Machine Intelligence Research
www.mi-research.net
DOI: 10.1007/s11633-022-1373-9

DynamicRetriever: A Pre-trained Model-based IR System Without an Explicit Index

Yu-Jia Zhou[1*] Jing Yao[1*] Zhi-Cheng Dou[1] Ledell Wu[2] Ji-Rong Wen[1]

[1] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China

[2] Beijing Academy of Artificial Intelligence, Beijing 100084, China

Abstract: Web search provides a promising way for people to obtain information and has been extensively studied. With the surge of deep learning and large-scale pre-training techniques, various neural information retrieval models are proposed, and they have demonstrated the power for improving search (especially, the ranking) quality. All these existing search methods follow a common paradigm, i.e., index-retrieve-rerank, where they first build an index of all documents based on document terms (i.e., sparse inverted index) or representation vectors (i.e., dense vector index), then retrieve and rerank retrieved documents based on the similarity between the query and documents via ranking models. In this paper, we explore a new paradigm of information retrieval without an explicit index but only with a pre-trained model. Instead, all of the knowledge of the documents is encoded into model parameters, which can be regarded as a differentiable indexer and optimized in an end-to-end manner. Specifically, we propose a pre-trained model-based information retrieval (IR) system called DynamicRetriever, which directly returns document identifiers for a given query. Under such a framework, we implement two variants to explore how to train the model from scratch and how to combine the advantages of dense retrieval models. Compared with existing search methods, the model-based IR system parameterizes the traditional static index with a pre-training model, which converts the document semantic mapping into a dynamic and updatable process. Extensive experiments conducted on the public search benchmark Microsoft machine reading comprehension (MS MARCO) verify the effectiveness and potential of our proposed new paradigm for information retrieval.

Keywords: Information retrieval (IR), document retrieval, model-based IR, pre-trained language model, differentiable search index.

Citation: Y. J. Zhou, J. Yao, Z. C. Dou, L. Wu, J. R. Wen. Dynamicretriever: A pre-trained model-based IR system without an explicit index. *Machine Intelligence Research*. http://doi.org/10.1007/s11633-022-1373-9

1 Introduction

The web search, the typical information retrieval (IR) system, has become one of the main approaches for users to obtain information in their daily life. Given a query issued by the user, it retrieves relevant documents from massive web pages and returns a document ranking list as the final result. The traditional IR algorithms rely on the inverted index to complete the above process. With the inverted index, the IR algorithms can calculate features like term frequencies, term positions, proximity, etc., of each document. Relevant documents are retrieved by counting the co-occurrence relationship between the query terms and document terms. The representative of this type of IR algorithm is the BM25[1] model, which suffers from the challenge of word mismatching. With the development of natural language processing techniques, the understanding of terms has been elevated to the semantic level, alleviating the mismatch problem. Word embedding techniques, such as word2vec[2], allow models to measure the semantic similarity between any two words. Based on this method, various neural matching models have been proposed to compute the relevance of the query term sequence and the document sequence[3, 4]. They greatly improve search engine retrieval quality and user satisfaction.

Over the past few years, advances in representation learning have led to a shift from the traditional inverted index to the dense vector index, where the IR system first encodes all documents into dense vectors and retrieves relevant documents based on the matching score between the vectors of the query and documents[5–8]. Recently, state-of-the-art pre-trained language models (PLM) show a strong capability of involving contextual information to understand text sequences better[9–11]. Motivated by this, some studies tried to explore the use of PLM for IR[12–16], especially for the dense retrieval task. Considering that using matching tasks for pre-training is more suitable for the IR scenario, pseudo query-document pairs are constructed from the large corpus based on several strategies. These studies show that leveraging pre-trained language models can generate more accurate query and document representations to improve retrieval performance[12, 15–17].

Research Article
Manuscript received June 30, 2022; accepted August 31, 2022
Recommended by Associate Editor Zhi-Yuan Liu
*These authors contribute equally to this work
© Institute of Automation, Chinese Academy of Sciences and Springer-Verlag GmbH Germany, part of Springer Nature 2023

