Front.Comput.Sci.
DOI

SUPPLEMENTARY FILE

A Novel Dense Retrieval Framework for Long Document Retrieval

Jiajia Wang [1] **Weizhong Zhao**[2,3] **Xinhui Tu** (✉)[2,3] **Tingting He** (✉)[2,3]

1 School of Mathematics and Statistics, Central China Normal University, Wuhan 430079, China
2 Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, Central China Normal University, Wuhan, Hubei 430079, PR China
3 National Language Resources Monitoring & Research Center for Network Media, Central China Normal University, Wuhan, Hubei 430079, PR China

Abstract Dense retrieval models have achieved impressive success in passage retrieval, retrieving only the portions of a document that are relevant to a specific information need. However, existing models fail to effectively retrieve long documents that consist of several passages. The main challenge is that their direct application to long documents ignores the semantic associations among segments, which is essential to derive accurate ranking scores. To address this challenge, we propose a framework, dense retrieval via segment correlation matrix (DRSCM), specifically for long document retrieval, which explicitly incorporates the semantic associations between pairs of segments to calculate the relevance score between the query and the document. More specifically, dense vectors of segments in each document are derived by pretrained dense retrieval models, and the semantic relevance scores between pairs of segments are computed accordingly. Several strategies are designed to aggregate the relevance score between each segment and query, considering both the local context information and the correlations among segments of a document (i.e., global context information). Through these strategies, the different contributions of segments are captured to obtain the final ranking score, for improving the performance at long document retrieval. Results on two standard TREC collections demonstrate that the proposed framework is effective and efficient at the task of long document retrieval.

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *tuxinhui@mail.ccnu.edu.cn*.

Keywords Dense Retrieval, Semantic Association, Long Document Retrieval

1 Introduction

Inspired by the impressive success of BERT [4] in various NLP applications (e.g., natural language inference [2, 5], question answering [15], named entity recognition [17], and document ranking [22,23]), researchers have attempted to apply pretrained language models to information retrieval. Since documents are first transformed by BERT to dense vectors for calculating ranking scores, the model is usually called a *dense retrieval model*. However, BERT has the limitation that the maximum length of tokens is only 512. Most dense retrieval models focus on passage retrieval [9, 13, 16, 19, 24] (in which each document in a collection typically consists of a single passage), which is not suitable for information retrieval for long documents consisting of several passages.

To address long document retrieval, researchers split long documents into small segments, and obtain the representation of each by BERT. The final retrieval score of a document is calculated by integrating the similarity of the query and each segment in the long document according to the utilized aggregation strategies. Based on the granularity of dividing documents, sentences and passages are selected as two types of segments, where a passage refers to an overlapping sequence of text with a fixed length. Thus, a passage may consist of more than one sentence in a long document. According to the

granularity of dividing segments, approaches to long document retrieval can be classified as either sentence- or passage-level aggregation. The main steps include splitting each whole document into several small segments, e.g., sentences or overlapping passages; computing similarity between each segment and the query; and aggregating segment similarities as the final similarity score of the document.

According to different strategies of aggregation, methods can be divided into three subgroups: 1) Avg (average): all segments of a document contribute equally to the similarity score, i.e., the average similarity of all segments is the score of the document [25]; 2) Max: the segment most relevant to the query represents the whole document [3]; 3) TopK: segments with the top-k similarities represent the whole document, and the sum of their similarities is the final ranking score [22,23]. Since a long document usually consists of many segments, the association weights of segments to its theme or label are inevitably different. It turns out that for methods based on Avg aggregation, the ranking score of a document against certain queries is inaccurate, leading to undesirable performance at long document retrieval tasks. Hence, Max and TopK are widely used for long document retrieval. However, these strategies consider only the local context (i.e., text in each segment) to compute the retrieval score of each segment, while ignoring the global context (e.g., semantic associations among segments) of the whole document. This can lead to a phenomenon called "topic drift", by which some segments that are irrelevant to the topic of the whole document will dominate the calculation of the final ranking score.

We employ a toy example in Figure 1 to illustrate this idea. The query is, "what are the advantages of the bank in school?" For the (long) document *Doc*1, its topic or label is "school bank," and almost every sentence in *Doc*1 is about the bank in school. The topic or label of *Doc*2 is "campus life," and most of its sentences are about school activities. However, three sentences mention banks in school. By comparing the details of the two documents, it is clear that *Doc*1 is more related than *Doc*2 to the query. However, a retrieval model using Max or TopK aggregation will derive an inaccurate result. To illustrate more clearly, the top-3 most relevant sentences in each document are highlighted and labeled with ①, ②, and ③, respectively, in decreasing order of association scores, and we calculate the relevance scores between the query and the sentences of each document using the Sentence-BERT dense retrieval model [16].

According to the Max aggregation method, *Doc2* has a greater ranking score than *Doc1*, since the derived segment score of the sentence with ① in *Doc1* is 0.4367, while its counterpart in *Doc2* is 0.6677. Hence, a suboptimal result of long document retrieval is derived due to the Max aggregation method. For TopK aggregation, when k=*3*, the top-3 scores of sentences are ① 0.4367, ② 0.4325, and ③ 0.4251 for *Doc1*, and ① 0.6677, ② 0.5121, and ③ 0.4250 for *Doc2*. Hence, a retrieval model with TopK aggregation concludes incorrectly that *Doc2* is more related to the query than *Doc1*, even though most sentences in *Doc2* discuss the topic other than the query topic "school bank." Although k can be assigned a greater value to address this issue, there is no appropriate k in some cases. We argue that Max and TopK fail because they consider only the local context (i.e., local semantic information) and ignore the global context (relationships among segments), which leads to topic drift when calculating the association scores of segments (i.e., sentences or passages). A feasible solution is to consider both local and global information for representation learning. For example, Zhu and Yang [26] introduce the ActBERT model, which uses the global action information to enrich interactions between linguistic texts and local regional objects for learning better video-text representations, and improves the performance on several downstream video-and-language tasks, such as text-video clip retrieval, video captioning, and video question answering.

