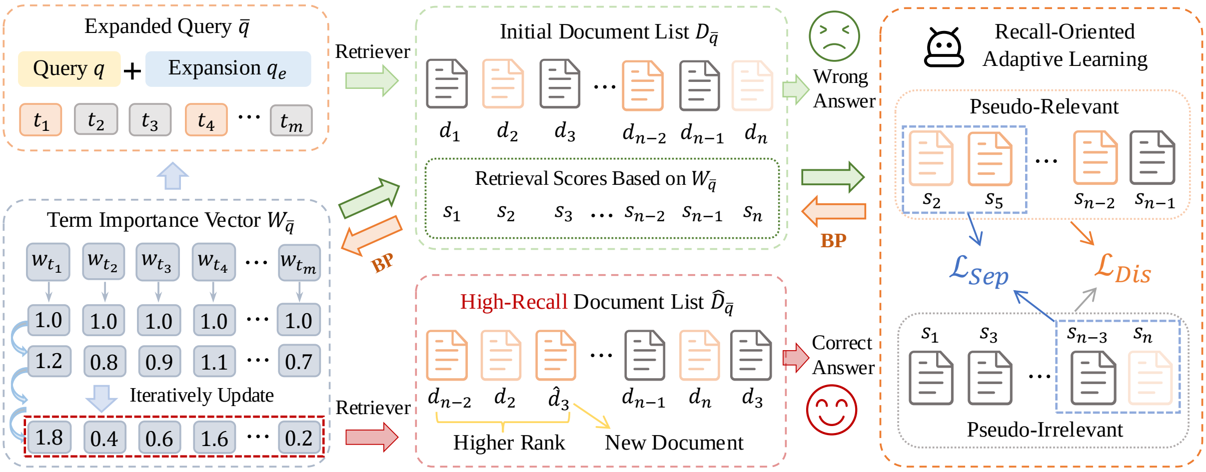

Figure 2: Overview of ReAL. Firstly an initial set of documents is retrieved through sparse retriever with expand query. Then an iterative optimization through a recall-oriented adaptive learning is used for term importance vector.

formance. Our approach, ReAL, addresses this by introducing an adaptive learning process that optimizes term importance based on relevance signals, resulting in improved retrieval effectiveness.

2.2 Term Weighting in Sparse Retrieval

Sparse retrieval methods are foundational to many information retrieval systems, widely adopted for their simplicity and efficiency. These methods offer a straightforward approach to retrieving relevant documents (Salton and Buckley, 1988; Robertson and Zaragoza, 2009), making them particularly well-suited for large-scale applications where retrieval speed is crucial. However, when integrated with modern PLM-aided QE methods, they struggle with dynamically adapting term importance based on the relevance of retrieved documents (Lv and Zhai, 2011). Although traditional studies about term weighting in sparse retrieval like relevance models (Lavrenko and Croft, 2001) and SPLADE (Formal et al., 2021a,b), assign term weights as part of the query expansion process, they are not well-suited for PLM-aided QE approaches.

In contrast, ReAL offers a more efficient, adaptive learning strategy based on relevance-aware feedback. This allows for real-time adjustments with minimal computational overhead, enhancing retrieval precision in ODQA tasks and the performance of end-to-end QA systems.

3 Method

3.1 Overview

As shown in Figure 2, given an original query q and its query expansion q_e generated by a QE technique such as Query2Doc (Wang et al., 2023), the final input query $\bar{q}$ of ReAL is a concatenation of q and q_e, containing m query terms: $\bar{q} = q + q_e = \{t_1, t_2, \ldots, t_m\}$. ReAL first utilizes a sparse retriever capable of providing token-level scores to retrieve the top-n relevant documents $D_{\bar{q}} = \{d_1, d_2, \ldots, d_n\}$ from the corpus, while obtaining a token-level scores vector as $\mathbf{S}_{\bar{q}} = [\mathbf{s}_{t_1}, \mathbf{s}_{t_2}, \ldots, \mathbf{s}_{t_m}]$. However, the initial vector $\mathbf{S}_{\bar{q}}$, while indicative of statistical importance, is not differentiable and poses challenges for dynamic optimization through feedback signals. Therefore, ReAL introduces a weight vector $\mathbf{W}_{\bar{q}} = [\mathbf{w}_{t_1}, \mathbf{w}_{t_2}, \ldots, \mathbf{w}_{t_m}]$ as an additional factor to enable dynamic adaptation. Ultimately, the optimized query $\bar{q}$, along with the optimized weight vector $\mathbf{w}_{\bar{q}}^{last}$ in adaptive learning, is applied to calculate the score for each document d_k via Eq. 1, improving the final retrieval precision of $\hat{D}_{\bar{q}}$.

$$S_k = \text{Retriever}(\bar{q}, \mathbf{W}_{\bar{q}}, d_k) = \sum_{t_i \in \bar{q} \cap d_k} \mathbf{w}_{t_i} \times \mathbf{s}_{t_i} \tag{1}$$

where $\bar{q} \cap d_k$ means the shared terms for expanded query $\bar{q}$ and document d_k.