Despite the great progress made by previous research, advanced IR models have the same framework[1, 18, 19], i.e., index-retrieval-rerank, as traditional IR systems from decades ago, which includes three steps: 1) building an index for each document in the corpus; 2) retrieving a set of documents based on the query; 3) computing the relevance and reranking the candidate documents. This framework enables search engines to retrieve a small number of documents with low query latency, and then rerank them through deep semantic matching. Recently, Metzler et al.[20] proposed that this fixed framework can be optimized using a unified model. Under this framework, differentiable search index (DSI)[21] was presented and tried a variety of document identifiers and training strategies. Their experimental results showed the potential of model-based IR for document retrieval. However, the design of their training samples is relatively simple, resulting in a model that is not robust enough on datasets with insufficient supervised data. In this paper, we intend to delve deeper into the pre-training strategies of this model-based framework for document retrieval. As for language models, the pre-training stage can be seen as the process of learning the basic meaning of each word and the dependencies between words. Similarly, for our model-based IR system, we can regard each document as an individual token and encode the knowledge of all documents in the corpus into the model through pre-training. With such a model being aware of both the semantic knowledge and document identifiers, we can complete a variety of downstream IR tasks, including document retrieval, response generation, document summarization, etc.

Specifically, we propose a pre-training model-based IR system without an explicit index, called DynamicRetriever. It comprises two modules: A PLM encoder to obtain the semantic representation of text sequences, i.e., queries and document passages, and a docid decoder, which keeps a vocabulary of document identifiers and learns a vector for each docid. For a given query, DynamicRetriever first encodes this query into a context-aware representation vector and then directly outputs document identifiers through the docid decoder, which is different from previous index-based retrieval methods. This framework has several advantages compared to traditional index-based IR systems. First, the model-based approach parametrizes the traditional static index. This allows the model's understanding of the document content to be a dynamic process that can be updated during training. Second, the separated indexing and retrieval stages can be optimized in an end-to-end manner, which will increase the fitness of the model for the document retrieval task.

Similar to advanced language models, the training process of our model-based IR system includes pre-training and fine-tuning. At the pre-training stage, the semantic information of each document identifier can be memorized in the model through multiple pre-training tasks. At the fine-tuning stage, the model attempts to learn the query-docid relations with labeled query-document pairs. Specifically, we implement two variants of the DynamicRetriever model. The first variant is called Vanilla model, which trains the docid decoder module from scratch. However, since each document identifier is independent and learned separately, there is no obvious relatedness between the vectors of two different document identifiers, even though they share similar semantics. Due to the weak generalizability, the model struggles to predict documents correctly for those who lack fine-tuning data. Therefore, we propose the Over-Dense model, which combines the advantages of the model-based IR system and dense retrieval models. It uses existing vectorized indexes to initialize the docid decoder, strengthening the model's understanding of each doc identifier.

We conducted experiments on Microsoft machine reading comprehension (MS MARCO) and natural question datasets to test the performance of the document retrieval task. Experimental results show that our proposed DynamicRetriever, which involves document identifiers into model parameters, is helpful for improving the retrieval results and has the potential to be scaled up.

In conclusion, the contributions of this paper are three-fold:

1) Along with the model-based IR blueprint, we propose DynamicRetriever, an end-to-end document retrieval model with various pre-training tasks. To alleviate the problem of limited supervised data, we devise diverse pre-training tasks to encode richer knowledge of documents into model parameters and enhance the downstream retrieval quality.

2) To enhance the robustness of the model, we propose to combine the advantages of dense retrieval models. We apply dense vectors to initialize the parameters of the model, which will make the model converge faster and work better.

3) Experimental results on two public datasets show the significant improvement of our model over DSI, and also reveal that initializing the parameters with dense vectors can enhance the robustness of the model for document retrieval.

The remainder of the paper is arranged as follows. Related works are summarized in Section 2. The proposed models are introduced in Section 3. The experimental settings and results are shown in Sections 4 and 5. Finally, the conclusion is drawn in Section 6.

2 Related work

As stated in Section 1, existing information retrieval models follow the index-retrieve-rerank paradigm. The indexing and retrieval components are crucial and have been widely studied, especially in recent years when deep

learning and large-scale pre-trained language models are developing. In this section, we briefly review the related works of this paper, including sparse retrieval and dense retrieval.

2.1 Sparse retrieval

Sparse retrieval is a traditional method for document indexing and retrieval. It first builds an inverted index based on all documents in the corpus, which encodes term frequencies, term position, document structure information, document length, etc. Then, it retrieves relevant documents based on the matching between query terms and document terms. How to measure the relevance between terms and the weights of different terms is the main challenge for sparse retrieval. The classical TF-TDF and BM25[1] methods employ the term frequency and precise word matching, achieving great results. Furthermore, several works apply neural networks to improve the performance of sparse retrieval from the semantic aspect. Word embedding techniques[2, 22–25] are introduced to better measure the semantic similarity between different query terms and document terms[26], alleviating the mismatch problem. Some deep-learning based models[27, 28] also expand possible terms for the issued query to improve the recall. Deep contextualized term weighting framework (DeepCT)[29] and context-aware document term weighting framework (HDCT)[30] employ the large-scale language model bidirectional encoder representations from transformers (BERT)[9] to predict the term weights, instead of traditional term frequency.