We propose a dense retrieval framework for long documents that considers both local and global semantic contexts to derive more accurate retrieval scores. Each long document is divided into small segments (sentences or passages) in an offline module, dense representations of segments are obtained by pretrained dense retrieval models, and the semantic association among divided segments is calculated and represented as a correlation matrix. In the online module (for long document retrieval), the retrieval score for each segment is calculated while considering both the dense representation of the segment (local context) and the correlation matrix (global context). Aggregation strategies combine the retrieval scores for segments to derive the final retrieval score for each long document, so as to obtain the retrieval results. Experiments on two standard TREC collections demonstrate the effectiveness of the proposed framework.

This paper makes the following contributions.

• We propose a dense retrieval framework called DRSCM for long documents, which uses a pretrained dense retrieval model to learn representations of short segments in an offline pattern, and improves the efficiency of long document retrieval;

Query	What are the advantages of the bank in school?	
Document	Doc1	Doc2
Content	Every Tuesday, Carina goes to work in a bank. She knows all her customers very well, because they are students of a school. ①Carina' s bank is in the school in Chicago. The bank is a branch of the national bank, it is open for 30 minutes a week. …… ③Every Tuesday morning, she opens the bank. ……The idea came from a teacher. Mr. Bassett took his idea to the national bank. ②The bank agreed to his plan. "We want to give the children a chance to learn about money." said Mr. Bassett. "We are training them to look after their money carefully." The next plan for the bank is a credit card. Many people in the USA don't use credit cards very well." said Mr. Basset.	Campus activities have been organized in many universities and colleges. ①There are some national banks in school. These activities range from academic to recreational, such as academic reports, speech contests, poet's club, painting clubs, singing and dancing groups, etc. ……. From these activities, …… classroom and get to know the society. ③In addition, the banks mainly include "bank of china" and "industrial and commercial bank of china" in school. ②The banks open five days a week for students. All these offer an important method for students to broaden their horizons. By participating in campus activities, they have fulfilled university life and in turn help campus activities to grow and flourish.
Topic	School bank	Campus life

Fig. 1: A toy example of topic drift.

• Both local and global contexts are considered when calculating retrieval scores for short segments, which addresses the issue of topic drift to improve the effectiveness of long document retrieval;

• Comprehensive experiments are conducted to evaluate our proposed framework by comparing with different pre-trained dense retrieval models, different dividing methods of long documents, and different aggregation strategies. In addition, the contribution of the segment correlation matrix to the performance improvements is demonstrated.

The remainder of this paper is organized as follows. Related work is reviewed in section 2. Our framework is described in section 3. We discuss our experiments and analysis in section 4. Our conclusions and future work are provided in section 5.

2 Related work

In this section, we briefly review the related works to the task of long document retrieval, including cross-encoder based models and dual-encoder based models.

2.1 Cross-encoder based models

Typically, cross-encoder based models take simultaneously both query and each document as input to integrate the rich interactions between the query and the context of the corresponding document (i.e., one candidate) for the task of long document retrieval. Because BERT limits the sequence length to 512, a long document is split into small segments, and the final retrieval score is calculated by integrating the similarity of the query and each segment according to the utilized aggregation strategies. According to the selected granularity of segments, existing cross-encoder based models for long document retrieval can be classified as sentence-level models or passage-level models.

Sentence-level Models

For the task of long document retrieval, Birch [23] is the most representative sentence-level model, in which it takes the aggregation score of top-k sentences as the final document score. To aggregate the semantic context from different

sentences, Birch assigns distinct weights (w_1, w_2, w_3) to top-3 sentences, where w_1 is set fixedly as 1, and w_2 and w_3 are tuned via grid search in the range of [0, 1] with step size 0.1. Moreover, it proves by experimental results that only three most associated sentences are good enough to represent the whole document, since the contributing weights of other sentences than the top-3 sentences are nearly zero [23].

Passage-level Models

After segmenting a long document into passages, long document retrieval can be realized by aggregating passage scores or representations [10, 14]. We can compute the similarity between the query and each passage of a long document, and indirectly obtain the document-level semantic retrieval score by aggregating passage similarities, or we can first obtain the document-level representation by aggregating passage representations, and directly obtain the document-level semantic retrieval score by computing the similarity between it and the query.

• Passage-level score aggregation: Dai and Callan [3] divide a long document into overlapping passages within the input length limitation of BERT. Methods to determine the final document retrieval score include selecting the score of the first passage (BERT-FirstP), selecting the maximum passage score (BERT-MaxP), and taking the sum of all passage scores (BERT-SumP). Zhang et al. [25] propose passage-level score aggregation (AvgP), which takes the mean of all passage scores as the final document retrieval score. Similar to [23], MaxP and TopKP apply an expensive pretrained model such as BERT to obtain the relevance scores of all passages in a document to select the top-k passages. To simplify the process, Hofstätter et al. [7] select the top-k passages via a lightweight and fast selection model, i.e., the efficient student model (EST), and use an expensive model to estimate the relevance scores of the selected passages.

• Passage-level representation aggregation: From the perspective of the contextualized embeddings of BERT, Li et al. [10] propose passage representation aggregation for document reranking (PARADE), which has several approaches: 1) Similar to CEDR [14], $\text{PARADE}_{\text{Avg}}$ takes the measure of average pooling for passage representations to obtain the final document representation; 2) $\text{PARADE}_{\text{Sum}}$ performs additive pooling across passage representations; 3) $\text{PARADE}_{\text{Max}}$ obtains the final document representation via element-wise max pooling on all passage representations; 4) $\text{PARADE}_{\text{Attn}}$ produces an attention weight for each passage representation via a feedforward network; 5) $\text{PARADE}_{\text{CNN}}$ uses a stack of convolutional neural networks (CNNs) to repeatedly aggregate pairs of passage representations until only one remains; 6) $\text{PARADE}_{\text{Transformer}}$ aggregates passage representations via a stack of two randomly-initialized transformer encoders.

The above works integrate fine-grained independent passage-level information for the document-level signal, and lack global context information, which may lead to suboptimal estimates of passage-level relevance. To obtain context-aware passage-level signals, Wu et al. [20] introduce the Passage-level Cumulative Gain Model (PCGM) for document ranking, applying BERT to obtain passage-level (CLS) representations, and aggregating them via LSTM [6].

