Word2Passage : Word-level Importance Re-weighting for Query Expansion

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) enhances the quality of LLM generation by providing relevant chunks, but retrieving accurately from external knowledge remains challenging due to missing contextually important words in query. We present **Word2Passage**, a novel approach that improves retrieval accuracy by optimizing word importance in query expansion. Our method generates references at word, sentence, and passage levels for query expansion, then determines word importance by considering both their reference level origin and characteristics derived from query types and corpus analysis. Specifically, our method assigns distinct importance scores to words based on whether they originate from word, sentence, or passage-level references. Extensive experiments demonstrate that Word2Passage outperforms existing methods across various datasets and LLM configurations, effectively enhancing both retrieval accuracy and generation quality. The code will be released publicly.

1 Introduction

The advent of Large Language Models (LLMs) has significantly influenced the field of Information Retrieval (IR). One notable advancement in this domain is Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), which integrates retrievers with generative models. By leveraging external knowledge sources during response generation (Gao et al., 2023b), RAG effectively mitigates key challenges of LLMs, such as hallucination (Ji et al., 2023).

Within the evolving landscape of RAG, query expansion has become a key technique for improving retrieval performance (Ma et al., 2023; Mao et al., 2024). It enhances retrieval by either reformulating the original query or generating pseudo passages – artificially created text that captures semantically relevant information. For instance, HyDE (Gao et al., 2023a) leverages a LLM to generate a pseudo passage, which serves as an enriched query containing contextually relevant words. Query2doc (Wang et al., 2023) improves retrieval by repeating the original query a fixed number of times alongside the pseudo passage. These studies highlight that generating pseudo passage helps augment highly relevant words, enhancing retrieval performance (Gao et al., 2023a; Wang et al., 2023, 2024).

Building upon them, recent studies have focused on optimizing the integration of pseudo passages with the original query. Specifically, MuGI (Zhang et al., 2024) calculates query importance based on the lengths of both the query and generated pseudo passages, ensuring balanced integration and improved retrieval performance.

Despite advancements in query expansion, methods like HyDE, Query2doc, and MuGI rely on *passage-level*, treating all words in a pseudo passage equally (Song and Zheng, 2024), failing to differentiate high-importance words that are crucial for retrieval. Also, when determining importance, solely relying on frequency overemphasizes common words and overlooks rare but meaningful ones, leading to query drift. Therefore, low-importance or misleading words by them lead to reduced retrieval effectiveness in RAG. This highlights the need to properly adjust word importance in query expansion (Chen et al., 2024; Kim et al., 2023).

To address this, we propose a novel approach named **Word2Passage**, which introduces a *word-level* importance re-weighting for query expansion. It generates pseudo references at three different levels, forming a hierarchical structure that progresses from `words`→`sentences`→`passages`. This hierarchical structure enables a gradual expansion, capturing the importance of query-relevant words more accurately. As illustrated in Figure 1, Word2Passage assesses the importance of individual words by finely adjusting them based on the varying significance of each level, while also incorporating query type and domain characteristics to enhance word im-

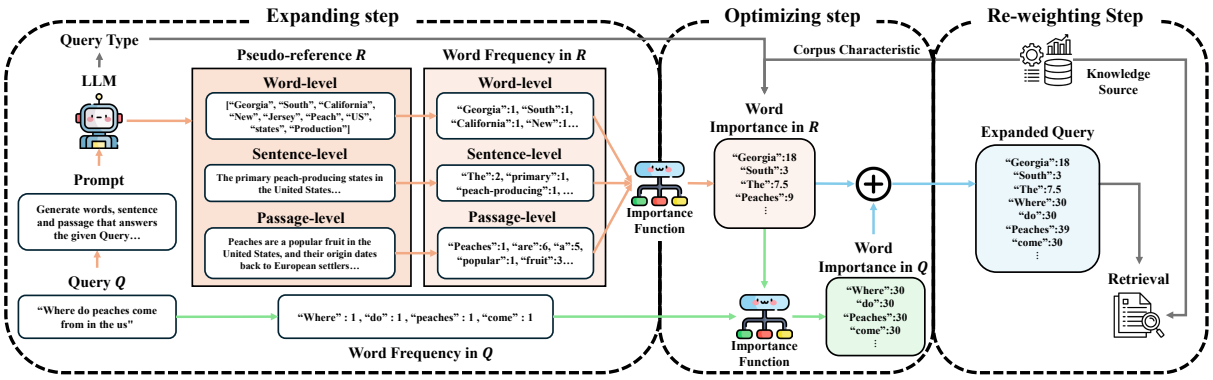

Figure 1: Overview of Word2Passage: The framework consists of three main steps. 1) Expanding Step, LLM generates word, sentence, and passage-level references based on query type. 2) Optimizing Step, importance scores of words are computed using corpus characteristics and query type-dependent reference-level weights. 3) Re-weighting Step, final word weights are determined by aggregating significance scores from both references and original query.

portance estimation. Specifically, Word2Passage performs through a three-step process:

- **Expanding Step**: This step expands the words in the original query Q by generating multi-level pseudo references using LLMs.
- **Optimizing Step**: This step estimates the importance I of each word in the query and reference. A word's weight in the pseudo reference is determined by two aspects: (1) its frequency at each level and (2) the significance scores of the three levels, adjusted based on the query type and domain characteristics. In contrast, words in the query are assigned importance weights solely based on their frequency. Finally, these weights are combined to produce the expanded query with world-level importance.
- **Re-weighting Step**: This step incorporates the word importance weights obtained earlier into the retriever score computation, ensuring they are reflected in the query-chunk scoring process. Then, we perform generation using the retrieved chunks following the standard RAG pipeline.

In particular, we reveal that the significance of each reference level depends on the query type, e.g., description and entity, rather than adhering a single standard (see Table 7). Therefore, we define five query categories, which can be easily classified by LLMs, allowing us to dynamically adjust the significance across words, sentences, and passages instantaneously. In addition, we provide an analysis of the impact of domain characteristics on our importance re-weighting method. It reveals that domain-specific lexical diversity is essential to consider, and can be captured by analyzing the average number of unique words per chunk across the corpus (see Section 4.3), as domains with repetitive terminology (e.g., Legal) tend to have lower unique word counts per chunk, while those with diverse expressions (e.g., News) exhibit higher counts.

Our main contributions are as follows:
(1) Word2Passage: We are the first to present a multi-perspective query expansion method that prompts LLMs to generate word, sentence, and passage level references. This multi-level approach enables a more fine-grained analysis of word importance than existing passage-only methods.
(2) Multi-level Adjustment: We propose a scheme that can reflects the varying contribution of multi-level references in importance re-weighting based on query type, effectively adapting the contribution of each reference level to enhance retrieval performance.
(3) Domain-aware Adjustment: We refine word importance weighting by considering the number of unique words in chunks, effectively capturing domain-specific lexical diversity. This prevents the incorrect overemphasis of words from references, enabling a more balanced expansion.

2 Related work

Information retrieval (IR) Information retrieval (IR) is a key component in RAG, where retrieval effectiveness directly impacts generation quality. Existing retrievers can be categorized into lexical-based (sparse) retrievers and embedding-based (dense) retrievers. *Lexical*-based retrievers, such as BM25 (E. Robertson et al., 2009), are efficient, interpretable, and robust to domain shifts, but struggle with semantic variations due to exact word matching. *Embedding*-based retrievers, such as DPR (Karpukhin et al., 2020), ANCE(Xiong et al., 2021), overcome this limitation by capturing semantic similarity, but require large-scale training and are sensitive to domain shifts.

To improve retrieval effectiveness, ensemble retrievers (Karpukhin et al., 2020; Xiong et al., 2021; Thakur et al., 2021) combine BM25's efficiency with dense retrieval's semantic capabilities, enhancing retrieval perfomance. Despite these advances, BM25 remains widely used for its zero-shot performance, efficiency, and interpretability, but struggles with semantic variations due to exact word matching. To address this, we propose Word2Passage, which enhances BM25 retrieval by enriching queries with semantically relevant words and re-weighting word importance.