2.2 Dense retrieval

Dense retrieval is a representation-based method for indexing and retrieval. First, it applies a neural network to encode each document into a dense vector and builds a vectorized index. Then, it embeds the issued query into the same latent space and computes the similarity between the query representation and document vectors to efficiently retrieve relevant documents[31], where the inner product, cosine similarity, and efficient K-nearest neighbor search[32] could be used. Compared to sparse retrieval, encoding the query and documents into low-dimensional vectors for matching is promising to capture rich semantic and contextual information and provide a way to alleviate the vocabulary mismatch problem. However, due to the precise match between tokens being ignored, the precision may be sacrificed. With the development of neural models and large-scale pre-trained language models to better learn contextual information of documents, such as Transformer[33], BERT[9], and generative pre-training (GPT)[10], dense retrieval is receiving more and more attention, which has been demonstrated to outperform sparse retrieval[18, 34–37]. Some studies attempt to explore the use of PLM for IR[12, 15–17]. They design various strategies to construct pseudo query-document pairs from the large corpus and pre-train a model to generate query and document representations more accurately, improving retrieval performance.

Different from the above retrieval approaches with a separate index, we explore a new search paradigm and propose a model-based IR system without an explicit index in this paper.

2.3 Model-based retrieval

In order to solve the problem that traditional dense retrieval cannot be optimized end-to-end, model-based retrieval[20] is proposed to encode the knowledge of all documents into a model and retrieve the relevant document identifiers directly. Under this framework, Cao et al.[38] try to retrieve entities through a seq-to-seq model, which regards the query as input and outputs the entity names. Tay et al.[21] devise DSI to regard Transformer as a differentiable search index, and directly output docids for document retrieval. Chen et al.[39] propose generative evidence retrieval (GENE) to retrieve evidence by a generative language model. However, existing works suffer from limited supervised data. In this paper, we attempt to devise more reasonable pre-training tasks and explore the fusion of model-based retrieval and dense retrieval.

3 DynamicRetriever: A pre-training model-based IR framework

Existing search methods follow an index-retrieval-rerank framework that has dominated IR systems for decades. They first encode the document content into a sparse inverted index or a dense vector index, then retrieve and rerank documents based on the similarity between the query and documents, where the index is always a necessary part. With the advent of pre-training techniques, we envision that the model can involve both the semantic information and corresponding document identifiers through pre-training, thereby replacing traditional static indexes. In such a model-based IR system, given a query, the document identifiers can be directly generated as a result. In this section, we propose a basic pre-training model-based IR system named DynamicRetriever and implement two variant models.

3.1 Model architecture

The whole architecture of DynamicRetriever is shown in the right part of Fig. 1. It works in two main steps: Given a query, the model first encodes it into a vector with a PLM encoder and then maps it to the docid vocabulary through a docid decoder and outputs document identifiers. Compared to the dual encoder model presented in the left part of Fig. 1, which computes a static vectorized index for all documents using a trained

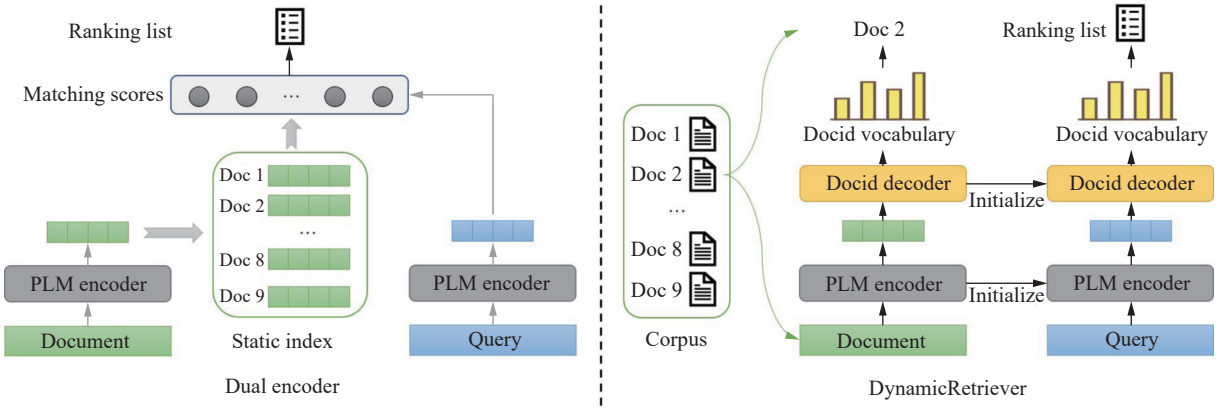

Fig. 1 Comparison between the dual encoder model and DynamicRetriever

encoder, DynamicRetriever stores the semantic information and all document identifiers in the parameters of the docid decoder. This can be viewed as the dynamic index that is updated directly as the model training. In the following, we describe the workflow of DynamicRetriever in detail.

