Complex & Intelligent Systems (2024) 10:963–979
https://doi.org/10.1007/s40747-023-01192-3

ORIGINAL ARTICLE

An efficient long-text semantic retrieval approach via utilizing presentation learning on short-text

Junmei Wang[1,3] · **Jimmy X. Huang**[2] · **Jinhua Sheng**[1,3]

Received: 20 March 2023 / Accepted: 15 July 2023 / Published online: 14 August 2023

Abstract
Although the short-text retrieval model by BERT achieves significant performance improvement, research on the efficiency and performance of long-text retrieval still faces challenges. Therefore, this study proposes an efficient long-text retrieval model based on BERT (called LTR-BERT). This model achieves speed improvement while retaining most of the long-text retrieval performance. In particular, The LTR-BERT model is trained by using the relevance between short texts. Then, the long text is segmented and stored off-line. In the retrieval stage, only the coding of the query and the matching scores are calculated, which speeds up the retrieval. Moreover, a query expansion strategy is designed to enhance the representation of the original query and reserve the encoding region for the query. It is beneficial for learning missing information in the representation stage. The interaction mechanism without training parameters takes into account the local semantic details and the whole relevance to ensure the accuracy of retrieval and further shorten the response time. Experiments are carried out on MS MARCO Document Ranking dataset, which is specially designed for long-text retrieval. Compared with the interaction-focused semantic matching method by BERT-CLS, the MRR@10 values of the proposed LTR-BERT method are increased by 2.74%. Moreover, the number of documents processed per millisecond increased by 333 times.

Keywords Neural information retrieval · Long-text similarity · Pretrained language model · Efficiency

Introduction

In recent years, some large-scale pretrained language models such as embeddings from language models (called ELMo) [1] and bidirectional encoder representation from transformers (called BERT) [2]) have continuously updated results on many fields [3, 4]. These pretrained language models can be fine-tuned to estimate the semantic relevance between two texts. BERT is the most representative pretrained language model. After BERT's emergence, many BERT-based ranking models have achieved state-of-the-art results on various retrieval benchmarks within less than 1 year [5–8]. It is beneficial that the BERT-based ranking models calculate the semantic interaction to obtain the semantic matching degree between two texts; these models can thereby bridge the gap of common lexical mismatch between documents and queries [9, 10]. The input of the BERT retriever for interactive semantic matching is a join for query and each passage from the retrieved document (e.g., "[CLS] Query [SEP] Passage [SEP]"). The probability of relevance between each passage and query is obtained from [CLS]. Given the transformer-based architecture, BERT's memory and time consumption increase exponentially as the input length increases. Therefore, the architecture of the BERT model limits the input length of text to 512 words.

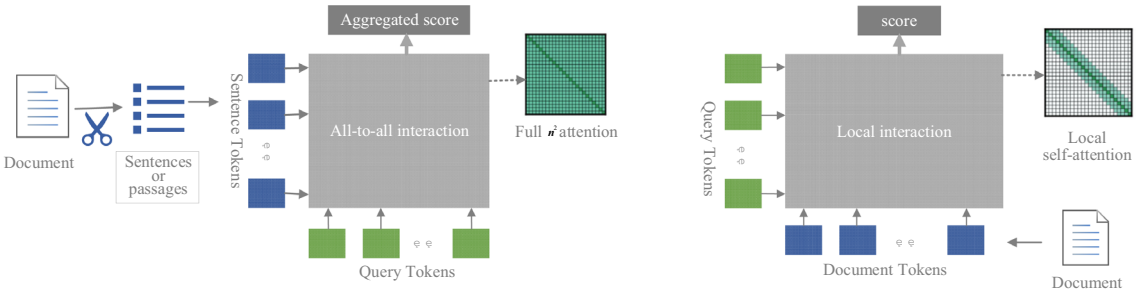
In document retrieval, the average lengths of texts in document collections greatly exceed the lengths of texts in passage collections [11]. Figure 1 shows query document matching paradigms in transformer-based information retrieval (IR). There are two mainstream long-text retrieval schemes. One is to divide the document into sentences or passages and interact with the query one by one, as shown in Fig. 1a. Another, which is similar to a sliding window, is to apply local self-attention mechanisms that interact with queries segment

✉ Junmei Wang
jmwang@hdu.edu.cn

[1] School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China

[2] Information Retrieval and Knowledge Management Research Lab, School of Information Technology, York University, Toronto, Canada

[3] Key Laboratory of Intelligent Image Analysis for Sensory and Cognitive Health, Ministry of Industry and Information Technology of China, Hangzhou 310018, China



(a) All-to-all interaction
(e.g., Birch, CEDR)

(b) Local interaction
(e.g., TKL, Longformer)

Fig. 1 Query document matching paradigms in transformer-based IR

by segment. This scheme can reduce model complexity and make it possible to input entire documents. However, the interaction is time-consuming and costly. In a real-world scenario, a user wait time of more than 100 ms for a query is not a good user experience [12].

Recently, some scholars have used representation learning via BERT to learn the representation of text in open-domain question—answer tasks and short-text information retrieval tasks [12, 13]. These scholars obtain semantic matching values between the query and sentences by delayed interaction [12, 13]. However, the study of interactive semantic matching still continues to impede long-text retrieval [8, 14, 15], and the improvement of retrieval results is at the cost of a considerable computing time. Therefore, this paper designs a compromise method considering retrieval performance and retrieval time. We propose an efficient long-text retrieval model based on BERT (called LTR-BERT). Our major contributions are listed as follows:

- The proposed LTR-BERT model preserves the vector semantic representation of documents in advance. This process needs to be calculated only once, so the cost is lower than that of interaction-based semantic matching.
- Aiming at the missing query semantics caused by the large difference between the query length and the document length, a query expansion strategy is designed to improve the semantic matching ability of a query and related documents.
- Inspired by exact term matching, the proposed LTR-BERT model uses a cheap interaction mechanism without training parameters. The interaction mechanism considers the fine-grained relevance of documents and queries while saving computing time during matching.

Related work

The extensive application of neural network in speech recognition, computer vision, natural language processing, pattern recognition and other fields has aroused great interest of researchers, and the study of neural network has become more in-depth [3, 4, 16, 17]. At this time, a number of neural ranking models have been emerging [18]. Unlike traditional information retrieval models, which consider exact matching only at the term level, neural ranking models can capture relevance between queries and documents at the semantic level. These neural IR models are divided into two main types: interaction-focused neural IR models and representation-focused neural IR models. The interaction-focused neural IR models use the embedded representation of queries and documents to directly model local contextual interactions. Before the advent of BERT models, interaction-based neural IR models dominated. After the advent of the BERT model, the performance of interaction-focused neural IR models has improved significantly [19]. However, its time inefficiency is prohibitive, especially for long-text retrieval tasks.

To address the issue of time efficiency in long-text retrieval, this work combines BERT's understanding of sentence context and the characteristics of the representation learning twin tower structure to apply BERT to a representation-focused approach. Several kinds of research work related to our work will be explained below. We will analyze the advantages and disadvantages of several existing interaction-focused neural IR models and representation-based neural IR models in "Interaction-focused neural IR model" and "Representation-focused neural IR model", respectively. Furthermore, since our main research object is long-text retrieval tasks, we will mention some studies on long-text inference models in "Long-text inference model". In the comparison experiment, to show the advantages of piecewise coding, we apply the proposed method to the long-

text inference model for retrieval. Finally, we introduce the work of existing long-text retrieval models and summarize their advantages and disadvantages in "Long-text retrieval model".

Interaction-focused neural IR model

The interaction-based neural IR method extracts meaningful matching patterns from words, phrases and sentences and generates matching scores. The method first establishes local interactions (i.e., local matching signals) between two texts and then uses deep neural networks to learn hierarchical interaction patterns for matching. Interaction-based neural ranking models include ARC-II [20], MatchPyramid [21], PACRR [22], CO-PACRR [23], K-NRM [24] and CONV-KNRM [25]. These models focus more on modeling-related signals than on sentence-level representation. Although interaction-focused neural IR models have achieved some improvement in retrieval performance, the achievements of these models have been overshadowed by the emergence of pretrained language models. Take BERT, a standard pretrained language model, for example. It has 12 transformer layers with 340 M parameters and has real bidirectional context encoding capability. In addition, training on a large set of unsupervised data enables understanding common texts.

Yilmaz et al. [8] first applied the BERT model to information retrieval tasks. This approach presents a challenge for long-text retrieval, which generally has a longer input length than BERT allows. The authors solved this problem by inferring sentences independently and then aggregating sentence scores to generate document scores. The experimental results show that the method was simple and effective. Later, Yilmaz et al. [19] demonstrated Birch. This model applies BERT to document retrieval by integrating BERT with the open-source Anserini toolkit [26] to demonstrate end-to-end search on large document collections. Birch adopts a simple ranking model, and the researchers replicate the state-of-the-art document ranking results proposed by Yilmaz et al. [8].

Although the availability of massive datasets and computing power has enabled data-driven deep neural network approaches to significantly affect research in information retrieval, the computational time for each instant query individually to interact deeply with the document is still prohibitively long.

Representation-focused neural IR model

Before the advent of representation learning, researchers usually needed to manually annotate features or manually design rules from the domain knowledge of raw data to construct features and then deploy these features into relevant machine learning algorithms. Although effective for machine learning, this approach is difficult, expensive and time-consuming and relies on strong expertise. Representation learning makes up for these shortcomings by enabling machines not only to learn features of the data but also to use these features to accomplish specific tasks. The neural IR model based on representation learning first learns the representation of the query and document separately and then computes the semantic similarity between the query and the document by using a simple interactive method.

