Fusion-in-T5: Unifying Document Ranking Signals for Improved Information Retrieval

Shi Yu[1*], Chenghao Fan[2*], Chenyan Xiong[3], David Jin[4], Zhiyuan Liu[15], and Zhenghao Liu[6]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
[2]School of Comp. Sci., Huazhong University of Science and Technology, Wuhan, China
[3]Microsoft Research, Redmond, USA [4]MIT, Cambridge, USA
[5]Beijing National Research Center for Information Science and Technology, Beijing, China
[6]Dept. of Comp. Sci. & Tech., Northeastern University, Shenyang, China

Abstract

Common IR pipelines are typically cascade systems that may involve multiple rankers and/or fusion models to integrate different information step-by-step. In this paper, we propose a novel re-ranker named Fusion-in-T5 (FiT5), which integrates document text information, retrieval features, and global document information into a single unified model using templated-based input and global attention. Experiments on passage ranking benchmarks MS MARCO and TREC DL show that FiT5 significantly improves ranking performance over prior pipelines. Analyses find that through global attention, FiT5 is able to jointly utilize the ranking features via gradually attending to related documents, and thus improve the detection of subtle nuances between them. Our code will be open-sourced.

1 Introduction

Information retrieval (IR) aims to retrieve a relevant set of documents from a large collection, given a user query (Croft et al., 2010). The task poses challenges for researchers to build models that are able to process vast amounts of information in response to a single input query.

The information-rich nature of IR motivates researchers to construct intricate, *cascade* systems (Yates et al., 2021; Zhang et al., 2021; Dai et al., 2018). Neural IR models often serve as the foundation of such systems, directly capturing text relevance in a coarse-to-fine approach (Yates et al., 2021; Nogueira et al., 2019; Pradeep et al., 2021). To capture ranking features observed from the data or the rankers, a learning-to-rank (LeToR) module is often applied (Zhang et al., 2020; Sun et al., 2021; Zhang et al., 2021, 2022; Dai et al., 2018). Further approaches are introduced to incorporate global information from the documents at the cost of an additional ranking stage, such as designing an extra pair/list-wise re-ranker (Nogueira et al., 2019; Zhang et al., 2022), or using pseudo relevance feedback (PRF) to expand the query with potentially relevant document information (Zheng et al., 2020; Yu et al., 2021; Li et al., 2023). These techniques ultimately transform the ranking process into a task that demands careful engineering in order to achieve optimal performance.

In this paper, we introduce Fusion-in-T5 (FiT5), a T5-based (Raffel et al., 2020) re-ranking model that collects ranking signals and ranks documents within a unified framework. FiT5 is designed to consolidate multiple IR features, including document texts, ranking features, and global document information, into a single learnable model. Specifically, the input to FiT5 is formulated using a template that incorporates the document text with the ranking feature, which is represented as discretized integers. Additionally, to leverage information from other documents, we introduce global attention on the representation token from the late layers of FiT5 encoders, enabling document-wise information flow during encoding while mitigating the increase in computational cost. FiT5 functions as a re-ranking model within a typical two-stage retrieve-and-rerank pipeline, without the need for additional stages or hyperparameters.

Experimental results on widely-used IR benchmarks, namely MS MARCO (Nguyen et al., 2016) and TREC DL 2019 (Craswell et al., 2020) & 2020 (Craswell et al., 2021), demonstrate that FiT5 exhibits substantial improvements over traditional retrieve-and-rerank pipelines. Furthermore, FiT5 outperforms systems with more re-ranking stages and/or larger models on the MS MARCO dataset. Further analysis reveals that FiT5 effectively leverages ranking features through its global attention architecture, enabling the model to better differentiate between similar documents and ultimately produce better ranking results.

[*]Equal contribution.

