Incorporating Discarded Candidate Tokens of Large Language Models for Efficient Query Expansion

Anonymous ACL submission

Abstract

In the era of Large Language Models (LLMs), efficient retrieval is crucial for integration with modern retrieval-augmented LLM systems, making sparse retrieval modules a popular choice due to their efficiency and robustness in low-resource settings. To enhance sparse retrieval performance, LLM-based Query Expansion (QE) has emerged as a solution to bridge the lexical gap between queries and documents. However, existing QE methods face a fundamental trade-off between efficiency and effectiveness, driven by the length of generated tokens. To address this, we propose Discarded candidate Tokens Query Expansion (DTQE), a novel query expansion method that leverages conventionally unselected tokens from the LLM's decoding process by indexing them separately. Experimental results demonstrate that DTQE maintains high efficiency compared to more resource-intensive baselines while significantly outperforming keyword-based expansion ones.

1 Introduction

Information Retrieval (IR) aims to retrieve relevant documents in response to queries from a large corpus. In recent years, Dense retrievers (Xiong et al., 2021; Izacard et al., 2022) using semantic embeddings excel with substantial labeled training data (Karpukhin et al., 2020). However, lexical-based sparse approaches (Robertson et al., 1994) offer key advantages: faster retrieval, efficient memory usage, and competitive performance on low-resource datasets (Arabzadeh et al., 2021; Luo et al., 2023; Thakur et al., 2021). Nevertheless, a key limitation of sparse retrievers is their reliance on exact term matching, which makes them susceptible to lexical mismatch (Nogueira et al., 2019).

Query Expansion (QE) is a widely used technique for improving sparse retrieval performance by adding related terms to the original query (Robertson, 1991; Amati and Van Rijsbergen, 2002). Recently, Large Language Models (LLMs) (Brown et al., 2020; Touvron et al., 2023) have been leveraged for QE by generating pseudo-documents to enhance retrieval effectiveness (Gao et al., 2023; Weller et al., 2024). While generating long pseudo-documents can improve retrieval performance, it significantly increases the overall latency due to the computational overhead of document generation (Wang et al., 2023). An alternative approach utilizes LLMs for *keyword-based expansions*, where only the most relevant, short keywords are generated (Jagerman et al., 2023; Mackie et al., 2023; Li et al., 2024). This method conserves tokens, enabling faster query expansion; however, the reduced number of tokens results in limited performance gain. Consequently, LLM-based QE exhibits *a clear trade-off between performance and latency* dictated by the length of generated tokens.

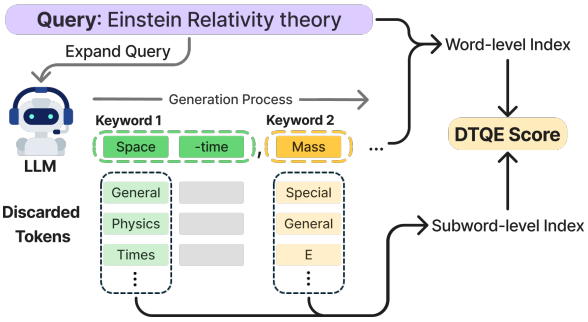

Figure 1: An illustration of our DTQE method.

To overcome the trade-off between latency and performance, we propose Discarded candidate Tokens Query Expansion (DTQE). Our method extends keyword-based expansion by leveraging both the *selected* keywords and the *discarded candidate tokens* generated by the LLM at each step of QE. Although a single token is ultimately chosen at each step, the ranked lists of unselected tokens contain semantically related terms to the original query. These related terms can significantly enhance the retrieval performance. By filtering and incorporating these unselected tokens, our method captures additional lexical variety without increasing the

LLM's final output length.

Building on this insight, we design our retrieval framework as illustrated in Figure 1. Our approach begins with prompting the LLM to generate a fixed number of semantically relevant keywords for the original query. We extract both the final keywords and potential candidate tokens from the LLM's generation step. Our two-stage retrieval process first retrieves documents using concatenated text of the original query and selected keywords on a standard word-level inverted index. The second stage employs the extracted candidate tokens on a specialized tokenizer-based inverted index. We then enhance retrieval performance by interpolating the scores from the candidate tokens with those from existing query expansion approaches.

