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

Jeonghwan Choi[1], **Minjeong Ban**[1], **Minseok Kim**[2], **Hwanjun Song**[1*]
[1]KAIST [2]Meta
{hwani.choi, songhwanjun}@kaist.ac.kr

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 is publicly available at https://github.com/DISL-Lab/Word2Passage

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 (see Appendix A). Therefore, low-importance or misleading words lead to reduced retrieval effectiveness. This highlights the need to properly adjust word importance in query expansion (Kim et al., 2023; Chen et al., 2024).

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 signifi-

[*]Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 8276–8296
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

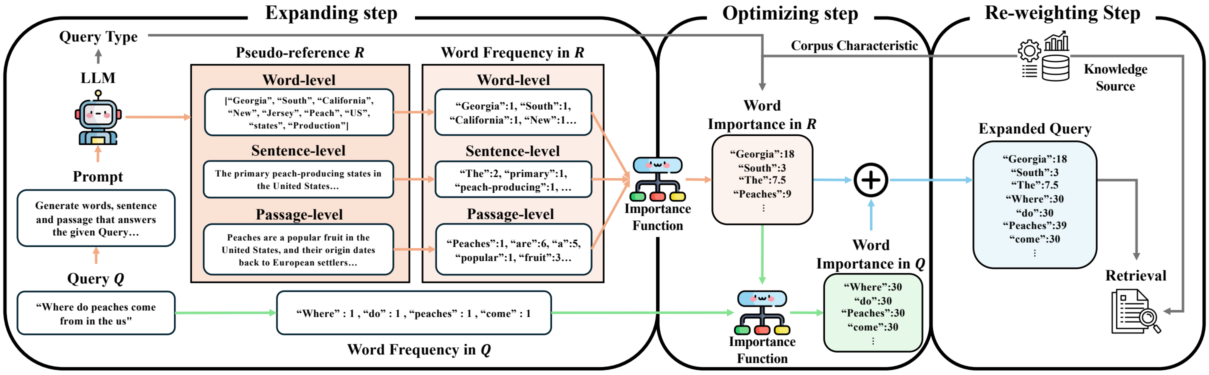
Expanding step
Query Type
LLM
Prompt
Generate words, sentence and passage that answers the given Query…
Query *Q*
"Where do peaches come from in the us"
Pseudo-reference *R*
Word-level
[“Georgia”, “South”, “California”, “New”, “Jersey”, “Peach”, “US”, “states”, “Production”]
Sentence-level
The primary peach-producing states in the United States…
Passage-level
Peaches are a popular fruit in the United States, and their origin dates back to European settlers…
Word Frequency in *R*
Word-level
“Georgia”:1, “South”:1, “California”:1, “New”:1…
Sentence-level
“The”:2, “primary”:1, “peach-producing”:1, …
Passage-level
“Peaches”:1, “are”:6, “a”:5, “popular”:1, “fruit”:3…
“Where” : 1 , “do” : 1 , “peaches” : 1 , “come” : 1
Word Frequency in *Q*
Importance Function
Optimizing step
Corpus Characteristic
Word Importance in *R*
“Georgia”:18
“South”:3
“The”:7.5
“Peaches”:9
⋮
Word Importance in *Q*
“Where”:30
“do”:30
“Peaches”:30
“come”:30
⋮
Importance Function
Re-weighting Step
Knowledge Source
Expanded Query
“Georgia”:18
“South”:3
“The”:7.5
“Where”:30
“do”:30
“Peaches”:39
“come”:30
⋮
Retrieval

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.

cance of each level, while also incorporating query type and domain characteristics to enhance word importance 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 LLM.
- **Optimizing Step**: This step estimates the importance 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 10). 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 LLM 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 reflect the varying contribution of multi-level references in importance re-weighting, effectively adapting the contribution of each reference level based on different query types 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 limitation, we propose Word2Passage, which enhances BM25 retrieval by enriching queries with semantically relevant words and re-weighting word importance.

Generation in RAG In RAG, the combination of IR and LLMs allows the system to leverage external knowledge for tasks like question-answering (QA), improving the quality of generated responses. RAG first retrieves relevant documents based on the query, then uses these documents as context for the LLM to generate appropriate answers. The effectiveness depends heavily on the LLM's ability to synthesize information from multiple sources while maintaining factual consistency. Several LLMs, including GPT-4 (Hurst et al., 2024), Llama (Grattafiori et al., 2024), and Qwen (Yang et al., 2024), have been widely adopted in RAG systems due to their strong generation capabilities and ability to handle long context windows. However, the generation quality is fundamentally constrained by quality of retrieved documents, making effective retrieval a critical prerequisite for successful RAG implementation.