First, given a query containing n tokens, i.e., $q = \{w_1, w2, \cdots, w_n\}$, we are supposed to understand this query for analyzing the user's information need. The queries issued by users are often very short, which leads to much ambiguity in understanding. Therefore, it is crucial to model queries in a fine-grained way. We apply a Transformer-based PLM encoder to compute the sentence embedding of q, denoted as

$$V^q = \text{Transformer}^{cls}([w_1, w_2, \cdots, w_n]). \tag{1}$$

We take the output of *cls* as the query representation V^q.

Second, with the encoded query representation, the target of our model is to directly generate the most relevant document identifiers in the entire corpus. To implement this goal, we feed V^q into the docid decoder to obtain a probability distribution over all the document identifiers. Formally, assuming that there are D document identifiers in the corpus, the probability distribution is calculated by simulating the output layer of generative language models:

$$O^q = \text{softmax}(W_{doc}^{\mathrm{T}} \times V^q) \tag{2}$$

where $W_{doc} \in \mathbf{R}^{d_{model} \times D}$ is a project matrix to map the query representations to the probability of each docid. It can be viewed as dynamic indexes that can be updated during the model training. According to the output O^q, we are able to retrieve the top-k document identifiers by sorting the probability for the given query q.

3.2 Encoding document identifiers into model

The training of pre-trained language models such as BERT concludes the pre-training stage and the fine-tuning stage. The pre-training of the language model focuses on learning the basic semantics of words and the semantic dependencies between words. At the fine-tuning stage, the model will enhance the ability to handle specific tasks. Similar to these PLMs, at the pre-training stage of DynamicRetriever, we hope the semantic information of each document identifier can be memorized in the model through multiple pre-training tasks. The fine-tuning stage is used to learn the matching relationships between queries and document identifiers. During this process, a large number of docid-level meta information can be captured over term-level semantics, thereby improving the ranking quality. Under such a training framework, we propose two variant models with different training strategies: 1) Vanilla model, which warms up the model parameters with pre-training tasks and fine-tunes the model over the query-docid matching data. 2) OverDense model, which initializes the docid decoder parameters over trained dense vectors and then continues training with query-docid relations. The workflows of the current dense retrieval method and the two variant models of DynamicRetriever are shown in Fig. 2.

3.3 Vanilla model: Training from scratch

The Vanilla model initializes the projection matrix W_{doc} randomly and trains it from scratch. Firstly, we devise three pre-training tasks to encode the semantics of each document identifier into the model. Then, we use labeled query-document pairs to fine-tune the model parameters by capturing docid-level features.

Pre-training. This stage is designed to learn the knowledge from the large corpus, which is used to pre-train the semantics of each docid in the large corpus. A critical step is to extract self-supervised signals from the corpus and construct (term sequence-docid) pairs. We try three strategies that may contribute to pre-training. As shown in Fig. 3, they are:

1) Training with passages. Previous studies have

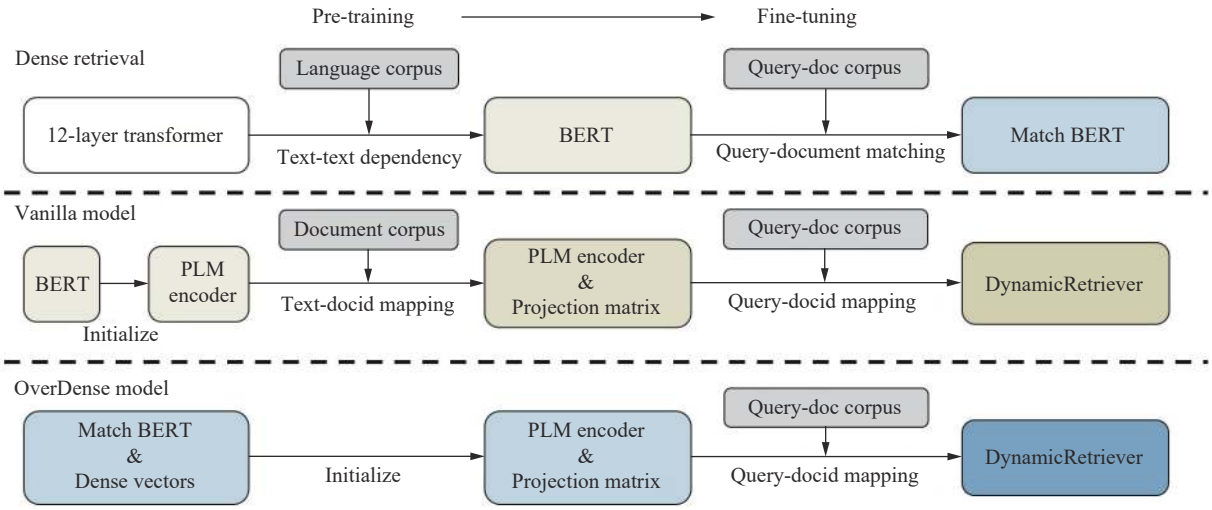

Fig. 2 Model training workflows of dense retrieval and two variant models of DynamicRetriever

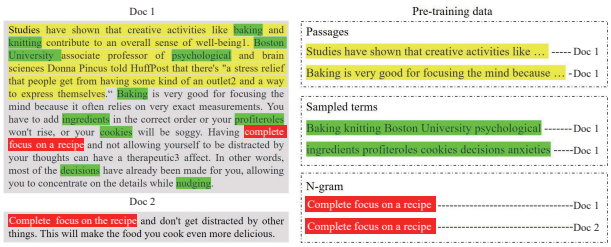

Fig. 3 Pre-training tasks of the Vanilla model

shown that using passage-level evidence for document ranking can enhance the ranking quality[40]. Inspired by this, we attempt to segment the document text into multiple passages with fixed-length windows. Each passage can reflect a local view of the document content. For the document $doc1$, whose content can be divided into m passages, we can construct m pairs for training, i.e., $(\text{passage}_1, doc1), \cdots, (\text{passage}_m, doc1)$.

2) Training with sampled terms. The importance of each word in the document is different, and often some important words can reflect the basic content of the document. Therefore, we sample terms according to the word importance with random length (from 10 to 512). Formally, for the document $doc1$, after m times of sampling, we obtain m sets of terms for training, i.e., $(\text{set}_1, doc1), \cdots, (\text{set}_m, doc1)$.

3) Training with ngram. Ngram is a sequence of n words, which may appear in multiple documents. To some extent, this sequence can characterize the similarity between multiple documents. It can be seen as an enhanced version of the inverted table. For a sequence of n words, supposing that we find it in m documents, then the training pairs can be formed as $(n\text{-gram}, doc1), \cdots, (n\text{-gram}, docm)$.

Fine-tuning. After pre-training, the model has memorized the basic semantics of each docid. The fine-tuning stage is used to learn the query-docid relations with supervised matching data. Different from traditional two-tower matching models focusing on text matching, our model pays more attention to bridging the gap between terms and document identifiers through training. For the query q, its representation is denoted as V_q. We choose cross entropy as the loss function:

$$\mathcal{L} = \sum_i y_i \times \frac{\exp(u_i^{\mathrm{T}} \times V^q)}{\sum_{j=1}^{\mathrm{D}} \exp(u_j^{\mathrm{T}} \times V^q)} \tag{3}$$

where u_i is the i-th column of the projection matrix W_{doc}.

However, there are two shortcomings of the Vanilla model. 1) The learning of mapping relations from query text to docid is overly dependent on the fine-tuning task. With limited fine-tuning data, a large number of document identifiers can only be learned from a small number of samples in the pre-training tasks. If we lack enough fine-tuning data, the model will perform poorly. 2) The dependencies between docids are difficult to capture, which leads to a poor generalization ability of the model. To overcome these issues, we propose the OverDense model to incorporate the benefits of dense retrieval into the model-based IR system.

3.4 OverDense model: Training over dense vectors

By comparing the DynamicRetriever and the dense retrieval models, we find that their advantages are complementary. For example, a major advantage of dense retrieval is that it can enhance the generalization of the model. Moreover, it is good at extracting term-level features in a fine-grained manner. These advantages are exactly what our model lacks. If we combine their advantages, a model with strong generalizability and multi-level feature extraction can be trained to enhance the performance. Based on this consideration, we attempt to integrate the advantages of dense retrieval models into our framework. We devise the OverDense model, which has

different parameter initialization strategies compared with the Vanilla model.