3.2 Adaptive Learning

Relevance Classifier The relevance classifier plays a pivotal role in the ReAL framework by assessing the relevance of retrieved documents $D_{\bar{q}} = \{d_1, d_2, \ldots, d_n\}$ based on the expanded query $\bar{q}$. It categorizes $D_{\bar{q}}$ into pseudo-relevant (D_{pr}) and pseudo-irrelevant (D_{pi}) sets. This classification process continues until D_{pr} contains s relevant documents and D_{pi} holds the remaining $n - s$

documents. The output of the relevance classifier provides foundational feedback for the subsequent optimization process. By analyzing the distributional differences in document terms between the D_{pr} and D_{pi} sets, as well as their respective subsets, the term weights of important query words are dynamically adjusted during optimization. Various types of relevance classifiers can be employed in the ReAL framework and comparative performance is discussed in Section 4.3.

Loss Function Design To optimize the term importance vector based on the relevance classifier's output, we propose two complementary loss functions: *Distinction of Slight Related Term* and *Separation of Clear Relevant Term*. These functions work together to ensure that key query terms are prioritized while avoiding overfitting to incorrect terms.

Distinction of Slight Related Term establishes a broad separation between relevant and irrelevant documents. It ensures that query terms appearing exclusively in D_{pr}, and not in D_{pi}, are assigned higher weights. These terms, which are typically common across many relevant documents, play a crucial role in enhancing retrieval accuracy. This design is implemented through the following loss function, which penalizes cases where D_{pr} does not consistently achieve higher scores than D_{pi}.

$$\mathcal{L}_{Dis} = \sum_{d_i \in D_{pr}} \sum_{d_j \in D_{pi}} -\log\left(Sig(\mathbf{s}_i - \mathbf{s}_j)\right) \quad (2)$$

where s_i is the revised retrieval score of document d_i as defined in Eq. 1, and Sig is the sigmoid function to adjust the score difference into [0, 1].

Separation of Clear Relevant Term further refines the optimization by narrowing the focus to the most relevant terms. This function specifically targets terms that appear in the most relevant documents D^t_{pr} but are unlikely to be present in the bottom-ranked pseudo-irrelevant documents D^b_{pi}. By emphasizing these critical terms, it increases their weight, ensuring they are properly prioritized. The loss function is formulated as:

$$\mathcal{L}_{Sep} = \sum_{d_i \in D^t_{pr}} \sum_{d_j \in D^b_{pi}} \max\left(0, 1 - \frac{\mathbf{s}_i - \mathbf{s}_j}{\tau}\right) \quad (3)$$

where τ is the score difference between the median scores of document sets D^t_{pr} and D^b_{pi}.

While $\mathcal{L}_{Sep}$ focuses on a narrow set of highly relevant terms, it can lead to bias, especially if incorrect answer related terms are overly emphasized. To mitigate this risk, $\mathcal{L}_{Dis}$ provides a broader adjustment to the term weights, ensuring that the importance of relevant terms is not overestimated at the cost of others. Together, these two loss functions complement each other: $\mathcal{L}_{Dis}$ ensures a wide, foundational separation, while $\mathcal{L}_{Sep}$ sharpens the focus on the most crucial terms, avoiding bias and overfitting. The effectiveness of both functions is discussed in Section 4.3.

3.3 Iterative Optimization

Given an expanded query $\overline{q}$ and the retrieved document list $D_{\overline{q}} = \{d_1, d_2, \ldots, d_n\}$ by the retrieval model with initial scores $S^{(0)}_{D_{\overline{q}}} = [\mathbf{s}^{(0)}_1, \mathbf{s}^{(0)}_2, \ldots, \mathbf{s}^{(0)}_n]$ for each document, we initialize the term importance vector as $\mathbf{W}^{(0)}_{\overline{q}} = [1, 1, \ldots, 1]_m$, and iteratively optimize it by minimizing a combined objective of Eq. 2 and Eq. 3 with a weight factor $\alpha \in [0, 1]$ as in Eq. 4.

$$\mathcal{L}_{ReAL} = \alpha \times \mathcal{L}_{Dis} + (1 - \alpha) \times \mathcal{L}_{Sep} \quad (4)$$

During the i-th iteration, we compute the document scores using the term importance vector $\mathbf{W}^{(i-1)}_{\overline{q}}$ from the previous iteration, and update it using a gradient descent algorithm with learning rate lr as in Eq. 5. The iteration continues until the loss converges (i.e., $\mathcal{L}^{(i)}_{ReAL} - \mathcal{L}^{(i-1)}_{ReAL} \leq \delta$) or the maximum number of steps is reached (i.e., $i = N$).

$$\mathbf{W}^{(i)}_{\overline{q}} = \mathbf{W}^{(i-1)}_{\overline{q}} - lr \times \frac{\partial \mathcal{L}^{(i)}_{ReAL}}{\partial \mathbf{W}^{(i-1)}_{\overline{q}}} \quad (5)$$

After stopping the iteration, the optimized term importance vector $\mathbf{W}^{(j)}_{\overline{q}}$ undergoes a scaling operation, including proportional adjustment and averaging regression. This operation restores the importance of certain key terms whose significance may have diminished during optimization due to frequent occurrence.

$$\mathbf{W}^{last}_{\overline{q}} = \frac{\frac{\sum_{d_k \in D_{\overline{q}}} \mathbf{s}^{(0)}_k}{\sum_{d_k \in D_{\overline{q}}} \mathbf{s}^{(j)}_k} \times \mathbf{W}^{(j)}_{\overline{q}} + \mathbf{W}^{(0)}_{\overline{q}}}{2} \quad (6)$$

where $\mathbf{s}^{(j)}_k$ is the weighted retrieval score for document $d_k \in D_{\overline{q}}$ using term importance vector $\mathbf{W}^{(j)}_{\overline{q}}$.