2.2 Dual-encoder based models

Since cross-encoder based models require complex computations and have long interaction time, the efficiency of existing methods is undesirable. Consequently, researchers have attempted to retrieve documents in a low-dimensional embedding space [21, 24] through a dense retrieval model (DRM), in which queries and documents are mapped into low-dimensional embeddings via dual-encoders (i.e., Query-Encoder and Document-Encoder), adopting the inner product or cosine similarity to retrieve relevant documents.

Since the maximum length of BERT is 512, dense retrieval models focus on dense passage retrieval [9, 13, 16, 19, 24] (where each document consists typically of a single passage). Reimers and Gurevych [16] propose the Sentence-BERT (S-BERT) model, which is a modification of the pretrained BERT network, using siamese and triplet network structures to derive sentence embeddings. Khattab et al. [9] introduce the ColBERT model, which greatly reduces the retrieval latency with a slight degradation in performance compared to cross-encoder based models. Furthermore, Lin et al. [12] improve ColBERT for ranking with dense representations by applying knowledge distillation (KD). Zhan et al. [24] propose the RepBERT model, which represents queries and documents with fixed-length embeddings by the BERT model in first-stage retrieval, and the token embeddings of the passage are averaged to produce the final passage representation. Karpukhin et al. [8] propose the DPR model, in which two separated encoders are trained for queries and documents on several commonly used datasets.

In addition, Luan et al. [13] discuss the relationship among sparse retrieval models (e.g., BM25), dense retrieval models, and attention neural networks. The importance of BM25 to dense retrieval models is presented as well in [19].

To effectively retrieve long documents via dense passage

retrieval, we propose a framework for long document retrieval, explicitly incorporating the semantic associations between pairs of segments to calculate the relevance between the query and documents. Dense vectors of segments in each document are derived by dense retrieval models (DRMs), and the semantic relevance between pairs of segments is computed accordingly.

3 Dense retrieval via segment correlation matrix

We describe the proposed framework called Dense Retrieval via Segment Correlation Matrix (DRSCM), whose schematic diagram is presented in Figure 2. DRSCM includes preprocessing and retrieval modules, which are discussed in the next two sections. In the preprocessing module, each document D is split into several small segments, e.g., sentences or overlapping passages, and each segment (i.e., local context) is embedded in a dense vector by the Document-Encoder (E_D) of DRM. To obtain the global context information, the correlation matrix M for segment embeddings is calculated via a similarity operation (e.g., cosine similarity for SBERT and inner product for RepBERT). In the retrieval module, the dense vector of a query q can be obtained via the Query-Encoder (E_Q) of DRM, and the relevance scores between the query embedding and all segment embeddings of each document are computed according to the selected similarity function. To address the issue of topic drift, the final segment scores are calculated by combining the semantic associations between each segment and the other segments in the long document (i.e., the correlation matrix M) with the original segment relevance scores. Finally, the semantic retrieval score for each document S_D is obtained via the selected aggregation strategy.

3.1 Preprocessing module

In the preprocessing module, a long document D is split into a set of segments, $SS=\{seg_1, \cdots, seg_i, \cdots, seg_n\}$. We use a Document-Encoder (E_D) of DRM, such as SBERT [16] or RepBERT [24], to produce dense representations for segments, denoted by $\{E_D(seg_1), \cdots, E_D(seg_i), \cdots, E_D(seg_n)\}$. The reason why we test two document encoders (i.e., SBERT and RepBERT) lies in that there are two main typical differences between them, which can be used to evaluate the robustness of our proposed framework. First, SBERT employs siamese and triple network on BERT to derive sentence semantic embeddings, while RepBERT uses an average pooling operator on the contextualized token embeddings derived from BERT of the passage to obtain the passage-level embedding. Second, SBERT uses cosine similarity to compute relevance scores between queries and documents, while RepBERT regards inner products of queries and documents as relevance scores.

Generally, the correlation matrix M consists of the relevance score between each pair of segments, which is viewed as the global context information. Assume there are n segments in document D, the correlation matrix M will be an $n \times n$ matrix which is illustrated as follows:

$$M = \begin{pmatrix} m_{11} & \cdots & m_{1j} & \cdots & m_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{i1} & \cdots & m_{ij} & \cdots & m_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n1} & \cdots & m_{nj} & \cdots & m_{nn} \end{pmatrix} \quad (1)$$

where m_{ij} is the relevance score between the i-th segment and the j-th segment.

This procedure can be completed offline, which guarantees the efficiency of the retrieval module.

3.2 Retrieval module

The retrieval module includes a linear combination phase for computing segment scores, and an aggregation phase for computing the semantic score of each document.

3.2.1 Linear combination phase

Given a query q, the dense representation can be obtained through the Query-Encoder $E_Q(q)$ of DRM. Relevance scores $\{s'_1, \cdots, s'_i, \cdots, s'_n\}$ between query and segment representations can be calculated via a similarity function (e.g., cosine similarity for SBERT and inner product for RepBERT), and the i-th element relevant score can be calculated as follows.

$$s'_i = similarity(E_Q(q), E_D(seg_i)) \quad (2)$$

According to the correlation matrix M (eq. 1), the weight of the i-th segment in the entire long document is defined as:

$$w_i = \frac{1}{n} \sum_{j=1}^{n} m_{ij} \quad (3)$$

We can find that the weight definition of each segment takes into account the semantic associations between the segment and all other segments in the long document. This weight definition can be used to address topic drift to some extent.