Query Expansion Query expansion enhances retrieval results by reformulating the original query to include additional relevant words, addressing issues like vocabulary mismatch between queries and documents (Huang et al., 2021). Traditional methods, such as Pseudo-Relevance Feedback (PRF), assume that top-ranked documents from an initial retrieval are relevant and use words from these documents to expand the query (Lavrenko and Croft, 2001; Li et al., 2022). However, PRF can introduce noise if the initial retrieval includes irrelevant documents, highlighting the need for more advanced expansion methods such as HyDE, Query2doc, and MuGI.

Advancements in LLMs have introduced new avenues for query expansion (Kim et al., 2023; Gao et al., 2023a; Wang et al., 2023; Song and Zheng, 2024; Chen et al., 2024; 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). These approaches represent a shift towards utilizing LLMs for more effective query expansion.

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_t, 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, $\tilde{Q}$ is represented as a set of tuples (t, I_t), where each tuple contains a unique word t from the expanded query and its corresponding importance score I_t. Here, $\text{BM25}(t, \text{Chunk})$ represents the BM25 score of word t with respect to a given chunk. Based on this formulation, we identify two key factors that 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., 2023) 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 the importance score of each word I_t in both the original query Q and pseudo references R, which construct the expanded query $\tilde{Q}$. From the $\tilde{Q}$ perspective, we introduce multi-level references with a 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,

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.

while also separately inspecting importance based on whether the word is from the original query or the pseudo references. This enables precise word re-weighting, improving retrieval effectiveness.

The following sections detail the three components to implement WORD2PASSAGE.

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 (see Table 14 in Appendix F). 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:

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

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

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

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

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 importance within the pseudo references $\mathcal{R}$ and the original query Q. This approach ensures that expanded queries retain essential words from Q while incorporating relevant contextual words from $\mathcal{R}$. Note that Q and $\mathcal{R}$ 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 incorporate query type information into the computation of 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 G): *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:

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

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

and $0 \leq I_{q,w}, I_{q,s}, I_{q,p}$; and q represents the query type. Here, $F_{t,w_i}, F_{t,s_i}, F_{t,p_i}$ serve 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 query type identification, we prompt the same LLM (used for query expansion) using the prompt in Table 15 of Appendix F. The corresponding importance scores are then assigned to the reference levels of the word, sentence and passage, as determined by our empirical analysis in Appendix B and Appendix D. This design allows us to reflect the uneven importance 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 Expanding 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:

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

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:

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

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

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

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 $S(\tilde{Q}, d)$.

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 $D_{\tilde{Q}}$.

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, TREC-COVID(Covid), Touche-2020(Touche), SciFact, NFCorpus(NFC), Arguana(Arg), SCIDOCS(SCI), HotpotQA(Hotpot), NQ, and FiQA. For the QA task, we use 5 QA datasets based on the WikiCorpus. In particular, the QA datasets are categorized according to their reasoning complexity:

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

Among these, three datasets, *i.e.*, NQ, Hotpot, and FiQA, are used for QA while also included in the BEIR for IR evaluation. We randomly sampled 500 test examples from each of these datasets for 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 13 in Appendix F) (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 HyDE (Gao et al., 2023a), Query2doc (Q2D) (Wang et al., 2023), 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 GPT4o is accessed via its API. For the QA task, all answer generation is performed using Llama3.1-8B-Instruct. More details about model checkpoints are described in Table 12.

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 4.6).

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 10. The detailed analysis is presented in Appendix D.

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).

LLM	Method	IR											Avg.
		DL19	DL20	Covid	Touche	SciFact	NFC	Arg	SCI	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	42.3
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	40.9
	MuGI	66.5	61.1	73.1	49.6	72.2	**36.4**	29.4	**15.3**	65.2	50.2	24.3	49.4
	W2P	**68.1**	**62.5**	**78.4**	**50.7**	**72.4**	36.1	**32.5**	**15.3**	**71.9**	**50.4**	**26.1**	**51.3**
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	36.8
	MuGI	65.8	**62.9**	67.7	44.2	**71.6**	36.5	28.9	14.7	67.8	43.9	24.7	48.1
	W2P	**67.5**	62.8	**77.0**	**49.6**	71.5	**36.5**	**32.7**	**15.6**	**70.1**	**44.4**	**26.3**	**50.4**
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	41.6
	MuGI	69.4	62.7	70.3	47.3	**72.8**	36.4	28.3	15.2	72.2	49.7	25.5	50.0
	W2P	**69.7**	**64.1**	**75.6**	**48.1**	72.1	**36.9**	**33.3**	**15.5**	**73.6**	**50.3**	**26.6**	**51.4**

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.