To initialize the parameters of each document identifier in the model, the Vanilla model constructs self-supervised data to learn the text-docid relations. However, with this approach, it is difficult to exploit the relationship between document identifiers. Therefore, we propose a new framework to train the model, which has three steps:

Fine-tuning the dense model. At this step, we fine-tune a two-tower BERT with query-document pairs. Two-tower BERT is a typical framework of PLM-based dense retrieval. Endowed with the benefit of PLM's powerful semantic modeling capability, this framework greatly improves retrieval quality while enhancing the generalization ability of the model compared to sparse retrieval. To strengthen the performance of BERT on the query-document matching task, we use labeled query-document pairs to fine-tune the two-tower BERT, which will be used in the next step.

Initializing DynamicRetriever. After fine-tuning the BERT for matching, we can compute the dense vectors of each document. These vectors fully integrate the textual semantic information of the documents, so documents with similar texts have higher vector similarity. If we initialize the projection matrix of the docid decoder with dense vectors, the problem of poor model generalization can be alleviated. After initializing W_{doc} with dense vectors, our model can achieve the same performance as dense retrieval. Continuing to train on this basis, our model can pay more attention to docid-level information, thereby improving the model's performance on document retrieval.

Fine-tuning DynamicRetriever. The initialized parameters have fully recorded the semantic information of each docid, and also have integrated the text-text matching information. Next, we will extract text-docid matching relations from supervised data, and continue training the model. We expect the model to capture docid-level features such as authority. For example, there are many forwarded news on the Internet, and the content is roughly the same, but the publishers are different. Users tend to choose the more authoritative one. Traditional PLM-based methods try to model such information at the term-level, which is an indirect way of information loss. In contrast, our model can model the docid-level features by directly updating the representation of the document identifier without relying on terms.

3.5 Potential of scaling up

The model-based IR system uses model parameters in place of traditional static document indexes. A natural problem is how to scale the model to a larger corpora. As the number of documents increases, the model has to use more and more parameters to memorize document identifiers. Due to memory constraints, the number of parameters of our single model cannot grow infinitely. This prompts us to think about how to deal with large-scale corpus scenarios.

Distributed model. Now that our model can work on small-scale data, we can train multiple sub-models distributedly and then fuse their predictions to get the final document ranking list. As shown in Fig. 4, we devise multiple models with the same structure, and each model is responsible for learning the mapping relationships of a part of the document identifiers. Finally, given a query, each model can compute a probability distribution over some document identifiers. By a merge function, we can get the whole probabilities and generate the most relevant docid. However, merging the outputs in such a simple way may cause another problem. Due to different sub-models being trained independently, the scale of document scores of different sub-models is not consistent. Therefore, more suitable merge functions or training strategies are needed. A possible solution is to add some common documents into different sub-models so as to scale the space of each one to the same level. This question requires more exploration in the future.

3.6 Discussion

As more and more researchers focus on model-based IR, we would like to share some of our reflections and experiences in the following discussion.

What are the advantages of model-centric framework over traditional index-based IR methods?

We summarize the advantages of model-based IR in three aspects. For retrieval tasks, the model-centric ap-

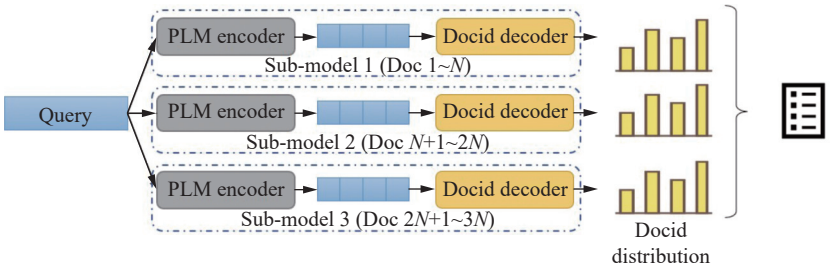

Fig. 4 Architecture of the distributed model

proach can break the gap between terms and docids and optimize the model end-to-end. For IR scenarios, a consolidated model can perceive the knowledge of all documents in the corpus, thereby enhancing a variety of IR tasks, like question answering, and document summarization. At the cognitive level, end-to-end generative models are closer to the way humans think. Such models are more likely to have the ability to understand and reason, not just memorize and match.

How to deal with incremental documents, whose identifiers are completely new to the model?

Since each docid is a random integer, the newly added docids bring great challenges to model-based IR systems. A possible solution is to periodically update the decoder part of the model, especially the dimension of the mapping matrix. We can take the strategy of the Over-Dense model, use the dense vector to initialize this part of the newly added parameters, and then update them with the model training.