Generation in RAG In Retrieval-Augmented Generation (RAG), the combination of Information Retrieval (IR) and LLMs allows the system to leverage external knowledge, improving the quality and accuracy of generated responses. Retrieval first retrieves relevant documents based on the query, and these documents are then used by the LLM to generate contextually appropriate answers. Thus, the performance of LLMs is crucial in generating coherent, contextually relevant answers based on the retrieved information. Several LLMs, including GPT-4 (OpenAI et al., 2024), LLaMA (Grattafiori et al., 2024), and Qwen (Bai et al., 2023), have been widely adopted in RAG-based systems. These models excel at generating natural language outputs, utilizing both the retrieved documents and the query to produce contextually appropriate responses.

Query Expansion Query expansion enhances search results by reformulating the original query to include additional relevant terms, addressing issues like vocabulary mismatch between users and documents (Huang et al., 2021). Traditional methods, such as Pseudo-Relevance Feedback (PRF), assume that top-ranked documents from an initial query are relevant and use terms from these documents to expand the query (Li et al., 2022; Lavrenko and Croft, 2017). However, PRF can be susceptible to errors if the initial retrieval includes irrelevant documents.

Advancements in LLMs have introduced new avenues for query expansion (Gao et al., 2023a; Wang et al., 2023; Song and Zheng, 2024; Chen et al., 2024; Kim et al., 2023; Lei et al., 2024). One approach leverages the generative capabilities of LLMs to expand queries, differing from traditional methods by relying on the model's inherent knowledge (Jagerman et al., 2023). Another method introduces a framework that employs LLMs to generate multiple pseudo-references, enhancing both sparse and dense retrieval systems (Zhang et al., 2024). They highlight a shift towards utilizing LLMs to improve query expansion, offering promising directions in information retrieval.

3 Proposed Method: Word2Passage

In this section, we start with formulating the impact of importance re-weighting in query expansion on the `<query, chunk>` score in retrieval, specifically within the BM25 framework. Next, we outline the three key components of Word2Passage: Expanding Q, Optimizing I, and Re-weighting t Steps.

3.1 Formulation of Query Expansion

The BM25 framework can be re-formulated to illustrate the impact of word-level importance re-weighting in query expansion as:

$$S(\tilde{Q}, \text{Chunk}) = \sum_{\forall (t, I_t) \in \tilde{Q}} I_t \cdot \text{BM25}(t, \text{Chunk}), \quad (1)$$

where $\tilde{Q}$ denotes an expanded query derived from the original query Q, using a query expansion method. Specifically, it is a set of tuples, each containing a unique word t inside and its corresponding importance score I_t. Here, $\text{BM25}(t, \text{Chunk})$ represents the original BM25 score of word t with respect to a given chunk. Accordingly, we confirm that two factors influence retrieval results in query expansion: (1) the set of unique words and (2) the importance score of each word.

HyDE (Gao et al., 2023a) focuses solely on generating an expanded query $\tilde{Q}$ without adjusting importance. Query2doc (Wang et al., 2013) and MuGI (Zhang et al., 2024) applies a uniform constant (or dynamic) weight only to the words in the original query Q to adjust the importance I_t. This family of methods do not make any weight adjustments to the unique word in pseudo passages.

Unlike these studies, we explicitly optimize both the expanded query $\tilde{Q}$ and the individual word weights I_t. From the Q perspective, we introduce multi-level references with hierarchical structure spanning three levels: word, sentence, and passage. From the I_t perspective, we define the word weight I_t by accounting for the varying contributions of different levels to word importance, while also separately inspecting importance based on whether the word is from the original query or the references. This enables precise word re-weighting, improving retrieval effectiveness.

The following sections detail the three components to implement Word2Passage.

Category	Generated
Query	"who played the predator in the movie predator"
Type	"Person"
Word	["Predator", "Kevin Peter Hall", "John McTiernan", "movie", "actor", "1987"]
Sentence	"Kevin Peter Hall portrayed the Predator in the 1987 movie Predator."
Passage	"The movie Predator is a science fiction action thriller directed by John McTiernan, released in 1987. The iconic alien species, the Predator, was brought to life by actor Kevin Peter Hall in the first film of the series. Known for its advanced technology and lethal hunting skills, the creature became a popular figure in popular culture. Throughout the series, the role of the Predator has been played by several actors, using elaborate costumes and prosthetics to recreate the creature's terrifying appearance. Kevin Peter Hall was the original actor to wear the Predator suit and is well-remembered for his portrayal in the 1987 film."

Table 1: Example multi-level references generated by Word2Passage in Expanding Q Step.

3.2 Expanding Q Step

We generate multiple pseudo references set, denoted as $\mathcal{R} = \{r_i \mid 1 \leq i \leq N\}$, where each pseudo reference r_i consists of three different levels of granularity (i.e., word, sentence, and passage) generated by a LLM[1] with our prompt (in Table 8 in Appendix). Here, N represents the number of generated pseudo references.

As demonstrated in Table 1, the word, sentence, and passage levels provide distinct contextual perspectives, enabling the extraction of diverse query-relevant words from the LLM's internal knowledge. Specifically, at each pseudo reference r_i:

• **Word**$_i$: A list of keywords likely to serve as answer candidates, extracted based on query relevance and concatenated into a single string.

• **Sentence**$_i$: A knowledge-intensive sentence that captures essential query-related context while preserving semantic coherence.

• **Passage**$_i$: A longer, more structured passage that provides additional supporting details and broader contextual information.

Then, each pseudo reference r_i is formulated as:

$$r_i = \text{Concat}(\text{Word}_i \ \text{Sentence}_i \ \text{Passage}_i), \quad (2)$$

where Concat denotes concatenation, combining the word, sentence, and passage-level outputs within each r_i as a single structured reference.

[1]Following prior work (Wang et al., 2023; Zhang et al., 2024), we use the same LLM employed for RAG.

Finally, we construct the set R of unique words appearing in all pseudo references in $\mathcal{R}$ as:

$$R = \text{Split}\big(\text{Concat}(r_i \mid r_i \in \mathcal{R})\big), \quad (3)$$

where Split$(\cdot)$ splits textual sequences into a set of words using a single space as the delimiter.

By expanding the query with semantically relevant words extracted across different granularity levels, our approach enhances BM25-based retrieval while maintaining interpretability and efficiency.

3.3 Optimizing I_t Step

We compute the importance scores of each word by separately evaluating their contributions to the pseudo references $\mathcal{R}$ and the original query Q. This step ensures that expanded queries retain essential words while incorporating relevant contextual terms. For simplicity, Q and R can represent either sets of words or textual sequences, corresponding to the original query and the combined reference text, respectively.

Importance of Words in Reference ($I_{t,R}$) To determine the importance score of each word t in the reference text R, we first compute its importance score I_{t,r_i} for each pseudo reference $r_i \in \mathcal{R}$. That is, we evaluate I_{t,r_i} for each word t appearing in r_i across $\mathcal{R}$, where $1 \leq i \leq N$. Here, since the effectiveness of extracting query-relevant words varies across different granularity levels depending on the query type, we define the significance score of each reference, I_{t,r_i}.