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 weighting 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.

Specifically, W2P significantly outperforms other methods (*e.g.*, 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

Since HyDE, Q2D, and MuGI (Gao et al., 2023a; Wang et al., 2023; Zhang et al., 2024) 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. For more details on this analysis, see Section 4.7.

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

LLM	Method	IR					IR & QA			Avg.
		Covid	Touche	NFC	Arg	SCI	Hotpot	NQ	FiQA	
Llama3.1-8B-Inst.	W2P	78.4	50.7	36.1	32.5	15.3	71.9	50.4	26.1	45.2
	(-) Multi-level	73.4	50.2	35.8	31.5	15.4	71.9	49.5	25.2	44.1
	(-) Domain-aware	73.6	47.9	36.0	30.9	15.4	71.3	49.5	24.8	43.7
Qwen2.5-7B-Inst.	W2P	77.0	49.6	36.5	32.7	15.6	70.1	44.4	26.3	44.1
	(-) Multi-level	74.7	48.9	36.5	31.9	15.2	69.6	44.3	25.8	43.4
	(-) Domain-aware	74.2	46.6	36.5	31.4	15.3	69.9	44.1	25.6	43.0

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

Retrieval Method	Llama3.1-8B-Inst.	Qwen2.5-7B-Inst.
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.

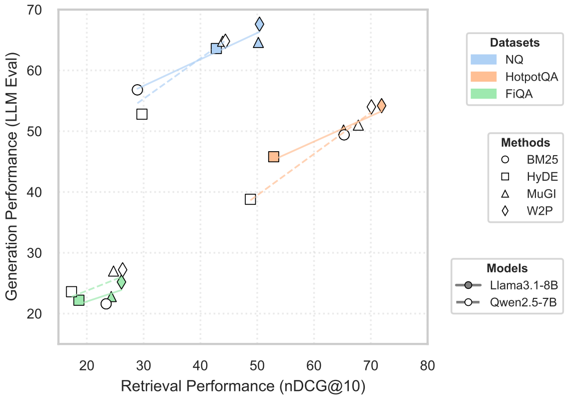

Figure 2: Analysis on the correlation between nDCG@10 in IR and LLM Eval in QA task across three datasets.

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 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 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.

4.7 Misalignment between IR and QA

To further understand the interaction between IR and QA, we analyze their performance relationship on three datasets (NQ, HotpotQA, and FiQA) that provide ground-truth chunk IDs and answers. Figure 2 shows the nDCG@10 and LLM Eval scores for IR and QA tasks, respectively. To understand the misaligned cases, Table 7 analyzes the ranking positions of ground-truth chunks in the NQ dataset.

Overall IR-QA Performance Correlation Figure 2 reveals a strong positive correlation (0.69–0.96) between IR and QA performance across the three datasets, confirming that **higher-quality retrieval generally leads to better answer generation in RAG**. However, as discussed in Section 4.3, this relationship does not always hold consistently.

Notably, Word2Passage consistently achieves the highest performance in both IR and QA tasks, positioning at the top-right of the correlation plots across all datasets. This demonstrates that Word2Passage's word-level re-weighting effec-

LLM	Method	IR											Avg.
		DL19	DL20	Covid	Touche	SciFact	NFC	Arg	SCI	Hotpot	NQ	FiQA	
-	Q	50.6	48.0	59.5	44.2	67.9	32.2	30.5	14.9	65.3	28.9	23.4	42.3
Llama3.1-8B-Inst.	R	59.7	55.0	71.6	42.2	71.2	35.1	31.1	14.7	68.9	48.0	20.6	47.1
	R_w	59.0	55.8	72.6	42.8	71.3	36.1	31.7	14.7	69.6	49.4	21.8	47.7
	W2P	**68.1**	**62.5**	**78.4**	**50.7**	**72.4**	36.1	**32.5**	**15.3**	**71.9**	**50.4**	**26.1**	**51.3**
Qwen2.5-7B-Inst.	R	57.5	55.5	68.8	40.7	71.2	34.7	31.9	14.8	68.1	40.7	22.7	46.1
	R_w	55.1	54.9	71.2	41.3	70.8	35.2	32.1	15.3	68.3	42.0	23.8	46.4
	W2P	**67.5**	**62.8**	**77.0**	**49.6**	**71.5**	**36.5**	**32.7**	**15.6**	**70.1**	**44.4**	**26.3**	**50.4**

Table 6: Comparing retrieval performance of different query configurations: original query (Q), pseudo reference (R), re-weighted reference (R_w), and Word2Passage (W2P). Performance measured using nDCG@10.