4 Experimental settings

4.1 Dataset

We complete all experiments on two public datasets for the document retrieval task. Their statistics are presented in Table 1.

Table 1 Statistics of MS MARCO and different data subsets

Dataset	#Doc	#Passage	#Train	#Valid
MS MARCO				
Top 100 K	100 K	955 586	147 086	466
Top 200 K	200 K	1 763 726	247 086	636
Top 300 K	300 K	2 721 974	347 086	778
Random 100 K	100 K	838 527	11 262	156
Random 200 K	200 K	1 656 273	22 907	317
Random 300 K	300 K	2 477 582	34 290	487
Whole	3.2M	25 600 715	367 013	5 193
NQ	231 695	2 281 472	307 373	7 830

MS MARCO is a large-scale dataset collected from the field of machine reading comprehension, which is widely used on various tasks after its release, including question answering, passage ranking, document ranking, and so on. In this paper, our experiments are conducted on the MS MARCO Document Ranking benchmark, and we mainly focus on the results of Ad-Hoc retrieval. This dataset contains a total of 3.2 million candidate documents with a mean document length of 1 600 terms. The training set has 367 013 queries, while the testing set has 5 193 queries. For each training or testing query, there is a positive document in the document set, which contains at least one passage manually annotated as positive to answer the corresponding query. All the testing queries are distinguished from the training queries, and there is little overlap between their corresponding positive document sets. According to the above statistics, about 300K+ documents are annotated as relevant and can be used to construct query-document pairs for model training. Thus, to evaluate the performance and scalability of our model on datasets with different distributions, we consider different sampling methods to construct candidate document sets and conduct extensive experiments for comparison. At first, we rank all candidate documents based on their click frequency and select the top 100 K, 200 K, 300 K documents to construct different document sets for evaluation (corresponding to Top 100 K, 200 K, and 300 K in Table 1). In addition, we also randomly sample 100 K, 200 K, and 300 K documents from the whole document set for analysis (corresponding to Random 100 K, 200 K, 300 K in Table 1), as well as the whole set with 3.2 million documents. As for all these subsets, we only consider the training queries and testing queries whose clicked documents exist in this set.

Natural Question (NQ)[41] is a challenging dataset collected for question-answering research. Each piece of data includes a question from a real user and a Wikipedia article that is used to extract answer to the question. Using the uniform resource locator (URL) as a distinction, there are a total of 231 695 Wikipedia articles in the corpus. NQ contains 307 K training pairs of questions and Wikipedia pages, and 8 K development pairs. We use this dataset to evaluate the performance of models on document retrieval, where the task is to retrieve the Wikipedia page paired with the question.

4.2 Evaluation metrics

For the document retrieval task, we use the typical metric Recall@k where $k = 20$ to evaluate the recall power of our model and all the baselines. In addition, we also pay attention to the document ranking quality and apply mean reciprocal rank (MRR) for evaluation.

4.3 Baselines

In order to confirm the effectiveness of our model, we select several baselines for comparison, including the classical BM25 algorithm for sparse retrieval, the recent dense retrieval methods based on BERT and the preliminary work for model-based IR.

BM25. It is a bag-of-word retrieval method that ranks the candidate documents based on the TF-IDF weights of the query terms appearing in each document, traditional but effective[1].

BERT-Dual. With the development of large-scale pre-trained language models such as BERT, the dual en-

coder framework for dense retrieval has become popular. It encodes the query and document into representation vectors, and computes the dot product between them as the ranking score. We first use the query-document pairs constructed from the training data to fine-tune the parameters of the BERT dual encoder and generate the representation vectors for all candidate documents. When testing, we encode each query into a vector and retrieve the documents with the largest ranking scores. For all input sequences, we set the max sequence length as 512[9, 18]. The learning rate is set to 5E–5, and the batch size is 64.

DSI-Semantic. As a preliminary exploration of model-based IR, DSI[21] attempts to accomplish information retrieval with a single Transformer. It applies a text-to-text PLMs (T5 in their paper) to encode the corpus into the parameters and learns to directly map queries to relevant docids. The variations in document identifiers are the key problems, including atomic, naive string, and semantic clusters. In this paper, we take the best variant DSI-Semantic for comparison, which clusters all documents into a decimal tree based on semantics and uses the paths as docids. We follow the settings stated in the original paper for reproduction.

D-Vanilla and D-OverDense indicate the two variants of our proposed DynamicRetriever IR system.