In 2013, Huang et al. [27] proposed the first deep structured semantic model (DSSM) based on a representation learning framework. In this model, one-hot sparse word vectors are converted into dense vector representations by the word hashing technique. However, it cannot represent the context information of the word. In 2014, Shen and He et al. [28, 29] proposed a new convolutional latent semantic model (named CLSM or CDSSM). CDSSM is an extension of DSSM. When capturing the context information, convolutional neural network is used to better preserve local word order information, and then a max pooling strategy is used to filter semantic concepts to form sentence-level representations. The advantage of CDSSM over the DSSM model is that the CDSSM can make up for the lack of contextual information in DSSM and convert variable-length text information into vectors of the same length. However, DSSM and CDSSM consider only semantic matching between queries and documents [30]. In 2015, Hu and Lu et al. [20] proposed two related convolutional architectures, namely, Architecture-I (ARC-I) and Architect-II (ARC-II), to semantically match two sentences. ARC-I is a neural network model based on representation learning. The difference between the ARC-I model and the DSSM model is that the ARC-I model can extract n-gram information of words by using convolutional layers and express the word order information of sentences by using a layer-by-layer combination. However, the ARC-I model has a nonnegligible disadvantage in that it delays the interaction between two sentences (in the final MLP) until their respective representations mature (in the convolution model); consequently, details of representing sentences may be lost in matching tasks. In 2018, Zamani and Dehghani et al. [31] proposed a self-contained neural ranking model (SNRM). The model learns a latent sparse representation for each query and document by introducing sparsity features. This representation constrains the semantic relationship between queries and documents, but is sparse enough to construct an inverted index for the entire collection. The authors generate a retrieval model that is as efficient as the traditional term-based models by using the sparsity of the parameterized model. Without losing the validity of the model, the efficiency of the model is improved.

In 2020, Khattab et al. [12] proposed an efficient contextualized late interaction over BERT (ColBERT for short-text

retrieval). The method uses an off-line BERT to encode the context of the paragraph and an online BERT to encode the query separately and then uses a delayed interaction method to obtain the relevance score between the query and the passage. ColBERT has a competitive advantage with existing BERT-based models in terms of the accuracy of retrieval results. More importantly, it turned out to be a more efficient model. Additionally, Nie et al. [13] proposed a new decoupled contextual encoding framework with dual BERT models (DC-BERT). It pre-encodes all documents by using off-line BERT and saves their encoding only once. The query is then encoded using online BERT. Both methods employ delayed interaction to address the long interaction time, but only for passage retrieval and question and answer (QA) tasks. These methods lack research on long-text retrieval, such as documents.

Long-text inference model

The limitation of the maximum input length of BERT reminds us that the capacity of human working memory is limited. How do humans effectively perceive long texts? In recent years, some scholars have conducted in-depth research on long-text reasoning. Beltagy and Peters et al. [32] propose a long-text pretraining model called Longformer. The model processes sequences linearly, making it easier to process documents containing sequences of thousands of words or longer. Subsequently, Ding et al. proposed a general BERT model to cognize long texts (CogLTX) [33]. The authors train a judgment model to identify key sentences and concatenate them for reasoning. The CogLTX model performs best in downstream tasks, such as reading comprehension, question answering and text reasoning. However, these long-text inference methods have not been applied in document information retrieval tasks, because for long text, part of the content in the document is relevant to the query; that is, the document is considered relevant to the query. Modeling long-distance dependence between paragraphs and the semantic information of the entire document ignores the information that is truly relevant to the query, resulting in errors in the calculated semantic relevance between document topics and query topics.

Long-text retrieval model

At present, the research on long-text retrieval includes the following main aspects: (1) Nogueira et al. [14] consider the interactive semantic matching between the top 512 terms of the document and the query, ignoring the relevance between the query and the rest of the document. (2) Yilmaz et al. [8] applied the relevance estimator based on passage-level BERT to long-text ranking. Specifically, the authors first split the document into fragments and then interact with the query segment by segment. Then, the semantic matching value of the paragraph with the highest interactive matching value or the sum of the semantic matching values of multiple paragraphs is used as the semantic matching value of the document. Finally, the documents are ranked in decreasing order of their semantic matching values. (3) Hofstatter et al. [15] propose a transformer–kernel pooling model for long text (TKL). The model adopts a local self-attention method that uses a fixed-size moving window to move over the document, allowing long-text input by reducing the computational complexity of the self-attention mechanism. The TKL model also interacts directly with the query. When the number of documents is large and the length of documents is long, the time and memory overhead of contextual semantic matching between the query and the full text is high.

These studies share some common features: these studies all use interaction-focused semantic matching for long-text retrieval. For each query, the interaction with each long-text and even each paragraph needs to be calculated; such calculations are time-consuming and expensive. However, in real scenarios, if the user waits more than 100 ms for a query, the user experience will suffer [12]. In view of this, this work explores a more efficient long-text retrieval model by combining the characteristics of the two-tower structure and the advantages of the BERT model for contextual representation. Inspired by the research of Zhuang et al. [34] and Zhou et al. [35], it is very necessary to study the stability and experimental parameters of the system. We conducted an ablation study on the model and studied the influence of document input length, query expansion and other factors on the model results.

Representation-focused long-text retrieval method

An overview of the architecture

Figure 2 shows the architecture of the proposed LTR-BERT model, which consists of three parts: (1) off-line BERT for long-text representation; (2) online BERT for query representation; and (3) an interaction mechanism without training parameters. Overall, the execution process of the model is as follows: first, the contextual semantic representation of the long text is obtained by using BERT. The representation of the long text is stored off-line. This process includes a compression layer, which is also designed to save storage space. It needs to be calculated and merged only once. Second, for ad hoc queries, online BERT is used to obtain a representation of the query, which is usually short, so the time cost of this process is not very high. Additionally, aiming at the matching problem caused by the large difference between the

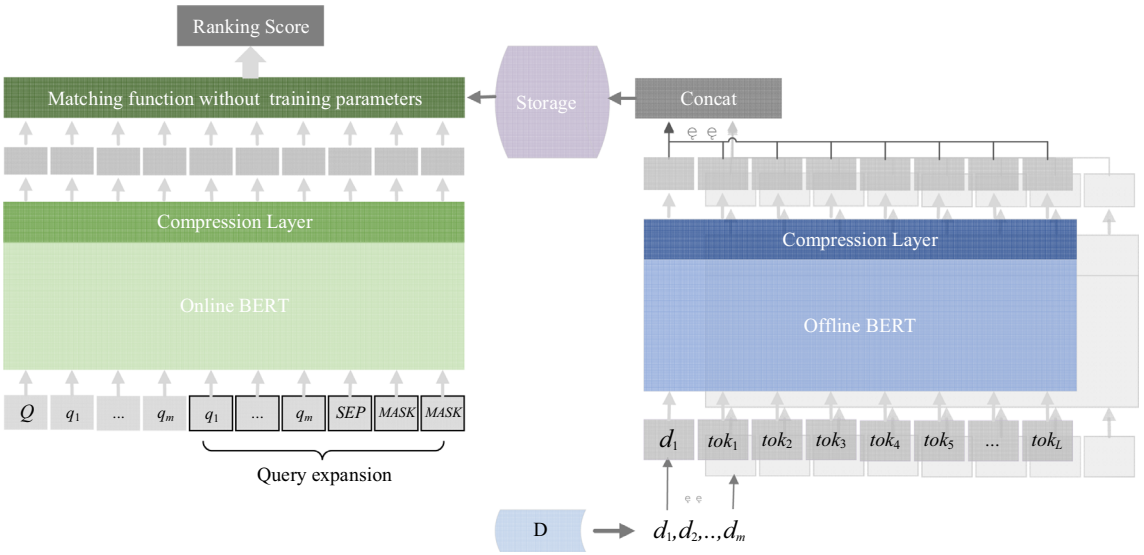


Fig. 2 Structure diagram of LTR-BERT

length of the query and long text, a query expansion strategy is designed to improve the semantic matching ability between the query and the long text. Finally, to further reduce the computational time of semantic matching, a matching mechanism without training parameters is proposed. We will describe the design of these three parts in detail in later chapters of this paper.

Query representation via online BERT

The input of BERT for representing a single query is shown in Fig. 2. Specifically, for a query Q: "Buy apples on iPhone", the query is split into "Buy", "apples", "on" and "iPhone", and these query terms are recorded as q_1, q_2, ..., q_m. The starting position of the input is marked with [Q]. The user's queries are based on the user's store of real-world knowledge and understanding, which the search engine does not have. Therefore, the input query has difficulty expressing the user's real intention clearly due to the lack of information. In addition, the query length is usually much shorter than the document length. To address these questions, we design a query expansion strategy. Query expansion has been shown to be an effective strategy to better find relevant documents by complementing the missing information in the query [36]. The specific processing of our query expansion strategy is: the maximum length of the query is set as L_Q. For queries whose length is less than L_Q, the query expansion strategy is adopted, the query terms are input repeatedly once and [SEP] is used to indicate the end position. The remaining positions less than L_Q are filled with a special marker [MASK] until the length reaches L_Q. The strategy's goal is to preserve the original query semantics while enabling the model to learn the ability to query missing information during training. Simultaneously, we enhance the input of the original query to avoid query topic drift caused by the semantics of the supplementary query.