Rank Range/Metric	Methods			
	Default	HyDE	MuGI	W2P
Rank Position Distribution				
Ranks 1–3	161	235	273	284
Ranks 4–7	52	70	63	55
Ranks 8–10	30	16	28	27
Success cases	243	321	364	366
Performance Metrics				
nDCG@10 (IR)	28.9	42.8	50.2	50.4
LLM Eval (QA)	56.8	63.6	64.6	68.0

Table 7: Rank distribution and performance comparison across expansion methods on the NQ dataset. The table shows the position of ground-truth chunks in retrieval results and corresponding IR-QA performance metrics.

tively enhances both retrieval quality and downstream QA performance simultaneously.

Deep Analysis of IR-QA Misalignment Table 7 ana the number of ground-truth chunk positions in NQ dataset to explain the observed IR-QA misalignment. Despite nearly identical nDCG@10 scores (50.4 vs 50.2), WORD2PASSAGE significantly outperforms MuGI in LLM Eval (68.0 vs 64.6).

The critical difference lies in rank distribution. WORD2PASSAGE places more ground-truth chunks in top-3 positions (284 vs 273) and fewer in middle ranks (4–7). This positioning disparity contributes to the LLM Eval performance gap, reflecting the lost-in-the-middle phenomenon where LLMs exhibit reduced attention to middle-ranked passages during generation. Hence, **the precise positioning of relevant chunks matters due to the lost in the middle**, which causes misalignment between IR and QA performance.

4.8 Isolation Study of Query Components

Table 6 presents an isolation study examining the individual contributions of expanded query components in WORD2PASSAGE. We evaluate four configurations: original query (Q), pseudo reference (R), re-weighted reference (R_w), and the complete WORD2PASSAGE approach.

The pseudo reference alone consistently outperforms the original query across most datasets, demonstrating the effectiveness of LLM-generated references as standalone queries. Word-level importance re-weighting further enhances reference alone performance (*e.g.*, 71.6(R) vs 72.6(R_w) in Covid dataset), confirming that our re-weighting mechanism effectively emphasizes important words.

The most substantial gains emerge from combining the original query with the re-weighted reference in WORD2PASSAGE, which outperforms the original query by approximately 9 points and the re-weighted reference alone by nearly 4 points on average. This synergy is particularly pronounced on challenging datasets like DL19 and DL20, where W2P achieves 68.1 compared to Q achieving 50.6 and R achieving 59.7 individually.

These results reveal that **the original query and pseudo reference provide complementary information**. The original query preserves user intent while the re-weighted reference supplies contextual expansion, and their integration is essential for robust performance across diverse retrieval scenarios.

5 Conclusion

We introduce WORD2PASSAGE, a word-level reweighting 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. Our analysis of IR-QA performance alignment and component-wise evaluation further validate the effectiveness of WORD2PASSAGE's approach.

Limitation. 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.

Ethics 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 our experiments by using Llama-3.1-8B-Instruct, Qwen-2.5-7B/72B-Instruct. We utilized GPT-4o to evaluate our approach across models of different scales. The three open-source models were loaded from their Hugging Face checkpoints, while GPT-4o was accessed via the OpenAI API. All prompts are listed in Appendix F, and additional implementation details are summarized in Table 12.

Acknowledgements. KAIST was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea goverment (MSIT) (No. RS-2024-00445087, Enhancing AI Model Reliability Through Domain-Specific Automated Value Alignment Assessment). Additionally, KAIST was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00334343).

References

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, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv 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*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *SIGIR*.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *EACL*.

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*.

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*.

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*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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*.