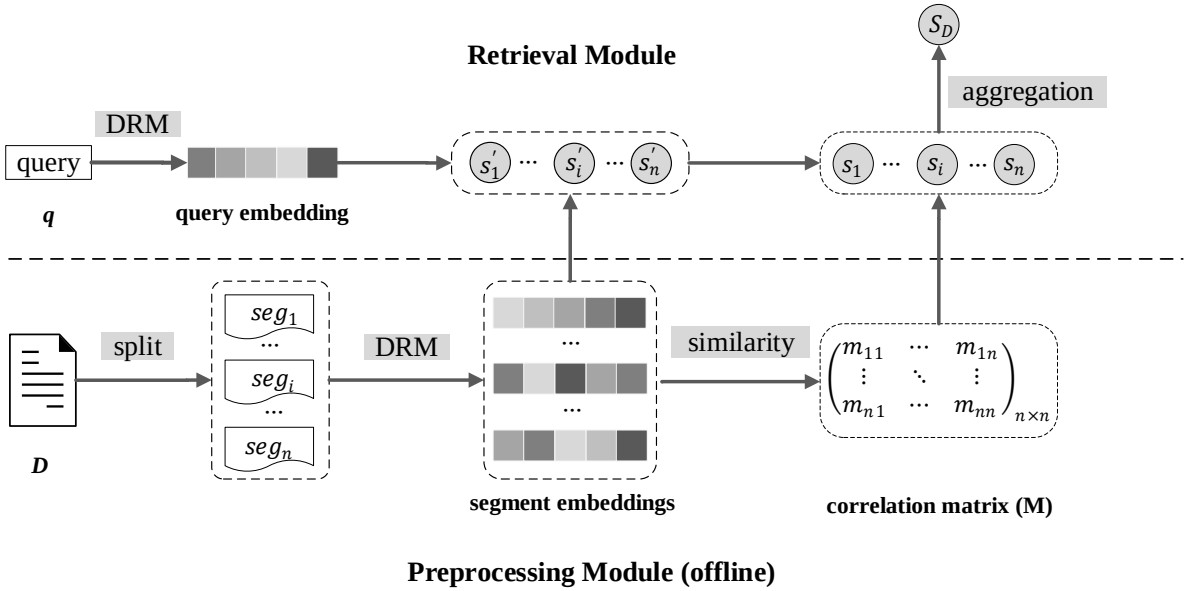


Fig. 2: DRSCM architecture includes preprocessing and retrieval modules. In preprocessing, DRSCM splits long document D into n small segments ($\{seg_1, \cdots, seg_i, \cdots, seg_n\}$), and segment embeddings are obtained via DRM. The correlation matrix for segment embeddings is computed via a similarity operation. For the retrieval module, the embedding of query q is obtained via DRM, and relevance scores between query and segment embeddings are calculated, as denoted by $\{s_1^{'}, \cdots, s_i^{'}, \cdots, s_n^{'}\}$). Final semantic scores for segments are obtained as linear combinations of the correlation matrix and segment relevance scores, and denoted by $\{s_1, \cdots, s_i, \cdots, s_n\}$. Document-level semantic score S_D is derived based on final segment scores via aggregation strategies.

The final score for segment seg_i ($i \in \{1, \ldots, n\}$) is

$$s_i = \alpha \cdot s_i' + (1 - \alpha) \cdot w_i \tag{4}$$

where α is the hyperparameter of the proposed model, which is used to trade off the contributions of the relevance score and segment weight for long document retrieval. Note that α is tuned via grid search in the range of [0,1].

3.2.2 Aggregation phase

After obtaining the final segment scores, the semantic score S_D for the long document can be derived through aggregation. We investigate four aggregation strategies: Max, 2sum, 3sum, and Mean. It is worth noting that 2sum and 3sum are variants of TopK, which we describe below.

• **Max**: The segment with the highest score is selected to represent the semantic score for D,

$$S_{max}(D) = \max_{seg_i \in SS} s_i \tag{5}$$

• **2sum**: The scores of the two highest-scoring segments are aggregated to derive the semantic score for D. Without loss of generality, seg_i and seg_j are the segments with the highest and second-highest scores, respectively. The semantic score of D is defined as:

$$S_{2sum}(D) = \beta_1 \cdot s_i + \beta_2 \cdot s_j \tag{6}$$

where β_1 and β_2 are hyperparameters that determine the contributions of the segments with the top-2 scores for D. These are tuned via grid search in the range of [0,1], with step size 0.1.

• **3sum**: Similar to $2sum$, the top-3 scores of segments are aggregated to derive the semantic score for D. Without loss of generality, seg_i, seg_j, and seg_k are the segments with the top-3 scores in descending order. The semantic score of D is defined as:

$$S_{3sum}(D) = \beta_1 \cdot s_i + \beta_2 \cdot s_j + \beta_3 \cdot s_k \tag{7}$$

Similar to $2sum$, hyperparameters β_1, β_2, and β_3 are tuned via grid search in the range of [0,1] with the step size 0.1.

• **Mean**: The average score of all segments represents the semantic score for D,

$$S_{mean}(D) = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{8}$$

With the dense retrieval model described above, the retrieval scores for all long documents are obtained, which can

be combined with retrieval scores derived based on sparse retrieval models (e.g., BM25) to further enhance the reranking performance for long document retrieval. More specifically, we combine the BM25 score and semantic score to compute the final document score, which is defined formally as follows.

$$S_D^f = \gamma \cdot S_D + (1 - \gamma) \cdot BM25(D) \quad (9)$$

where S_D denotes the semantic score derived by one of the above mentioned aggregation strategies, and γ denotes the adjusting factor for contributions of BM25 score and semantic score.

4 Experiments

We conduct experiments on two widely-used collections to evaluate the proposed DRSCM framework, including a general performance comparison between DRSCM and selected baselines, evaluation of the influence of pretrained dense retrieval model, segment length, and aggregation method on performance, and evaluation of the contribution of the segment correlation matrix to the performance improvement. We describe the experimental setting and provide our analysis below.

4.1 Experimental settings

4.1.1 Datasets

We utilize two standard TREC collections to evaluate our framework.

- **Robust04** is a newswire corpus from TREC 2004 Robust Track [18], a standard ad hoc retrieval dataset consisting of 250 topics among about 0.5M documents.
- **GOV2** is a crawl of the .gov domain webpages with 25M documents. The collection has 150 queries derived from TREC Terabyte 2004, 2005, and 2006 (topics 701–750, 751–800, and 801–850, respectively).

Table 1: Statistics of the collections.

Collection	# Queries	# Documents	# Words/document
Robust04	250	500000	771
GOV2	150	25000000	3882

4.1.2 Baselines and evaluation metrics

To experimentally evaluate DRSCM, we select the traditional retrieval model BM25 and three cross-encoder pretrained models.

For BM25, we use the implementation in Anserini[1)], whose two hyperparameters (k_1 and b), called fraction adjusters, must be set appropriately for different datasets. For GOV2, we set k_1 and b according to the queries, and used ($k_1 = 1.1$, $b = 0.41$), ($k_1 = 0.9$, $b = 0.4$), and ($k_1 = 0.9$, $b = 0.4$) for topics 701–750, 751–800, and 801–850, respectively. The default values ($k_1 = 0.9$, $b = 0.4$) are used for Robust04.