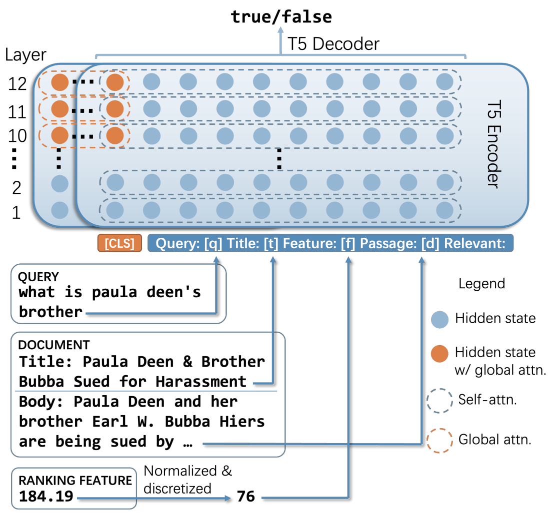


Figure 1: Architecture of Fusion-in-T5. The query, document, and ranking feature are filled in the input template to form the input. We use retrieval score as the ranking feature.

2 Related Work

IR Pipelines Recent IR pipelines are often cascade systems consisting of a retriever and one/multiple re-ranker(s) (Yates et al., 2021). The simplest form of a re-ranker is a pre-trained language model (PLM)-based model, which takes a pair of (query, document) texts as input and outputs a relevance score, e.g. BERT Re-ranker (Nogueira and Cho, 2019) and monoT5 (Nogueira et al., 2020). Learning-to-rank (LeToR) models (Liu et al., 2009) are often used to learn a final ranking score based on a set of data or ranker features, such as linear models (Vogt and Cottrell, 1999; Metzler and Bruce Croft, 2007) and neural networks (Han et al., 2020; Burges et al., 2005; Zhang et al., 2022). To leverage features from other candidate documents, researchers have proposed pseudo relevance feedback (PRF) to expand the query (Zheng et al., 2020; Yu et al., 2021; Li et al., 2023), and pair/list-wise re-ranking models duoT5 (Pradeep et al., 2021) and HLATR (Zhang et al., 2022). Despite their effectiveness, these methods introduce an extra stage in ranking, which may bring an additional efficiency burden.

Attention over Multiple Texts Our work leverages similar ideas from Fusion-in-Decoder (FiD) (Izacard and Grave, 2021) and Transformer-XH (Zhao et al., 2020) to incorporate global information. FiD adds a T5 decoder model on top of the independent T5 document encoders to fuse all the text evidences through the decoder-encoder attention and generate the answer for open-domain QA. Transformer-XH builds eXtra Hop attention across the text evidences inside the BERT layers to model the structure of texts for multi-hop QA. In this paper, we integrate the similar attention mechanism into the T5 encoder and build a fully-connected attention graph to model all the mutual relationships between candidate documents.

3 Methodology

3.1 Overview

FiT5 performs re-ranking on a set of candidate documents $D = \{d_1, d_2, ..., d_n\}$ retrieved by a first-stage retriever, given a query q. Unlike typical re-ranking models which calculate s_i solely based on the query and one document text, denoted as $s_i = f(q, d_i)$, FiT5 goes beyond the approach by further incorporating the ranking feature r_i and the information from all the documents in D, which can be formulated as $s_i = f(q, d_i, r_i, D)$.

Figure 1 presents the overall architecture of FiT5. FiT5 is based on the encoder-decoder model T5 (Raffel et al., 2020). It takes a triple of (q, d_i, r_i) as the input and outputs a relevant score s_i. Global attention is introduced in the late layers of the encoder to incorporate information from other documents in D. We describe the input and output format in §3.2 and the global attention in §3.3.

3.2 Input and Output

We pack (q, d_i, r_i) using a template to form the input to FiT5. The template consists of slots for input data and several prompt tokens, defined as

$$\text{Query: } \texttt{[q]} \text{ Title: } \texttt{[t]} \text{ Feature: } \texttt{[f]} \text{ Passage: } \texttt{[d]} \text{ Relevant:} \tag{1}$$

where, `[q]`, `[t]` and `[d]` are slots for text features, corresponding to the query q, the title and the body of the document d_i, respectively. `[f]` is the slot for the feature r_i (i.e. the retrieval score in this paper), represented as a normalized, discretized integer.

The model is fine-tuned to decode the token "true" or "false" according to the input. During inference, the final relevance score is obtained from the normalized probability of the token "true".

