# Enhancing Fusion-in-Decoder for Multi-Granularity Ranking

Haeju Park[1,*], Kyungjae Lee[2], Sunghyun Park[3] and Moontae Lee[4]

[1]LG AI Research, Republic of Korea
[2]LG AI Research, Republic of Korea
[3]LG AI Research, Republic of Korea
[4]LG AI Research, Republic of Korea

**Abstract**

Large Language Models (LLMs) have demonstrated exceptional performance across various natural language tasks, leveraging extensive knowledge from massive datasets. However, their reliance solely on parametric knowledge often leads to the generation of inaccurate or outdated content, particularly in domain-specific tasks. Retrieval Augmented Generation (RAG) has emerged as a promising approach to address this limitation by incorporating external knowledge without necessitating re-training. While RAG enhances the accuracy of LLM-generated content, effectively retrieving external knowledge remains a challenge due to potential noise and computational costs. To address this, traditional information retrieval systems adopt two-stage approaches, utilizing efficient retrievers followed by reranking mechanisms. Recently, transformer-based architectures, including BERT and T5 models, have shown promise as effective rerankers. However, such models have limited context size and only perform single-granularity ranking at a time, hindering their effectiveness and efficiency. In this paper, we first explore the existing rerankers such as RankT5 and RFiD, highlighting challenges in multi-granularity ranking. Subsequently, we introduce PFiD (Passage Fusion-in-Decoder), a simple yet efficient approach aimed at effectively ranking both document and passage simultaneously. Through empirical evaluation, we demonstrate the efficacy of PFiD in improving effectiveness and efficiency, offering a promising direction for further research in this domain.

**Keywords**

Information Systems, Retrieval Augmented Generation, Large Language Model

## 1. Introduction

Despite their remarkable capabilities and growth, Large Language Models (LLMs) [1, 2, 3, 4] still tend to generate factually incorrect or outdated content as their knowledge solely relies on their parametric knowledge, especially in domain-specific or knowledge-intensive tasks [5, 6, 7]. Retrieval Augmented Generation (RAG) approaches [8, 9, 10, 11] have gained significant attention, which improve the quality of LLM-generated output by grounding on external knowledge to supplement the LLMs' parametric knowledge, without having to re-train the LLMs. RAG leverages a powerful information retrieval model, which is designed to search large datasets or knowledge bases. The retrieved information is then incorporated into LLMs, enabling it to generate more accurate and contextually relevant content. By incorporating external knowledge, RAG can effectively reduce the problem of generating factually incorrect or outdated content in LLMs [12, 13].

However, current RAG frameworks have major challenges when it comes to the effectiveness and efficiency of information retrieval systems: First, LLMs tend to generate inaccurate responses on distracting (or noisy) contexts, thus the performance of retrieval models has a significant impact on the quality of RAG's responses [14, 15, 11]. Second, the retrieval component of RAG requires searching through large-scale knowledge bases or the web, which can be computationally expensive and slow [11]. Due to the above challenges, existing retrieval systems adopt two-stage approaches, an efficient first-stage retriever such as BM25 [16] and DPR [17] retrieves a set of documents from a larger dataset, and then a second-stage reranker is used to rerank retrieved documents for precise ranking. Recently, with the advent of transformer-based models such as BERT [18] and T5 [19], more architectures including bi-encoder [17], cross-encoder [20], encoder-decoder [21, 22], and decoder-only models [23], have gradually shown their effectiveness as a reranker. However, these models have limited context size and only perform single-granularity ranking during inference, which hinders their effectiveness and efficiency in real-world RAG scenarios.

To this end, in this paper, we focus on the multi-granularity ranking task, which ranks both document and passage simultaneously. Specifically, we first investigate the single-passage cross-encoder models such as MonoT5 [22] and RankT5 [21]. It achieves superior performance across various ranking tasks, but due to the constraint of input tokens, its efficiency is limited in real-world RAG scenarios. Next, we present the use of multi-passage cross-encoder, such as FiD [9] and RFiD [24]. These models alleviate the input tokens limit by leveraging multi-passage, but they directly use the cross-attention score of the decoder as a passage relevance, which is implicitly learned, and encounter difficulty with distinguishing relative differences between passages. Thereafter, we propose a simple and effective PFiD (Passage Fusion-in-Decoder) for multi-granularity ranking. PFiD extends the FiD model by generating a document-level relevance token, enabling both document retrieval and passage ranking. Furthermore, PFiD adopts the inter-passage attention mechanism to learn relative passage relevance explicitly, using the special tokens at the beginning of the input text to represent the entire context.