Method	MuGI	W2P
Query	*What cuisine is described as a cultural blending of Mediterranean influences (such as those created by Italian and Spanish populations)?*	
Gold Chunk	**Document ID**: 2222 **Title**: *Argentine cuisine* **Text**: "Argentine **cuisine** is **described** as a **cultural blending** of **Mediterranean influences (such** as those **created** by **Italian** and **Spanish** populations) with and very small inflows (mainly in border areas), Indigenous, within the wide scope of agricultural products that are abundant in the country. Argentine annual consumption of beef has averaged 100 kg (220 lbs) per capita, approaching 180 kg (396 lbs) per capita during the 19th century; consumption averaged 67.7 kg (149 lbs) in 2007. Beyond "asado" (the Argentine barbecue), no other dish more genuinely matches the national identity. Nevertheless, the country's vast area, and its **cultural** diversity, have led to a **local cuisine** of various dishes. **The** great immigratory waves consequently imprinted a large **influence** in the Argentine **cuisine**, after all Argentina was the second country in the world with the most immigrants with 6.6 million, only second to the United States with 27 million, and ahead of other immigratory receptor countries such as Canada, Brazil, Australia, etc."	**Document ID**: 2222 **Title**: *Argentine cuisine* **Text**: "Argentine **cuisine** is **described** as a **cultural blending** of **Mediterranean influences (such** as those **created** by **Italian** and **Spanish** populations) with and very small inflows (mainly in border areas), Indigenous, within the **wide** scope of agricultural products that are abundant in the country. Argentine annual consumption of beef has averaged 100 kg (220 lbs) per capita, approaching 180 kg (396 lbs) per capita during the 19th century; consumption averaged 67.7 kg (149 lbs) in 2007. Beyond "asado" (the Argentine barbecue), no other dish more genuinely matches the national **identity.** Nevertheless, the **country's** vast area, and its **cultural** diversity, have led to a local **cuisine** of **various** dishes. **The** great immigratory waves consequently imprinted a large influence in the Argentine **cuisine**, after all Argentina was the second country in the **world** with the most immigrants with 6.6 million, only second to the United States with 27 million, and ahead of other immigratory receptor countries such as Canada, Brazil, Australia, etc."
Word Weights	'**Mediterranean**': 3.944, '**cuisine**': 3.662, '**Italian**': 3.662, '**blending**': 3.380, '**Spanish**': 3.380, '**cultural**': 3.099, '**The**': 2.535, '**influences**': 2.254, 'This': 1.690, 'What': 1.408, '**described**': 1.408, '**(such**': 1.408, '**created**': 1.408, 'populations)?': 1.408, 'use': 1.408, 'olive': 1.408, 'oil,': 1.408, 'region': 1.127, 'unique': 1.127, 'garlic,': 1.127, 'flavors': 1.127, 'culinary': 1.127, 'Provençal': 1.127, '**dishes**': 1.127, 'rich': 1.127, ⋮ 'aromas': 0.282, 'hearty': 0.282, 'ingredients': 0.282, 'countryside.': 0.282, 'herbs': 0.282, 'thyme': 0.282, 'rosemary,': 0.282, 'features': 0.282, 'like': 0.282, 'tradition': 0.282, 'culture': 0.282, 'Cuisine': 0.282, 'populations': 0.282, 'characteristic': 0.282, 'specifically': 0.282, 'Provence-Alpes-Côte': 0.282, 'dÁzur': 0.282, 'France.': 0.282, 'simplicity': 0.282, 'incorporating': 0.282, '**local**': 0.282, 'produce.': 0.282, 'results': 0.282, 'rustic': 0.282, 'refined,': 0.282	'**Mediterranean**': 6.660, '**cuisine**': 5.184, '**cultural**': 4.199, '**Spanish**': 4.199, '**Italian**': 3.346, '**blending**': 3.018, '**influences**': 2.198, '**created**': 2.034, '**described**': 1.870, 'olive': 1.640, 'culinary': 1.476, 'What': 1.378, '**(such**': 1.378, 'populations)?': 1.378, 'influences,': 1.312, 'oil,': 1.148, '**The**': 0.984, 'characterized': 0.984, 'use': 0.984, 'garlic,': 0.984, 'traditions': 0.984, 'dishes': 0.984, 'including': 0.984, 'populations,': 0.82, 'often': 0.82, ⋮ 'profile': 0.164, 'seafood': 0.164, 'tomatoes': 0.164, 'history': 0.164, 'distinctive': 0.164, 'emerged': 0.164, 'result': 0.164, 'Its': 0.164, 'profiles': 0.164, 'techniques': 0.164, 'history,': 0.164, 'popular': 0.164, 'featuring': 0.164, 'tomatoes.': 0.164, 'cuisine's': 0.164, 'testament': 0.164, 'Catalonia,': 0.164, 'situated': 0.164, 'nexus': 0.164, 'Mediterranean,': 0.164, 'trade': 0.164, 'exchange.': 0.164, 'resulted': 0.164, 'exotic': 0.164, 'inviting.': 0.164

Table 8: Comparison of MuGI and WORD2PASSAGE (W2P) expansions for a HotpotQA query. Boldface words in each expansion also appear in the gold chunk; weights are relative proportions (%).

A Case Study on Word Importance

To examine word importance differences between the two methods, we compare the frequency-based expansion of MuGI against our word-level reweighting approach WORD2PASSAGE using a HotpotQA example. Table 8 presents the words in expanded query with their proportional weights (percentage of total weight), demonstrating the difference in how each method weights important and common words.