Our experimental evaluation encompasses two web search datasets and eight low-resource datasets. The results demonstrate that DTQE achieves competitive retrieval performance compared to pseudo-document-based query expansion techniques, while utilizing significantly fewer tokens. Furthermore, comprehensive analyses across varying keyword counts and different LLM sizes validate the robustness of DTQE. These findings suggest promising applications for scalable and efficient LLM-driven query expansion in real-world search systems.

2 Related Work

Query expansion is a long-standing strategy in IR that aims to mitigate the lexical gap between user queries and relevant documents. Traditional approaches often utilize relevance feedback signals—either explicit (Lavrenko and Croft, 2001) or pseudo-relevant (Robertson, 1991; Lv and Zhai, 2009). However, they still struggle with word-sense disambiguation and domain-specific terminology (Jeong et al., 2024).

More recently, LLMs have emerged as powerful tools for query expansion (Mao et al., 2021; Chuang et al., 2023; Gao et al., 2023; Li et al., 2024), by generating pseudo documents to enrich contextual cues in zero-shot or limited-labeled scenarios. Building on this, Jagerman et al. (2023) proposed incorporating PRF-derived documents into the LLM's input, thereby creating context-aware pseudo-documents. While these approaches substantially enhance retrieval effectiveness, they also introduce considerable latency (Wang et al., 2023). One way to reduce inference time is to restrict the LLM to producing only a set of relevant keywords (Mackie et al., 2023), but this typically leads to lower performance due to reduced contextual information. We address this trade-off by exploiting candidate tokens that are normally discarded during LLM generation, offering improved retrieval performance without incurring the additional latency cost.

3 Methodology

In this section, we describe our approach for expanding a user's query $\boldsymbol{q}$ with both (1) a set of semantically relevant keywords generated by the LLM and (2) discarded candidate tokens that were not selected as part of the final output.

3.1 Preliminary: Keyword Generation for Query Expansion

Following Jagerman et al. (2023), we first generate a set of keywords, $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_i\}$, to expand the original query $\boldsymbol{q}$. Each keyword consists of multiple tokens, represented as $\boldsymbol{w}_i = w_{i,1}||\ldots||w_{i,j}$, where $w_{i,j}$ denotes the j-th token in keyword $\boldsymbol{w}_i$. An LLM generates the keywords based on either (1) the query $\boldsymbol{q}$ alone or (2) pseudo-relevance feedback (PRF), where the LLM incorporates retrieved results from the original query. Further details on keyword generation are provided in Appendix B.

However, the generated keywords are limited by the small number of generated tokens, potentially omitting relevant terms. To address this issue, we aim to utilize a set of *discarded candidate tokens* from the keyword generation process.

3.2 Discarded Candidate Token Query Expansion (DTQE)

We present DTQE, a simple yet effective method for enhancing retrieval performance by incorporating discarded candidate tokens into query expansion. For each token position j within a keyword $\boldsymbol{w}_i$, the LLM considers multiple candidates before selecting the final token $w_{i,j}$. We define the set of discarded candidates as:

$$\mathcal{W}'_{i,j} = \{w'_{i,j,1}, w'_{i,j,2}, \ldots, w'_{i,j,k}\}$$

where $\mathcal{W}'_{i,j}$ represents the set of alternative tokens discarded for the j-th position in keyword $\boldsymbol{w}_i$, and k is the number of discarded candidate tokens at that position. To maximize the utility of $\mathcal{W}'_{i,j}$, we perform three steps: (1) filtering discarded tokens, (2) computing relevance scores for discarded tokens, and (3) interpolating these scores with BM25 scores for keyword-expanded queries.