Note that since different query types require varying scopes and contextual depths of information, the significance score is influenced by the query type, which falls into five categories defined in MS MARCO (Bajaj et al., 2016) (see Appendix E): *description*, *person*, *entity*, *numeric*, and *location*. Let $F_{t,w_i}, F_{t,s_i}, F_{t,p_i}$ denote the frequency of a word t appearing at the word, sentence, and passage levels of r_i. Then, the importance score of a word t for a pseudo reference r_i is formulated as:

$$I_{t,r_i} = I_{q,w} F_{t,w_i} + I_{q,s} F_{t,s_i} + I_{q,p} \ F_{t,p_i}, \quad \text{where } t \in r_i \text{ and } 1 \leq i \leq N, \quad (4)$$

and $0 \leq I_{q,w}, I_{q,s}, I_{q,p}$; and q represents the query type. Here, $F_{t,w_i}, F_{t,s_i}, F_{t,p_i}$ serves as *intra*-level importance scores, measuring word importance within each respective level. Meanwhile, $I_{q,w}, I_{q,s}, I_{q,p}$ serve as the significant scores for word, sentence, and passage, respectively, controlling the *inter*-level relative contributions. With this

level-aware adjustment, we account for the varying significance of reference levels based on the query type, enabling dynamic weighting.

Specifically for the query type, it is first identified using an the same LLM (used for query expansion) by prompting the prompt in Table 9 of Appendix. Then, the corresponding importance scores are assigned to word-, sentence-, and passage-level references, as determined by our empirical analysis in Section A. This design allows us to reflect the uneven distribution of words across different levels, ensuring that words from more important levels receive higher weights while maintaining proportional contributions from less significant levels.

Next, we aggregate the score of a word t for a single reference r_i across all pseudo references in $\mathcal{R}$ to obtain the overall word importance score to the reference text. Instead of simple averaging, we apply *domain-aware* averaging, introducing a scaling factor α and W, which represents the average number of unique words per chunk within each corpus, to decay the original importance as:

$$I_{t,R} = \frac{\alpha}{\sqrt{W}} \sum_i I_{t,r_i} \quad {}^\forall t \in R. \tag{5}$$

Without this adjustment, a corpus (i.e., domain) with a high number of unique word can cause excessive expansion during the Expansion Q step. As a result, the importance of words in the original query is marginalized, as their relative importance is diluted by the large number of expanded words from pseudo-references. This also prevents the expanded query from unfairly favoring longer chunks, ensuring a more balanced retrieval process.

Importance of Words in Query ($I_{t,Q}$) Now we determine the contribution of a word t to the original query Q. We compute the importance of words in the original query Q, denoted as $I_{t,Q}$, where $t \in Q$, and being formulated as:

$$I_{t,Q} = \frac{\sum_{t' \in \mathcal{R}} F_{t',\mathcal{R}}}{\sum_{t' \in Q} F_{t',Q}} \cdot F_{t,Q} \quad {}^\forall t \in Q, \tag{6}$$

where F is the frequency of a word t in either the reference text R or the original query Q. The rightmost term ($F_{t,Q}$) is the original contribution of a word t to the query. Contrary to the right one, the left term acts as a normalization mechanism between the query and its pseudo references, adjusting the importance score of words in the query to balance the influence of the two word sets: one from the query (typically smaller) and the other from the pseudo-references (typically larger).

Aggregation for Expanded Query (I_t) The overall word importance score is computed by integrating contributions from both the reference text and the original query:

$$\begin{aligned} I_t = I_{t,R} + I_{t,Q} \quad {}^\forall t \in R \cup Q, \\ \text{where } I_{t,Q} = 0 \text{ if } t \notin Q, \\ I_{t,R} = 0 \text{ if } t \notin R. \end{aligned} \tag{7}$$

Note that we optimize word importance separately for the pseudo reference and the query, followed by aggregating their importance. This approach ensures a well-balanced importance aggregation between words from the two sources.

3.4 Re-weighting t step

We refine word importance to enhance relevant words while suppressing less informative ones under the BM25 framework. Therefore, the expanded query $\tilde{Q}$ is formed by aligning the unique word t in $R \cup Q$ with its final word-level importance score computed in Eq. (7). Therefore, the re-weighting is applied to the BM25-like retrieval as:

$$\begin{aligned} S(\tilde{Q}, \text{Chunk}) = \sum_{{}^\forall (t, I_t) \in \tilde{Q}} I_t \cdot \text{BM25}(t, \text{Chunk}) \\ \text{where } \tilde{Q} = \{(t, I_t) \mid t \in R \cup Q\}. \end{aligned} \tag{8}$$

Retrieval and Generation Pipeline Given the score function $S(\tilde{Q}, \text{Chunk})$, we select the top-$K$ ranked chunks defined as:

$$D_{\tilde{Q}} = \{d \in C \mid \text{rank}(d; S, \tilde{Q}) \leq K\}, \tag{9}$$

where C is the entire corpus of chunks and rank is a function that returns the rank of a chunk d based on the score function $S(\tilde{Q}, d)$.

Then, the retrieved chunks are then utilized as context for response generation as:

$$\text{Response} = \text{LLM}(Q, D_{\tilde{Q}}), \tag{10}$$

where the language model generates a response conditioned on both the original query Q and the top-K ranked chunks $D_{\tilde{Q}}$.

$$D_{\tilde{Q}} = \{d \in C \mid \text{rank}(S(\tilde{Q}, d)) \leq K\}, \tag{11}$$

4 Experiments

In this section, we conduct experiments for IR and QA tasks, the two main tasks of RAG.

4.1 Experiment Setup

Datasets For the IR task, we conduct experiments on 11 IR datasets from the BeIR benchmark (Thakur et al., 2021), including DL19–20, Covid, Toche, SciFact, NFC, Arguana, Scidocs, Hotpot, NQ, and FiQA. For QA task, 5 QA datasets based on the WikiCorpus. In particular, the QA datasets are categorized according to their reasoning complexity:

- **Single-hop QA**: NQ (Kwiatkowski et al., 2019), SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017)
- **Multi-hop QA**: HotpotQA (Yang et al., 2018)
- **Long-form QA**: FiQA (Maia et al., 2018)

Among these, three datasets, i.e., NQ, HotpotQA, and FiQA, are used for QA while also being included in the BeIR benchmark for IR evaluation. This allows us to analyze how well retrieval performance aligns with answer generation quality. Other IR datasets are not suitable for the QA task as they lack corresponding QA pairs.

Metrics We evaluate retrieval effectiveness using nDCG@10, a widely adopted metric for IR. For QA, we measure performance using Accuracy (Acc) and LLM-based evaluation (LLM Eval) (See Table 10) (Rau et al., 2024), which assesses the quality of generated responses beyond traditional lexical overlap metrics. Regardig LLM Eval, we employ GPT4o as the LLM evaluation model for assessing the quality of generated responses.

4.2 Implementation Details

Baselines To assess impact the retrieval effectiveness and generation quality, we compare Word2Passage (W2P) with four existing retrieval methods: one canonical lexical retreival, BM25 (E. Robertson et al., 2009), and three latest query expansion approaches, including Query2Doc (Q2D) (Wang et al., 2023), HyDE (Gao et al., 2023a), and MuGI (Zhang et al., 2024).

LLM Backbones For IR and QA datasets, we conduct experiments using three instruction-tuned LLMs: Llama3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Qwen2.5-72B-Instruct, which serve as the backbone for generating the pseudo passages for HyDE, Q2D, and MuGI; or the pseudo references of Word2Passage. The 7B, 8B and 72B models are run on NVIDIA L40s GPUs, while the GPT4o models are accessed via their respective APIs. For the QA task, all answer generation is performed using Llama3.1-8B-Instruct.

Retriever For retrieval, we use LuceneSearcher (Pérez-Iglesias et al., 2009; Lin et al., 2021) as the BM25 retriever with default BM25 parameters, following the literature (Gao et al., 2023a; Zhang et al., 2024; Shen et al., 2024). We set the top-k to be 10 for all experiments. Our method is tailored for BM25-like retrieval but achieves synergy when combined with dense retrieval. This adaptability is another strength (see Section 4.7).

Corpus We use the Wikipedia corpus from DPR (Karpukhin et al., 2020), which contains 21M processed chunks, as the document corpus for SQuAD and TriviaQA. For all other datasets, we use their respective corpora from BeIR (Thakur et al., 2021) to ensure consistency with prior work.