Our analysis reveals three key findings. First, WORD2PASSAGE assigns substantially higher weights to important words present in the ground-truth chunk, such as *Mediterranean* (3.94% vs 6.66%) and *cultural* (3.10% vs 4.20%). Second, common words with lower relevance receive appropriately diminished emphasis in our approach, as demonstrated by the reduced weighting of words like *use* (1.41% vs 0.98%). Finally, WORD2PASSAGE assigns lower weights to irrelevant words (0.282% vs 0.164%), thereby reducing query drift from ubiquitous but irrelevant words compared to MuGI's frequency-based approach.

B Query Type Analysis for Reference Level Significance

Different query types benefit from different reference levels. We determine optimal weighting

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 9: Average relative ratio of significance scores for word-level, sentence-level, and passage-level references across all datasets.

schemes through grid search across five MS MARCO query categories (*e.g.*, description, entity, person, numeric, and location). In this section, we analyze the characteristics of each query type and their correspondence with obtained optimal reference level significance.

In Table 9, our analysis reveals distinct patterns in optimal reference level usage across query types. Description queries seek comprehensive explanations and benefit most from passage-level expansion. Entity queries target specific named entities typically well-defined within single sentences, making sentence-level expansion most effective. Person and location queries retrieve information where key details are captured effectively through balanced word- and sentence-level expansion. Numeric queries focus on numerical values requiring surrounding context for interpretation, making sentence-level expansion optimal.

The results confirm that optimal weights align with query characteristics:

- **Description**: passage-level dominant (0.43).
- **Entity**: sentence-level optimal (0.41).
- **Person/Location**: balanced word- and sentence-level (0.38 each).
- **Numeric**: sentence-level preferred (0.40).

These results validate that our multi-level reference approach effectively adapts to the distinct characteristics of different query types.

C Domain-aware Adjustments

In this section, we analyze the differences in corpus characteristics across datasets and examine how these influence our domain-aware weighting factor. We first analyze word distribution patterns across different corpora. Then, we investigate the sensitivity of retrieval performance to the balance between reference and query importance, demonstrating how the domain-aware factor W adapts to corpus complexity.

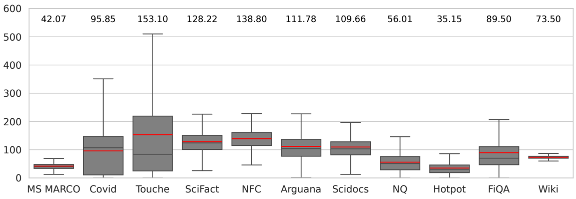

Figure 3: Distribution of unique words per chunk across different datasets. The y-axis shows the average number of unique words per chunk, with averages displayed above each box.

C.1 Characteristics of each corpus

Figure 3 shows the distribution of unique words per chunk across different datasets through box plots. The red line indicates the average number of unique words per chunk, and the box represents the interquartile range (IQR).

The datasets exhibit three distinct patterns in word distribution. Short-length datasets include MS MARCO (42.07), Hotpot (35.15), and NQ (56.01), showing consistent chunk sizes with small IQRs. Medium-length datasets comprise Wiki (73.50), FiQA (89.50), and Covid (95.85), with Covid displaying notably larger variance through its extended box and whiskers. High-legnth datasets include academic corpora such as Scidocs (109.66), Arguana (111.78), SciFact (128.22), NFC (138.80), and Touche (153.10), with Touche showing the largest variance overall. This analysis reveals significant differences in corpus characteristics across datasets, highlighting the need for domain-aware adjustments in our word-level re-weighting approach.

C.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):

$$\begin{aligned} I_t &= I_{t,R} + I_{t,Q} \\ &= \frac{\alpha}{\sqrt{W}} \sum_i I_{t,r_i} + \frac{\sum_{t' \in \mathcal{R}} F_{t',\mathcal{R}}}{\sum_{t' \in Q} F_{t',Q}} \cdot F_{t,Q} \\ &\text{where } \; I_{t,Q} = 0 \; \text{ if } \; t \notin Q, \\ &\qquad\quad\; I_{t,R} = 0 \; \text{ if } \; t \notin R, \\ &\qquad\quad\; {}^{\forall} t \in R \cup Q. \end{aligned} \tag{11}$$

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 $I_{t,Q}$. Specifically, 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.