3.3 Global Attention

In the document set D, there may exist many related documents that may share similar content with the current example. The distinctions between

Model	# Params	MS MARCO		TREC DL'19		TREC DL'20	
		MRR@10	MAP	NDCG@10	MRR	NDCG@10	MRR
First Stage Retrieval							
BM25	–	18.7	19.5	50.58	70.36	47.96	65.85
coCondenser (2022)	–	38.3	37.6	71.45	86.75	67.97	84.41
*Two-stage Ranking (coCondenser → *)*							
BERT Re-ranker (2019)	110M	39.2	38.6	70.12	83.80	69.23	82.26
monoT5 (2020)	220M	40.6	39.9	72.55	84.79	67.73	85.05
FiT5	227M	**43.9**	**43.3**	**77.63**	**87.40**	**75.24**	**85.48**
Three-stage Ranking (For Reference)							
HLATR-base (2022)	132M	42.5	–	–	–	–	–
HLATR-large (2022)	342M	43.7	–	–	–	–	–
Expando-Mono-Duo (2021)	2×3B	42.0	–	–	–	78.37	87.98

Table 1: Overall results on MS MARCO and TREC DL 19 & 20.

these documents may not be captured effectively via point-wise inference over the "local" information (q, d_i, r_i). To enhance the effectiveness of ranking, we propose *global* attention in FiT5 to enable the model to better comprehend and differentiate these documents in the ranking process.

In FiT5, each (q, d_i, r_i) pair first runs through $l-1$ transformer encoder layers independently, as in vanilla T5. Global attention is injected into every layer $j \geq l$. The representation of the first token [CLS] (prepended to the input), denoted as $h^j_{i,\text{[CLS]}} \in \mathbb{R}^c$, is picked out from the normal self-attention:

$$h^j_{i,\text{[CLS]}}, \hat{\mathbf{H}}^j_i = \text{TF}(\mathbf{H}^{j-1}_i) \tag{2}$$

where $\hat{\mathbf{H}}^j_i$ denotes the remaining part of the hidden representation, c is the hidden size and TF is the transformer layer. The representations of the first tokens from all n encoders are then fed into a global attention layer:

$$\begin{aligned}&\hat{h}^j_{1,\text{[CLS]}}, ..., \hat{h}^j_{n,\text{[CLS]}} \\ =&\text{Global\_Attn}(h^j_{1,\text{[CLS]}}, ..., h^j_{n,\text{[CLS]}})\end{aligned} \tag{3}$$

Finally, the globally-attended representation $\hat{h}^j_{i,\text{[CLS]}}$ is added back to the hidden representation:

$$\mathbf{H}^j_i = [h^j_{i,\text{[CLS]}} + \hat{h}^j_{i,\text{[CLS]}}; \hat{\mathbf{H}}^j_i] \tag{4}$$

In this way, the global information is modeled in the representation of the [CLS] token and can be further leveraged by the following layer(s). This provides a chance for the model to adjust the representation according to other relating documents.

4 Experimental Methodology

Datasets and Metrics We train FiT5 on MS MARCO passage ranking dataset (Nguyen et al., 2016) and evaluate it on the development set and TREC Deep Learning Tracks (TREC DL) 2019 & 2020 (Craswell et al., 2020, 2021). MS MARCO labels are binary sparse labels derived from click data with often one positive document per query. TREC DL labels are dense judgments on a four-point scale from irrelevant to perfectly relevant and thus are more comprehensive (Craswell et al., 2020, 2021). We report MRR@10, MAP and MS MARCO, and NDCG@10, MRR on TREC DL.

Implementation Details We use T5-base model (Raffel et al., 2020) as the backbone of our model. Global attention modules are added starting from the third to last layer (i.e. $l = 10$). We re-rank the top 100 documents from coCondenser (Gao and Callan, 2022) ans use coCondenser retrieval score as ranking features in the template (Eq. 1). We first train a FiT5 without the features to warm-up the model for 400k steps, and then train it with features for 1.5k steps to obtain the final model. It is acceptable to incorporate more additional ranking features in a template to optimize the model.

Baselines We compare FiT5 with typical two-stage retrieve-and-rerank pipelines including BERT Re-ranker (Nogueira and Cho, 2019) and monoT5 (Nogueira et al., 2020). These re-rankers simply assign a score for a (q, d_i) text pair. To have a fair comparison, the first-stage retrieval for such pipelines is kept the same as FiT5. We also report the performance of three-stage ranking pipelines HLATR (Zhang et al., 2022) and Expando-Mono-Duo (Pradeep et al., 2021) for reference.