Experiments on MIRACL passage ranking dataset [25] demonstrate that PFiD improves effectiveness and efficiency compared to existing approaches, especially in RAG scenarios.

## 2. Preliminaries

### 2.1. Task definition

Given a user query $q$ and a document (or passage) corpus $C = \{D_1, D_2, ..., D_n\}$, the goal of document retrieval is to find the $k$ documents that are most relevant to the query $q$. In our multi-granularity ranking setting, which consists of document retrieval and passage ranking tasks, the document retrieval task is to perform reranking on BM25 retrieved

*Information Retrieval's Role in RAG Systems (IR-RAG) - 2024*
*Corresponding author.

top-$k$ documents. While traditional passage ranking tasks typically involve ranking entire passages, in this paper, the passage ranking task focuses solely on ranking passages within the retrieved document itself, which aligns more closely with real-world RAG scenarios and is thus more feasible.

### 2.2. Ranking models

Pre-trained Language Models (PLMs) are currently the most effective ranking models, which can be categorized into: bi-encoders and cross-encoders. Bi-encoders encode a query and a passage separately to obtain semantic representations [17], emerging as powerful first-stage retrievers by pre-computing the passage representations offline. Instead, cross-encoders take the concatenation of the query and a passage, and perform query-passage interactions [20], which have been conceived as second-stage rerankers, designed to explicitly refine the results provided by the first-stage retrieval. In this paper, for brevity, we also refer other PLMs, such as encoder-only [17, 20], decoder-only [23], and encoder-decoder [21, 22] models that perform query-passage interactions simultaneously, as cross-encoders.

There are several PLM-based cross-encoders, including sequence-to-sequence language models such as MonoT5 [22] and RankT5 [21] for ranking task, as well as multi-passage reader models like FiD [9] and RFiD [24] for RAG tasks, which have been demonstrated superior effectiveness.

**MonoT5.** MonoT5 [22] is the first work to define a ranking task as a text generation task by leveraging T5 [19] encoder-decoder model. A query-document pair is concatenated into an input sequence `Query` : $q$ `Document` : $D_n$ `Relevant:`, and utilizes `true` and `false` as target tokens to represent their relevance. Then, the model is fine-tuned for text generation task. After training, the ranking scores are derived from the logits of `true` token, based on the softmax applied only on the logits of the `true` and `false` tokens.

**RankT5.** Following MonoT5 [22], the input sequence is similar except that RankT5 do not include the `Relevant:` postfix. Then, the model use the `<extra_id_10>` as target token to learn unnormalized ranking score. The model is trained with list-wise ranking loss directly, instead of using text generation loss as in MonoT5 [22]. However, these models cannot be directly used for long document retrieval due to the maximum input length constraint as in most PLMs, which hinders their effectiveness in the document retrieval task.

**FiD.** The FiD model further extends T5 [19] encoder-decoder model, taking multiple $k$ passages as input, encoding separately, and then feeds the concatenated $k$ encoder hidden states into a T5 decoder to generate the answer. Relevance scores for passages are computed using cross-attention scores, which entail averaging the attention score across all tokens within the passage and all layers and heads within the decoder [26].

**RFiD.** While FiD [9] treats all passages equally within its encoders, solely depending on the cross-attention mechanism to establish correlations between the decoder and encoders, which may identify the incorrect answer by referring to spurious passages. Instead, RFiD [24] improves FiD by identifying potential answer-containing passages (or rationale) among the candidates and guiding the decoder with the identified rationales. Afterward, cross-attention scores are directly regarded as passage relevance scores the same as in [9]. However, even with the rationale, the cross-attention mechanism still lacks for distinguishing relative differences between passages, as it is implicitly guided by a rationale classifier solely trained on point-wise binary classification loss.