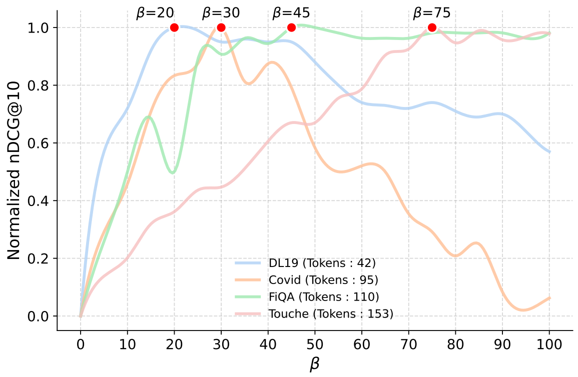

Figure 4: Relationship between β and retrieval performance. The x-axis represents β values, and the y-axis shows normalized nDCG@10 scores (scaled to [0,1] for each dataset). The plot demonstrates how optimal β values increase with the average number of unique words per chunk.

As shown in the figure 4, datasets with higher average numbers of unique words 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 words per chunk (larger W), the term $\frac{\alpha}{\sqrt{W}} \sum_i I_{t,r_i}$ decreases, effectively reducing the influence of reference words. This confirms that our domain-aware factor W appropriately adapts to varying corpus characteristics.

D 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 10.

D.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 corpus characteristics. Specifically, we formed four groups sharing similar corpus properties:

- **MS MARCO group (DL19, DL20)**: Web documents with diverse topics and general domain knowledge.

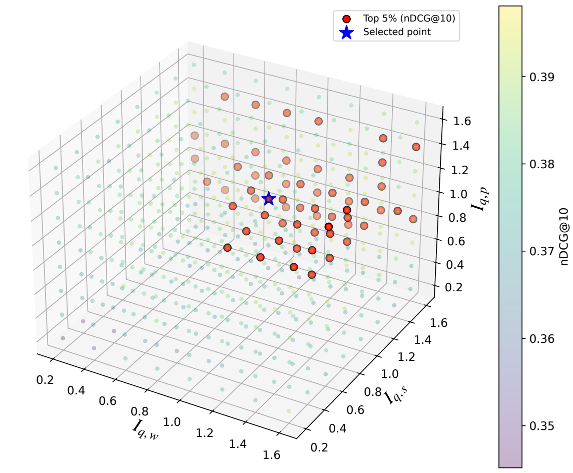

Figure 5: Distribution of nDCG@10 scores on the training set across different combinations of word, sentence, and passage significance scores for person query type in DL19 dataset.

- **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.

D.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.

D.3 Analysis of Score Distribution

Figure 5 illustrates the distribution of nDCG@10 scores on the training set across different combina-

Query Type	DL19-20	Covid	NFC	Touche	SciFact	Arg
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)	(1.0, 1.0, 1.0)	(1.0, 1.0, 1.0)
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)	(1.0, 1.0, 1.0)	(1.0, 1.0, 1.0)	(1.0, 1.0, 1.0)	(1.0, 1.0, 1.0)

Query Type	SCI	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	(1.0, 1.0, 1.0)	(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	(1.0, 1.0, 1.0)	(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 10: Significance scores ($I_{q,w}$, $I_{q,s}$, $I_{q,p}$) for word, sentence, and passage-level references generated by Llama3.1-8B-Instruct, obtained by grid search on a balanced subset of 500 queries from the training set. Note that (1.0, 1.0, 1.0) is assigned for query types absent in the training set to ensure stability.

# of Gen.	IR								Avg.	Inf. Time (s)
	DL19	DL20	Covid	Touche	SciFact	NFC	Arg	SCI		
1	63.3	57.6	73.1	45.2	70.3	35.6	27.4	14.4	48.4	0.89
2	66.6	60.5	75.5	48.4	71.3	**36.2**	31.7	14.8	50.6	1.79
3	68.1	60.6	76.2	49.4	**72.4**	36.1	32.3	15.2	51.3	2.69
4	**68.5**	62.0	77.5	49.6	72.1	36.1	32.4	**15.4**	51.8	3.59
5	68.1	**62.6**	**78.4**	**50.7**	**72.4**	36.1	**32.5**	15.3	**52.1**	4.49

Table 11: Retrieval performance with varying numbers of pseudo-references generated by Llama3.1-8B-Instruct. Inference time increases linearly with the number of generated references.

tions 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.

E Number of references

To investigate the optimal number of references for balancing retrieval effectiveness and computational efficiency, we conducted an ablation study examining how performance changes with varying numbers of generated pseudo-references. Our findings indicate that generating three references can achieve near-optimal performance while substantially reducing computational overhead.

Table 11 demonstrates the relationship between the number of references and retrieval effectiveness across different datasets. Performance consistently improves as the number of references increases from one to five, with average nDCG@10 scores rising from 48.4 to 52.1. However, the rate of improvement diminishes significantly after three references. The performance gain from one to three references is substantial (48.4 vs 51.3), while the improvement from three to five references is marginal (51.3 vs 52.1). This pattern holds across most individual datasets: for instance, on Covid, the gain from one to three references is +3.1 points (73.1 vs 76.2), compared to only +2.2 points from three to five references (76.2 vs 78.4).