For the output of online BERT, we use a linear layer for compression. The specific role of this layer is described in "Long-text representation via off-line BERT". The representation of query Q is denoted as E_Q, and the size of E_Q is $L_Q \times$ dim, where dim is the compressed dimension of each word embedding (the dimension of each word embedding is 768 before compression).

$$E_Q = \begin{cases} \text{Linear(BERT([}Q\text{]}q_0, ..., q_m, q_0, ..., q_m, \text{[sep], [mask], ..., [mask]))} & \text{len}(Q) < L_Q \\ \text{Linear(BERT([}Q\text{]}q_0, ..., q_m,))} & \text{len}(Q) \geq L_Q . \end{cases} \quad (1)$$

Long-text representation via off-line BERT

Due to BERT's input length limitation, the entire content of the document cannot be fed into the model at once. The size of the shard is set to L_d, and the document is segmented if the length of the document exceeds L_d. For a document up to L_d in length, the special marker [mask] is used for padding. If a document is longer than L_d, the document is divided into a series of paragraphs of length L_d; these paragraphs are denoted as $d = \{p_1, p_2, ..., p_k\}$. p_i represents the ith segment after segmentation, where $p_i \in d, i \in \{1, 2, .., k\}$. $\text{tok}_{i,j}$ represents the word embedding of the jth token in the ith segment of document d. For the beginning of each input paragraph, we use the [D] marker. Although off-line computation can reduce the running time when a new query is entered, the off-line storage approach can be expensive due to

the large storage requirements for embedding representations of long text (for example, 768 floating point values per token in BERT-Base). Inspired by MacAvaney et al. [37], the proposed LTR-BERT model solves this problem by compressing and storing the representation of documents to reduce the storage space. We use a simple linear layer and a regularization process to compress the representation of the document. This approach not only fits well with the transformer networks, but also reduces the dimensionality of representing each word embedding without costing too much. The length of the output vector of each token by BERT is 768 dimensions, and direct off-line storage requires considerable space. The linear layer is the simplest layer of the neural network to reduce the number of dimensions. The linear layer can easily reduce the dimension of the vector and reduce the storage overhead. Such reductions just require a matrix multiplication and an addition. Finally, we obtain the compressed and concatenated representation of BERT's last hidden layer, which is stored in each document according to the document number. The representation process of documents is shown in Eq. (2).

$$E_d = \underset{tok_{i,j} \in p_i,\, p_i \in d}{concat} \begin{matrix} \left(\text{Linear}\left(\text{BERT}\left([D],\ \text{tok}_{i,1},\ ...,\ \text{tok}_{i,L_d}\right)\right)\right) \\ \text{len}(d) \geq L_d. \end{matrix} \tag{2}$$

The representation of document d obtained by off-line BERT is denoted as E_d. The size of the first dimension of E_d is $k \times L_d$, which is the total length of k paragraphs from document d. The second dimension of E_d is dim, which represents the compressed dimension of each word embedding (the dimension of each word embedding is 768 before compression). Similarly, when len(d) of a document does not exceed L_d, [mask] is used to pad the length until the length is L_d. The representation process of documents is shown in Eq. (3). In Eq. (3), tok_i represents the embedded representation of the ith word in the document.

$$\begin{aligned} E_d = \underset{tok_i \in d}{concat} \Big(&\text{Linear}\Big(\text{BERT}\big([\text{D}],\ \text{tok}_1,\ ...,\ \text{tok}_{\text{len}(d)},\\ &[\text{sep}],\ [\text{mask}],\ ...,\ [\text{mask}]\big)\Big)\Big) \text{len}(d) < L_d. \end{aligned} \tag{3}$$

Matching mechanism without training parameters

Traditional relevance matching methods can quickly find the documents that may be relevant to the query by exact matching based on query keywords. This approach is still widely used in industry today. Inspired by the idea of exact term matching, each word embedding already contains contextual information via BERT-based representation learning. Therefore, we use the average of several word embeddings most relevant to the query as the representation of paragraphs to match the representation of the query, as show in Fig. 3. Specifically, we search for word embeddings with the closest cosine similarity to each word embedding of query, take the average of these word embeddings as the representation of the passage p and take the average of the query embeddings as the representation of the query.

The matching score between a query and a single paragraph is shown in Eq. (4). This interaction mechanism without training parameters can reduce the time spent matching the semantic representation of the document and query.

$$\text{Score}(Q,\ p) = \text{cosine}\left(\frac{1}{|L_Q|}\sum_{e_m \in Bt} e_m,\ \frac{1}{|L_Q|}\sum\nolimits_{e_j \in Q} e_j\right),\ p \in d. \tag{4}$$

In Eq. (4), $Bt = \{e_1, ..., e_m, ..., e_{|L_Q|}\}$ represents the set of paragraph word embeddings closest to the query embedding, which is a subset of document word embeddings. e_q represents the embedding representation of word q in query Q, and e_m represents the word item in document d with the largest cosine similarity to the word embedding of query word item e_q. For the final score of document d, the maximum score of the query and each fragment of the document is taken as the semantic relevance score of the document, and the calculation method is shown in Eq. (5). In Eq. (5), $d = \{p_1,\ p_2,\ ...,\ p_k\}$, the fragment $p_i \in d, i \in \{1,\ 2,\ ..,\ k\}$.

$$\text{Score}(Q,\ d) = \underset{p_i \in d}{max}\text{Score}(Q,\ p_i). \tag{5}$$

Model training

For the proposed LTR-BERT model, we fine-tune the BERT encoder by using the Adam optimizer and train additional parameters from scratch (i.e., the compression layer and the embedding of [Q] and [D] markers). The interaction mechanism of this model has no trainable parameters. The training samples for our model are in the form of triples$< Q$, Passage$^+$, Passage$^-$ $>$. In the triple, Passage$^+$ is the positive passage of the query Q, while Passage$^-$ is the passage ranked in the top 100 passages but not marked as relevant.

$$\begin{aligned} \text{Loss} = -\sum_{(Q,\,p)} &(y(Q,\ p)\log(\text{Score}(Q,\ p)) \\ &+(1 - y(Q,\ p))\log(1 - \text{Score}(Q,\ p))), \end{aligned} \tag{6}$$

where $y(Q,\ p)$ is the relevance of passage and query and Score($Q,\ p$) is calculated as shown in Eq. (4).

Combining relevance matching and semantic matching

This work also studies the effect of the LTR-BERT model and the BM25 model on the final results. Equation (7) is

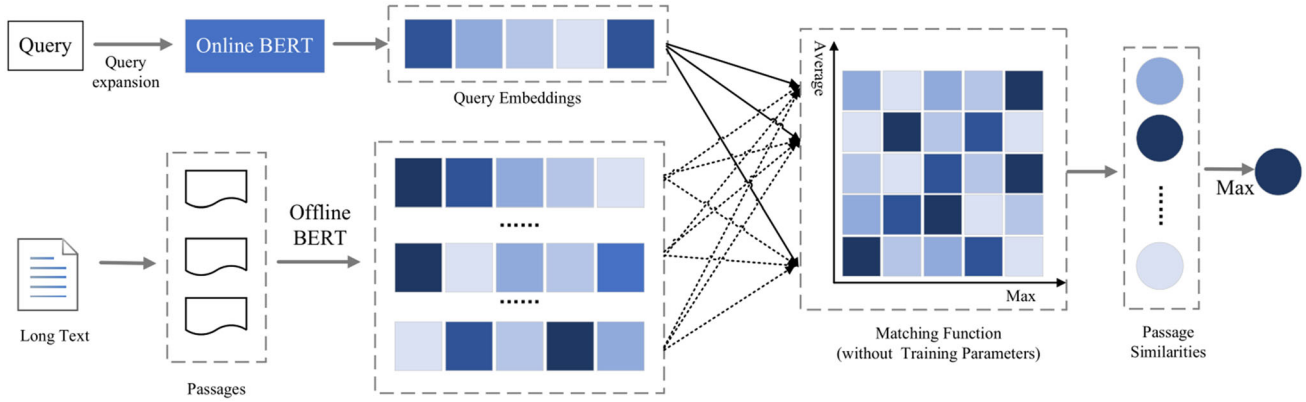


Fig. 3 Matching mechanism without training parameters

a combination of relevance matching and semantic matching. Score(Q, d) represents the final matching value of document d and query Q; this score combines the contributions of relevance matching and semantic matching based on presentation learning. After normalizing the two contributions with max–min, a linear regulatory factor α, $\alpha \in \{0, 0.1, 0.2, \ldots, 1\}$, is introduced to study the effect of α.

$$\text{Score}(Q, d) = \alpha \times S_r + (1 - \alpha) \times S_b, \tag{7}$$

where S_b is the semantic matching value based on representation learning and refers to the semantic matching value between the query and document calculated by LTR-BERT, as shown in Formula (5); S_r refers to the relevance matching value obtained by the Anserini tool[1] based on BM25. The calculation method is shown in Eq. (8).

$$\text{Score}(Q, d) = \sum_{q \in Q} \frac{(k_1 + 1) \times \text{TF}}{k_1 + \text{TF}} \times \frac{(k_3 + 1) \times \text{qtf}}{k_3 + \text{qtf}} \times \text{IDF}(t), \tag{8}$$

where $\text{TF} = \text{tf}/((1 - b) + b \times dl/\text{avdl})$ is the regularized document length; avdl is the average length of the document; k_1 and k_3 are constants; qtf is the frequency of the query term q; and b is the regulating factor, which balances the effect of the document length dl. The inverse document frequency IDF(t) is used mainly to measure the importance of a term t in the document set. $\text{IDF}(t) = \log\left(\left(N' - df_t + 0.5\right)/\left(df_t + 0.5\right)\right)$, where N' represents the number of all documents in the index and df_t represents the number of documents in which the term t appears.

