Intra-document Block Pre-ranking for BERT-based Long Document Information Retrieval - Abstract

Minghan Li[1], Eric Gaussier[1]

[1]Univ. Grenoble Alpes, CNRS, LIG, Grenoble, France

Abstract

Information retrieval using transformer architectures, especially pretrained models like BERT, has seen great improvements. However, due to the quadratic complexity of the self-attention mechanism, for long documents, directly using such models is unsatisfactory. Truncating long documents is a widely adopted approach. Other researchers also propose to separate a long document into passages, each of which can be treated by a standard BERT model. The other solution is modifying the self-attention mechanism to make it sparser. However, these approaches either lose information or have high computational complexity and memory requirement. We propose a slightly different approach that firstly pre-ranks passages within a long document according to the query, after which the filtered top-ranking passages are combined for later ranking to obtain the document relevance score. Experiments on IR collections demonstrate the SOTA level effectiveness of the proposed approach.

Keywords

Neural IR, Document Representation for IR, BERT-based Models

1. Introduction

Document information retrieval (IR) is used in many applications in our daily lives, including as web search. Benefiting from deep neural networks, Neural Information Retrieval (Neural IR) has led to the development of numerous interesting IR models. The transformer model [1], which is based on the multi-head attention mechanism, has shown to be more parallelizable and of greater quality than recurrent neural network models. Based on the transformer encoder, Devlin et al. [2] propose Bidirectional Encoder Representations from Transformers (BERT) by pre-training it on large scale corpus using self-supervised learning. Fine-tuning on BERT-like models enables one to produce cutting-edge models on a variety of tasks including information retrieval [3, 4, 5, 6]. Despite its effectiveness and intuitive characteristics, the amount of input tokens is restricted to 512 due to the quadratic complexity of the self-attention mechanism, which is less than the length of a long document.

To tackle this issue in IR, three techniques have been presented. The first kind is truncation [3, 4] which directly uses the beginning tokens in long documents. The second type involves segmenting long documents into shorter passages, where a hierarchical architecture can be used. The last focuses on modifying the self-attention to use a sparser attention mechanism.

CIRCLE (Joint Conference of the Information Retrieval Communities in Europe), July 04–07, 2022, Samatan, Gers, France
minghan.li@univ-grenoble-alpes.fr (M. Li); eric.gaussier@imag.fr (E. Gaussier)
0000-0002-1041-8887 (M. Li); 0000-0002-8858-3233 (E. Gaussier)

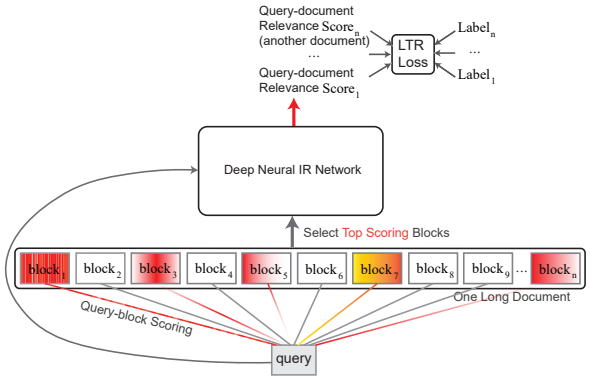

Figure 1: An illustration of the architecture of KeyB (e.g., BM25).

However, all three techniques have drawbacks such as information loss, high computational costs, memory requirements or requiring modifying CUDA kernels [7, 8].

From a different perspective, we propose a new framework [7, 9] for long document information retrieval. Similar to human judgement process [10], this framework firstly searches relevance blocks within a long document according to the query. Then top ranking blocks are combined as the query-directed summary which suits the BERT's capacity. Finally the relevance score of a long document is obtained via a BERT model on the query and summary which can be regarded as an aggregation of local relevance information.

This extended abstract describes this framework and the remainder of the paper is organized as follows: Section 2 will describe the proposed framework with classical IR approach for block ranking. While section 3 will describe two semantic block matching approaches. Then Section 4 will show the experimental results.