Three cross-encoder based models are selected as baselines for performance evaluation.

- **Birch** completes the task of long document retrieval via sentence-level score aggregation [23], which allows for interaction between the query and document;
- **Vanilla BERT** is a BERT-based baseline in which each long document is truncated to the first 512 or 1024 tokens [11];
- **CEDR_KNRM** is a state-of-the-art model that incorporates the BERT classification vector in a KNRM neural model; each long document is split into segments as evenly as possible [11].

It is worth noting that the three cross-encoder based models and our method are all implemented on the initial results derived by BM25. Specifically, the top 1000 candidates for each query obtained by BM25 are selected, from which the semantic scores of documents are derived by each model.

To evaluate retrieval performance, we report precision (P@10, P@20) and discounted cumulative gain (NDCG@10, NDCG@20) as the evaluation measures to compare these baselines with our proposed framework DRSCM. We provide the average reranking time (in ms) for all queries in each dataset to compare the efficiency of models.

4.2 Experimental results

We evaluate DRSCM from five aspects, including the general performance comparison, influence of different pretrained dense retrieval models, influence of different segment lengths on retrieval performance, influence of different aggregation strategies on retrieval performance, and contribution of the segment correlation matrix to performance improvement. Note that two embedding models, SBERT [16]

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

and RepBERT [24], are tested as pretrained models for segments and queries. In our experiments, DRSCM(SBERT) and DRSCM(RepBERT) denote the two variants of DRSCM, respectively.

4.2.1 General performance comparison

We estimate the performance of DRSCM by comparison to the traditional bag-of-words model (BM25) and several cross-encoder pretrained language models (BERT-based models). Since achievements based on cross-encoder pretrained language models come at the expense of quadratic complexity and longer interaction time, the efficiency of existing methods is undesirable. To reduce the computational complexity of long document retrieval and improve its effectiveness, we propose the DRSCM framework, which derives more accurate retrieval scores for segments by considering both local and global semantic contexts. In Table 2, we compare the results of DRSCM under the four aggregation methods (2sum, 3sum, Max, Mean) with baseline models.

On the GOV2 collection, both DRSCM(SBERT) and DRSCM(RepBERT) under four aggregation strategies outperform clearly the traditional bag-of-words model (BM25) and three cross-encoder models (i.e., Birch(MS MARCO), Vinilla_BERT, and CEDR_KNRM) in terms of effectiveness and efficiency.

Similar patterns can be found on the Robust04 collection. More specifically, both DRSCM(SBERT) and DRSCM(RepBERT) under "2sum", "3sum", and "Max" aggregation strategies perform better than the traditional bag-of-words model (BM25) and three cross-encoder models (i.e., Birch(MS MARCO), Vinilla_BERT, and CEDR_KNRM) in terms of effectiveness and efficiency. As for the "Mean" aggregation method, the proposed framework drives comparable performance with selected baselines.

In most cases, our proposed framework DRSCM outperforms stably the traditional bag-of-words model (i.e., BM25), and the cross-encoder pretrained language models. Moreover, the retrieval time of DRSCM can be tens of thousands of times less than that of most cross-encoder models.

4.2.2 Influence of different pretrained dense retrieval models

As discussed in Section 3, two dense retrieval models, SBERT [16] and RepBERT [24], are tested as the pretrained model for short segments and queries. In Table 2, we present the results of DRSCM with two different pretrained dense retrieval models (i.e., SBERT and RepBERT) under the four aggregation strategies (i.e., 2sum, 3sum, Max, and Mean), and we can find the following observations.

On the GOV2 collection, DRSCM(SBERT) achieves the best performances in terms of P@20, NDCG@10, and NDCG@20. In addition, the best P@10 value of DRSCM(SBERT) is slightly less than that of DRSCM(RepBERT). Thereafter, in general, DRSCM(SBERT) performs better than DRSCM(RepBERT) on the GOV2 collection.

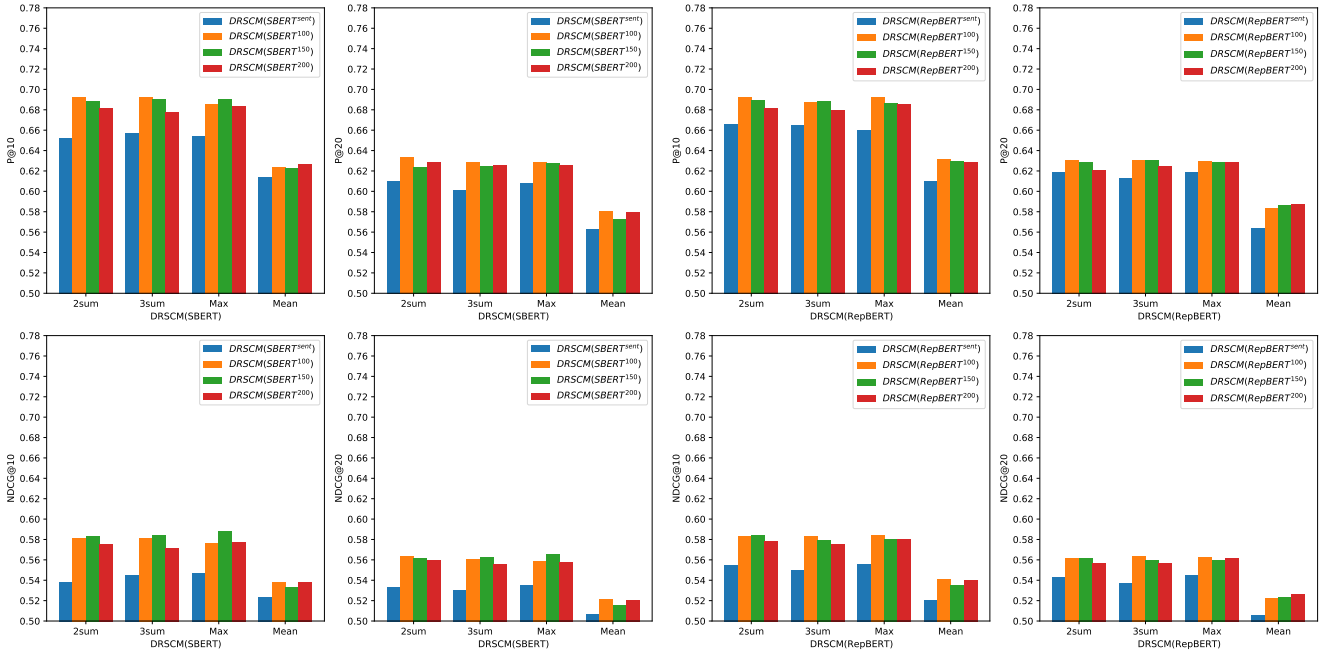
On the Robust04 collection, DRSCM(SBERT) derives the best performances in terms of P@10 and NDCG@10, while DRSCM(RepBERT) performs the best in terms of P@20 and NDCG@20. Hence, we can reach the conclusion that DRSCM(SBERT) and DRSCM(RepBERT) perform comparably on the Robust04, and both perform better than selected baselines.