Hyperparameters Our method introduces hyperparameters: the scaling factor α, and the significance scores for different levels of word generation, i.e., $I_{q,w}$ (word-level), $I_{q,s}$ (sentence-level), and $I_{q,p}$ (passage-level), where N represents the number of pseudo reference generations. We fix $\alpha = 30$ and $N = 5$ across all datasets.

We determine the best values of $I_{q,w}$, $I_{q,s}$, and $I_{q,p}$ through grid search on a balanced subset of 500 queries from the training set, sampling 100 queries for each of the five query types. While these parameters may add complexity, the process remains efficient, requiring only a few hundred data points and typically completing within 1–2 hours, depending on the corpus size. Our analysis reveals that the best values varies depending on query type, as presented in Table 7. The detailed analysis is presented in Appendix C.

4.3 Task 1: Information Retrieval

Table 2 shows the IR performance of four query expansion methods, along with the canonical BM25 as a reference. Among the four methods, Q2D uniquely applies few-shot demonstration in passage generation, thus we borrow the results from the original paper (Wang et al., 2023).

In general, **Word2Passage achieves the highest nDCG@10 scores over other baselines** in most cases, consistently outperforming across both backbone types and datasets. This suggests that Word2Passage's word-level re-weighting method is more effective than the passage-level re-weighting methods employed by other methods. That is, finely adjusting word importance scores by incorporating multi-level references alongside the original query is essential for achieving higher IR performance.

LLM	Method	IR										
		DL19	DL20	Covid	Touche	SciFact	NFC	Arguana	Scidocs	Hotpot	NQ	FiQA
-	BM25	50.6	48.0	59.5	44.2	67.9	32.2	30.5	14.9	65.3	28.9	23.4
ChatGPT-3.5	Q2D	66.2	62.9	72.2	39.8	68.6	34.9	-	-	-	-	-
Llama3.1-8B-Inst.	HyDE	47.6	48.8	59.9	41.8	67.0	31.8	25.6	12.9	52.9	42.8	18.6
	MuGI	66.5	61.1	73.1	49.6	72.2	**36.4**	29.4	**15.3**	65.2	50.2	24.3
	W2P	**68.1**	**62.5**	**78.4**	**50.7**	**72.4**	36.1	**32.5**	**15.3**	**71.9**	**50.4**	**26.1**
Qwen2.5-7B-Inst.	HyDE	43.9	41.8	56.4	34.8	67.0	28.4	24.8	12.0	48.8	29.7	17.3
	MuGI	65.8	**62.9**	67.7	44.2	**71.6**	36.5	28.9	14.7	67.8	43.9	24.7
	W2P	**67.5**	62.8	**77.0**	**49.6**	71.5	**36.5**	**32.7**	**15.6**	**70.1**	**44.4**	**26.3**
Qwen2.5-72B-Inst.	HyDE	52.9	52.5	59.2	38.6	68.5	32.2	26.0	13.1	56.9	39.1	18.2
	MuGI	69.4	62.7	70.3	47.3	**72.8**	36.4	28.3	15.2	72.2	49.7	25.5
	W2P	**69.7**	**64.1**	**75.6**	**48.1**	72.1	**36.9**	**33.3**	**15.5**	**73.6**	**50.3**	**26.6**

Table 2: IR performance for four different retrieval methods using varying LLM backbones. Performance is measured using nDCG@10. The best nDCG@10 score is marked in bold for each dataset, as well as for each backbone.

LLM	Method	Hotpot		NQ		FiQA		SQuAD		Trivia	
		Acc	LLM Eval	Acc	LLM Eval	Acc	LLM Eval	Acc	LLM Eval	Acc	LLM Eval
-	BM25	31.2	49.4	42.4	56.8	-	21.6	28.8	48.8	52.4	71.8
Llama3.1-8B-Inst.	HyDE	30.3	45.8	46.2	63.6	-	22.2	23.8	39.8	53.6	72.6
	MuGI	32.2	50.2	49.2	64.6	-	22.8	27.6	48.6	55.2	75.2
	W2P	**35.8**	**54.2**	**51.8**	**68.0**	-	**25.2**	**31.4**	**52.0**	**56.0**	**76.6**
Qwen2.5-7B-Inst.	HyDE	26.0	38.8	38.0	52.8	-	22.4	21.6	37.4	49.6	68.6
	MuGI	34.2	51.0	47.2	64.8	-	24.2	31.4	51.6	55.8	75.4
	W2P	**34.8**	**54.0**	**48.2**	**66.0**	-	**26.8**	**31.8**	**51.8**	**56.6**	**77.2**

Table 3: QA performance for three retrieval methods using two LLM backbones. We reports both accuracy (Acc) and LLM-based metric (LLMEval). The best values are marked in bold for each dataset, as well as for each backbone.

W2P significantly outperforms other methods (HyDE and MuGI) in Covid data. This dataset belong to the Medical domain, focusing on biomedical literature. Unlike other domains (e.g., News and Simple QA), the medical domain exhibits distinct domain characteristics, particularly in terms of lexical diversity—it contains highly specialized terminology, frequent abbreviations, and complex multi-word expressions that are uncommon in general text. Therefore, it confirms that **Word2Passage effectively handles domain-specific lexical diversity in word-level re-weighting**, thereby achieving significantly higher IR performance than others.

4.4 Task 2: Question and Answering

While HyDE, Q2D, and MuGI (Wang et al., 2023; Zhang et al., 2024; Gao et al., 2023a) have focused primarily on IR evaluation without reporting their QA performance in RAG, it remains uncertain whether gains in IR performance directly lead to better QA results. Therefore, evaluating both IR and QA performance is crucial. Table 3 shows the QA performance of three query expansion methods, along with the canonical BM25 as a reference. Note that we omit the Acc scores for the FiQA dataset, as all values are 0 due to its long-form QA nature.

Interestingly, **performance gains in the IR task do not translate proportionally to improvements in the QA task**, indicating that enhanced retrieval does not always lead to a corresponding level of QA improvement. This is evident in the NQ dataset, where Word2Passage shows only marginal improvement of 0.2–0.5 (in nDCG@10) over MuGI in the IR task (see the 2nd last column in Table 2), yet achieves a significantly larger performance boost of 1.0–2.6 (in Acc) and 1.2–3.4 (in LLMEval) in the QA task. This demonstrates that Word2Passage's word-level re-weighting is likely to yield greater performance gains in QA tasks than in IR tasks.

Overall, across all datasets, Word2Passage consistently outperforms the other three query expansion methods across both evaluation metrics and LLM backbones.

4.5 Component Ablation Study

In Table 4, we analyze the effects of two main components of Word2Passage:

(1) "Multi-level:" Removing contribution differences among word-, sentence-, and passage-levels by assigning a uniform significance of "1" in Eq. (4).

(2) "Domain-aware:" Removing the domain-aware adjustment factor. We simply set "$W = 1$" in Eq. (5).

Firstly, the results show that, across most datasets, equalizing the contribution of multi-level references in Word2Passage leads to a decline in nDCG@10 performance. Notably, the extent of performance

LLM	Method	IR					IR & QA		
		Covid	Touche	NFC	Arguana	Scidocs	Hotpot	NQ	FiQA
Llama3.1-8B-Inst.	W2P	78.4	50.7	36.1	32.5	15.3	71.9	50.4	26.1
	(-) Multi-level	73.4	50.2	35.8	31.5	15.4	71.9	49.5	25.2
	(-) Domain-aware	73.6	47.9	36.0	30.9	15.4	71.3	49.5	24.8
Qwen2.5-7B-Inst.	W2P	77.0	49.6	36.5	32.7	15.6	70.1	44.4	26.3
	(-) Multi-level	74.7	48.9	36.5	31.9	15.2	69.6	44.3	25.8
	(-) Domain-aware	74.2	46.6	36.5	31.4	15.3	69.9	44.1	25.6

Table 4: Ablation study on Word2Passage in the IR task (nDCG@10), excluding (1) the contribution differences among three-level references and (2) the contribution differences and domain-aware adjustment.