In general, combining relevance matching and semantic matching based on representation learning for information retrieval can be divided into three stages: in the first stage, the LTR-BERT model is trained. In the second stage, the semantic matching of documents and queries via the LTR-BERT model is evaluated. In the third stage, the documents are reranked by combining relevance matching and semantic matching.

Experimental data and parameter setting

Experimental data and evaluation

The machine reading comprehension (MS MARCO) dataset was designed to benchmark large-scale deep learning models [38]. The document set and the queries are derived from real user search scenarios. Therefore, MS MARCO differs from other known publicly available datasets for machine reading comprehension and QA. MS MARCO contains 8,841,823 passages extracted from 3,563,535 web documents retrieved by Bing. In 2019, "Deep Learning Tracking Task[2]" was published at the Text Retrieval Conference (TREC). This task uses the MS MARCO dataset. The task is divided into two subtasks: the document ranking task and paragraph ranking task. We use relevance labels containing 3.2 million documents and their corresponding query document pairs. The number of queries in the training set is 367,013, the number of queries in the development set is 5193 and the number of queries in the test set is 200. For each query, the labels relevant to it are transferred from the MS MARCO passage ranking task by mapping a positive passage to the document containing the relevant passage.

Figure 4 shows that long documents outnumber short ones. The number of documents ranging from 0 to 500 terms accounts for 43.47%; therefore, more than half of the documents cannot be entered into the BERT model at one time. Therefore, the approach similar to Nogueira et al.'s [14] approach, which considers the interaction between only the

[1] https://github.com/castorini/anserini.

[2] https://microsoft.github.io/msmarco/TREC-Deep-Learning.

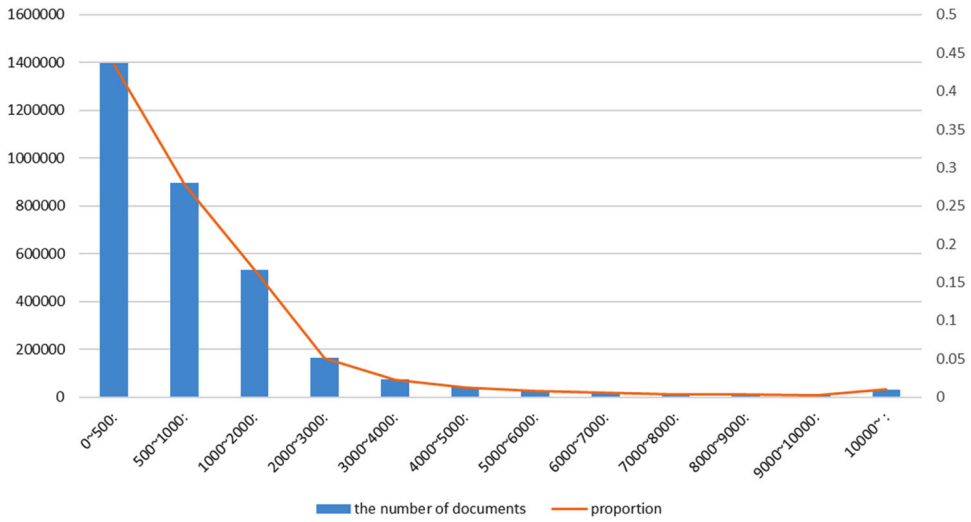


Fig. 4 Document length statistics on the MS MARCO dataset

first approximately 500 terms of the document and the query, is unfair to most long documents, because if the query-related content appears at the end of the document important information will be ignored, and the relevance of the document will not be properly evaluated.

The development set in the 2019 TREC deep learning tracking task contains binary judgment results for 5193 queries, where each query has only one relevant document. The main evaluation metric officially recommended is the mean reciprocal rank at position 10 (MRR@10). In addition, the average accuracy of the top 100 documents (MAP@100) is also used.

The 2019 TREC deep learning tracking task test query set has 200 queries, including 43 queries with relevance labels, with an average of 153 judgments per query. The level of judgment has four scales: "irrelevance" is indicated by 0; "relevance" is indicated by 1; "high relevance" is indicated by 2; and "perfect relevance" is indicated by 3. Therefore, the main average metric recommended by the test query set is normalized discounted cumulative gain at position 10 (nDCG@10). To unify the evaluation indicators, all comparison models use these three indicators for evaluation.

Compare models and parameter settings

With the application and development of deep neural networks in image processing, machine vision and natural language processing, many neural information retrieval models have emerged in information retrieval tasks (these models include MatchPyramid [21], PACRR [22], CO-PACRR [23], K-NRM [24], CONV-KNRM [25] and TKL [15]). These models differ from traditional information retrieval methods in that they capture relevance between queries and documents at the semantic level, not just exact matching at the term level. In addition, BM25 was also used as a representative of the relevance matching model to show the level of the baseline model for easy comparison with other models. With the development of pretrained language models, interactive semantic matching methods represented by BERT have been widely used and have achieved better results. Therefore, we also used BERT-base [CLS] [14] and bm25_marcomb [8] for comparison.

Based on the above considerations, the comparative models used in this experiment include BM25 (tuned Anserini) [26], MatchPyramid [21], PACRR [22], CO-PACRR [23], K-NRM [24], CONV-K-NRM [25], BERT-base[CLS] [14], bm25_marcomb [8], ColBERT [12], TKL [15], DRSCM [39] and LTR-Longformer. MatchPyramid, PACRR, CO-PACRR, K-NRM, CONV-KNRM, BM25_Marcomb and TKL are interactive neural information retrieval models, and LTR-Longformer adopts a representation learning-based method. The BERT model was replaced by the Longformer model [32]. BM25 (tuned Anserini) uses the results of the top 100 documents per query run by Anserini [26]. BM25 is optimized. k_1 is 4.46, and b is 0.82. Other models rerank this result. The exception is bm25_marcomb (the result of a successful TREC run in 2019), using a stronger first-round retrieval model. All the models except bm25_marcomb rerun on the same hardware and environment; these models include the MatchPyramid, PACRR, CO-PACRR, K-NRM and CONV-KNRM models, with a maximum document length of 500. TKL, LTR-Longformer and LTR-BERT are specially designed for long-document information retrieval tasks and can process longer text. In our experiment, the lengths of documents are set to 1000 or 2000, and the length of queries is set as 50. The compared experiment in this section uses the same server equipped with an NVIDIA RTX 8000 GPU graphics card.

Table 1 Results of the proposed LTR-BERT model and several related models on queries with binary label

Model	2019 TREC deep learning track dev—binary relevance labels				
	Max doc length	nDCG@10	MRR@10	MAP@100	Average docs./ms
BM25 (tuned)	–	0.3251	0.2646	0.2769	–
MatchPyramid	500	0.3364	0.2716	0.2789	27
PACRR	500	0.3344	0.2729	0.2816	22
CO-PACRR	500	0.3436	0.2824	0.2819	14
K-NRM	500	0.3214	0.2609	0.2638	**49**
CONV-KNRM	500	0.3424	0.2836	0.2866	10
BERT-base [CLS]	500	0.4165	0.3535	0.3594	0.1
ColBERT	500	0.4057	0.3425	0.3498	31.3
TKL	1000	0.3758	0.3125	0.3213	1.5
TKL	2000	0.3396	0.2790	0.2892	1.1
DRSCM (2 sum)	1000	0.4216	0.3433	0.3567	10.0
DRSCM (4 sum)	2000	0.4178	0.3325	0.3425	10.0
LTR-Longformer	1000	0.3347	0.2658	0.2734	26
LTR-Longformer	2000	0.3364	0.2712	0.2776	26
LTR-BERT	1000	**0.4338**	**0.3632**	**0.3701**	33.3
LTR-BERT	2000	0.4289	0.3574	0.3647	33.3

"Average docs/ms" indicates the average number of documents processed per millisecond during retrieval. The bold font indicates the optimal result under the corresponding metric

Experimental results and analysis

Our experiment answers the following main research questions. Additionally, each subsection of "Experimental Results and Analysis" analyzes and answers a research question.

RQ1: For long-text retrieval tasks, can LTR-BERT simultaneously ensure the highest retrieval performance and improve efficiency ("Study on semantic matching for long text")?

RQ2: How well does the LTR-BERT model perform on the more fine-grained relevance discrimination task ("Study on fine-grained semantic matching for long text")?

RQ3: How does each LTR-BERT component (e.g., interaction mechanism and query expansion) contribute to the model's results ("Ablation study")?

RQ4: How does the model perform on different types of long-text datasets ("Study on different types of datasets")?

RQ5: What is the indexing cost of LTR-BERT in terms of documents' off-line storage and memory overhead ("Indexing throughput and footprint")?

RQ6: How much does BM25 affect the first round of sorting results ("Sensitivity analysis of parameter α")?

Study on semantic matching for long text

In this section, we mainly test the results of the proposed model on binary labels. The 2019 TREC deep learning tracking development set contains binary judgment results on 5193 queries, where each query has only one relevant document. The major evaluation metric is the MRR@10. Table 1 compares the proposed LTR-BERT model with several other baselines on the TREC deep learning document task.