5 Evaluation Results

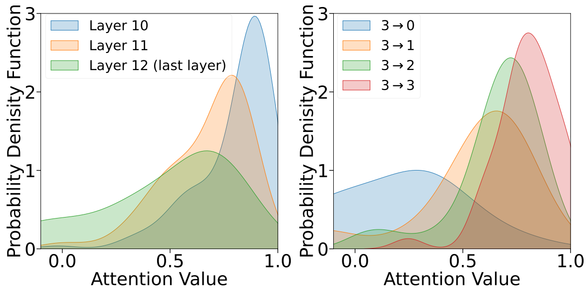
This section presents the overall results of FiT5, and analyses its effectiveness.

Model	MARCO	DL'19	DL'20
monoT5	40.56	72.55	67.73
monoT5 (w/ feature)	40.95	72.12	68.73
FiT5 (w/o feature)	42.79	74.94	70.02
FiT5 (linear combination)	43.65	75.41	70.95
FiT5	**43.93**	**77.63**	**75.24**

Table 2: Ablation study of FiT5. The evaluation metric is MRR@10 on MS MARCO and NDCG@10 on TREC DL.

Model	FiT5 (w/o feature)	FiT5
All layers ($l = 1$)	41.23	40.83
Top-6 layers ($l = 7$)	42.49	43.36
Top-3 layers ($l = 10$)	42.79	**43.93**
Top-2 layers ($l = 11$)	**42.95**	43.43
Top-1 layer ($l = 12$)	42.78	43.07
No global attention	41.49	40.95

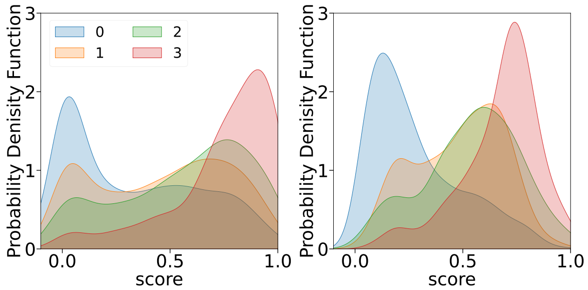
Table 3: Performance on MS MARCO with global attention started to introduce at top-k transformer layers. The evaluation metric is MRR@10.



(a) At each layer.

(b) At last layer.

Figure 2: Attention weights distribution on TREC DL 20. (a) presents the attention weights from passages labeled 3 (perfectly relevant) to other passages in each global attention layer. (b) depicts the last-layer attention weights between perfectly-relevant docs and others.



(a) monoT5 (w/ ret. score).

(b) FiT5.

Figure 3: Output score distributions on passages at different annotated relevance levels from TREC DL 20. 0, 1, 2, and 3 are relevance levels from irrelevant (0) to perfectly relevant (3).

5.1 Overall Performance

The results of passage ranking on MS MARCO and TREC DL are presented in Table 1. By incorporating multiple types of ranking information, FiT5 greatly improves over the first-stage retrieval model coCondenser, and outperforms typical BERT Re-ranker and monoT5 that re-rank on top of the same retriever. On MS MARCO, FiT5 further outperforms three-stage ranking pipelines HLATR-large and Expando-Mono-Duo, which uses significantly larger models (RoBERTa-large (Liu et al., 2019) / 2×T5-3B) and one additional re-ranking stage.

5.2 Ablation Study

In this section, we study the contribution of additional ranking features (retrieval score) and global attention in the effectiveness of FiT5 and present the results in Table 2. Removing the feature score (FiT5 (w/o feature)) or the global attention (monoT5 (w/ feature)) both results in significant performance drop. Notably, monoT5 (w/ feature) doesn't have a significant performance gain over monoT5, indicating that the ranking features can't be effectively captured in a point-wise model. Using linear combination of the re-ranker score and features still lags behind FiT5, revealing that the use of global attention is the key to effectively integrating the information from the retriever and other documents.

5.3 Attention Depth

In this experiment, we investigate the impact of the number of transformer layers with global attention on the model's performance. We re-train FiT5 on MS MARCO with top 1, 2, 3, 6, and 12 layer(s) incorporated with global attention, respectively. The results presented in Table 3 reveal that starting to integrate global attention from a late layer ($l = 10$) is an optimal choice. Starting the integration too early may mislead the model from the beginning, whereas starting too late may provide insufficient paths for reasoning over ranking features.