## 3. Method

In this section, we briefly discuss a simple but effective Passage Fusion-in-Decoder (PFiD) for multi-granularity ranking. PFiD adopts the FiD [9] architecture as a base model, further extends FiD by utilizing `true` and `false` token as a target token to model document-level relevance, enabling multi-granularity ranking simultaneously. Additionally, PFiD integrates inter-passage attention to learn relative passage relevances explicitly, which is similar to the list-wise training objective of RankT5 [21].

**Fusion-in-Decoder for Document Retrieval.** Formally, Given a question $q$ and a set of $k$ passages within the document $D_n = \{P_1^n, P_2^n, ..., P_k^n\}$, the FiD encoder outputs the $k$-th passage embeddings $\mathsf{H_k} \in \mathbb{R}^{L\times d}$, where $L$ denotes the maximum token length, and $d$ denotes the dimension of hidden states, which are then concatenated as the input of the fusion decoder $[\mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H_k}]$.

$$\mathsf{H_k} = \texttt{FiD-Encoder}(q + P_k^n) \tag{1}$$

The FiD decoder utilizes $[\mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H_k}]$ to generate the target token $T =$ `true` *or* `false`. Therefore, the loss function can be defined as follows:

$$\mathcal{L}_{FiD} = -\sum_{i=1}^{T} \log p(y_i|y_1, y_2, ...., y_{i-1}, [\mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H_k}]) \tag{2}$$

**Inter-passage Attention.** Previous work [24] tackled the issue of spurious passages by employing a binary classifier on the first token's encoder hidden states $\mathsf{H}_{k,1}$, to determine whether the passage is a rationale passage to the query. Then, guide the decoder by appending the additional embeddings toward the end of the encoder's hidden states $[\mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H_k}, \mathsf{H}_{k+1}]$, where $\mathsf{H}_{k+1} \in \mathbb{R}^{2\times d}$ is trainable rationale embedding. However, as Table 2 shows, it drastically underperforms in passage ranking tasks by a large margin, as it does not explicitly model relative passage relevance.

Instead, to mitigate this, we utilize inter-passage attention to model interactions between passages explicitly. PFiD builds a set of input sequences by appending the first token hidden states of each pair as $\mathsf{B} = [\mathsf{H}_{1,1}, \mathsf{H}_{2,1}, ..., \mathsf{H}_{k,1}]$, where $\mathsf{H}_{i,j}$ denotes the $j$-th token embeddings of $i$-th passage. In a standard cross-encoder, the first token of the encoder aggregate query-passage information to compute a relevance score. We further use this token to depict the relative semantics via self-attention mechanism. Inspried by [27], we consider single-layer transformer model to depict relative passage relevance as follows:

$$\widetilde{\mathsf{B}} = \texttt{softmax}\left(\frac{\mathsf{QK}^\intercal}{\sqrt{d}}\right)\mathsf{V}, \ \texttt{where} \ \mathsf{Q} = \mathsf{BW_Q}, \mathsf{K} = \mathsf{BW_K}, \mathsf{V} = \mathsf{BW_V} \tag{3}$$

in which matrices $\mathsf{W_Q}, \mathsf{W_K}, \mathsf{W_V} \in \mathbb{R}^{d\times d}$ are learnable parameters. The information from different passages is fused

and exchanged via the self-attention mechanism. The training loss used for inter-passage attention can be defined as follows:

$$p_k = \texttt{softmax}(\widetilde{\mathsf{B}}_k \mathsf{W}_B) \in \mathbb{R}^2,$$
$$\mathcal{L}_{passage} = -(y \log(p_k) + (1-y)\log(1-p_k)) \tag{4}$$

where $y$ is the passage relevance label, and the overall training objective of PFiD is:

$$\mathcal{L}_{all} = \mathcal{L}_{FiD} + \lambda \mathcal{L}_{passage}, \tag{5}$$

where $\lambda$ is a hyperparameter to balance two losses.

# 4. Experimental setup

## 4.1. Datasets

We use MIRACL [25] passage ranking dataset for our experiments. The MIRACL [25] dataset is a large-scale, open-domain, human-generated multi-document ranking dataset which is similar to MS MARCO [28], but MIRACL owns its advantage by providing segmented document collection, enabling both document retrieval and passage ranking.[1] For the document retrieval task, we construct the document retrieval dataset by regarding a document with at least one positive passage, as a positive document. Table 1 shows the statistics of the datasets.