We compare the LTR-BERT with MatchPyramid [21], PACRR [22], CO-PACRR [23], K-NRM [24], CONV-KNRM [25], BERT-base [CLS] [14] and ColBERT [12]. Experimental results show that on documents of 1000–2000 words, the LTR-BERT model is more effective than the BERT-based model and other neural network ranking models that consider only 500 words. Notably, LTR-BERT increases documents' length of processing without increasing GPU cost.

The TKL model is a neural ranking model designed for document information retrieval [15]. The comparison of the retrieval results of LTR-BERT and TKL shows that the retrieval time of the LTR-BERT model is more than 30 times shorter than that of the TKL model. When the document length is 2000, the results of the TKL-BERT model on MRR@10, nDCG@10 and MAP are significantly lower compared to the results when the document length is 1000, because although the TKL model allows processing long text,

the TKL model performs interaction-based semantic matching, and each query needs to interact with each passage in the document. Although the local self-attention mechanism can save time, the mechanism still needs to wait for every ad hoc query that interacts with all documents. However, the LTR-BERT model calculates the representation of the documents off-line and stores the representation in advance so that the representation of the documents is calculated only once. When a new query arises, only the representation of the short query needs to be calculated. Since the matching mechanism is simple enough, ranked documents can be obtained quickly. Therefore, the long-text retrieval method based on presentation learning is more efficient than the retrieval method based on interaction.

A comparison of the two proposed retrieval models according to representation learning (LTR-BERT and LTR-Longformer) shows that the retrieval results of the LTR-Longformer differ from those of the LTR-BERT model, because Longformer models can handle longer text than BERT models can, but long-text inference models are not necessarily bad at handling long-text information retrieval tasks. In document information retrieval tasks, the part of the content related to the query is regarded as the document related to the query. However, the LTR-Longformer model considers the information of the whole document, thereby possibly reducing the importance of the information relevant to the query. Consequently, it is difficult for the LTR-Longformer model to distinguish the truly query-relevant parts of the long text. Based on the above comparison results, the proposed LTR-BERT model ensures the effectiveness and efficiency of the retrieval result. Wang et al. [39] proposed a DRSCM model, which uses a linear combination of the segment correlation score and segment correlation matrix to obtain the final document score. The experimental results show that compared with the latest DRSCM model, the proposed LTR-BERT model still has advantages in long-text retrieval, not only in efficiency but also in performance (nDCG@10, MRR@10, and MAP@100). It shows that our fine-grained interaction can not only obtain fine-grained similarity information, but also save time in the interaction stage.

Study on fine-grained semantic matching for long text

In "Study on semantic matching for long text", we test the retrieval performance of the proposed LTR-BERT model and the comparison models on 5193 binary labeled queries. In this section, we test the retrieval performance of the proposed LTR-BERT model and comparison models on continuous relevance labeled queries. Different levels of relevance labels are used to evaluate the model's results in terms of the nDCG metrics. Therefore, the main evaluation for queries with continuous relevance judgment is nDCG@10. To be consistent with the evaluation in Table 1, the experimental results in this section also show the results of the MRR@10 and the MAP.

In Table 2, the results of the bm25_marcomb model are the results of a successful run on the 2019 TREC deep learning track [8]. The bm25_marcomb model also uses the BERT model for long-text retrieval. The difference is that the bm25_marcomb model divides the document into passage chunks, generates scores for each passage and combines these scores into a document score. Therefore, the results obtained by this model can also be compared with those obtained by the LTR-BERT model and the TKL model in terms of nDCG@10. In terms of MRR@10 and nDCG@10, the results of the LTR-BERT model are higher than those of the bm25_marcob and TKL models. However, the bm25_marcob model has a higher MAP, but this result is due to a stronger first-round retrieval. In other words, all models except bm25_marcob rerank the top 100 documents provided by the track organizer; bm25_marcob reranks the full documents using a stronger first-round retrieval model. As the experimental results in Table 2 show, the proposed LTR-BERT model significantly improves the nDCG@10 obtained by other neural ranking models. The results of the four evaluation indices show that the proposed LTR-BERT model has the best comprehensive performance. Therefore, long-text retrieval approach utilizing presentation learning on short text is not only effective, but also efficient.

Furthermore, to compare the results of each query more clearly, Fig. 5 shows the MAP results for 43 queries by the BM25 model, TKL model and the proposed LTR-BERT model. Comparing the MAP results of the three models shows that for most queries, the MAP results of the LTR-BERT model are higher than those of the BM25 and TKL models. The title of the query in number 39 is "What is theraderm used for". In the annotation results, five documents (D2536093, D3494217, D3494218, D3494220 and D3494221) are labeled as relevant, where documents D3494217 and D3494218 are labeled as perfectly relevant. Figure 5 shows the MAP values of different models for this query: the result of TKL is 0.8909, the result of BM25 is 1 and the result of the proposed LTR-BERT model is 1. As the statistical analysis reveals, the word "theraderm" appeared in a few documents marked as relevant. In addition, the word "theraderm" is not ambiguous, so the analysis shows that the relevant documents can be found for query No. 39 only by using the keyword-based exact matching method (e.g., BM25). In long-text retrieval tasks, a document often has multiple topics. In general, if part of the content of a document is related to the query, users think that the relevant document has been found. However, the TKL models the entire document, focusing too much on the long-distance dependence of the document, and ignores the topic that should be considered. Of course, this result is only a relatively rare case.

Table 2 Results of the proposed LTR-BERT model and several neural ranking models on queries with continuous relevance labels

Model	2019 TREC deep learning track test—continuous relevance labels				
	Max doc length	nDCG@10	MRR@10	MAP@100	Average docs/ms
BM25 (tuned)	–	0.5234	0.8632	0.2339	
MatchPyramid	500	0.5741	0.9011	0.2324	27
PACRR	500	0.5960	0.8591	0.2183	22
CO-PACRR	500	0.5349	0.8845	0.2231	14
K-NRM	500	0.4936	0.7631	0.2124	**49**
CONV-KNRM	500	0.5465	0.8993	0.2341	10
BERT-base [CLS]	500	0.6512	**0.9436**	0.2613	0.1
bm25_marcomb	–	0.640	0.913	**0.323**	< 0.1
ColBERT	500	0.6439	0.9279	0.2610	31.3
TKL	1000	0.5284	0.910	0.2278	1.5
TKL	2000	0.5475	0.915	0.2351	1.1
DRSCM (2 sum)	1000	0.6434	0.9193	0.2531	10.0
DRSCM (4 sum)	2000	0.6375	0.9108	0.2452	10.0
LTR-Longformer	1000	0.5366	0.9085	0.2286	26
LTR-Longformer	2000	0.5424	0.9128	0.2347	26
LTR-BERT	1000	**0.6674**	0.9341	0.2711	33.3
LTR-BERT	2000	0.6666	0.9341	0.2734	33.3

"Average docs/ms" indicate the average number of documents processed per millisecond during retrieval. The bold font indicates the optimal result under the corresponding metric

Overall, when compared with the latest DRSCM model, the proposed LTR-BERT model still has advantages in long-text retrieval, not only in efficiency but also in performance (nDCG@10, MRR@10, and MAP@100). This is because the LTR-BERT model divides the document into passages, matches the representation of each passage with the query and takes the content most relevant to the current query as being relevant to the whole document. Therefore, as long as the LTR-BERT model finds that the most relevant content is relevant to the query, the content can be identified as the relevant document. Therefore, the LTR-BERT model is more suitable than the TKL model for long-text retrieval for whole document modeling.

Ablation study

In the context of deep neural networks, ablation studies can be used to remove parts of the network structure to better understand the source of validity. This work studies two sources (query expansion and the matching method).

The results of the ablation study on the query expansion of the LTR-BERT model and the effectiveness of the matching method are shown in Fig. 6. "[A] LTR-BERT-Average Similarity" indicates changing the matching mechanism of LTR-BERT to the average cosine similarity of queries and all word embedding of documents. "[B] LTR-BERT-[CLS]-cosine" means that the CLS in LTR-BERT is used as the relevance score between the query and document, and the cosine is used to express the similarity of the text. "[D]" is the proposed LTR-BERT model. Comparing [A], [B] and [D] shows that the cosine similarity of the matching mechanism changed to calculate the average value of word embedding or that the cosine similarity of the CLS vector is not as good as that of the proposed matching method. Therefore, it is effective for the matching method to take the average of the word embeddings most similar to the query as the representation of the passage.

Then, we study whether the query expansion has a gain effect on the proposed LTR-BERT model. "[C] LTR-BERT w/o query expansion" indicates that the LTR-BERT model does not use query expansion, and the query length is still fixed at 30. When the length is insufficient, the random mask is set to zero. [C] and [D] show that the query extension can help users find more relevant documents. The results show that our query expansion strategy is more effective because it can automatically learn and supplement the missing information during training, which helps to find more relevant documents.