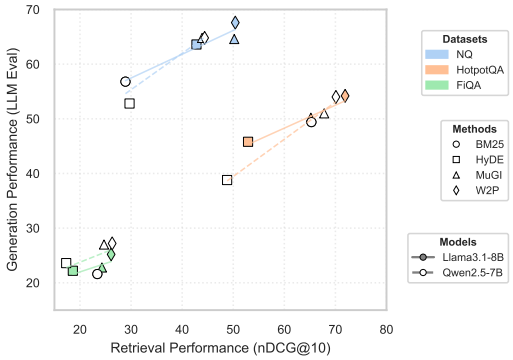

Figure 2: Analysis on the correlation between nDCG@10 in IR and LLMEval in QA task across three datasets.

drop varies across datasets, indicating that the **significance scores of the three levels exhibit high variability within each dataset**. This highlights the crucial role of properly defining these scores in achieving performance improvements.

Secondly, when both key components, Multi-level and Domain-aware, are eliminated, performance generally drops, but the decline is less pronounced compared to removing only the multi-level contribution adjustment. This highlights that **the use of multi-level significance scoring plays a more critical role in performance improvements** than domain-aware adjustment, suggesting that capturing hierarchical importance is essential for effective ranking.

4.6 Alignment between IR and QA

Lastly, we perform an analysis on the alignment between IR and QA performance. This analysis is conducted on three datasets, i.e., NQ, HotpotQA, and FiQA, all of which provide ground-truth chunk IDs for IR and answers for QA. Figure 2 shows the nDCG@10 and LLMEval scores for IR and QA tasks, respectively. To ensure a sufficient number of data points, we plot the results across four methods and two backbone architectures for each dataset.

The results show a fairly high positive correlation of 0.69–0.96 between retrieval and generation performance in three datasets. This indicates that **higher-quality retrieval is likely to lead to better answer generation in RAG.** However, as mentioned in Section 4.3, a closer examination reveals that there are cases where this trend does not hold.

Additionally, for each dataset, we observe that Word2Passage locates the top-most and right-most points in the plot. This indicates that it generally achieves the highest retrieval and generation performance. This suggests that Word2Passage effectively enhances both retrieval quality and downstream QA performance by enabling effective word-level re-weighting in retrieval.

Retrieval Method	Llama3.1-8B-Instruct	Qwen2.5-7B-Instruct
BM25 (Sparse)	23.4	23.4
Dense	16.0	16.0
BM25 + Dense	25.5	25.5
W2P (Sparse)	26.1	26.3
W2P + Dense	**27.8**	**27.7**

Table 5: IR (nDCG@10) performance for sparse and dense retrieval methods and their ensemble on FiQA. The best value score is marked in bold for each backbone.

4.7 Ensemble with Dense Retrieval

Another advantage of Word2Passage is its synergy with dense retrieval, where the ensemble selects the top-5 from both sparse and dense, removing duplicates. Table 5 shows that, while dense retrieval alone performs poorly, integrating it with the canonical BM25 enhances performance. This improvement persists when BM25 is replaced with W2P, and ensembling further amplifies the synergy, achieving the highest nDCG@10 score.

5 Conclusion

We introduce Word2Passage, a word-level re-weighting approach for query expansion. By generating multi-level references and optimizing word-level importance, Word2Passage enhances query expansion effectiveness and improves retrieval performance. Experimental results demonstrate that Word2Passage consistently outperforms existing methods, including HyDE, Q2D, and MuGI, across diverse datasets and LLM backbones. Furthermore, Word2Passage exhibits synergy when integrated with a dense retrieval approach.

6 Limitations

Our method has several limitations that motivate future work.

First, the word-level significance scores are tuned via grid search on a per-dataset basis, which is computationally expensive. Future work could explore developing a model that directly predicts optimal importance scores given a query and its multi-level references, potentially offering more general and precise tuning than grid search.

Second, our approach is currently limited to BM25-based retrieval. While BM25 is a powerful retriever, hybrid approaches combining sparse and dense retrievers have shown superior performance. Extending Word2Passage to dense retrievers or developing a hybrid approach remains an important direction for future work.

Additionally, the method requires multiple LLM calls for generating multi-level references, which can be time-consuming and costly. Future research could investigate more efficient reference generation strategies or methods to reduce the number of required LLM queries while maintaining performance. Finally, while our query type-based importance scoring is effective, it relies on predefined query categories. Developing a more flexible and fine-grained query analysis system could potentially lead to better word importance estimation.

Ethical Statement. Our research centers on query expansion to improve retrieval performance through Word2Passage. As our study relies predominantly on outputs generated by well-established open-source models and publicly accessible datasets, it does not involve the collection of sensitive or personally identifiable information. Consequently, our work does not present any immediate ethical concerns regarding privacy or data security.

Scientific Artifacts. We conducted experiments with Llama3.1-8B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-72B-Instruct, and GPT4o to evaluate our approach across varying model scales (see Section 4.2). We used Hugging Face checkpoints for Llama3.1-8B-Inst. and Qwen2.5-7B-Instruct, Qwen2.5-72B-Instruct and accessed GPT4o via the OpenAI API. The prompts are provided in Table 8 in the Appendix for reproducibility.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, et al. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. Ms marco: A human generated machine reading comprehension dataset. In *NIPS*.

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 *ACL*.

Stephen E. Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023a. Precise zero-shot dense retrieval without relevance labels. In *ACL*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Minghui Huang, Dong Wang, Shuang Liu, and Meizhen Ding. 2021. Gqe-prf: Generative query expansion with pseudo-relevance feedback. *arXiv preprint arXiv:2108.06010*.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *EMNLP*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur 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. In *ACL*.

Victor Lavrenko and W. Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 3–15. ACM.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1–10. Association for Computational Linguistics.

Patrick 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 *NeurIPS*.

Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2022. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–35.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *SIGIR*.

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

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Open challenge: Financial opinion mining and question answering. In *WWW*.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. Rafe: Ranking feedback improves query rewriting for rag. In *EMNLP*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Joaquín Pérez-Iglesias, José R. Pérez-Agüera, Víctor Fresno, and Yuval Z. Feinstein. 2009. Integrating the probabilistic model bm25/bm25f into lucene. *arXiv preprint arXiv:0911.5046*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

David Rau, Hervé Déjean, Nadezhda Chirkova, Thibault Formal, Shuai Wang, Stéphane Clinchant, and Vassilina Nikoulina. 2024. Bergen: A benchmarking library for retrieval-augmented generation. In *EMNLP*.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *ACL*.

Mingyang Song and Mao Zheng. 2024. A survey of query optimization in large language models. *arXiv preprint arXiv:2412.17558*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS*.

Haoyu Wang, Ruirui Li, Haoming Jiang, Jinjin Tian, Zhengyang Wang, Chen Luo, Xianfeng Tang, Monica Xiao Cheng, Tuo Zhao, and Jing Gao. 2024. Blendfilter: Advancing retrieval-augmented large language models via query generation blending and knowledge filtering. In *EMNLP*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *EMNLP*.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. 2013. A theoretical analysis of ndcg type ranking measures. In *COLT*.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024. Exploring the best practices of query expansion with large language models. In *EMNLP*.

A Analysis of Reference Level Significance by Query Type

A.1 Methodology

In our approach to query expansion using pseudo references, we assign different weights to word, sentence, and passage-level expansions based on the query type. The weighting scheme was determined through grid search for each dataset and query type, which includes five categories: description, entity, person, numeric, and location.

A.2 Query Type Characteristics

Description Queries Description-type queries seek explanatory information, requiring a comprehensive understanding of the topic. Passage-level expansion plays a dominant role as it provides richer context and more complete answers. While key terms are important, they alone lack sufficient context. Sentence-level expansion provides additional meaning but remains less effective than passage-level expansion.