2. The proposed architecture with classical IR functions

To begin, we introduce the proposed framework, which utilizes classical IR functions such as BM25 for intra block ranking. It is illustrated in Figure 1. The query-block scoring part, as can be seen, is used to identify relevant blocks across the whole document, which may be regarded as a pre-ranking strategy. A neural IR network that generates relevance scores for a learning-to-rank (LTR) loss is represented by the Deep Neural IR Network. In this study, we focus here on two state-of-the-art neural IR models, namely Vanilla BERT [3] and PARADE [6]. A long document is firstly segmented into blocks, then the query-block scoring step firstly picks the relevant blocks according its retrieval status value (RSV) with the query using e.g. BM25: $RSV(q,b)_{BM25} = \sum_{w \in q \cap b} IDF(w) \cdot \frac{tf_w^b}{k_1 \cdot (1-b+b \cdot \frac{l_b}{l_{avg}}) + tf_w^b}$, where l_b is the length of block b, l_{avg} the average length of the blocks in d, and k_1 and b are two hyperparameters.

The IDF is based on documents rather than blocks, as using blocks instead of documents might lead to bias, as important words in a document are likely to appear in many blocks.

KeyB(vBERT) We call the model KeyB(vBERT) when the deep neural IR network is a BERT model. The most relevant blocks are concatenated together in their order of appearance in

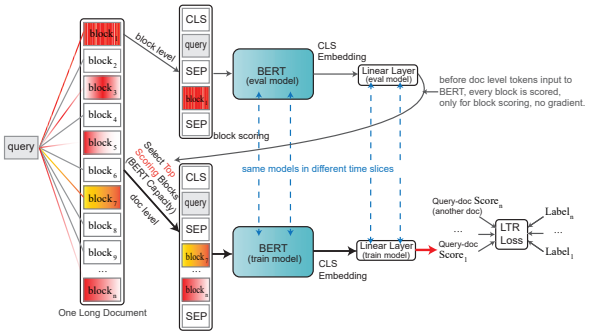

Figure 2: The architecture of KeyB(vBERT)$_{BinB}$.

the document and with the query to form the input of BERT. The number n of selected blocks depends on the capacity of BERT (512 tokens).

KeyB(PARADEk) PARADE [6] is a cutting-edge model that produces a query-document representation from query-passage representations. Denoting by p_i the i^{th} passage and p_i^{cls} the corresponding query-passage representation, one has: $p_i^{cls} = BERT(q, p_i)$. The query-passage representations are then aggregated to obtain the query-document representation. We propose here to select a fixed, small number of passages, denoted by k, to address PARADE model's high complexity for large numbers of passages and the potential issue of including noise signals. As shown in Figure 1, the selecting key block stage allows for efficiency and effectiveness.

3. Learning to select blocks

KeyB(vBERT)$_{BinB}$ We propose here a strategy that aims at exploiting BERT to compute the relevance score of a block. The overall architecture of the model proposed is depicted in Figure 2, in which the same BERT model and linear layer are used at different time slices, first to compute a query-block representation, from which ([CLS] embedding from BERT) the relevance score of the block is derived, and then to compute the query-document representation ([CLS] embedding) based on the top ranked blocks, finally to obtain the score of the document. This second part is identical to the KeyB(vBERT) model, the only difference lying in the way the blocks are selected. For the first part, both the BERT model and the linear layer are just utilized for scoring and are not trained (hence the phrase "eval model" used in Figure 2) which reduces the complexity.

Extend for late interaction approach (ICLI) Previous approaches are interaction based methods for long documents which is computation expensive. We propose to seek a solution for late interaction [11] based approach that can handle long documents. The late interaction based method pre-stores the contextualized tokens which are learned and interacts with query tokens. To be specific, each query token interacts with the document tokens and obtains the maximum similarity, then all query tokens' obtained similarities are summed as the final query-document relevance score. Despite its efficiency, ColBERT [11] cannot handle long documents.