4.2.3 Influence of different segment lengths

To compare the performances of DRSCM at different segment lengths, long documents are divided into small segments via two splitting strategies: sentences and overlapping passages. Based on splitting strategies, for each pretrained model, four variants are derived. More specifically, DRSCM($\text{SBERT}^{\text{sent}}$) denotes our model with splitting documents into sentences via a NLTK [2]. DRSCM(SBERT^{100}), DRSCM(SBERT^{150}) and DRSCM(SBERT^{200}) denote DRSCM(SBERT) with splitting documents into overlapping segments of 100 words, 150 words and 200 words, respectively. For DRSCM(RepBERT), four variants are obtained in the same way. The results are presented in Figure 3 and Figure 4.

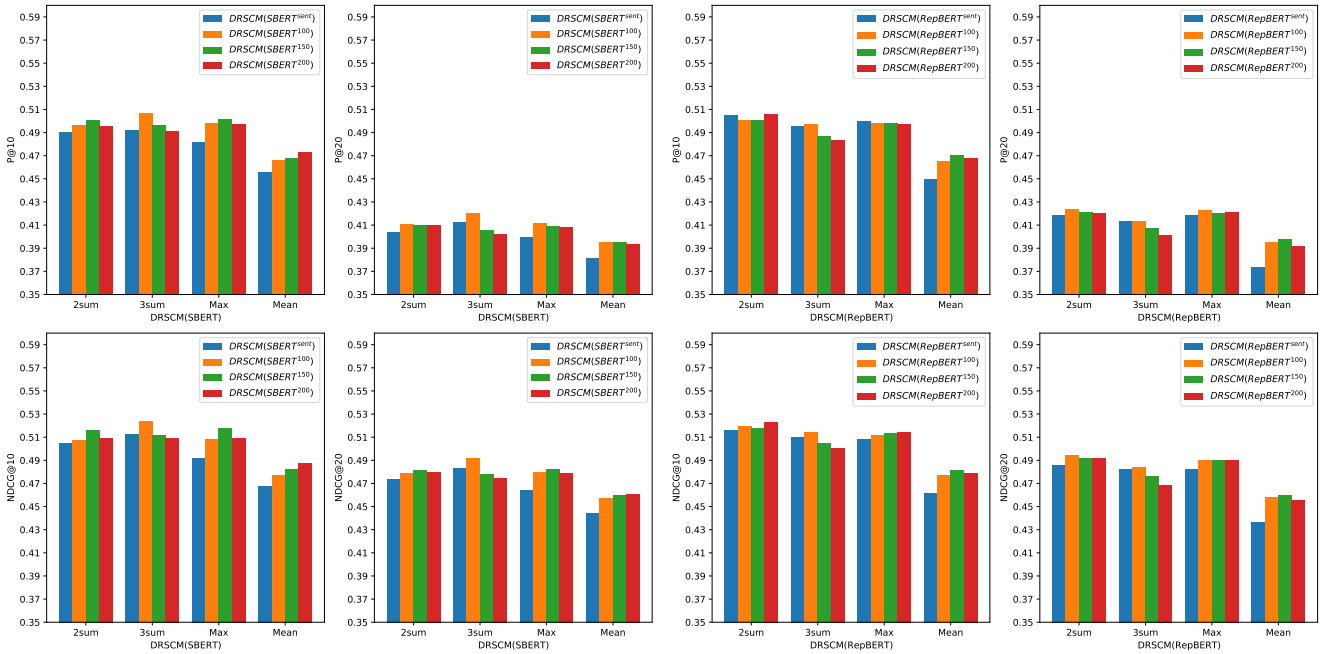
Figure 3 and Figure 4 illustrate the comparative performances on two collections (i.e., GOV2 and Robust04 collections) of different variants under 2sum, 3sum, Max, and Mean aggregation strategies. Generally speaking, for both pretrained models, the two variants with segments of 100 words and 150 words achieve better performance over other variants on the GOV2 collection. On the Robust04 collection, for both pretrained models, the two variants with segments of 100 words and 200 words achieve better performance over other variants.

[2] https://www.nltk.org/api/nltk.tokenize.html



(a) Comparative performances of DRSCM(SBERT) (b) Comparative performances of DRSCM(RepBERT)

Fig. 3: Comparative performances of DRSCM with different segment lengths on GOV2 collection.



(a) Comparative performances of DRSCM(SBERT) (b) Comparative performances of DRSCM(RepBERT)

Fig. 4: Comparative performances of DRSCM with different segment lengths on Robust04 collection.

Table 2: The comparison results of DRSCM with two different dense retrieval models and four aggregation strategies on GOV2 and Robust04.