5.4 Attention and Score Distribution

In this experiment, we study the attention and scoring behavior of FiT5. In Figure 2, we analyze the distribution of the global attention values. As shown in Figure 2a, as the layer depth increases, the attention values between passages labeled 3 (perfectly relevant) and other passages become closer to 0. As shown in Figure 2b, in the last layer, the attention values between most relevant passages are significantly larger than those with less relevant passages. The attention patterns indicate that, by passing through multiple global attention layers, our model learns to gradually attend to the related relevant documents. In Figure 3, we present the scores of documents with different labels. It can be observed that FiT5 produces more distinguishable distributions, indicating that it can better capture the nuances between similar documents.

6 Conclusion

In this paper, we propose Fusion-in-T5 (FiT5), which collects and unifies IR features on the re-ranking stage. We conduct experiments on MS MARCO and TREC DL, demonstrating FiT5's advantage in effectiveness. In addition, we provide an analytical demonstration to show the rationale of the effectiveness of FiT5 in incorporating global document information and ranking features.

References

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of ICML*, pages 89–96.

N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. 2021. Overview of the trec 2020 deep learning track. In *TREC*.

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

W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search Engines: Information Retrieval in Practice*, volume 520. Addison-Wesley Reading.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of WSDM*, pages 126–134.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of ACL*, pages 2843–2853.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Tevatron: An efficient and flexible toolkit for dense retrieval. *arXiv preprint arXiv:2203.05765*.

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-rank with bert in tf-ranking. *arXiv preprint arXiv:2004.08476*.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of EACL*, pages 874–880.

Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *TOIS*, 41(3):1–40.

Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Donald Metzler and W Bruce Croft. 2007. Linear feature-based models for information retrieval. *Information Retrieval*, 10:257–274.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of EMNLP*, pages 708–718.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. 2021. Few-shot text ranking with meta adapted synthetic weak supervision. In *Proceedings of ACL*, pages 5030–5043.

Christopher C Vogt and Garrison W Cottrell. 1999. Fusion via a linear combination of scores. *Information retrieval*, 1(3):151–173.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of WSDM*, pages 1154–1156.

HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. *arXiv preprint arXiv:2108.13454*.

Kaitao Zhang, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2020. Selective weak supervision for neural information retrieval. In *Proceedings of The Web Conference 2020*, pages 474–485.

Yanzhao Zhang, Dingkun Long, Guangwei Xu, and Pengjun Xie. 2022. Hlatr: enhance multi-stage text retrieval with hybrid list aware transformer reranking. *arXiv preprint arXiv:2205.10569*.

Yue Zhang, ChengCheng Hu, Yuqi Liu, Hui Fang, and Jimmy Lin. 2021. Learning to rank in the age of muppets: Effectiveness–efficiency tradeoffs in multi-stage ranking. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 64–73.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-xh: Multi-evidence reasoning with extra hop attention. In *Proceedings of ICLR*.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. Bert-qe: Contextualized query expansion for document re-ranking. In *Findings of EMNLP*, pages 4718–4728.

A Datasets

MS MARCO passage (Nguyen et al., 2016) is a ranking dataset with 8.8M passages, which is constructed from Bing's search query logs and web documents retrieved by Bing. The training set has about 530K queries. The development sets contain 6,900 respectively.We train FiT5 on MS MARCO passage ranking. For every query, we take top-100 documents retrieved using coCondenser for re-ranking, which is implemented with Tevatron (Gao et al., 2022). We divide MS MARCO training into a training set of 495k samples and our own validation set of 3195 samples.

TREC Deep Learning Tracks are the test collections designed to study ad hoc ranking in a large data regime. The passage corpus of MSMARCO is shared with TREC-DL'19 (Craswell et al., 2020) and TREC-DL'20 (Craswell et al., 2021) collections with 43 and 54 queries respectively. We evaluate our model on these collections.