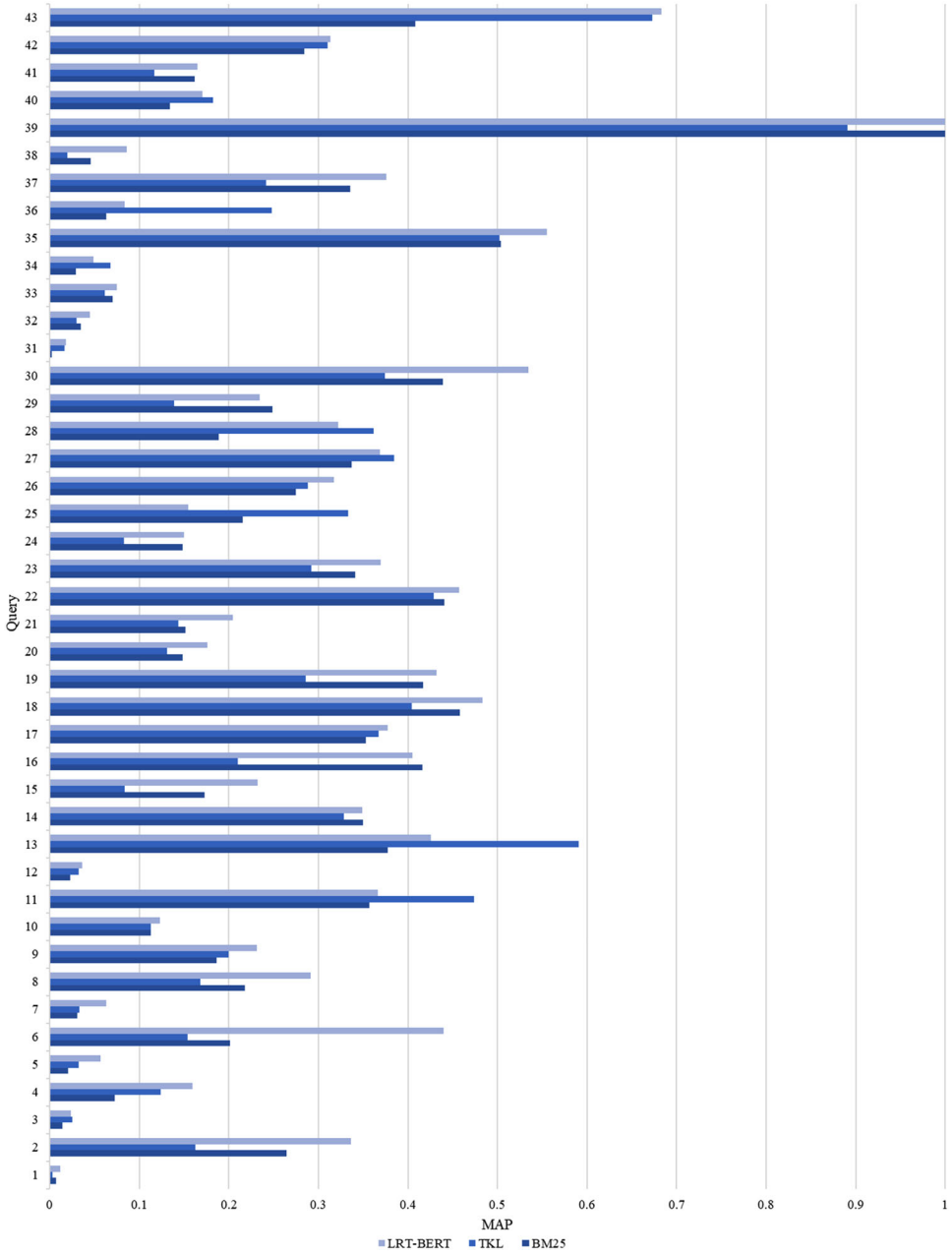


Fig. 5 Comparison of MAP results for the BM25 model, TKL model and LTR-BERT model on the 2019 TREC deep learning tracking task test on 43 queries with continuous relevance labels

Study on different types of datasets

To test the applicability of the short-text training model for long-text retrieval tasks, we also test the LTR-BERT model on four datasets from different sources[3] (FBIS, SJMN, Disk1&2, and LA). In the experiments, the MS MARCO passage training dataset was still used to train the LTR-BERT model. In addition, due to BERT's input length limitation, the entire document cannot be fed into the model at once. We assume the long-text input to be n terms, $n = \{500, 1000, 1500, 2000\}$; here, we take the passage size of 500. The comparison results between the LTR-BERT model and BM25 are shown in Table 3.

As shown in Table 3, in terms of nDCG@10, MRR@10 and MAP, the results for document lengths from 1000 to 2000 are mostly better than those for document lengths of 500. According to the document length statistics, the lengths of 70% of the documents are less than 1000, and the lengths of more than 90% of the documents are less than 2000. Considering only the first 500 terms of the document will ignore some content that may be relevant to the query. However, when the calculated document length is too long, too much

[3] https://pan.baidu.com/s/4rcD131U.

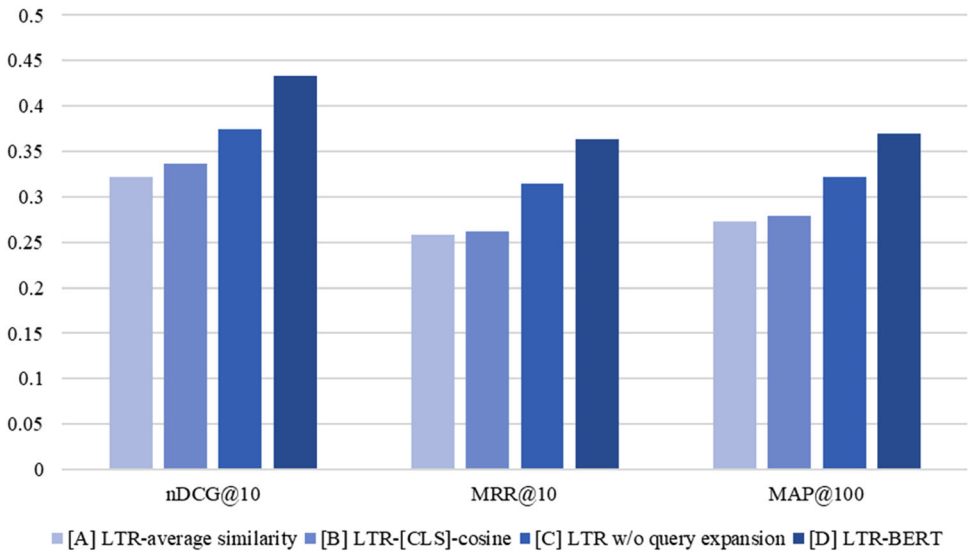


Fig. 6 Ablation results of the LTR-BERT model on the MS MARCO development set

irrelevant information may be introduced, affecting the representation of the document and interfering with the final result.

Indexing throughput and footprint

In this section, we test the index throughput and footprint to comprehensively evaluate the performance of the proposed LTR-BERT model. Table 4 shows the throughput and footprint of the LTR-BERT model for indexing on the LA dataset.

The results in Table 4 show the space occupation of the index by the LTR-BERT model under word embedding of various dimensions. Notably, the higher the word embedding dimension is, the better the MAP results of the model. However, the index footprint is higher. The most recommended setting is when we use 24-dimensional word embeddings to store the representation of documents. At this point, the MAP result is only approximately 1% worse than the most space-consuming setting (the MAP value is 0.2536 when the model uses 128-dimensional word embeddings), but the space footprint is reduced by approximately 81%.

Sensitivity analysis of parameter α

In "Study on semantic matching for long text" and "Study on fine-grained semantic matching for long text", the mentioned models rerank the results according to the first-round ranking. Therefore, in this section, we also study the impact of the document scores added in the first round of ranking on the model. In this work, a linear adjustment factor α is introduced to study the effect of different α values on the MRR@10, nDCG@10 and MAP results. The value range of α is [0,1], and the step is 0.1. When the α value is 0, the BM25 algorithm is not used, and only the reranking score of representation learning-based BERT is used. When the value of α is 1, only BM25 is used.

As Fig. 7 shows, α is approximately 0.5–0.6, with MRR@10, NDCG@10 and MAP reaching maximum values, which indicate increases of 39.08%, 34.56% and 35.46%, respectively, over the results obtained by BM25 on queries with binary relevance labels. α is approximately 0.4–0.5, with MRR@10, NDCG@10 and MAP reaching their maximum values, which indicate increases of 13.15%, 31.10% and 16.12%, respectively, over the results obtained by BM25 on queries with continuous relevance labels. It shows that relevance matching and semantic matching seem to be equally important in this experiment. However, past studies on other datasets have found that the optimal setting requires a greater weighting of BM25 [11]. The experimental data in Fig. 7 used the same document datasets, and the queries were different. Therefore, the optimal value of α is affected by the queries and type of datasets. The results on queries with binary relevance labels show that the reranking results of using BERT alone without adding the first-round retrieval scores are better than the results of using BM25 alone. When the α value is between 0 and 0.7, the results are relatively stable, and when the α value is greater than 0.8, the results decrease significantly. When α is set to 0.5, the LTR-BERT model performs better in terms of MRR@10, nDCG@10 and MAP values on both query datasets.