	High Resource			Low Resource									
Method	**DL19**	**DL20**	**Tokens Avg**	**Covid**	**NFCorpus**	**Scifact**	**DBPedia**	**FiQA**	**Arguana**	**News**	**Robust04**	**BEIR Avg**	**Tokens Avg**
w/o relevance judgement													
BM25	49.7	48.8	–	59.5	32.2	67.9	31.8	23.6	39.7	39.5	40.7	41.9	–
BM25 + RM3	51.5	49.2	–	59.3	34.6	64.6	30.8	19.2	38.0	42.6	42.6	41.4	–
Contriever	44.5	42.1	–	27.3	31.7	64.9	29.2	24.5	37.9	34.8	31.6	35.2	–
Contriever + Hyde	61.3	57.9	–	59.3	–	69.1	36.8	27.3	**46.6**	44.0	–	–	–
BM25 + Q2D	**68.4**	**63.5**	99.5	70.0	35.8	70.4	**39.6**	25.8	40.0	46.9	**49.4**	47.2	100.0
BM25 + Q2D/PRF	65.0	61.8	86.3	72.2	37.2	71.0	36.1	**27.4**	40.2	47.0	47.8	47.4	96.6
Promptreps (Llama3-70B-I)	–	–	–	63.0	29.7	61.5	28.3	22.2	24.7	–	–	–	–
BM25 + Q2K (5)	60.1	55.5	15.5	65.3	36.0	69.7	36.0	24.8	40.3	44.9	46.8	45.5	15.7
w/ DTQE (Ours)	62.9	56.9	15.5	70.8	37.2	70.8	37.0	26.4	40.8	47.6	49.3	47.5	15.7
BM25 + Q2K/PRF (5)	59.4	56.9	15.3	70.6	36.7	70.3	36.4	25.2	40.0	45.9	46.3	46.4	17.3
w/ DTQE (Ours)	62.4	57.6	15.3	**73.0**	**37.9**	**71.3**	37.7	26.0	40.8	**48.6**	48.8	**48.0**	17.3
w/ relevance judgement													
DPR	62.2	**65.3**	–	33.2	18.9	31.8	26.3	29.5	17.5	16.1	25.2	24.8	–
ANCE	**64.5**	64.6	–	**65.4**	23.7	50.7	28.1	30.0	41.5	38.2	39.2	39.6	–
Contriever-FT	62.1	63.2	–	59.6	**32.8**	**67.7**	**41.3**	**32.9**	**44.6**	**42.8**	**47.3**	**46.1**	–

Table 1: NDCG@10 on TREC and 8 low resource datasets from BEIR. The first or second highest performances in each category (w/o and w/ relevance judgment) are highlighted in **bold** or underlined.

Token filtering. To enhance lexical diversity while minimizing redundancy, we first remove duplicate tokens from $\mathcal{W}'_{i,j}$. Since discarded tokens in $\mathcal{W}'_{i,j}$ for $j > 1$ are highly dependent on the keyword's first token $w_{i,1}$, we retain only discarded candidates from $j = 1$ to maximize semantic diversity. Finally, we discard tokens shorter than two characters to prevent trivial expansions.

Relevance Score for Discarded Tokens. To quantify the impact of discarded tokens on the retrieval process, we compute their relevance scores based on how frequently they appear in the document collection. Because many of these subword-based tokens do not appear in the standard word-level BM25 vocabulary, we build an additional index by processing documents through a subword-based tokenizer. This lets us assess the importance of discarded tokens in much the same way as BM25.

Score Interpolation. To integrate scores from keyword-expanded queries and discarded tokens, we define the final document relevance score as:

$$S_{\text{combined}}(d) = \alpha \cdot \tilde{S}_{\text{W}}(d) + (1 - \alpha) \cdot S_{\text{T}}(d),$$

where $\tilde{S}_{\text{W}}(d)$ is the normalized BM25 score for keyword-expanded queries, $S_{\text{T}}(d)$ is the relevance score for the filtered discarded tokens, and α is a hyperparameter controlling their relative contribution. We describe more details of our proposed method, DTQE, in Appendix A.