**Table 1**
Statistics of Datasets.

| Task | # train | # dev | # avg judgement | # corpus |
|---|---|---|---|---|
| Document Retrieval | 22,548 | 6,404 | 2.22 | 5,758,285 |
| Passage Ranking | 29,416 | 8,350 | 2.75 | 32,893,221 |

## 4.2. Baselines

We compare PFiD against the following three types of ranking baselines. The first is Single-Passage Cross-encoder (SPC) baselines, including MonoT5 [22], and RankT5 [21]. Due to the constraint of input tokens, we only take the first-$k$ tokens in the document retrieval task. An alternative approach is to score each passage independently, and then take the passage with the highest score as the representative for ranking the document, or directly perform retrieval over the segmented passages. However, we will omit these approaches as the former lacks efficiency, and the latter is not scalable for real-world RAG scenarios. Then, the model is trained list-wisely with randomly sampled negatives from the entire passage sets; The second is Multi-Passage Cross-encoder (MPC) baselines, including FiD [9] and RFiD [24]. For comparison in our experimental setting, both FiD and RFiD models are trained with the target token of `true` *or* `false`, enabling both document retrieval and passage ranking. All SPC and MPC baselines used in this experiment are initialized with T5-base model; The third is the most frequently employed lexical ranker BM25 [16]. We use the Elasticsearch engine with default parameters $k_1$ = 1.2, and $b$ = 0.75.

[1]MS MARCO also provide segmented document collection, but the segmented corpus do not align with passages in passage ranking tasks.

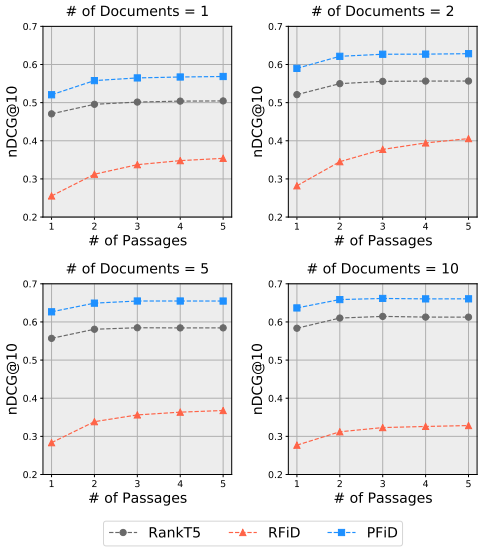


**Figure 1:** Passage ranking results on the real-world RAG scenarios. We first retrieve # of documents and rerank # passages within the retrieved documents.

## 4.3. Experimental Details

We adopt T5-base [19] as our base model, using Adam [29] with a learning rate of $10^{-4}$ and a dropout rate of 0.1. For both training and inference, we use the top-100 passages and truncate them to 200 of the maximum token length. The hyperparameter $\lambda$ is set to 0.5. For the document retrieval task, we perform ranking on BM25 top-100 retrieved documents, whereas passage ranking ranks the passages within the given positive document. We also conduct experiments on real-world RAG scenarios, considering both document retrieval and passage ranking simultaneously. We use the evaluation metric of the nDCG [30], Recall, and MRR scores to evaluate the effectiveness. All experiments are conducted on a single NVIDIA A100 GPU (40GB). In this work, we do not consider other training approaches including data augmentation, knowledge distillation, or negative sampling strategies as delving into their effects falls outside the scope of our objectives.

# 5. Results and Analysis

**Retrieval and Ranking.** Table 2 presents our evaluation results on document retrieval and passage ranking tasks. The key observations are as follows: (i) MPC significantly outperforms SPC in document retrieval task by aggregating multiple $k$ passages, alleviating the problem of limited context size in SPC. In particular, one can see that PFiD outperforms RFiD by a large margin on both document ranking and passage ranking task. This indicates that by leveraging passage-wise context to guide the decoder, we can better identify relative passage relevance. Note that compared with the existing SPC baselines, our method achieves ranking efficiency by explicitly removing the need for each granularity ranking. PFiD directly consumes the entire document, and scores the relevance of the entire passages and document simultaneously. (ii) RFiD, implicitly guiding the decoder with rationale embedding shows improvement over FiD by a large margin, however, it is still even worse than BM25. It suggests that implicitly guiding indeed benefits the model's ranking ability to some extent. However, when ranking