The final weight vector $\mathbf{W}^{last}_{\overline{q}}$ is then used in a new retrieval round to obtain more relevant documents, improving both retrieval precision and overall end-to-end QA performance.

Dataset	Natural Questions		TriviaQA		WebQuestion		CuratedTREC	
Method	Hit@20	Hit@100	Hit@20	Hit@100	Hit@20	Hit@100	Hit@20	Hit@100
w/o QE	62.99	78.22	76.40	83.04	62.30	75.49	80.69	89.91
+ ReAL	**65.59**	**79.36**	**77.51**	**83.85**	**65.35**	**77.21**	**83.43**	**91.21**
Query2Doc	71.77	83.96	79.26	84.81	75.39	83.21	89.91	93.94
+ ReAL	**73.43**	**84.71**	**80.11**	**85.54**	**76.62**	**83.65**	**90.78**	**94.38**
GAR	74.40	83.60	73.56	81.60	66.14	77.31	82.85	90.34
+ ReAL	**76.23**	**85.01**	**75.87**	**82.56**	**68.11**	**78.69**	**84.73**	**91.79**
EAR-RI	72.57	83.51	78.21	84.27	64.86	78.64	85.59	92.79
+ ReAL	**74.13**	**84.35**	**79.17**	**84.73**	**66.98**	**79.43**	**87.61**	**93.18**
EAR-RD	75.45	84.12	79.55	84.47	68.01	79.57	89.19	93.37
+ ReAL	**76.84**	**85.04**	**80.07**	**84.96**	**69.19**	**80.56**	**90.05**	**93.69**
AGR	77.25	85.76	81.87	86.01	74.55	82.82	93.37	94.95
+ ReAL	**78.14**	**86.04****	**82.43**	**86.39****	**75.25**	**83.23**	**93.80**	**95.39**

Table 1: Hit@k retrieval accuracy (%) on test sets across four open-domain QA datasets. "**+ ReAL**" indicates the application of our ReAL method to various QE approaches or original queries (w/o QE). All improvements are statistically significant at $p < 0.01$ according to the paired t-test, except for those marked with * where $p < 0.1$.

4 Experiments

4.1 Experimental Setup

Datasets For the evaluation, we select four diverse datasets pertinent to ODQA task, including Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Trivia) (Joshi et al., 2017), WebQuestions (WebQ) (Berant et al., 2013), and CuratedTREC (TREC) (Baudis and Sedivý, 2015). A comparative analysis is conducted to assess the improvements achieved by the ReAL method across different PLM-aided QE approaches, with a focus on its impact on sparse retriever performance across all datasets. Furthermore, within the Retriever-Reader framework for ODQA, we evaluate the end-to-end performance of ReAL on NQ and TriviaQA, measuring its overall effect on the complete ODQA pipeline.

Details of ReAL In this study, we adopt the BM25 model (Robertson and Zaragoza, 2009) as the retriever, due to its widespread use and efficient retrieval speed, particularly for models that provide token-level scores. As for the relevance classifier, our primary implementation employs a cross-encoder model, specifically the "cross-encoder/ms-marco-MiniLM-L-12" provided by Sentence Transformers (Reimers and Gurevych, 2019). In addition, we evaluate two alternative sources of relevance signals: a bi-encoder model "BAAI/bge-base-en-v1.5" (Xiao et al., 2023), and a large language model "Mistral-7B-Instruct-v0.2" (Jiang et al., 2023). These variants are analyzed in Section 4.3 to assess their impact on retrieval performance within the ReAL framework. During the iterative optimization process, the gradient descent optimization algorithm Adam (Kingma and Ba, 2015) is used, the number of pseudo-relevant documents s used in $\mathcal{L}_{Dis}$ objective is set as 30, the range parameter c for defining the top and bottom documents in the $\mathcal{L}_{Sep}$ objective is set as 10, the loss weighting factor α is set to 0.5, and the learning rate lr is configured at 0.5. The influence of these hyper-parameters is thoroughly analyzed in Section 4.3. Additionally, we use the Fusion-in-Decoder (FiD) model (Izacard and Grave, 2021) as the reader for end-to-end QA experiments.