Model	GOV2					Robust04				
	P@10	P@20	NDCG@10	NDCG@20	Latency (ms)	P@10	P@20	NDCG@10	NDCG@20	Latency (ms)
Bag-of-words										
BM25(*Anserini*)	0.5775	0.5381	0.4856	0.4784	132	0.4382	0.3631	0.4485	0.4240	88
BERT-based Models										
Birch (MS MARCO) [23]	-	-	-	-	-	0.4578	0.3964	0.4645	0.4512	290000
Vinilla_BERT [11]	0.5666	0.5483	0.4714	0.4670	129963	0.4633	0.4050	0.4750	0.4685	58400
CEDR_KNRM [11]	0.5746	0.5437	0.4618	0.4626	475940	0.4936	0.4175	0.5101	0.4832	146000
Aggregation Method "2sum"										
DRSCM(SBERT)	0.6919	**0.6335**	0.5810	0.5637	55	0.5008	0.4098	0.5157	0.4816	34
DRSCM(RepBERT)	**0.6924**	0.6300	0.5832	0.5620	28	0.5008	**0.4237**	0.5192	**0.4941**	11
Aggregation Method "3sum"										
DRSCM(SBERT)	0.6918	0.6284	0.5814	0.5606	55	**0.5064**	0.4205	**0.5238**	0.4919	34
DRSCM(RepBERT)	0.6870	0.6300	0.5828	0.5633	28	0.4972	0.4133	0.5140	0.4836	11
Aggregation Method "Max"										
DRSCM(SBERT)	0.6898	0.6270	**0.5878**	**0.5653**	55	0.5016	0.4088	0.5177	0.4821	34
DRSCM(RepBERT)	**0.6924**	0.6290	0.5840	0.5625	28	0.4980	0.4229	0.5118	0.4897	11
Aggregation Method "Mean"										
DRSCM(SBERT)	0.6233	0.5807	0.5377	0.5211	55	0.4731	0.3932	0.4873	0.4601	34
DRSCM(RepBERT)	0.6313	0.5828	0.5408	0.5228	28	0.4707	0.3974	0.4809	0.4594	11

Table 3: The evaluation results of segment correlation matrix on GOV2 and Robust04.

Models	GOV2				Robust04			
	P@10	P@20	NDCG@10	NDCG@20	P@10	P@20	NDCG@10	NDCG@20
Aggregation Method "2sum"								
SBERT^{100}	0.6118	0.5501	0.4993	0.4838	0.4550	0.3751	0.4793	0.4475
DRSCM(SBERT^{100})	**0.6186**	**0.5538**	**0.5049**	**0.4879**	0.4550	0.3743	0.4792	0.4472
RepBERT^{100}	0.6086	0.5508	0.4833	0.4718	0.4450	0.3687	0.4601	0.4318
DRSCM(RepBERT^{100})	0.6086	**0.5538**	**0.4892**	**0.4776**	0.4396	0.3673	0.4573	0.4308
Aggregation Method "3sum"								
SBERT^{100}	0.6233	0.5455	0.5109	0.4833	0.4510	0.3713	0.4835	0.4482
DRSCM(SBERT^{100})	**0.6307**	**0.5542**	**0.5176**	**0.4905**	**0.4570**	**0.3725**	**0.4852**	**0.4483**
RepBERT^{100}	0.6112	0.5515	0.4924	0.4793	0.4337	0.3596	0.4544	0.4251
DRSCM(RepBERT^{100})	**0.6186**	**0.5552**	**0.4982**	**0.4818**	0.4333	**0.3622**	0.4532	**0.4260**
Aggregation Method "Max"								
SBERT^{100}	0.5924	0.5391	0.4798	0.4684	0.4382	0.3671	0.4462	0.4245
DRSCM(SBERT^{100})	**0.5999**	**0.5425**	**0.4905**	**0.4777**	0.4382	**0.3689**	**0.4476**	**0.4275**
RepBERT^{100}	0.5824	0.5370	0.4610	0.4582	0.4229	0.3608	0.4314	0.4144
DRSCM(RepBERT^{100})	**0.5859**	**0.5411**	**0.4717**	**0.4653**	0.4221	**0.3622**	**0.4316**	**0.4155**
Aggregation Method "Mean"								
SBERT^{100}	0.4537	0.4168	0.3843	0.3694	0.3518	0.2892	0.3627	0.3392
DRSCM(SBERT^{100})	0.4523	0.4162	0.3822	0.3679	0.3502	0.2880	0.3605	0.3372
RepBERT^{100}	0.4301	0.3893	0.3657	0.3457	0.3289	0.2825	0.3471	0.3307
DRSCM(RepBERT^{100})	0.4227	0.3856	0.3602	0.3425	0.3273	0.2783	0.3440	0.3262

4.2.4 Influence of different aggregation strategies

In this subsection, we evaluate the effectiveness of different aggregation strategies. Since including more sentences has no effect on the performance improvement [23], we only consider top-k segments with $k \leq 3$ for aggregation strategies, i.e., Max, 2sum, and 3sum.

As shown in Figure 3 and Figure 4, the aggregation strategy "Mean" performs the worst on the four evaluation metrics for both pretrained dense retrieval models (i.e., DRSCM(SBERT) and DRSCM(RepBERT)). The possible reason lies in that it introduces significant noises by the Mean aggregation strategy, which considers all segments for document ranking. However, most segments in each document

might be irrelevant to the query.

The results in Figure 3 and Figure 4 demonstrate that the "2sum" aggregation strategy performs the best on almost all cases (i.e., with different segment lengths, and different pre-trained models). More specifically, on GOV2 (in Figure 3), the "2sum" aggregation strategy performs stably better than other aggregation strategies. On Robust04 (in Figure 4), the "2sum" aggregation strategy performs better than other aggregation strategies when combining with RepBERT. However, when combining with SBERT, the "2sum" aggregation strategy performs slightly worse than "3sum" aggregation strategy which is the best strategy among all aggregation strategies.

4.2.5 Effectiveness of segment correlation matrix

In this subsection, we investigate the contribution of segment correlation matrix to performance improvement. Since similar patterns are observed on the proposed framework with different segment lengths, we only present the comparative performances of DRSCM and the original dense retrieval models (i.e., SBERT and RepBERT) with segments of 100 words due to the limit of space. Note that SBERT^{100} and RepBERT^{100} denote the models without integrating segment correlation matrix M, while DRSCM(SBERT^{100}) and DRSCM(RepBERT^{100}) denote the counterparts with integrating segment correlation matrix M. The results are shown in Table 3.