**Table 2**
The evaluation results of different baselines. As for the document retrieval, we rank the top 100 documents retrieved by BM25, while the passage ranking task ranks the passage within the retrieved document. $N$ denotes the number of documents to rank, whereas $P$ denotes the number of passages in the document. The best performances are in †. Latency indicates the total inference time from document retrieval to passage ranking, which is measured by averaging the time taken for each query with a single thread and a single batch on the GPU.

| Model | Category | Document Retrieval | | | | Passage Ranking | | | Complexity | Latency (s) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | top-$k$ | MRR@10 | Recall@5 | Recall@10 | MRR@10 | nDCG@5 | nDCG@10 | | |
| BM25 | - | \|C\| | 0.3951 | 0.3683 | 0.4736 | 0.7366 | 0.7718 | 0.7856 | - | 0.32 (x1.00) |
| MonoT5 | SPC | 100 | 0.6204 | 0.5141 | 0.5794 | 0.8571 | 0.8774 | 0.8803 | $O(N+NP)$ | 5.65 (x17.65) |
| RankT5 | SPC | 100 | 0.6352 | 0.4992 | 0.5605 | 0.8778† | 0.8916† | 0.8952† | $O(N+NP)$ | 5.64 (x17.62) |
| FiD | MPC | 100 | 0.6322 | 0.5139 | 0.5821 | 0.3464 | 0.3725 | 0.4260 | $O(N)$ | 1.17 (x3.65) |
| RFiD | MPC | 100 | 0.7177 | 0.5743 | 0.6407 | 0.5617 | 0.6036 | 0.6359 | $O(N)$ | 1.21 (x3.78) |
| PFiD (Ours) | MPC | 100 | 0.7231† | 0.5937† | 0.6516† | 0.8530 | 0.8726 | 0.8780 | $O(N)$ | 1.23 (x3.84) |

various passages from multi-documents, traditional MPC is completely indistinguishable, suggesting cross-attention score from the decoder is not suited for the passage ranking task. (iii) SPC achieves superior performance over MPC in passage ranking task, as it is trained with rich negative samples from other documents, while MPC is only trained with in-document negatives. Additionally, even with in-document negatives, when trained with inter-passage attention, PFiD can achieve ranking effectiveness that rivals that of SPC, suggesting that incorporating an additional module to identify relevant passages is more effective than relying solely on the cross-attention mechanism of the decoder.

**Results on real-world RAG scenario.** Next, we investigate the effectiveness of PFiD in real-world RAG scenarios. We first retrieve # documents from the candidates, and rerank # passages within the retrieved documents. Figure 1 represents the result of our evaluation. Notably, from Table 2 we observed that MPC outperforms SPC in document retrieval tasks, however, the performance drastically drops in this setting, as cross-attention scores from the decoder are indistinguishable across passages from multi-documents. Additionally, despite RankT5 reaching the best effectiveness on the passage ranking task, it did not exhibit any improvements over our method in real-world RAG scenarios, suggesting the importance of the multi-granularity ranking. Instead, PFiD consistently outperforms all baselines, by leveraging the complementary nature of SPC and MPC. PFiD is capable of more efficiently retrieving documents and ranking passages, and capturing the relative semantic correlation between different passages, leading to superior performance.

**Cross-attention vs PFiD.** As discussed above, PFiD has the advantage of identifying relevant passages compared to previous models like RFiD since it explicitly models relative passage relevance. We investigate the effects of the cross-attention scores of the decoder and our passage ranking scores for the passage ranking task. Figure 2 illustrates the distribution of the rank of positive passages. As depicted in Figure 2, the PFiD is more strongly correlated with passage relevances than cross-attention scores, suggesting the PFiD focuses more on positive passages by explicitly learning relative passage relevance. Our experimental results show that the enhanced ability to identify relevant passages contributes to overall performance improvement.