Baselines We evaluate the performance of the ReAL method based on five retrieval approaches that process the original query q in different ways: **w/o QE** means directly using BM25 (Robertson and Zaragoza, 2009) model to retrieve without performing query expansion; **GAR** (Mao et al., 2021) adopts three types of query expansion generators based on trained seq2seq models; **EAR** (Chuang et al., 2023) further uses trained query rankers to reorganize the QEs by GAR; **Query2doc** (Wang et al., 2023) uses LLMs to generate answer-oriented passages as QEs; and **AGR** (Jagerman et al., 2023) proposes a multi-step generation framework with quality control mechanisms to produce more refined expansions. To ensure a fair comparison, we

Dataset	Natural Questions				TriviaQA			
Method	**EM@20**	**EM@100**	**LLM@20**	**LLM@100**	**EM@20**	**EM@100**	**LLM@20**	**LLM@100**
w/o QE	36.93	45.26	55.51	62.02	64.08	69.03	69.66	73.98
+ ReAL	**39.06**	**46.45**	**57.34**	**63.19**	**65.45**	**69.54**	**70.94**	**74.74**
Query2Doc	43.57	49.64	63.49	68.00	67.35	70.33	73.11	75.74
+ ReAL	**45.10**	**50.50**	**64.79**	**69.14**	**68.27**	**70.63**	**74.19**	**76.26**
GAR	46.09	50.42	63.79	68.03	59.64	65.17	65.04	69.97
+ ReAL	**47.06**	**51.22**	**64.82**	**68.50**	**61.99**	**66.82**	**67.26**	**71.65**
EAR-RI	44.79	49.17	62.79	66.12	65.54	69.51	71.13	74.15
+ ReAL	**45.71**	**49.64**	**63.43**	**66.68**	**66.56**	**69.89**	**71.76**	**74.95**
EAR-RD	46.29	49.92	63.40	66.86	66.69	69.62	71.75	74.49
+ ReAL	**47.04**	**50.44**	**64.68**	**67.84**	**67.22**	**70.08**	**72.63**	**75.03**
AGR	48.53	51.47	67.83	69.97	70.33	72.20	75.61	77.03
+ ReAL	**49.34**	**51.91**	**68.67**	**70.53**	**70.79**	**72.48**	**76.18**	**77.57**

Table 2: End-to-end performance on the NQ and TriviaQA test datasets. @20/100 refers to the evaluation setup where the top-20 or top-100 retrieved documents are fed into the FiD model, with EM representing the exact match metric and LLM denoting the evaluation metric based on a large language model (Mistral-7B). All improvements are statistically significant at $p < 0.01$ according to the paired t-test.

keep the hyper-parameters and semantic similarity model configurations consistent across all QE methods when combined with ReAL.

Metrics Building on prior research in ODQA, we employ two traditional metrics (Mao et al., 2021) and a novel LLMs-based metric (Kamalloo et al., 2024) within the retriever-reader task paradigm. For retrieval accuracy, *Hit@k* is defined as the proportion of queries in which at least one relevant answer span appears within the top-*k* retrieved documents. For end-to-end QA performance, exact match score *EM@k* is employed, assessing the proportion of instances where the predicted answer span exactly matches one of the ground-truth answers after string normalization. Meanwhile, to address the limitations of string-matching evaluation, *LLM@k* metric implemented by qa-eval (Kamalloo et al., 2024) based on Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) is used, it reflects the proportion of instances in which LLM with few-shot prompting determines that the predicted answer correctly aligns with the ground-truth content.

4.2 Results

Retrieval Evaluation As shown in Table 1, we assess ReAL's performance across four datasets under different baseline methods. For the WebQuestions and CuratedTREC experiments, GAR and EAR utilized seq2seq models transferred from the NQ dataset. The key findings from the retrieval evaluations are summarized as follows:

1) ReAL consistently enhances retrieval performance over all baseline methods. ReAL shows notable gains in retrieval accuracy, measured by Hit@20 and Hit@100, across various datasets and baseline methods, including direct retrieval without QE, supervised QE models like GAR and EAR, and LLM-based approaches such as Query2Doc and AGR. For instance, on the NQ dataset, ReAL achieves Hit@20 improvements between 0.9% and 2.6%, and even for Hit@100, where baseline values are already high, ReAL yields gains of 0.3% to 1.5%. Similar improvements are observed across other datasets, demonstrating ReAL's consistent effectiveness in optimizing query expansion and enhancing retrieval outcomes across different QE methods and datasets.

End-to-End QA Evaluations As shown in Table 2, we performed end-to-end QA evaluations using the Natural Questions and TriviaQA datasets. In addition to traditional exact match metrics, we employed automated evaluation using LLMs (Leval@20/100) for a more comprehensive assessment of answer quality. The key observations from these evaluations are as follows:

2) ReAL provides notable improvements in end-to-end QA performance across various datasets. This is evident from the EM score improvements on both the Natural Questions and TriviaQA datasets. On NQ, for example, ReAL im-