4 Experiments

4.1 Setup

Implementation Details. We employ GPT-4o (OpenAI, 2024) for keyword generation. In our main experiment, we generated a total of 5 keywords for each query. Following Jagerman et al. (2023), we repeat the original query terms five times and concatenate keywords for keyword-expanded queries. When utilizing discarded tokens, we use the top-20 candidate tokens for each keyword, which are ranked by logprobs. Also, we build the word-level inverted index with Pyserini (Lin et al., 2021) for expanded query and subword–level inverted index with Python's BM25S library (Lù, 2024) for discarded tokens. We set $\alpha = 0.9$ for score interpolation.

Dataset. We evaluate on two web search datasets from TREC-DL (Craswell et al., 2020, 2021) and 8 low-resource retrieval datasets from BEIR (Thakur et al., 2021) covering a variety of domains. We report nDCG@10 scores, the commonly employed evaluation measure for these datasets.

Baseline. We compare our approach against unsupervised retrievers, including (1) **BM25**, (2) **Contriever** (Izacard et al., 2022), and (3) **BM25 + RM3**, using Pyserini's default implementation. For LLM-based query expansion, we evaluate (1) **Contriever + HyDE** (Gao et al., 2023), which enhances Contriever with hypothetical documents generated by an LLM; (2) **BM25 + Q2D** (Jagerman et al., 2023), which generates hypothetical answer documents via a task-agnostic zero-shot prompt; (3) **BM25 + Q2K**, which expands queries with generated keywords instead of full documents; (4) **BM25 + Q2D/PRF** and (5) **BM25 + Q2K/PRF**, which refine Q2D and Q2K using initial retrieval results. We also evaluate (6) **PromptReps** (Zhuang et al., 2024), a sparse retrieval approach leveraging LLM-generated text representations for token-based matching. Additionally, we include three supervised dense retriev-

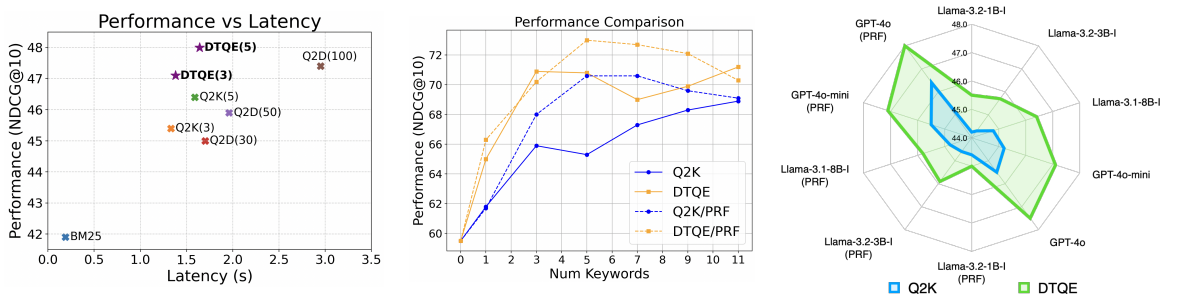

Figure 2: (Left). Retrieval Performance and Latency for different QE methods, averaged over 8 low-resource datasets. The number in parentheses indicates the keyword count for Q2K and DTQE and the maximum token count for Q2D. (Center). Performance on TREC-COVID using a different number of keywords. (Right). Retrieval performance for different LLMs, averaged over 8 low-resource datasets. We exploit the models from Llama-3 (Dubey et al., 2024) and GPT-4o family

ers trained on MS-MARCO: (1) **DPR** (Karpukhin et al., 2020), (2) **ANCE** (Xiong et al., 2021), and (3) **Contriever-FT** (Izacard et al., 2022). Further details of set up are in Appendix B

4.2 Results

Main Results. The results on TREC-DL and BEIR are shown in Table 1. DTQE consistently outperforms Q2K with or without PRF across all datasets, confirming the effectiveness of incorporating discarded tokens into query expansion. Specifically, our method is highly effective on low-resource datasets, even outperforming Q2D despite using fewer tokens. This result demonstrates that the discarded tokens in the low-resource domain often contain specialized terminologies critical to retrieval performance. While Q2D performs better on high-resource datasets, our method achieves comparable results while significantly reducing the token budget, leading to much faster execution. Our method's efficiency advantage makes it a compelling choice, especially in scenarios where computational cost is a critical concern. Finally, DTQE surpasses the Contriever-FT, which requires additional significant fine-tuning cost, by a notable margin, underscoring its robust zero-shot capabilities.