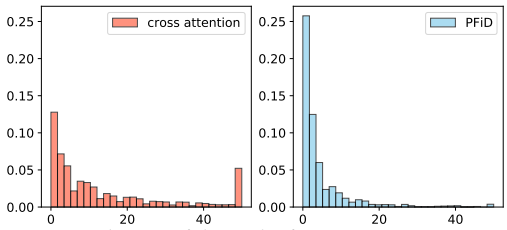

**Figure 2:** Distribution of the rank of positive passages.

## References


[1] OpenAI, Gpt-4 technical report, 2024. arXiv:2303.08774.

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

[3] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. C. et al., Palm 2 technical report, 2023. arXiv:2305.10403.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[5] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, H. Hajishirzi, When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 9802–9822. URL: https://aclanthology.org/2023.acl-long.546. doi:10.18653/v1/2023.acl-long.546.

[6] J. Maynez, S. Narayan, B. Bohnet, R. McDonald, On faithfulness and factuality in abstractive summarization, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1906–1919. URL: https://

aclanthology.org/2020.acl-main.173. doi:10.18653/v1/2020.acl-main.173.

[7] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023. arXiv:2311.05232.

[8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. arXiv:2005.11401.

[9] G. Izacard, E. Grave, Leveraging passage retrieval with generative models for open domain question answering, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 874–880. URL: https://aclanthology.org/2021.eacl-main.74. doi:10.18653/v1/2021.eacl-main.74.

[10] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, E. Grave, Atlas: Few-shot learning with retrieval augmented language models, Journal of Machine Learning Research 24 (2023) 1–43. URL: http://jmlr.org/papers/v24/23-0037.html.

[11] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2024. arXiv:2312.10997.

[12] H. He, H. Zhang, D. Roth, Rethinking with retrieval: Faithful large language model inference, 2022. arXiv:2301.00303.

[13] N. Thakur, L. Bonifacio, X. Zhang, O. Ogundepo, E. Kamalloo, D. Alfonso-Hermelo, X. Li, Q. Liu, B. Chen, M. Rezagholizadeh, J. Lin, Nomiracl: Knowing when you don't know for robust multilingual retrieval-augmented generation, 2024. arXiv:2312.11361.

[14] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi, N. Schärli, D. Zhou, Large language models can be easily distracted by irrelevant context, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), Proceedings of the 40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 31210–31227. URL: https://proceedings.mlr.press/v202/shi23a.html.

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

[16] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389. doi:10.1561/1500000019.

[17] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W. tau Yih, Dense passage retrieval for open-domain question answering, 2020. arXiv:2004.04906.

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. arXiv:1910.10683.

[20] R. Nogueira, K. Cho, Passage re-ranking with bert, 2020. arXiv:1901.04085.

[21] H. Zhuang, Z. Qin, R. Jagerman, K. Hui, J. Ma, J. Lu, J. Ni, X. Wang, M. Bendersky, Rankt5: Fine-tuning t5 for text ranking with ranking losses, 2022. arXiv:2210.10634.

[22] R. Nogueira, Z. Jiang, J. Lin, Document ranking with a pretrained sequence-to-sequence model, 2020. arXiv:2003.06713.

[23] X. Ma, L. Wang, N. Yang, F. Wei, J. Lin, Fine-tuning llama for multi-stage text retrieval, 2023. arXiv:2310.08319.

[24] C. Wang, H. Yu, Y. Zhang, Rfid: Towards rational fusion-in-decoder for open-domain question answering, 2023. arXiv:2305.17041.

[25] X. Zhang, N. Thakur, O. Ogundepo, E. Kamalloo, D. Alfonso-Hermelo, X. Li, Q. Liu, M. Rezagholizadeh, J. Lin, MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages, Transactions of the Association for Computational Linguistics 11 (2023) 1114–1131.

[26] G. Izacard, E. Grave, Distilling knowledge from reader to retriever for question answering, 2022. arXiv:2012.04584.

[27] J. Yang, Z. Liu, C. Li, G. Sun, X. Xie, Longtriever: a pre-trained long text encoder for dense document retrieval, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 3655–3665. URL: https://aclanthology.org/2023.emnlp-main.223. doi:10.18653/v1/2023.emnlp-main.223.

[28] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, T. Wang, Ms marco: A human generated machine reading comprehension dataset, 2018. arXiv:1611.09268.

[29] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.

[30] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Trans. Inf. Syst. 20 (2002) 422–446. URL: https://doi.org/10.1145/582415.582418. doi:10.1145/582415.582418.