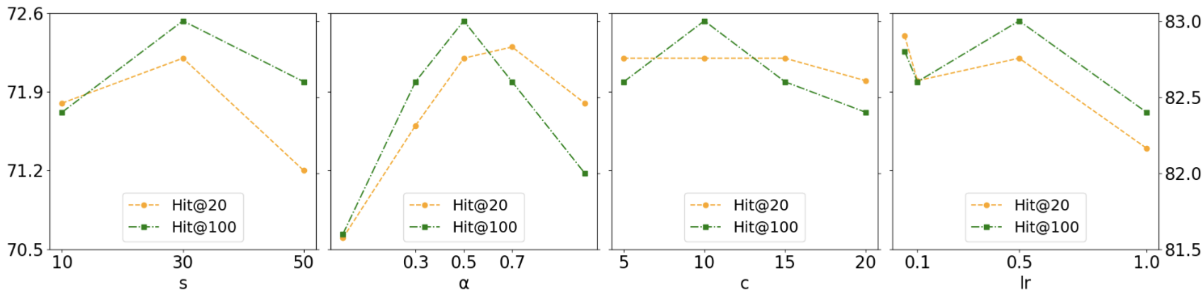

Figure 3: The impact of hyper-parameters on the performance of ReAL in terms of Hit@20 and Hit@100, including the number of pseudo-relevant documents s, the loss weighting factor α, the range parameter c for defining top and bottom documents, and the learning rate lr.

proves EM@20 by approximately 0.8% to 2.1% across various baseline methods, demonstrating substantial benefits for ODQA tasks. Meanwhile, ReAL also shows gains in EM@100, with improvements of approximately 0.5% to 1.2%. These enhancements are consistently observed on TriviaQA as well, underscoring ReAL's capacity to deliver more accurate answer predictions and elevate overall end-to-end QA performance.

3) LLMs-based QA evaluation further highlights the refined quality of ReAL in end-to-end tasks. The incorporation of LLMs for semantic-level evaluation offers a more comprehensive assessment of answer quality. A comparison of Leval@20/100 with EM@20/100 demonstrates that the LLM-based method more accurately evaluates cases where the generated answer partially aligns with the ground truth, capturing subtleties that traditional metrics may miss. Under this advanced evaluation approach, ReAL continues to deliver substantial improvements across all baseline methods on both the NQ and TriviaQA datasets, reinforcing its positive impact. These results further confirm that ReAL's optimization of the query term weighting vector effectively improves overall end-to-end performance in ODQA tasks.

4.3 Analysis

In this section, all analysis experiments are conducted on a randomly sampled subset of 500 queries from the NQ dev dataset, with Query2Doc employed as the query expansion method.

Ablation Study To better comprehend the utility of ReAL, we perform ablation studies to examine the contribution of key components within the method. Specifically, we establish three variants to investigate the necessity of each component: **a) w/o $\mathcal{L}_{Dis}$** means only the $\mathcal{L}_{Sep}$ loss is applied during iterative optimization; **b) w/o $\mathcal{L}_{Sep}$** means only the $\mathcal{L}_{Dis}$ loss is used; **c) w/o Scale** means the post-processing of scale operation on the term importance vector is omitted after iterative optimization.

Method	Hit@100	EM@100	LLM@100
Query2Doc	81.2	45.2	65.2
+ ReAL	**83.0**	**47.4**	**67.4**
w/o $\mathcal{L}_{Dis}$	81.6	46.4	65.8
w/o $\mathcal{L}_{Sep}$	82.0	46.8	66.2
w/o Scale	80.4	45.0	64.8

Table 3: Ablation study results of ReAL on the adaptive learning losses and scaling operation.

From Table 3, we can draw the following conclusions: a) ReAL outperforms the variants lacking certain components, validating the effectiveness of the complete ReAL method. The full configuration demonstrates a more substantial improvement in both sparse retrieval accuracy and end-to-end QA performance when compared to its incomplete counterparts. b) While each loss function individually contributes to some improvements, their combined use proves more effective in refining the term importance vector during iterative optimization, allowing the weighted query to better align with relevant documents. c) The post-processing of scale operation is crucial to the effectiveness of ReAL. Ablation results indicate that ReAL without this operation even performs worse than when ReAL is not applied. Through analysis of the updated weight vectors, we observe that the significance of certain important terms, which appear in both relevant and non-relevant documents, is reduced during iterative optimization due to their frequent occurrence. The scaling operation, similar to a residual connection, restores the importance of these terms, ensuring that the term importance

Group Setting	Good	Bad	#Query
top-30	24.8%	13.6%	368/132
top-50	23.1%	12.7%	389/111
top-100	26.1%	10.6%	406/94

Table 4: The impact of initial retrieval quality for ReAL. "Good" initial retrieval includes at least one ground-truth document in the top-k retrieved documents, while "Bad" initial retrieval does not.

Method	Hit@20	Hit@100
Query2Doc	69.0	81.2
+ ReAL (w/ LLM)	**74.0**	**83.0**
+ ReAL (w/ CE)	72.2	**83.0**
+ ReAL (w/ BE)	70.6	82.0

Table 5: The impact of relevance classifiers in ReAL, including large language model (LLM), cross-encoder (CE), and bi-encoder (BE) models.

vector accurately captures the relevant terms for optimized query performance.

Hyper-Parameter Sensitivity We further investigate the sensitivity of ReAL's performance to four key hyper-parameters during the iterative optimization phase, as detailed in Section 4.1. The experiments presented in Figure 3 demonstrate that for parameters s and c, a moderate increase in the number and range of pseudo-relevant document sets improves retrieval performance, while excessive values degrade it due to the inclusion of irrelevant documents. Accordingly, we set $s = 30$ and $c = 10$ as the optimal configuration. For hyper-parameters α and lr, we found that a higher α favoring the $\mathcal{L}_{Dis}$ improves Hit@20 but reduces improvements in Hit@100. To balance these effects, we set α to 0.5. Additionally, we observed that the learning rate (lr) influences the effectiveness of the iterative optimization. Setting lr to 0.5 yields a better retriever performance while reducing the number of iterations and accelerating the optimization process.