B Baselines

We compare against the following baselines:

BERT Re-ranker: (Nogueira and Cho, 2019) We use BERT-base to re-rank the top 100 documents from coCondenser and take the checkpoint at 100k steps as the result. In order to maintain consistency with FiT5, title information is also added during training of BERT-reranker.

monoT5: (Nogueira et al., 2020) We use monoT5 to re-rank the top 100 documents from coCondenser, with the same training details as monoT5. We take the checkpoint at 100k steps as the result. Then, following the training step as FiT5 (w/ feature), we use coCondenser retrieval scores as an additional ranking feature in the template (Eq 1).We train the model on the MS MARCO training set using the checkpoint obtained from the previous step, and use the checkpoint which that achieves the best performance on our validation set (i.e. monoT5 (w/ feature). In order to maintain consistency with FiT5, title information is also added during training of monoT5.

HLATR: (Zhang et al., 2022) HLATR is a model trained on the coCondenser retrieval results and also utilizes retrieval scores to enhance document information representation during the re-ranking stage. It is trained on MS MARCO using RoBERTa-base and RoBERTa-large, which we refer to as HLATR-base and HLATR-large, respectively.

Expando-Mono-Duo: (Pradeep et al., 2021) Expando-Mono-Duo is a multi-stage ranking model based on T5-3B, which requires pairwise comparison on the candidate documents.

C Training Details

In the training of FiT5 (w/o feature), the learning rate for the document ranking task is 2×10^{-5}, and the total batch size is 16. Each global attention module applies standard multi-head attention with 12 attention heads. We train the model for 400k steps on the MS MARCO training set and take the best-performing checkpoint on our validation set. In order to gain a deeper understanding of ranking features (retrieval scores in FiT5) and integrate them into the FiT5 model, we continue the training on FiT5 (w/o feature) using the template with feature-related components like Eq 1. Before incorporating feature scores, we normalize the coCondenser score to [0,1] using min-max normalization. To reduce the impact of extreme values, we set the minimum value at 165 and the maximum at 190 during normalization. The scores are then discretized to an integer in [0,100] by retaining two decimal places, input to the model as normal strings. In the training of FiT5, the learning rate for the document ranking task is 2×10^{-5}, and the total batch size is 256. We train FiT5 on the MS MARCO training set from the checkpoint saved of FiT5 (w/o feature) and use the result of the 1.5k steps as the final result.

In addition to incorporating feature information as text feature and fusing them with language model, we also employed a linear fusion method, as shown in the table 2 as FiT5 (linear combination). We used the linear fusion method in coor-ascent from RankLib[1] to fuse the ranking scores obtained from the first stage FiT5 (w/o feature) and the feature scores from coCondenser. Specifically, we randomly sampled 10k instances from the training data and trained RankLib to obtain the best linear fusion model, which was used as FiT5 (linear combination).

D Experiment Details

In the experiment analyzing attention distribution in §5.4, we compute attention values using the fol-

[1] https://sourceforge.net/p/lemur/code/HEAD/tree/RankLib/

lowing method. We assume that the global attention similarity between the i-th and k-th samples in the j-th layer of transformers is denoted by $A_{i,k}^{j}$:

$$A_{i,k}^{j} = \frac{\hat{h}_{i,\text{[CLS]}}^{j} \cdot \hat{h}_{k,\text{[CLS]}}^{j}}{||\hat{h}_{i,\text{[CLS]}}^{j}||||\hat{h}_{k,\text{[CLS]}}^{j}||} \tag{5}$$

Assuming the i-th sample is associated with a relevance label l_i for query q, we compute the mean value of global attention similarity $A_q^j(R_1, R_2)$ in the j-th layer between samples with relevance scores R_1 and R_2,which indicate the model's ability to distinguish between similar documents.

$$A_q^j(R_1, R_2) = \frac{\sum_{i=1,l_i=R_1}^{n} \sum_{k=1,r_k=R_2}^{n} A_{i,k}^{j}}{\sum_{i=1,l_i=R_1}^{n} \sum_{k=1,l_k=R_2}^{n} 1} \tag{6}$$

To facilitate smoother visualization of the results for all queries, we perform min-max normalization on the those scores in the same layer j.

$$\{A_q^j(R_1, R_2)\} = \text{Min-Max}(\{A_q^j(R_1, R_2)\}) \tag{7}$$

For j equal to 10, 11, and 12, with R_1 and R_2 ranging from 0 to 3, the outcomes are presented in Figure 2a. Additionally, for j equal to 12, with R_1 at 3 and R_2 ranging from 0 to 3, the outcomes are shown in Figure 2b.