Performance vs Latency. We analyze the end-to-end latency, including expansion and retrieval time, as well as retrieval performance for query expansion in Q2D, Q2K, and our method. As shown on the left side of Figure 2, Q2D requires significant time for document retrieval due to the overhead of pseudo-document generation, whereas Q2K effectively reduces latency but achieves a smaller performance gain. In contrast, our method strikes a balance between latency and performance, positioning itself on the Pareto front. This result highlights the importance of considering both effectiveness and efficiency in LLM-based query expansion, and our method successfully achieves both objectives.

Impact of number of Keywords. We examine how the number of keywords affects the performance of both Q2K and DTQE. As shown in the center of Figure 2, DTQE consistently outperforms Q2K, regardless of PRF usage or keyword count. Notably, DTQE matches Q2K's maximum performance with just three keywords, whereas Q2K requires significantly more. This finding demonstrates that improving performance is not solely a matter of adding more keywords; instead, finding the tokens with diverse semantics is much more critical to enhancing the retrieval performance of QE, as evidenced by our approach leveraging the discarded tokens during the generation process.

Impact of Different LLMs. The right side of Figure 2 presents the results of using various LLMs for keyword generation. Our proposed method, DTQE, consistently outperforms Q2K across all LLMs, demonstrating strong adaptability and flexibility. Furthermore, ours effectively leverages the capabilities of stronger LLMs, such as the GPT series, as evidenced by the significantly higher gains with these models. These results confirm that our method is generalizable across diverse models and benefits from advancements in LLMs.

5 Conclusion

In this paper, we propose DTQE, a novel LLM-based query expansion framework that leverages both the keywords generated by LLMs and the discarded candidate tokens, capturing additional lexical cues without increasing the output length. Extensive experiments across multiple datasets, including both high- and low-resource settings, demonstrate that DTQE consistently enhances performance and, in some cases, even outperforms pseudo-document expansion while using significantly fewer tokens, showcasing its balance of effectiveness and efficiency.

Limitations

While DTQE consistently outperforms Q2K, it requires an additional inverted index to compute relevance scores for discarded tokens, leading to increased memory consumption. However, this overhead is negligible compared to the cost of expansion methods that generate long texts using LLMs.

Ethics Statement

Our proposed method, DTQE, does not pose ethical concerns, as it aims to enhance the retrieval performance of sparse retrievers by expanding queries with keywords and discarded tokens. However, LLMs may generate offensive or toxic text due to inherent biases when producing keywords. As LLMs continue to evolve, these biases are expected to diminish, mitigating this issue in the near future.

References

Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389.

Negar Arabzadeh, Xinyi Yan, and Charles L. A. Clarke. 2021. Predicting efficiency/effectiveness trade-offs for dense vs. sparse retrieval strategy selection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2862–2866, New York, NY, USA. Association for Computing Machinery.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

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

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the trec 2019 deep learning track.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

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

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Database-augmented query representation for information

retrieval. *arXiv preprint arXiv:2406.16013*, abs/2406.16013.

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

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 120–127, New York, NY, USA. Association for Computing Machinery.

Minghan Li, Honglei Zhuang, Kai Hui, Zhen Qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. 2024. Can query expansion improve generalization of strong cross-encoder rankers? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2321–2326, New York, NY, USA. Association for Computing Machinery.

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 *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Man Luo, Shashank Jain, Anchit Gupta, Arash Einolghozati, Barlas Oguz, Debojeet Chatterjee, Xilun Chen, Chitta Baral, and Peyman Heidari. 2023. A study on the efficiency and generalization of light hybrid retrievers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1617–1626, Toronto, Canada. Association for Computational Linguistics.

Yuanhua Lv and ChengXiang Zhai. 2009. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1895–1898, New York, NY, USA. Association for Computing Machinery.

Xing Han Lù. 2024. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2026–2031, New York, NY, USA. Association for Computing Machinery.

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

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction.