Initial Retrieval Impact To better understand the dependency of ReAL on the quality of initial retrieval, we conduct a comparative analysis focusing on this aspect. Specifically, we conduct a comparative analysis by categorizing queries into two groups, *Good* and *Bad*, based on whether the initial retrieval can successfully retrieve relevant documents into top-30/50/100 results. We compare the retrieval results of the Query2Doc QE method

Query Length	Gen.	Retr.	ReAL-Cls	ReAL-Iter
≈ 10 tokens	-	0.27s	0.26s	0.44s
≈ 60 tokens	0.65s	0.80s	0.3s	1.07s
≈ 110 tokens	1.27s	1.95s	0.33s	1.67s

Table 6: Computational latency of ReAL in different stages, including query expansion (Gen.), sparse retrieval (Retr.), cross-encoder relevance classification (ReAL-Cls) and iterative optimization (ReA-Iter).

before and after applying ReAL to the QE terms and calculate improvement rates for each group to assess the impact of initial retrieval quality on the effectiveness of ReAL. As seen in Table 4, the results confirm that ReAL's performance is influenced by the quality of the initial retrieval, as improvements in the *Good* group were consistently double or more compared to those in the *Bad* group across all group settings. Nevertheless, despite variations in initial retrieval quality, ReAL consistently enhanced retrieval performance, further validating its effectiveness.

Relevance Model Impact As seen in Table 5, we conduct a comparative analysis of three relevance models within the ReAL framework on the NQ dev dataset, to assess their impact on retrieval performance combined with QE, including the bi-encoder model (i.e., "BAAI/bge-base-en-v1.5"), the cross-encoder model (i.e., "cross-encoder/ms-marco-MiniLM-L-12" in Sentence Transformers), and the large language model (LLM, i.e., Mistral-7B-Instruct-v0.2). The results reveal that all three models effectively serve as relevance classifiers in ReAL, enhancing retrieval accuracy and demonstrating the framework's effectiveness. Specifically, the LLM with prompt-based natural language inference deliver the highest performance, followed by the cross-encoder models, with the bi-encoder models being less effective. However, considering the higher latency of LLMs, which require multiple inference steps, we select the cross-encoder model in this study, offering a balance between accuracy and efficiency.

Computational Latency We report the computational latency of ReAL in Table 6, which correlates with input query token length. The analyzed queries have an average length of approximately 10 tokens, with expanded queries reaching around 60 and 110 tokens, depending on the max-token generation parameter in Query2Doc. Latency is evaluated across four stages: query expansion

Method	+ Rerank	+ ReAL
w/o QE	78.22	**79.36**
Query2Doc	83.96	**84.71**
GAR	83.60	**85.01**
EAR-RI	83.51	**84.35**
EAR-RD	84.12	**85.04**
AGR	85.76	**86.04**

Table 7: Comparison of Hit@100 results on NQ test set using Rerank and ReAL.

(Gen), sparse retrieval (Retr), cross-encoder relevance classification (ReAL-Cls), and iterative optimization (ReAL-Iter). The iterative optimization typically involves 50-90 iterations, each taking milliseconds due to the low-dimensional token weight vector, resulting from the retriever's tokenization. As the dimensionality of the weight vector corresponds to the reduced number of query tokens, the optimization occurs in a compact space, keeping the overall latency within acceptable limits. This increase in latency is balanced by the significant improvements in retrieval accuracy and end-to-end QA performance.

4.4 More Discussion

ReAL vs Rerank While re-ranking methods, using relevance classifiers as rerankers, reorder the top-ranked documents to improve retrieval accuracy, they are limited in scope. Re-ranking only refines the order within the static top-k documents, without expanding the set of retrieved documents. In contrast, ReAL dynamically optimizes query term weights during retrieval, allowing the framework to retrieve more relevant documents that may not have been included in the initial set. The advantage of ReAL is its ability to identify and retrieve additional relevant documents through iterative optimization, rather than just reordering existing results. This is demonstrated in the comparison in Table 7 between re-ranking and ReAL based on the same cross-encoder model (i.e., "cross-encoder/ms-marco-MiniLM-L-12" available in Sentence Transformers), where ReAL leads to higher retrieval accuracy, showing its potential to enhance retrieval performance by extending the scope of relevant document retrieval.

Future Extensions of ReAL While ReAL has proven effective with sparse retrieval models, its framework is highly adaptable to more advanced architectures, such as dense or neural retrieval models. Specifically, we can replace the token-weight vectors in sparse retrieval with dense representations, allowing ReAL to optimize term weights based on dense retrieval scores. This integration has the potential to improve retrieval performance in large-scale ODQA tasks, enhancing both accuracy and scalability. The ability to work seamlessly with both sparse and dense retrievers would make ReAL a versatile solution for a broader range of retrieval systems, addressing emerging challenges in future research.