Entity Queries Entity-type queries aim to identify specific named entities, such as organizations, products, or technologies. Sentence-level expansion proves most effective, as entities are typically well-defined within a single sentence. Word-level and passage-level expansions demonstrate comparable performance, with the former capturing core entity terms and the latter providing broader context without yielding substantial performance gains.

Person Queries Person-type queries retrieve information about specific individuals. Personal details are typically captured at the word or sentence level. Names are key identifying factors, making word-level expansion highly relevant. Sentences often contain structured information about individuals (e.g., "Marie Curie was a physicist"). While passages may offer biographical details, most queries are effectively answered at a finer granularity.

Numeric Queries Numeric-type queries focus on retrieving numerical values. These values are usually embedded within sentences that provide necessary context (e.g., "The GDP growth rate in 2023 was 3.1%"). Context from multiple sentences can be useful for trend analysis but is not always necessary. Numbers alone lack meaning without surrounding context, making word-level expansion least effective.

Location Queries Location-type queries target geographic names. Such information is often explicitly mentioned in individual words or concise sentences (e.g., "Seoul is the capital of South Korea"). Since location names themselves carry significant information, word-level expansion plays a crucial role. While additional background information may be useful, retrieval effectiveness is primarily determined at the sentence and word levels.

Query Type	Relative Ratio of Significance Scores		
	Word-level	Sentence-level	Passage-level
Description	0.32	0.25	0.43
Entity	0.29	0.41	0.30
Person	0.38	0.38	0.24
Numeric	0.28	0.40	0.32
Location	0.38	0.38	0.24

Table 6: Average relative ratio of significance scores across word-level, sentence-level, and passage-level references.

A.3 Impact on Reference Level Weights

Table 6 shows the relative ratios of significance scores across different reference levels for each query type. Our analysis reveals that the optimal weight distribution aligns with the intrinsic characteristics of each query type:

- Description queries benefit most from passage-level expansion (0.43) due to their need for rich contextual information
- Entity queries achieve optimal performance with sentence-level expansion (0.41), where entity-specific details are most prevalent
- Person and location queries rely on balanced word- and sentence-level expansions (both 0.38)
- Numeric queries are best captured at the sentence level (0.40), where values appear with necessary context

B Domain-aware Adjustments

In this section, we analyze the characteristics of different corpora and the effect of the domain-aware factor W on token importance calculation.

B.1 Characteristics of each corpus

Figure 3 shows the distribution of unique tokens per chunk across different datasets through box plots,

where the red line indicates the average number of unique tokens per chunk and the box represents the interquartile range (IQR).

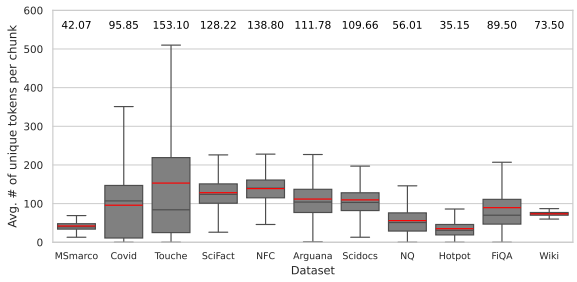

Figure 3: Distribution of unique tokens per chunk across different datasets. The numbers above each box represent the average number of unique tokens per chunk for each dataset.

The average number of unique tokens per chunk varies significantly across datasets. MSmarco (42.07), NQ (56.01), and Hotpot (35.15) show relatively low token counts with small IQRs, indicating consistent chunk sizes. In contrast, Touche (153.10), NFC (138.80), and SciFact (128.22) exhibit high token counts, with Touche showing the largest variance as evidenced by its extended whiskers.

A middle range of token counts is observed in several datasets: Covid (95.85), FiQA (89.50), and Wiki (73.50). Among these, Covid shows notably larger variance in its distribution compared to FiQA and Wiki, as shown by its larger box size and longer whiskers.

The academic datasets—SciFact (128.22), Arguana (111.78), and Scidocs (109.66)—demonstrate similar average token counts and relatively consistent distributions, as indicated by their comparable box sizes and whisker lengths.

B.2 Effect of W on Word Importance

To investigate the effect of domain-aware factor W on word importance I_t, we analyze Eq. (7):

$$\begin{aligned} I_t &= I_{t,R} + I_{t,Q} \\ &= \frac{\alpha}{\sqrt{W}} \sum_i I_{t,r_i} + \frac{\sum_{t' \in \mathcal{R}} F_{t',\mathcal{R}}}{\sum_{t' \in Q} F_{t',Q}} \cdot F_{t,Q} \\ &\text{where } I_{t,Q} = 0 \text{ if } t \notin Q, \\ &\qquad I_{t,R} = 0 \text{ if } t \notin R, \\ &\qquad \forall t \in R \cup Q. \end{aligned} \tag{12}$$

Since W directly affects $I_{t,R}$ but not $I_{t,Q}$, analyzing their relationship requires controlling the relative influence between $I_{t,R}$ and $I_{t,Q}$. We denote $\frac{\sum_{t' \in \mathcal{R}} F_{t',\mathcal{R}}}{\sum_{t' \in Q} F_{t',Q}}$ as β, which represents the relative word frequency between reference and query. By varying β, we can observe how different balances between importance of words in reference and query affect retrieval performance, indirectly revealing the optimal range for W.

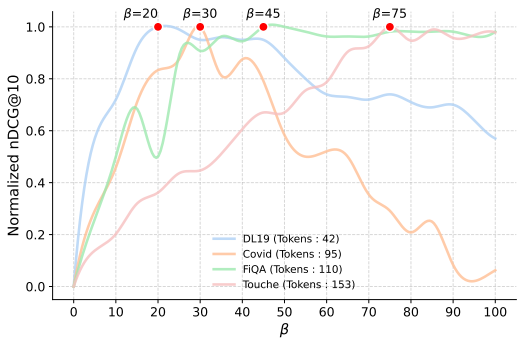

Figure 4: Relationship between β and retrieval performance. The x-axis represents β values, and the y-axis shows normalized nDCG@10 scores (scaled to [0,1] for each dataset). The plot demonstrates how optimal β values increase with the average number of unique tokens per chunk.

As shown in the figure 4, datasets with higher average numbers of unique tokens per chunk achieve optimal nDCG@10 scores at larger β values. This observation suggests that as corpus complexity increases, more emphasis needs to be placed on words in query relative words in reference. In our formulation, this balance is automatically achieved through W: when a corpus has more unique tokens per chunk (larger W), the term $\frac{\alpha}{\sqrt{W}} \sum_i I_{t,r_i}$ decreases, effectively reducing the influence of reference tokens. This confirms that our domain-aware factor W appropriately adapts to varying corpus characteristics.

C Grid Search Configuration

This section details our methodology for determining optimal significance scores across different reference levels (word, sentence, and passage) through a comprehensive grid search process, evaluated using the nDCG@10 metric. For the complete results showing optimal significance scores across different query types and datasets, see Table 7.

C.1 Dataset Preparation

For each dataset and query type, we implemented a systematic sampling approach using the training

data when available. In cases where only validation data was available, we utilized the validation set as our training data. For datasets lacking both training and validation splits, we employed a domain-based grouping strategy, clustering datasets with similar domain characteristics. Specifically, we formed four groups sharing similar corpus properties:

- MS MARCO group (DL19, DL20): Web documents with diverse topics and general domain knowledge
- Financial/Medical group (FiQA, Covid): Domain-specific documents with technical terminology
- Scientific/Academic group (Scifact, Scidocs, Arguana): Research papers, scientific articles, and academic arguments
- News/Factual group (NFC, Touche): News articles and fact-checking documents

These groupings reflect the inherent similarities in document structure, vocabulary, and information density within each domain. To ensure balanced representation across query types, we constructed a standardized training set comprising 100 queries per query type.