The results in Table 3 demonstrate that under three top-k aggregation strategies (i.e., Max, 2sum, and 3sum), both pre-trained models with integrating the segment correlation matrix derive clearly better performance over those counterparts without integrating the segment correlation matrix. As for the Mean aggregation strategy, integrating the segment correlation matrix provides no significant contribution to performance improvement on the task of long document retrieval. The main reason lies in that it introduces noises by the Mean aggregation strategy, leading to the failure of improving performance by integrating the segment correlation matrix.

5 Conclusions and future work

To reduce the computational complexity of long document retrieval and improve its effectiveness, we propose the DRSCM framework, which derives more accurate retrieval scores for segments by considering both local and global semantic contexts. To address the issue of topic drift, DRSCM utilizes a linear combination of the segment relevance scores, and a segment correlation matrix (M) to obtain the final segment scores, integrating the global context information. We split the document into segments of different lengths, and investigate their influence on the effectiveness of the model.

Our empirical analysis and comparative studies indicate that DRSCM significantly outperforms the traditional bag-of-words model (BM25) and cross-encoder-based models. In addition, the retrieval time of DRSCM can be thousands of times faster than the cross-encoder based models.

In the future, we plan to employ finer-grained information, such as words and phrases, to improve further the effectiveness of the task of long document retrieval.

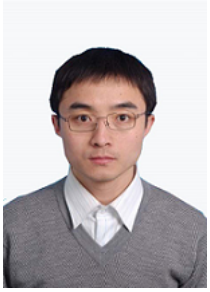
References

1. Bajaj P, Campos D, Craswell N, Deng L, Gao J, Majumder R, McNamara A, Mitra B, Nguyen T, Rosenberg M, Song X, Stoica A, Tiwary S, Wang T. Ms marco: A human generated machine reading comprehension dataset[J]. arXiv preprint arXiv:1611.09268, 2016.
2. Bowman S R, Angeli G, Potts C, Manning C D. A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326, 2015.
3. Dai Z, Callan J. 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, 985-988.
4. Devlin J, Chang M.-W, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational. 2019, 4171-4186.
5. Fedus W, Goodfellow I, Dai A M. Maskgan: better text generation via filling in the_____. arXiv preprint arXiv:1801.07736, 2018.
6. Graves A , Jürgen S. Framewise phoneme classification with bidirectional LSTM and other neural network architectures[J]. Neural Networks, 2005, 18(5-6): 602-610.
7. Hofstätter S, Mitra B, Zamani H, Craswell N, Hanbury A. Intra-document cascading: Learning to select passages for neural document ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 1349-1358.
8. Karpukhin V, Oğguz B, Min S, Lewis P, Wu L, Edunov S, Chen D, Yih W t. Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906, 2020.
9. Khattab O, Zaharia M. 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, 39-48.
10. Li C, Yates A, MacAvaney S, He B, Sun Y. Parade: Passage rep-

resentation aggregation for document reranking. arXiv preprint arXiv:2008.09093, 2020.

11. Li M, Popa D N, Chagnon J, Cinar Y G, Gaussier E. The Power of Selecting Key Blocks with Local Pre-ranking for Long Document Information Retrieval[J]. arXiv preprint arXiv:2111.09852, 2021.
12. Lin S C, Yang J H, Lin J. Distilling dense representations for ranking using tightly-coupled teachers. arXiv preprint arXiv:2010.11386, 2020.
13. Luan Y, Eisenstein J, Toutanova K, Collins M. Sparse, Dense, and Attentional Representations for Text Retrieval. Transactions of the Association for Computational Linguistics. 2021, 329-345.
14. MacAvaney S, Yates A, Cohan A, Goharian N. Cedr: Contextualized embeddings for document ranking. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019, 1101-1104.
15. Rajpurkar P, Zhang J, Lopyrev K, Liang P. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250, 2016.
16. Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. arXiv preprint arXiv:1908.10084, 2019.
17. Sang E F, De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition[J]. arXiv preprint cs/0306050, 2003.
18. Voorhees E M. Overview of TREC 2004[C]//Trec. 2004.
19. Wang S, Zhuang S, Zuccon G. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval. 2021, 317-324. doi:10.1145/3471158.3472233.
20. Wu Z, Mao J, Liu Y, Zhan J, Zheng Y, Zhang M, Ma S. Leveraging Passage-level Cumulative Gain for Document Ranking[C]// WWW '20: The Web Conference 2020. 2020, 2421-2431.
21. Xiong L, Xiong C, Li Y, Tang K F, Liu J, Bennett P, Ahmed J, Overwijk A. Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv preprint arXiv:2007.00808, 2020.
22. Yang W, Zhang H, Lin J. Simple applications of bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972, 2019.
23. Yilmaz Z A, Wang S, Yang W, Zhang H, Lin J. Applying BERT to document retrieval with birch[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations. 2019: 19-24.
24. Zhan J, Mao J, Liu Y, Zhang M, Ma S. Repbert: Contextualized text embeddings for first-stage retrieval. arXiv preprint arXiv:2006.15498, 2020.
25. Zhang X, Yates A, Lin J. Comparing score aggregation approaches for document retrieval with pretrained transformers. In: European Conference on Information Retrieval, Springer. 2021, 150-163.
26. Zhu L, Yang Y. Actbert: Learning global-local video-text representations[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 8746-8755.

Jiajia Wang is currently pursuing her Ph.D. degree with the School of Mathematics and Statistics, Central China Normal University, China. Her current research interests include information retrieval and natural language processing.



Weizhong Zhao is currently an associate professor with the School of Computer, Central China Normal University. His research interests include data mining, machine learning, and bioinformatics.

Xinhui Tu is currently an associate professor at School of Computer Science, Central China Normal University. His research interests include information retrieval and natural language processing. Since 2003, he has published more than 30 refereed papers in journals (such as the Journal of American Society for Information Science and Technology), book chapters, and international conference proceedings (such as ACM SIGIR, ACM CIKM and ECIR). He is also the principal investigator of an NSF project of China.

Tingting He received the B.E. degree and in Software Engineering from Wuhan University, Wuhan, China, in 1985, the M.Sc. degree in applied Software and Theory of Computer in 1988 from Wuhan University. and the Ph.D. degree in applied linguistics from Central China Normal University, Wuhan, China, in 2003. She is a Full Professor in School of Computer, Central China Normal University, China. Her areas of research interests are natural language processing, computational intelligence and deep learning.