We address this problem here through a BERT-based dense intra-ranking and contextualized late interaction (ICLI) with multi-task learning and the architecture is shown in Figure 3. Firstly a long document is also segmented into passages. Then each contextualized token in the passages

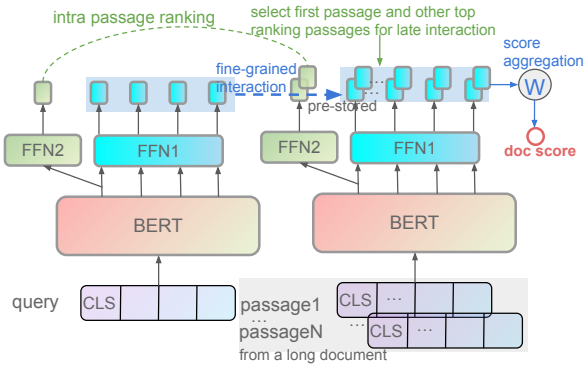

Figure 3: The architecture of ICLI-dot.

can be obtained by the BERT model. The tokens are passed to a one-layer feedforward neural network FFN_1 to obtain the compressed embeddings (dimension size 128). The query tokens and passage tokens are interacted as in ColBERT. A long document may contains many passages and plenty of potentially not relevant passages. Similar as above methods, we propose to select relevant passages before late interaction. In fact this can be done by BM25 and we call this ICLI-BM25. To deal with the potential issue of exact matching, we also want to include semantic matching in this approach. Inspired by the efficient dense retrieval, we want to take advantage of the [CLS] embeddings and use dot-product. To do so, the [CLS] embedding is also passed to a one-layer feedforward neural network FFN_2 to obtain the embedding (dimension size 128) for dense passage ranking. The first passage is always selected so that the [CLS] dot-product of the first passage can be compared with the document level label to obtain the loss $\mathcal{L}_1$ and train the model to generate good [CLS] embeddings. The document score is the aggregation of passage scores through a weighted sum, then another loss is obtained $\mathcal{L}_2$ for training good document embeddings. We use multi-task learning to train the model. As the two losses may have different scales, we combine them to obtain the final loss in a parameter learning way: $\mathcal{L} = \frac{1}{2\sigma_1^2}\mathcal{L}_1 + \frac{1}{2\sigma_2^2}\mathcal{L}_2 + \log(1+\sigma_1^2) + \log(1+\sigma_2^2)$, where σ_1 and σ_2 are two learning parameters for multi-task learning [12] which enforce positive regularization. During deployment, each document's contextualized tokens are pre-computed and stored for efficient late interaction.

4. Results and conclusion

We report here the results on the document reranking task of TREC 2019 Deep Learning Track [13].

We take other baseline results from QDS-Transformer [8] and implement PARADE-Transfomer [6] which is noted as PARADE and the proposed KeyB methods with pairwise hinge loss, each model is trained using Adam optimizer (the transformer layers are trained with a rate of 2e-5 while the linear layer with a rate of 1e-3) and each batch contains 2 positive-negative document pairs. Following [6], 16 passages are obtained for the original PARADE each with 225 tokens and stride size 200. For the variant of PARADE we have proposed, we have used BM25 to select the top 5 passages balancing effectiveness and efficiency. ColBERT is also implemented

Table 1
Experiment on TREC 2019 DL, comparison with baseline and sparse attention based models. Best results are in **bold**.

TREC Deep Learning Track Document Ranking		
Model	NDCG@10	MAP
Baseline models		
BM25	0.488	0.234
CO-PACRR [14]	0.550	0.231
RoBERTa (FirstP) [15, 5]	0.588	0.233
RoBERTa (MaxP) [15, 5]	0.630	0.246
PARADE [6]	0.655	0.280
ColBERT [11]	0.650	0.269
Sparse attention based models		
Longformer-QA [16]	0.627	0.255
QDS-Transformer [8]	0.667	0.278
Proposed select blocks models		
KeyB(vBERT)$_{BM25}$	0.678	0.277
KeyB(vBERT)$_{BinB}$	**0.707**	**0.281**
KeyB(PARADE5)$_{BM25}$	0.672	0.280
ICLI-BM25	0.681	0.270
ICLI-dot	0.705	0.277

and for the proposed ICLI methods, pairwise RankNet loss is used with a learning rate of 1e-5 and each batch contains 8 positive-negative pairs. The "BERT-Base, Uncased, L=12, H=768" pre-trained language model is used in all neural IR models based on BERT.