C.2 Significance Score Optimization

We conducted a grid search across all reference levels with a search range of (0, 1.6] and step size of 0.2, using nDCG@10 as our evaluation metric. All grid search experiments were performed exclusively on the training datasets to ensure fair evaluation. Rather than selecting the configuration with the highest nDCG@10 score, which might lead to overfitting on the training data, we opted for a more robust approach by identifying the 95th percentile point of performance across all configurations. The combination of word, sentence, and passage significance scores at this percentile was selected as our final configuration and applied to the test set without further adjustment.

C.3 Analysis of Score Distribution

Figure 5 illustrates the distribution of training set nDCG@10 scores across different combinations of word, sentence, and passage significance scores. The visualization reveals that the top 5% performing points form distinct clusters, indicating the existence of consistent patterns in the relationship between significance scores and query types. Our selected point, corresponding to the 95th percentile of training performance, is strategically positioned within these clusters. This positioning ensures that no single reference level dominates the others with an exceptionally high significance score, thereby promoting robust performance across different query scenarios. The balanced nature of our selected point suggests that it can effectively generalize to the test set while maintaining stable performance characteristics.

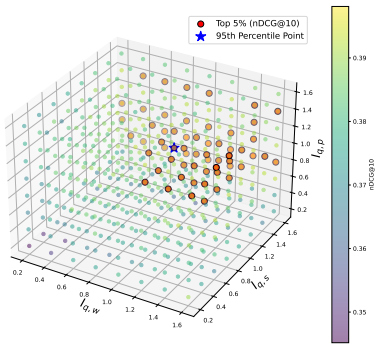

Figure 5: Distribution of nDCG@10 scores on training data across different combinations of word, sentence, and passage significance scores for person queries in DL19.

D Prompts

This section presents the prompts used in our experiments. Table 8 shows the prompts used for generating word, sentence, and passage-level references in Word2Passage. For query type classification, we use the prompts detailed in Table 9. The prompt used for LLM-based evaluation is provided in Table 10.

E Word2Passage Generation Examples by Query Type

We provide representative examples of Word2Passage's reference generation across different query types to illustrate how our method adapts to varying information needs. Tables 11-15 present examples for five different query types: description, person, entity, numeric, and location.

For each query type, word-level references extract key terms and concepts, sentence-level references provide concise but structured information, and passage-level references offer comprehensive context. Description queries focus on explanatory content, person queries capture hierarchical relationships, entity queries identify core characteristics,

numeric queries handle quantitative information, and location queries establish geographical context.

The highlighted spans in each example indicate word matches with ground truth chunks, demonstrating how different reference levels contribute to capturing relevant information while maintaining precision at different granularities.

Query Type	DL19-20	Covid	NFC	Touche	SciFact	Arguana
Description	(0.2, 0.6, 1.6)	(0.4, 0.6, 0.4)	(0.4, 0.2, 1.2)	(0.4, 0.2, 1.2)	(1.2, 0.4, 0.2)	(1.2, 0.4, 0.2)
Entity	(1.2, 0.8, 0.4)	(0.6, 1.4, 0.2)	(0.4, 0.4, 0.4)	(0.4, 0.4, 0.4)	(0.2, 0.2, 0.2)	(0.2, 0.2, 0.2)
Person	(0.8, 1.4, 0.8)	(1.2, 1.4, 0.2)	(0.8, 0.6, 0.4)	(0.8, 0.6, 0.4)	(1, 1, 1)	(1, 1, 1)
Numeric	(1.6, 1.4, 1.4)	(1.2, 1.2, 1.2)	(0.4, 0.6, 0.2)	(0.4, 0.6, 0.2)	(0.2, 0.8, 0.8)	(0.2, 0.8, 0.8)
Location	(1.2, 1.6, 0.2)	(0.8, 0.2, 0.4)	(1, 1, 1)	(1, 1, 1)	(1, 1, 1)	(1, 1, 1)

Query Type	Scidocs	Hotpot	NQ	FiQA	SQuAD	Trivia
Description	(1.2, 0.4, 0.2)	(1.4, 0.6, 1.0)	(0.2, 1.2, 1.6)	(0.4, 0.6, 0.4)	(1.0, 0.8, 1.6)	(1.6, 0.8, 1.2)
Entity	(0.2, 0.2, 0.2)	(0.4, 1.0, 1.2)	(0.6, 0.8, 1.2)	(0.6, 1.4, 0.2)	(0.4, 0.6, 1.0)	(0.8, 1.4, 0.2)
Person	(1, 1, 1)	(0.8, 1.6, 0.6)	(1.6, 1.2, 0.4)	(1.2, 1.4, 0.2)	(1.4, 0.6, 1.4)	(1.6, 1.2, 1.0)
Numeric	(0.2, 0.8, 0.8)	(1.4, 1.4, 1.2)	(1.6, 1.6, 0.2)	(1.2, 1.2, 1.2)	(0.4, 1.6, 1.2)	(0.6, 0.8, 1.6)
Location	(1, 1, 1)	(1.6, 1.2, 0.8)	(1.2, 1.4, 0.8)	(0.8, 0.2, 0.4)	(0.6, 1.4, 0.8)	(0.8, 1.0, 0.4)

Table 7: Significance scores for word, sentence, and passage-level references ($I_{q,w}$, $I_{q,s}$, $I_{q,p}$) obtained by grid search on a balanced subset of 500 queries from the training set. Note that (1,1,1) is assigned for query types absent in the training set to ensure stability.

Word2Passage Generation Prompt

Generate a passage, a sentence, and words that answer the given QUERY.
Terms that are important for answering the QUERY should frequently appear in the generation of the passage, the sentence, and words.

Definition:

passage: Answer the given QUERY in a passage perspective by generating an informative and clear passage.

sentence: Answer the given QUERY in a sentence perspective by generating a knowledge-intensive sentence.

word: Answer the given QUERY in a word perspective by generating a list of words.

QUERY:

{query}

FINAL OUTPUT JSON FORMAT (strictly follow this structure):

{{
"passage": "Your passage here",
"sentence": "Your sentence here",
"word": [Your words here],
}}

(From here on, only produce the final output in the specified JSON format.)

Table 8: Prompt of Word2Passage

Query Type Classification Prompt
You are given a dataset containing queries categorized into different types. Here are some examples: Query Type: description - Query: causes of inflamed pelvis - Query: name the two types of cells in the cortical collecting ducts and describe their function Query Type: numeric - Query: military family life consultant salary - Query: average amount of money spent on entertainment per month Query Type: location - Query: what is the biggest continent - Query: where is trinidad located Query Type: entity - Query: what kind of plants grow in oregon? - Query: what are therapy animals Query Type: person - Query: who is guardian angel cassiel - Query: interstellar film cast Now, classify the following query into one of the above categories. Choose only one of the following categories: [description, numeric, location, entity, person] Query : {query} ### OUTPUT FORMAT Query Type: your answer (must be one of the categories listed above)

Table 9: Prompt of query type classification

LLM Evaluation Prompt
You are an evaluation tool . Just answer by Yes or No . Here is a question , a golden answer and an AI-generated answer. Judge whether the AI-generated answer is correct according to the question and golden answer , answer with Yes or No . Question : { Question } Golden answer : { Golden answer } Generated answer : { Generated answer } Response :

Table 10: Prompt of LLM Evaluation from BERGEN(Rau et al., 2024)

Dataset	FiQA
Query Type	Description
Query	"How to pay with cash when car shopping?"
GT Answer	You could write a personal check after the final price has been set and you're ready to purchase. Another option would be to get the final price - then walk over to your bank and get a cashier's check.
Word	cash payment car shopping dealership policies payment procedures accept confirm purchase taxes fees process negotiate finance loan
Sentence	To pay with cash when car shopping it's essential to call ahead or check the dealership's website to confirm their payment policies as some may not accept large sums or require full payment in advance
Passage	When shopping for a car many consumers prefer to use cash as a payment method. However car dealerships may have varying policies and procedures for accepting cash payments. To determine if a dealership accepts cash payments it's best to call ahead or visit their website to confirm their payment policies. Additionally if a dealership does accept cash they may require payment in full for the purchase including any applicable taxes and fees. It's also essential to ensure that you have the exact amount of cash to avoid any issues or complications during the transaction process. Some car dealerships may not accept large sums of cash and this can cause problems when trying to purchase a car with a significant price tag. On the other hand some buyers may find it beneficial to negotiate for a better price when paying with cash as it's generally considered a more attractive offer than financing through a loan. Overall using cash to purchase a car can be a straightforward process but it's crucial to be prepared and informed about the dealership's policies to avoid any potential issues
GT Chunk	I have in the last few years purchased several used cars from dealers. They have handled it two different ways. They accepted a small check $1,000 now, and then gave me three business days to bring the rest as a cashiers check. They also insisted that I submit a application for credit, in case I needed a loan. They accepted a personal check on the spot. Ask them before you drive to the dealer. Of course they would love you to get a loan from them.