OpenAI. 2024. Gpt-4o system card.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

S. E. Robertson. 1991. On term selection for query expansion. *J. Doc.*, 46(4):359–364.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Text Retrieval Conference*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

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

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

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

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2024. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1987–2003, St. Julian's, Malta. Association for Computational Linguistics.

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

Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2024. PromptReps: Prompting large language models to generate dense and sparse representations for zero-shot document retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4375–4391, Miami, Florida, USA. Association for Computational Linguistics.

A Subword-tokenized index

LLMs rely on subword tokenization (Sennrich et al., 2016) to process text, meaning a single word can be split into multiple, smaller subword units that each carry semantic meaning. Discarded candidate tokens are likewise generated at the subword level. To effectively incorporate these subword tokens into retrieval, we segment each document into subword units and construct a new inverted index based on the frequency of these subwords. This index is exploited for calculating relevance score for discarded tokens.

B Implementation details

Keyword generation For our approach, we employ GPT-4o (OpenAI, 2024) as the LLM for keyword generation, given its status as a state-of-the-art model. When using PRF to generate keywords through an LLM, the top three documents from the initial search are provided as context to the LLM. Given that news datasets (TREC-News, Robust04) often contain very lengthy documents, we truncate each document to 512 tokens to maintain computational efficiency during LLM inference.

Index building We build (1) a word-level inverted index with Pyserini (Lin et al., 2021); (2) LLM tokenizer–based inverted index with Python's BM25S library (Lù, 2024), each using its default hyperparameters. For preprocessing, we apply stemming using NLTK (Bird et al., 2009) and leverage the gensim library (Řehůřek and Sojka, 2010) for lowercasing and punctuation elimination.

Retrieval Process We construct an expanded query by concatenating (i) the original query and (ii) generated keywords from the LLM. Following Jagerman et al. (2023), we repeat the original query terms five times to increase their relative importance. q' becomes:

$$q' = \underbrace{q \,\|\, \ldots \,\|\, q}_{5 \text{ times}} \,\|\, w_1 \,\|\, \ldots \,\|\, w_i.$$

We then apply this expanded query q' to a standard word-level index, yielding a BM25 score $S_{\mathrm{W}}(d)$ for each document d. Next, we normalize $S_{\mathrm{W}}(d)$ by dividing it with the same factor 5 used for repetition, resulting in $\tilde{S}_{\mathrm{W}}(d) = \frac{S_{\mathrm{W}}(d)}{5}$.

Method	Prompt

DTQE	`Write {num_keywords} keywords that are closely related to the given query:`
	
	`Query: {query}`
	
	`The output format is as follows:`
	`Keyword1, Keyword2, Keyword3 ...`

DTQE/PRF	`Write {num_keywords} keywords that are closely related to the given query based on the context:`
	
	`Context:`
	`{passage1}`
	`{passage2}`
	`{passage3}`
	
	`Query: {query}`
	
	`The output format is as follows:`
	`Keyword1, Keyword2, Keyword3 ...`

Table 2: Instructions of DTQE

Instructions We exploit two types of instruction conveyed to LLMs to generate keywords, as shown in Table 2.

C Effect of Token filtering

Methods	w/o PRF	w/ PRF
Q2K	45.5	46.4
DTQE w/o filter	45.0	47.1
+ duplicates removal	47.2	47.9
+ only first token	**47.5**	**48.0**

Table 3: Incremental ablation on the candidate token filtering process on 8 low resource datasets. Average NDCG@10 scores are reported.

Table 3 presents an ablation study of our DTQE approach on eight low-resource datasets, highlighting how each filtering process affect retrieval performance. First, DTQE without any filtering shows mixed results, even causing a performance drop in the w/o PRF setting compared to Q2K. This suggests that indiscriminately adding discarded token

candidates can introduce noise. When we apply duplicates removal, we observe a substantial performance improvement, indicating the importance of eliminating redundant or repetitive tokens. Finally, restricting the expansion to only the first discarded token provides additional gains. Overall, these findings validate the significance of systematic filtering in effectively harnessing discarded candidate tokens for query expansion.