To explore whether different levels of α values significantly affected the results, we performed one-way analysis of variance (ANOVA) on the results of MRR@10, NDCG@10 and MAP for different parameters α, and the results indicated that the p value was 0. We have reason to believe that different levels of parameter α still significantly affects the results. Therefore, we still recommend using joint relevance matching (BM25 scores used in the first round of retrieval)

Table 3 Results of the LTR-BERT model on six datasets where document lengths have different values

		NDCG@10		MRR@10		MAP	
BM25		0.3251		0.2646		0.2769	
MS docs dev	500	0.4329*	+ 33.16%	0.3627*	+ 37.07%	0.3696*	+ 33.48%
	1000	0.4338*	+ 33.44%	0.3632*	+ 37.26%	0.3701*	+ 33.66%
	1500	**0.4343***	+ 33.59%	**0.3636***	+ 37.41%	**0.3705***	+ 33.80%
	2000	0.4289*	+ 31.93%	0.3574*	+ 35.07%	0.3647*	+ 31.71%
BM25		0.5234		0.7843		0.2339	
MS docs Test2019	500	0.6602*	+ 26.14%	**0.9341***	+ 19.10%	0.2666*	+ 13.98%
	1000	**0.6674***	+ 27.51%	**0.9341***	+ 19.10%	0.2711*	+ 15.90%
	1500	0.6673*	+ 27.49%	**0.9341***	+ 19.10%	0.2726*	+ 16.55%
	2000	0.6666*	+ 27.36%	**0.9341***	+ 19.10%	**0.2734***	+ 16.89%
BM25		0.3190		0.4218		0.2188	
FBIS	500	**0.3557***	+ 11.50%	0.4495*	+ 6.57%	**0.2297***	+ 4.98%
	1000	0.3404*	+ 6.71%	0.4583*	+ 8.65%	0.2265*	+ 3.52%
	1500	0.3416*	+ 7.08%	**0.5755***	+ 36.44%	0.2286*	+ 4.48%
	2000	0.3383*	+ 6.05%	0.4568*	+ 8.30%	0.2280*	+ 4.20%
BM25		0.3276		0.5186		0.2019	
SJMN	500	0.3426*	+ 4.58%	0.5454	+ 5.17%	0.2074*	+ 2.72%
	1000	0.3435*	+ 4.85%	0.5428	+ 4.67%	0.2077*	+ 2.87%
	1500	**0.3468***	+ 5.86%	**0.5494**	+ 5.94%	**0.2081***	+ 3.07%
	2000	0.3468*	+ 5.86%	0.5494	+ 5.94%	0.2080*	+ 3.02%
BM25		0.5040		0.6327		0.2375	
Disk1&2	500	0.5302*	+ 5.20%	0.6720*	+ 6.21%	0.2502*	+ 5.35%
	1000	**0.5414***	+ 7.42%	**0.6867***	+ 8.53%	0.2561*	+ 7.83%
	1500	0.5381*	+ 6.77%	0.6866*	+ 8.52%	**0.2562***	+ 7.87%
	2000	0.5373*	+ 6.61%	0.6850*	+ 8.27%	0.2561*	+ 7.83%
BM25		0.3772		0.5750		0.2470	
LA	500	0.4019*	+ 6.55%	0.5853*	+ 1.79%	0.2649*	+ 7.25%
	1000	0.3973*	+ 5.33%	0.5899*	+ 2.59%	0.2642*	+ 6.96%
	1500	**0.4039***	+ 7.08%	**0.6030***	+ 4.87%	0.2674*	+ 8.26%
	2000	0.3977*	+ 5.43%	0.6001*	+ 4.37%	**0.2677***	+ 8.38%

"*"Indicates that the results of the LTR-BERT model are statistically significant improvements over those of the BM25 model (Wilcoxon signed-rank test, $p < 0.05$). The values shown in bold under each evaluation metric represent the optimal value on the current dataset. The percentage value on the right of each metric represents the percentage of improvement over the metric obtained by the BM25 model

Table 4 Throughput and footprint for indexing on the LA dataset

Method	Dim	Space (GiBs)	Throughput (documents/s)	MAP
LTR-BERT	128	8.96G	76.294	0.2536
LTR-BERT	96	6.87G	97.247	0.2528
LTR-BERT	48	3.45G	111.366	0.2517
LTR-BERT	24	1.74G	111.025	0.2496
LTR-BERT	12	0.95G	131.765	0.2204

"Dim" indicates the dimension of each word embedding in the document representation when the document index is stored off-line. "Space" indicates the disk space occupied by the index when the index is stored off-line. "Throughput" indicates the system throughput when creating the index, that is, the number of documents processed per second

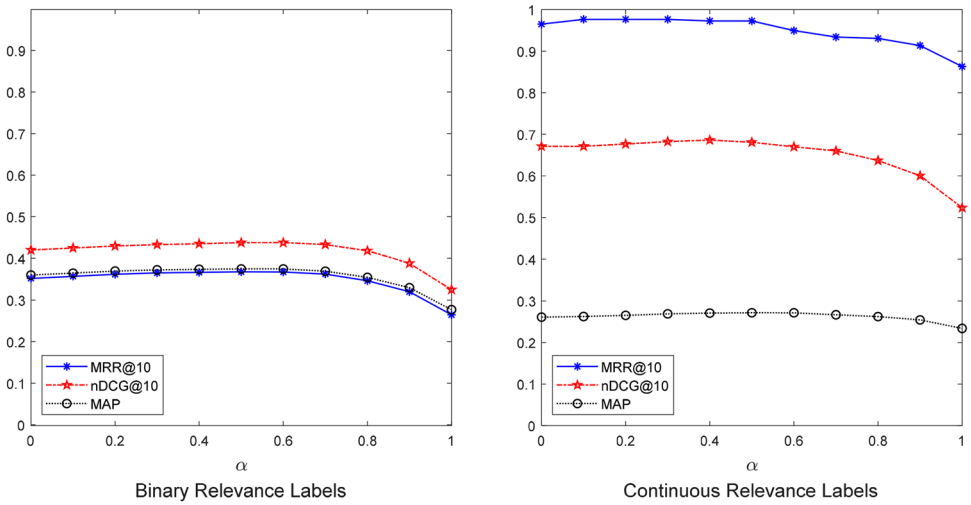


Fig. 7 Influence of parameter α on the reranking result of the LTR-BERT model on the queries with binary relevance labels and continuous relevance labels

and semantic matching (representation learning-based BERT scores used in the second round of retrieval) to obtain better retrieval results.

Conclusions and future work

The existing methods for long-text retrieval use mainly the interaction-focused neural IR method. Although the performance has been improved, interaction-focused semantic matching takes a long time. Therefore, we propose an efficient long-text retrieval by BERT (named LTR-BERT). The representation of query and long text is divided into online and off-line forms. First, the LTR-BERT model is trained with the relevance of short text and queries to obtain segmented text-encoded representations of long texts. A compression layer is designed so that the LTR-BERT model can use lower dimensional density word embedding to represent the semantics of long documents. This process is calculated only once. Second, a query expansion strategy is designed to compensate for the lack of query information and improve the matching ability between the query and the long text. Finally, benefiting from the idea of keyword exact matching in relevance matching, a cheap interaction mechanism without training parameters is designed. This mechanism speeds up response times when fine-grained relevance is considered. Combining representation-focused semantic matching and keywords-based relevance matching, the model is tested on MS MARCO document ranking datasets specifically designed for long-text retrieval. Experimental results show that compared with the interaction-focused neural IR method, the proposed method can guarantee better accuracy and can increase the number of long texts processed per unit time. This work also displays that segmented encoding is better at long-text retrieval than long-text encoding. The number of long-text processed per millisecond increased by 333 times when compared with the interaction-focused neural retrieval, therefore off-line pre-storage improves efficiency of long-text retrieval.

This work proposes a representation-focused method for efficient long-text retrieval. Therefore, we take only the BERT model as an example to perform representation-based semantic matching on queries and long text. In the future, we will first study and discuss the application of different semantic matching models to information retrieval tasks by using representation learning-based methods. Second, our proposed method will be applied for more applications (such as biomedicine and Clinical IR) in the future [40–42]. Further research and applications will be considered to explore the generality and limitations of the proposed method.

Acknowledgements This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LQ23F020014 and the National Natural Science Foundation of China under Grant No. 62271177. This research was also supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the York Research Chairs (YRC) program and an ORF-RE (Ontario Research Fund- Research Excellence) award in BRAIN Alliance.