Table 11: Example of generated references for FiQA dataset. The example shows how Word2Passage generates different levels of references for a description query. The highlighted spans indicate words that appear in the GT chunk.

Dataset	NQ
Query Type	Person
Query	"who is next in line to be the monarch of england"
GT Answer	"Charles, Prince of Wales"
Word	Prince William Queen Elizabeth II Prince Charles Camilla Parker Bowles Prince George Catherine Princess of Wales monarch england
Sentence	Prince William the Prince of Wales is currently second in line to the throne and next in line to become the monarch of England
Passage	The line of succession to the British monarchy is determined by the Act of Settlement of 1701 and the Succession to the Crown Act of 2013. Currently Charles III is the monarch of England having ascended to the throne after the passing of his mother Queen Elizabeth II. The next in line to the throne is Prince William the Prince of Wales who is the eldest son of Charles III and is currently second in line to the throne. Following Prince William is Prince George of Cambridge who is the eldest child and son of Prince William and his wife Catherine the Princess of Wales. Prince Charles's marriage to Camilla Parker Bowles has resulted in his son's children taking precedence over Camilla's previous marriage and children meaning Prince Harry and his children will be fifth and sixth in line to the throne
GT Chunk	Catholics are eligible. Queen Elizabeth II is the sovereign, and her heir apparent is her eldest son, Charles, Prince of Wales. Next in line after him is Prince William, Duke of Cambridge, the Prince of Wales's elder son. Third in line is Prince George, the eldest child of the Duke of Cambridge, followed by his sister, Princess Charlotte and younger brother, Prince Louis. Sixth in line is Prince Harry, Duke of Sussex, the younger son of the Prince of Wales. Under the Perth Agreement, which came into effect in 2015, only the first six in line of succession require the

Table 12: Example of generated references for the NQ dataset. The example shows how Word2Passage generates different levels of references for a person query. The highlighted spans indicate words that appear in the GT chunk.

Dataset	NQ
Query Type **Query** **GT Answer**	Entity "what is the number 1 sport in the usa" "American football"
Word **Sentence** **Passage**	American football National Football League USA most popular sport country The National Football League (NFL) is widely considered the most popular sport in the USA with American football being the country's most widely followed and watched sport The number 1 sport in the USA is American football also known as the National Football League (NFL). It is a contact team sport played with an oval-shaped ball and its popularity in the country can be attributed to its rich history high-energy gameplay and the enthusiasm of its fan base. The NFL has been a favorite among Americans for decades with millions of people attending games and watching broadcasts. The Super Bowl the championship game of the NFL is one of the most-watched television events in the world attracting a massive audience and generating massive revenue. As a result American football has become an integral part of American culture and is often referred to as 'football' in the USA much to the confusion of people from other countries who refer to the sport as'soccer
GT Chunk	$69 billion, roughly 50% larger than that of all of Europe, the Middle East, and Africa combined. All four enjoy wide-ranging domestic media coverage and are considered the preeminent leagues in their respective sports in the world, although American football does not have a substantial following in other nations. Three of those leagues have teams that represent Canadian cities, and all four are the most financially lucrative sports leagues of their sport. Major League Soccer (MLS), which also includes teams based in Canada, is sometimes included in a ̈top fiveöf leagues. Professional teams in all major sports in the

Table 13: Example of generated references for NQ dataset. The example shows how Word2Passage generates different levels of references for an Entity query. The highlighted spans indicate words that appear in the GT chunk.

Dataset	SQuAD
Query Type **Query** **GT Answer**	Numeric "How much did Sequoia Capital invest in youtube in November of 2005?" "$3.5 million"
Word **Sentence** **Passage**	Sequoia Capital November 2005 invest $3.5 million YouTube In November 2005 Sequoia Capital invested $3.5 million in YouTube Sequoia Capital played a crucial role in YouTube's early days providing significant financial backing to help the video-sharing platform expand its operations. In November 2005 Sequoia Capital invested in YouTube marking a pivotal moment in the company's growth. This strategic investment enabled YouTube to solidify its position in the emerging online video market and attracted new users to the platform. With Sequoia Capital's support YouTube was able to increase its staff improve infrastructure and enhance its content offerings. This collaboration ultimately propelled YouTube towards its current status as a leading video-sharing platform
GT Chunk	2005, and can still be viewed on the site. YouTube offered the public a beta test of the site in May 2005. The first video to reach one million views was a Nike advertisement featuring Ronaldinho in November 2005. Following a $3.5 million investment from Sequoia Capital in November, the site launched officially on December 15, 2005, by which time the site was receiving 8 million views a day. The site grew rapidly and, in July 2006, the company announced that more than 65,000 new videos were being uploaded every day, and that the site was receiving 100 million video

Table 14: Example of generated references for SQuAD dataset. The example shows how Word2Passage generates different levels of references for a Numeric query. The highlighted spans indicate words that appear in the GT chunk.

Dataset	TriviaQA
Query Type	Location
Query	"Why use accounting software like Quickbooks instead of Excel spreadsheets?"
GT Answer	"Harlem, NY", "Harlem (New York, N.Y.)", "Demographics of Harlem", "Central Harlem", "New Harlem", "Black Harlem", "Vinegar Hill, Manhattan", "Harlem, New York City", "Harlem, Manhattan", "Harlem, New York", "Harlem (Manhattan)", "Harlem", "Harlem (New York City)"
Word	Harlem Manhattan Apollo Theater neighborhood New York
Sentence	The Apollo Theater is situated in the historic Harlem neighborhood of Manhattan a legendary hub of music arts and culture in New York City
Passage	The Apollo Theater is located in the Harlem neighborhood of Manhattan in New York City. It has a rich history of hosting numerous iconic performers and is a cultural gem of the area. Known for its incredible acoustics and historic grandeur the Apollo Theater stands as a symbol of the city's rich musical heritage. Whether you're a music lover or just interested in exploring the city's vibrant culture a visit to the Apollo Theater is an absolute must
GT Chunk	Apollo Theater The Apollo Theater is a music hall located at 253 West 125th Street between Adam Clayton Powell Jr. Boulevard (formerly Seventh Avenue) and Frederick Douglass Boulevard (formerly Eighth Avenue) in the Harlem neighborhood of Manhattan, New York City. It is a noted venue for African-American performers, and is the home of Šhowtime at the Apollo, a nationally syndicated television variety show which showcased new talent, from 1987 to 2008, encompassing 1,093 episodes; the show was rebooted in 2018. The theater, which has a capacity of 1,506, opened in 1914 as Hurtig & Seamon's New Burlesque Theater, and was

Table 15: Example of generated references for TriviaQA dataset. The example shows how Word2Passage generates different levels of references for a location query. The highlighted spans indicate words that appear in the GT chunk.