Results Experimental results are displayed in Table 1. The overall average results of KeyB(vBERT) models, particularly KeyB(vBERT)$_{BinB}$, which employs the BERT itself to choose blocks, exceed the baseline models by a large margin and achieve SOTA level effectiveness, with a score of 0.707 for NDCG@10. Similar to KeyB(vBERT) models, experimental results show the proposed KeyB(PARADE5)$_{BM25}$ is also effective. In terms of NDCG@10, the proposed approach with five passages outperforms the original PARADE with 16 passages. In terms of the suggested late interaction based approach ICLI, the results reveal that the proposed approaches outperform baseline models when extended for long document retrieval, with the exception of PARADE in terms of MAP. ICLI surpasses the original ColBERT approach, with ICLI-dot achieving 0.705 in terms of NDCG@10, which is 8.46 percent greater than the original ColBERT method.

Comparing with sparse attention based methods, it is shown that the proposed select blocks models obtains comparable or better results. They all outperform QDS-Transformer in terms of NDCG@10. KeyB(vBERT)$_{BinB}$ and KeyB(PARADE5)$_{BM25}$ obtain better results in terms of MAP, while others are slightly lower. It's worth mentioning that, unlike QDS-Transformer, our methods don't necessitate altering CUDA kernels.

In conclusion, the proposed KeyB models are interaction based models while ICLI models are efficient late-interaction models for long document retrieval. These results show that the proposed pre-ranking framework for IR is effective, and that using learning or semantic

matching for block selection has more potential. In the future, we will also seek such a solution for PARADE model.

Acknowledgments

This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003) and the Chinese Scholarship Council (CSC) grant No.201906960018.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.

[2] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018).

[3] R. Nogueira, K. Cho, Passage re-ranking with bert, arXiv preprint arXiv:1901.04085 (2019).

[4] S. MacAvaney, A. Yates, A. Cohan, N. Goharian, Cedr: Contextualized embeddings for document ranking, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1101–1104.

[5] Z. Dai, J. Callan, Deeper text understanding for ir with contextual neural language modeling, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 985–988.

[6] C. Li, A. Yates, S. MacAvaney, B. He, Y. Sun, Parade: Passage representation aggregation for document reranking, arXiv preprint arXiv:2008.09093 (2020).

[7] M. Li, E. Gaussier, Keybld: Selecting key blocks with local pre-ranking for long document information retrieval, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 2207–2211.

[8] J.-Y. Jiang, C. Xiong, C.-J. Lee, W. Wang, Long document ranking with query-directed sparse transformer, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 2020, pp. 4594–4605.

[9] M. Li, D. N. Popa, J. Chagnon, Y. G. Cinar, É. Gaussier, The power of selecting key blocks with local pre-ranking for long document information retrieval, ArXiv abs/2111.09852 (2021).

[10] H. C. Wu, R. W. Luk, K.-F. Wong, K. Kwok, A retrospective study of a hybrid document-context based retrieval model, Information processing & management 43 (2007) 1308–1331.

[11] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.

[12] L. Liebel, M. Körner, Auxiliary tasks in multi-task learning, arXiv preprint arXiv:1805.06334 (2018).

[13] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, E. M. Voorhees, Overview of the trec 2019 deep learning track, arXiv preprint arXiv:2003.07820 (2020).

[14] K. Hui, A. Yates, K. Berberich, G. de Melo, Pacrr: A position-aware neural ir model for

relevance matching, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1049–1058.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[16] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, 2020. arXiv:2004.05150.