Data availability Data availability is not applicable to this article as no new data were created or analyzed in this study. Most of the author's data are from publicly available data sets, which have been explained in the paper.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proc. 16th conf. North Am. chapter assoc. comput. linguist., pp 2227–2237. http://arxiv.org/abs/1802.05365
2. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proc. 17th conf. North Am. chapter assoc. comput. linguist. hum. lang. technol., Minneapolis, USA, pp 4171–4186. http://arxiv.org/abs/1810.04805
3. Liu C, Zhu W, Zhang X, Zhai Q (2023) Sentence part-enhanced BERT with respect to downstream tasks. Complex Intell Syst 9:463–474. https://doi.org/10.1007/s40747-022-00819-1
4. Wang Y, Rong W, Zhang J, Zhou S, Xiong Z (2020) Multi-turn dialogue-oriented pretrained question generation model. Complex Intell Syst 6:493–505. https://doi.org/10.1007/s40747-020-00147-2
5. Dai Z, Callan J (2019) Deeper text understanding for IR with contextual neural language modeling. In: Proc. 42nd int. ACM SIGIR conf. res. dev. inf. Retrieval (SIGIR'19), pp 985–988. https://doi.org/10.1145/3331184.3331303
6. MacAvaney S, Yates A, Cohan A, Goharian N (2019) CEDR: contextualized embeddings for document ranking. In: Proc. 42nd int. ACM SIGIR conf. res. dev. inf. Retrieval (SIGIR'19). ACM, New York, USA, pp 1101–1104. https://doi.org/10.1145/3331184.3331317
7. Boualili L, Moreno JG, Boughanem M (2020) MarkedBERT: integrating traditional IR cues in pre-trained language models for passage retrieval. In: Proc. 43rd int. ACM SIGIR conf. res. dev. inf. Retrieval (SIGIR'20), pp 1977–1980. https://doi.org/10.1145/3397271.3401194
8. Akkalyoncu Yilmaz Z, Yang W, Zhang H, Lin J (2019) Cross-domain modeling of sentence-level evidence for document retrieval. In: Proc. 2019 conf. empir. methods nat. lang. process. 9th int. jt. conf. nat. lang. process. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 3488–3494. https://doi.org/10.18653/v1/D19-1352.
9. Mitra B, Craswell N (2018) An introduction to neural information retrieval. Found Inf Retr 13:1–126. https://doi.org/10.1561/1500000061
10. Pan M, Wang J, Huang JX, Huang AJ, Chen Q, Chen J (2022) A probabilistic framework for integrating sentence-level semantics via BERT into pseudo-relevance feedback. Inf Process Manage 59:102734. https://doi.org/10.1016/j.ipm.2021.102734
11. Wang J, Pan M, He T, Huang X, Wang X, Tu X (2020) A pseudo-relevance feedback framework combining relevance matching and semantic matching for information retrieval. Inf Process Manage 57:102342. https://doi.org/10.1016/j.ipm.2020.102342
12. Khattab O, Zaharia M (2020) ColBERT: efficient and effective passage search via contextualized late interaction over BERT. In: Proc. 43rd int. ACM SIGIR conf. res. dev. inf. retrieval (SIGIR'20), pp 39–48. https://doi.org/10.1145/3397271.3401075.
13. Nie P, Zhang Y, Geng X, Ramamurthy A, Song L, Jiang D (2020) DC-BERT: decoupling question and document for efficient contextual encoding. In: Proc. 43rd int. ACM SIGIR conf. res. dev. inf. retrieval (SIGIR'20), pp 1829–1832. https://doi.org/10.1145/3397271.3401271
14. Nogueira R, Cho K (2019) Passage re-ranking with BERT. arXiv: 1901.04085. http://arxiv.org/abs/1901.04085
15. Hofstätter S, Zamani H, Mitra B, Craswell N, Hanbury A (2020) Local self-attention over long text for efficient document retrieval. In: Proc. 43rd int. ACM SIGIR conf. res. dev. inf. retr. ACM, New York, USA, pp 2021–2024. https://doi.org/10.1145/3397271.3401224
16. Wei T, Li X, Stojanovic V (2021) Input-to-state stability of impulsive reaction–diffusion neural networks with infinite distributed delays. Nonlinear Dyn 103:1733–1755. https://doi.org/10.1007/s11071-021-06208-6
17. Xu Z, Li X, Stojanovic V (2021) Exponential stability of nonlinear state-dependent delayed impulsive systems with applications. Nonlinear Anal Hybrid Syst 42:101088. https://doi.org/10.1016/j.nahs.2021.101088
18. Xiao S, Liu Z, Han W, Zhang J, Shao Y, Lian D, Li C, Sun H, Deng D, Zhang L, Zhang Q, Xie X (2022) Progressively optimized bi-granular document representation for scalable embedding based retrieval. Assoc Comput Mach. https://doi.org/10.1145/3485447.3511957
19. Yilmaz ZA, Wang S, Yang W, Zhang H, Lin J (2020) Applying BERT to document retrieval with birch. In: Proc. conf. empir. methods nat. lang. process. 9th int. jt. conf. nat. lang. process., pp 19–24. https://doi.org/10.18653/v1/d19-3004
20. Hu B, Lu Z, Li H, Chen Q (2015) Convolutional neural network architectures for matching natural language sentences. Adv Neural Inf Process Syst 3:2042–2050
21. Pang L, Lan Y, Guo J, Xu J, Wan S, Cheng X (2016) Text matching as image recognition. In: Proc. 30th AAAI conf. artif. intell., pp 2793–2799. http://arxiv.org/abs/1602.06359
22. Hui K, Yates A, Berberich K, de Melo G (2017) PACRR: a position-aware neural IR model for relevance matching. In: Proc. 2017 conf. empir. methods nat. lang. process. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 1049–1058. https://doi.org/10.18653/v1/D17-1110.
23. Hui K, Yates A, Berberich K, de Melo G (2018) Co-PACRR: a context-aware neural IR model for ad-hoc retrieval. In: Proc. 11th ACM int. conf. web search data mining (WSDM'18). ACM, New York, NY, USA, pp 279–287. https://doi.org/10.1145/3159652.3159689
24. Xiong C, Dai Z, Callan J, Liu Z, Power R (2017) End-to-end neural ad-hoc ranking with kernel pooling. In: Proc. 40th int. ACM SIGIR conf. res. dev. inf. retr. Association for Computing Machinery, Inc, pp 55–64. https://doi.org/10.1145/3077136.3080809
25. Dai Z, Xiong C, Callan J, Liu Z (2018) Convolutional neural networks for soft-matching n-grams in ad-hoc search. In: Proc. 11th ACM int. conf. web search data mining (WSDM'18). ACM, New York, NY, USA, pp 126–134. https://doi.org/10.1145/3159652.3159659
26. Yang P, Fang H, Lin J (2018) Anserini: reproducible ranking baselines using lucene. J Data Inf Qual 10:1–20. https://doi.org/10.1145/3239571
27. Huang P-S, He X, Gao J, Deng L, Acero A, Heck L (2013) Learning deep structured semantic models for web search using click through data. In: Proc. 22nd ACM int. conf. conf. inf. knowl. manag., pp 2333–2338. https://doi.org/10.1145/2505515.2505665
28. Shen Y, He X, Gao J, Deng L, Mesnil G (2014) Learning semantic representations using convolutional neural networks for web search. In: Proc. 23rd int. conf. world wide web, pp 373–374. https://doi.org/10.1145/2567948.2577348.
29. Shen Y, He X, Gao J, Deng L, Mesnil G (2014) A latent semantic model with convolutional-pooling structure for information retrieval. In: Proc. 23rd ACM int. conf. inf. knowl. manag., pp 101–110. https://doi.org/10.1145/2661829.2661935

30. Guo J, Fan Y, Ai Q, Croft WB (2016) A deep relevance matching model for ad-hoc retrieval. In: Proc. 25th ACM int. conf. inf. knowl. manag. ACM, New York, USA, pp 55–64. https://doi.org/10.1145/2983323.2983769.
31. Zamani H, Dehghani M, Croft WB, Learned-Miller E, Kamps J (2018) From neural re-ranking to neural ranking: learning a sparse representation for inverted indexing hamed. In: Proc. 27th ACM int. conf. inf. knowl. manag. ACM, New York, USA, pp 497–506. https://doi.org/10.1145/3269206.3271800
32. Beltagy I, Peters ME, Cohan A (2020) Longformer: the long-document transformer, ArXiv: 2004.0515v1. http://arxiv.org/abs/2004.05150
33. Ding M, Zhou C, Yang H, Tang J (2020) CogLTX: applying BERT to long texts. In: Proc. 34th int. conf. neural inf. process. syst., pp 12792–12804. https://github.com/Sleepychord/CogLTX
34. Zhuang Z, Tao H, Chen Y, Stojanovic V, Paszke W (2022) An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints. IEEE Trans Syst Man Cybern Syst 53:3461–3473. https://doi.org/10.1109/TSMC.2022.3225381
35. Zhou C, Tao H, Chen Y, Stojanovic V, Paszke W (2022) Robust point-to-point iterative learning control for constrained systems: a minimum energy approach. Int J Robust Nonlinear Control 32:10139–10161. https://doi.org/10.1002/rnc.6354
36. Pan M, Zhang Y, Zhu Q, Sun B, He T, Jiang X (2019) An adaptive term proximity based Rocchio's model for clinical decision support retrieval. BMC Med Inform Decision Mak 19:251. https://doi.org/10.1186/s12911-019-0986-6
37. MacAvaney S, Nardini FM, Perego R, Tonellotto N, Goharian N, Frieder O (2020) Efficient document re-ranking for transformers by precomputing term representations. In: Proc. 43rd int. ACM SIGIR conf. res. dev. inf. retr., pp 49–58. https://doi.org/10.1145/3397271.3401093
38. Bajaj P, Campos D, Craswell N, Deng L, Gao J, Liu X, Majumder R, McNamara A, Mitra B, Nguyen T, Rosenberg M, Song X, Stoica A, Tiwary S, Wang T (2016) MS MARCO: a human generated machine reading comprehension dataset. In: Proc. 30th conf. neural inf. process. syst., pp 1–11. http://arxiv.org/abs/1611.09268
39. Wang J, Zhao W, Tu X, He T (2023) A novel dense retrieval framework for long document retrieval. Front Comput Sci 17:174609. https://doi.org/10.1007/s11704-022-2041-5
40. Yin X, Huang JX, Li Z, Zhou X (2013) A survival modeling approach to biomedical search result diversification using wikipedia. IEEE Trans Knowl Data Eng 25:1201–1212. https://doi.org/10.1109/TKDE.2012.24
41. Huang X, Zhong M, Si L (2005) York University at {TREC} 2005: Genomics track. In: Voorhees EM, Buckland LP (eds) Proceedings of the Fourteenth Text REtrieval Conference, National Institute of Standards and Technology (NIST), Gaithersburg, Maryland. http://trec.nist.gov/pubs/trec14/papers/yorkuhuang2.geo.pdf
42. Huang X, Hu Q (2009) A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval. In: Proceedings of the 32nd annual international acm sigir conference on research and development in information retrieval, SIGIR 2009, Boston, MA, USA. ACM Press, New York, USA, pp 307–314. https://doi.org/10.1145/1571941.1571995