The diminishing returns become particularly evident when considering computational cost. While using five references provides the best performance on most datasets, the marginal gains (typically less than 1 point in nDCG@10) may not justify the 67% increase in computational overhead compared to using three references. Based on these results, we recommend using three sets of references for practical applications, as this represents the optimal trade-off between retrieval accuracy and efficiency.

F Prompts

This section presents the prompts used in our experiments. The prompt used for LLM-based evaluation is provided in Table 13. Table 14 shows the prompts used for generating word, sentence, and passage-

Model	Checkpoints	Implementation Details	Precision
GPT-4o	gpt-4o-2024-08-06	API (OpenAI)	Default
Llama3.1-8B-Inst.	meta-llama/Llama-3.1-8B-Instruct	1 × NVIDIA L40S 48GB	BF16
Qwen2.5-7B-Inst.	Qwen/Qwen2.5-7B-Instruct	1 × NVIDIA L40S 48GB	BF16
Qwen2.5-72B-Inst.	Qwen/Qwen2.5-72B-Instruct	4 × NVIDIA L40S 48GB	BF16

Table 12: Details of the checkpoint and GPUs for implementation.

level references in WORD2PASSAGE. For query type classification, we use the prompt in Table 15.

G 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 16-20 present examples for five different query types: description, person, entity, numeric, and location.

For each query type, word-level references extract key words 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.

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

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 14: 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 15: Prompt of query type classification

Dataset	TREC-COVID
Query Type	Description
Query	"How does the coronavirus respond to changes in the weather?"
GT Answer	-
Word	**Coronavirus** Transmission Weather **Temperature Humidity Covid-19** Transmission Infectivity Respiratory Droplets Ventilation Air circulation Hygiene Contagious
Sentence	Research suggests that warmer temperatures, typically above 60°F (15.5°C), **and** more humid climates may reduce **the spread and** transmission **of Covid-19**, while colder temperatures below freezing may hinder **the spread of the** virus.
Passage	**The coronavirus**, specifically **Covid-19**, **has** shown mixed responses **to** changes **in the** weather. Research suggests that **the spread and** transmission **of the** virus may slow down **in** warmer **and** more humid climates, whereas colder temperatures may facilitate **the spread of the** virus. **For** instance, studies have found that **the** virus thrives **in** temperatures between 40°F **and** 80°F (4°C **and** 27°C), whereas low temperatures below freezing can reduce **the** infectivity **of the** virus. Furthermore, research suggests that **the** stability **of the** virus is affected by **temperature and humidity**, which can influence **the** effectiveness **of** outdoor respiratory droplets. While **the** exact relationship between weather **and coronavirus** is not fully understood, it is believed that warmer temperatures may reduce **the** transmission **of the** virus. However, it's essential **to** note that other factors such as ventilation, air circulation, **and** personal hygiene also play crucial roles **in** preventing **the spread of the** virus. Consequently, relying solely **on temperature to** mitigate **the spread of the coronavirus** is not a reliable approach **and** requires a multi-faceted approach **to** curb **the spread of the** disease. **In** general, governments **and** health authorities should consider various **environmental** factors when developing strategies **to** contain outbreaks **and** prevent **the** transmission **of the** virus.
GT Chunk	To date, the world perhaps has never waited for the summer so impatiently in the entire Anthropocene, owing to the debate whether increasing temperature and humidity will decrease the environmental endurance of SARS-CoV-2. We present the perspective on the seasonal change on SARS-CoV-2 decay and Covid-19 spread. Our arguments are based on: i) structural similarity of coronavirus with several enteric viruses, and its vulnerability; ii) reports related to decay of those similar transmissible gastroenteritis viruses (TGEV) like norovirus and iii) improvement in the human immunity during summer with respect to winter. We present reasons why we can be optimistic about the slowdown of corona in the upcoming summer.

Table 16: Example of generated references for TREC-COVID 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 17: 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	Entity
Query	"what is the number 1 sport in the usa"
GT Answer	"American football"
Word	**American football** National Football **League** USA **most** popular sport country
Sentence	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
Passage	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 ẗop fiveöf leagues. Professional teams in all major sports in the

Table 18: 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	Numeric
Query	"How much did Sequoia Capital invest in youtube in November of 2005?"
GT Answer	"$3.5 million"
Word	**Sequoia Capital November 2005** invest **$3.5 million YouTube**
Sentence	In **November 2005 Sequoia Capital** invested **$3.5 million in YouTube**
Passage	**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 19: 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 S̈howtime 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 20: 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.