5 Conclusion

In this paper, we introduce ReAL, a recall-oriented adaptive learning method that enhances query expansion through an adaptive learning based on relevance feedback, allowing for more precise alignment between query terms and relevant documents. This method addresses the limitations of current QE approaches, which often fail to account for the contextual significance of expanded terms, leading to suboptimal retrieval results. By adopting an adaptive learning strategy, ReAL improves the retrieval accuracy of sparse retrievers and enhances the overall performance of end-to-end QA systems, making it an practical solution for ODQA tasks. Future work will explore extending ReAL's applicability to more complex retrieval architectures and integrating it with deep retrieval models to further improve retrieval and QA performance.

Limitations

In this work, we focus on the combination of sparse retrieval (BM25) and current PLM-aided query expansion (QE), which is a prevalent and widely adopted approach in open-domain question answering. But actually, our ReAL framework is adaptable to a broader range of retrieval methods, owing to its design, which incorporates a term importance vector at the query level, facilitating seamless integration with additional retrieval models, such as dense retrieval models (e.g., ColBERT (Khattab and Zaharia, 2020)) and neural sparse retrieval models (e.g., SPLADE (Lassance et al., 2024)). Besides, given the computational constraints, the investigation is limited to widely used QE methods and smaller query token sizes, thereby restricting a comprehensive exploration of ReAL's full potential. With increased computational resources, it enables ReAL to better handle more complex and longer queries across diverse retrieval settings.

Acknowledgements

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work is supported by the National Natural Science Foundation of China (62272439) and the Fundamental Research Funds for the Central Universities.

References

Petr Baudis and Jan Sedivý. 2015. Modeling of the question answering task in the yodaqa system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, volume 9283 of *Lecture Notes in Computer Science*, pages 222–228. Springer.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Xinran Chen, Xuanang Chen, Ben He, Tengfei Wen, and Le Sun. 2024. Analyze, generate and refine: Query expansion with LLMs for zero-shot open-domain QA. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11908–11922, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xuanang Chen, Ben He, Kai Hui, Le Sun, and Yingfei Sun. 2023a. Dealing with textual noise for robust and effective BERT re-ranking. *Inf. Process. Manag.*, 60(1):103135.

Xuanang Chen, Ben He, Kai Hui, Yiran Wang, Le Sun, and Yingfei Sun. 2021. Contextualized offline relevance weighting for efficient and effective neural retrieval. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1617–1621. ACM.

Xuanang Chen, Ben He, Zheng Ye, Le Sun, and Yingfei Sun. 2023b. Towards imperceptible document manipulations against neural ranking models. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6648–6664. Association for Computational Linguistics.

Xuanang Chen, Jian Luo, Ben He, Le Sun, and Yingfei Sun. 2022. Towards robust dense retrieval via local ranking alignment. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 1980–1986. ijcai.org.

Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-tau Yih, and James Glass. 2023. Expand, rerank, and retrieve: Query reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12131–12147, Toronto, Canada. Association for Computational Linguistics.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR*, abs/2102.07662.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines - Information Retrieval in Practice*. Pearson Education.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR*, abs/2109.10086.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. SPLADE: sparse lexical and expansion model for first stage ranking. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2288–2292. ACM.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th*

Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, 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. *CoRR*, abs/2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Ehsan Kamalloo, Shivani Upadhyay, and Jimmy Lin. 2024. Towards robust qa evaluation via open llms. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2811–2816, New York, NY, USA. Association for Computing Machinery.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for SPLADE. *CoRR*, abs/2403.06789.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *SIGIR*, pages 120–127. ACM.

Victor Lavrenko and W. Bruce Croft. 2017. Relevance-based language models. *SIGIR Forum*, 51(2):260–267.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yuchen Li, Haoyi Xiong, Linghe Kong, Zeyi Sun, Hongyang Chen, Shuaiqiang Wang, and Dawei Yin. 2023a. Mpgraf: a modular and pre-trained graphformer for learning to rank at web-scale. In *IEEE International Conference on Data Mining, ICDM 2023, Shanghai, China, December 1-4, 2023*, pages 339–348. IEEE.

Yuchen Li, Haoyi Xiong, Linghe Kong, Qingzhong Wang, Shuaiqiang Wang, Guihai Chen, and Dawei Yin. 2023b. S^2phere: Semi-supervised pre-training for web search over heterogeneous learning to rank data. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 4437–4448. ACM.

Yuchen Li, Haoyi Xiong, Linghe Kong, Rui Zhang, Fanqin Xu, Guihai Chen, and Minglu Li. 2023c. MHRR: moocs recommender service with meta hierarchical reinforced ranking. *IEEE Trans. Serv. Comput.*, 16(6):4467–4480.

Yuchen Li, Haoyi Xiong, Qingzhong Wang, Linghe Kong, Hao Liu, Haifang Li, Jiang Bian, Shuaiqiang Wang, Guihai Chen, Dejing Dou, and Dawei Yin. 2023d. COLTR: semi-supervised learning to rank with co-training and over-parameterization for web search. *IEEE Trans. Knowl. Data Eng.*, 35(12):12542–12555.

Yuchen Li, Haoyi Xiong, Yongqi Zhang, Jiang Bian, Tianhao Peng, Xuhong Li, Shuaiqiang Wang, Linghe Kong, and Dawei Yin. 2025. Rankelectra: Semi-supervised pre-training of learning-to-rank electra for web-scale search. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pages 2415–2425. ACM.

Jian Luo, Xuanang Chen, Ben He, and Le Sun. 2024. Prp-graph: Pairwise ranking prompting to llms with graph aggregation for effective text re-ranking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5766–5776. Association for Computational Linguistics.

Yuanhua Lv and ChengXiang Zhai. 2011. When documents are very long, BM25 fails! In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 1103–1104. ACM.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *CoRR*, abs/1705.01509.

Shahrzad Naseri, Jeff Dalton, Andrew Yates, and James Allan. 2021. CEQE: contextualized embeddings for query expansion. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*, volume 12656 of *Lecture Notes in Computer Science*, pages 467–482. Springer.

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*. Association for Computational Linguistics.

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

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

Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, 24(5):513–523.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

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

Xueru Wen, Xiaoyang Chen, Xuanang Chen, Ben He, and Le Sun. 2023. Offline pseudo relevance feedback for efficient and effective single-pass dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2209–2214. ACM.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597.

Chunlei Xin, Shuheng Zhou, Xuanang Chen, Yaojie Lu, Huijia Zhu, Weiqiang Wang, Zhongyi Liu, Xianpei Han, and Le Sun. 2025. Aligning retrieval with reader needs: Reader-centered passage selection for open-domain question answering. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 1000–1012. Association for Computational Linguistics.

Haoyi Xiong, Jiang Bian, Yuchen Li, Xuhong Li, Mengnan Du, Shuaiqiang Wang, Dawei Yin, and Sumi Helal. 2024. When search engine services meet large language models: Visions and challenges. *IEEE Trans. Serv. Comput.*, 17(6):4558–4577.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: contextualized query expansion for document re-ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4718–4728. Association for Computational Linguistics.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 38–47. ACM.

A Dataset Information

Natural Questions (NQ) (Kwiatkowski et al., 2019) is a widely-used question answering dataset composed of real, anonymized queries submitted to Google. It contains 79,168 examples for training, 8,757 for development, and 3,610 for testing, making it a valuable resource for evaluating QA models on real-world search engine queries.

TriviaQA (Trivia) (Joshi et al., 2017) is a large-scale question answering dataset that includes over 950,000 question-answer pairs drawn from 662,000 Wikipedia articles and other web documents. It consists of 60,413 training examples, 8,377 development examples, and 11,313 test examples, offering a rich and diverse set of questions that challenge the breadth and adaptability of QA models.

WebQuestions (WebQ) (Berant et al., 2013) designed for question answering tasks, utilizes Freebase as its underlying knowledge base and consists of 6,642 question-answer pairs. This dataset was developed by sourcing questions through the Google Suggest API, followed by obtaining corresponding answers via Amazon Mechanical Turk. It is structured with an original split of 3,778 training examples and 2,032 testing examples. All answers are defined as Freebase entities.

CuratedTREC (TREC) (Baudis and Sedivý, 2015) is a benchmark dataset for QA systems, derived from TREC-8 (1999) to TREC-13 (2004) competitions. It includes 694 annotated entries, providing a concise yet focused set of examples that serve as a standard for evaluating QA system accuracy under controlled conditions.

B Evaluation on Information Retrieval Benchmarks

While our primary investigation focuses on PLM-aided query expansion within the context of ODQA, we additionally evaluated the broader applicability of the proposed ReAL method in general information retrieval tasks. For this purpose, we conducted experiments on two representative ad-hoc retrieval datasets, namely TREC-DL-2019 (Craswell et al., 2020) and TREC-DL-2020 (Craswell et al., 2021), both constructed from the MS MARCO corpus and widely used for benchmarking document ranking systems. In these supplementary experiments, we adopted the same QE generation pipeline as described in the main text. Query expansion terms were generated by an LLM (i.e., Mistral-7B-Instruct-v0.2) in a zero-shot setting, and subsequently reweighted using the ReAL framework without any additional fine-tuning. Table 8 presents the retrieval performance in terms of NDCG@10, MRR, and MAP.

Method	**NDCG@10**	**MRR**	**MAP**
	TREC-DL-2019		
BM25	50.58	82.45	29.93
+ ReAL	53.27	86.65	31.87
+ QE	57.57	88.29	35.46
+ QE + ReAL	**61.69**	**90.89**	**35.96**
	TREC-DL-2020		
BM25	47.96	82.69	30.27
+ ReAL	53.67	87.96	32.74
+ QE	51.04	85.00	32.34
+ QE + ReAL	**55.47**	**86.45**	**37.70**

Table 8: Retrieval performance on general Information Retrieval (IR) datasets. ReAL consistently improves results over baselines. All improvements are statistically significant at $p < 0.01$ according to the paired t-test.

The experimental results confirm that ReAL consistently improves retrieval performance across all evaluation metrics. Notably, the performance gains are more pronounced when ReAL is applied in combination with query expansions generated by LLMs. These findings underscore the potential of ReAL as a general-purpose term weighting framework that extends beyond ODQA, offering promising applicability to a wider range of information retrieval tasks.