4.4 Implementation details

In our DynamicRetriever model, the query understanding component is initialized by the pre-trained bert-base-uncased model (12 layers, hidden states 768-dimension) from the Transformers[1], and the document prediction module is a $768 \times N$ linear layer where N corresponds to the size of the candidate document set. The maximum length of tokens inputted into the BERT encoder is set to 512. In the pre-training stage, for each document, we constructed ten pieces of passages and one sampled term sequence as the training samples. Only the queries in the training set are used for fine-tuning. We perform ten epochs of pre-training and ten epochs of fine-tuning. For both stages, AdamW is applied to optimize the parameters with the learning rate of 5E–5. All experiments are completed with NVIDIA-V100 (32 GB). And the batch size is set as large as possible on NVIDIA-V100 (32 GB).

[1] https://huggingface.co/bert-base-uncased/tree/main

5 Experimental results

We conducted extensive experiments to confirm the advantages of our proposed DynamicRetriever. In this section, we present the experimental results and make some analyses.

5.1 Overall performance

To start with, we compare DynamicRetriever with the selected baselines on various data subsets with different scales and data distributions to verify its effectiveness and scalability. The results are illustrated in Tables 2 and 3. We come to several conclusions as follows.

1) In all data subsets, our proposed DynamicRetriever achieves better results than BM25, BERT dual encoder model and the DSI-Semantic. The D-OverDense model performs the best, which significantly outperforms all baselines with paired t-test at $p < 0.05$ level. BM25 applies a traditional sparse index and retrieves relevant doc-

Table 2 Overall performance on the MS MARCO dataset. "DSI-S", "D-Vani", and "D-OverD" indicate "DSI-Semantic", "D-Vanilla", and "D-OverDense", respectively. "†" denotes that the result is significantly better than other models in the t-test with a $p < 0.05$ level. The best results are in **bold**.

Model	Top 100K		Top 200K		Top 300K	
	Recall@20	MRR	Recall@20	MRR	Recall@20	MRR
BM25	0.548 (–33.8%)	0.281 (–33.7%)	0.469 (–41.7%)	0.197 (–51.0%)	0.419 (–49.0%)	0.174 (–58.2%)
BERT	0.828 (–)	0.424 (–)	0.803 (–)	0.402 (–)	0.820 (–)	0.417 (–)
DSI-S	0.702 (–15.2%)	0.394 (–7.0%)	0.612 (–23.7%)	0.345 (–14.0%)	0.626 (–23.7%)	0.344 (–17.5%)
D-Vani	0.878† (6.0%)	0.564† (33.0%)	0.756 (–5.7%)	0.462 (14.9%)	0.648 (–21.0%)	0.366 (–12.2%)
D-OverD	**0.886**† (7.0%)	**0.573**† (35.2%)	**0.871**† (8.5%)	**0.522**† (30.0%)	**0.853**† (3.9%)	**0.488**† (16.9%)
Model	Random 100K		Random 200K		Random 300K	
	Recall@20	MRR	Recall@20	MRR	Recall@20	MRR
BM25	0.582 (–26.3%)	0.361 (–34.0%)	0.520 (–25.3%)	0.311 (–29.5%)	0.486 (–23.8%)	0.281 (–24.5%)
BERT	0.790 (–)	0.546 (–)	0.697 (–)	0.441 (–)	0.639 (–)	0.372 (–)
DSI-S	0.424 (–46.3%)	0.251 (–54.0%)	0.384 (–44.8%)	0.182 (–58.7%)	0.342 (–46.4%)	0.138 (–62.8%)
D-Vani	0.692 (–12.4%)	0.499 (–8.8%)	0.213 (–69.5%)	0.143 (–67.5%)	0.123 (–80.9%)	0.104 (–72.2%)
D-OverD	**0.842**† (6.6%)	**0.645**† (18.0%)	**0.783**† (12.4%)	**0.495**† (12.4%)	**0.699**† (9.4%)	**0.409**† (9.8%)

Table 3 Overall performance on the NQ dataset

Model	Recall@20		MRR	
BM25	0.586	–22.4%	0.236	–34.7%
BERT	0.756	–	0.361	–
DSI-Semantic	0.592	–21.6%	0.237	–34.2%
D-Vanilla	0.115	–84.8%	0.053	–85.2%
D-OverDense	0.763	0.99%	0.362	0.19%

uments based on precise matching between the query terms and document terms, while the BERT-based dual encoder is state-of-the-art for dense retrieval, which builds a vectorized index based on the document semantics. In our DynamicRetriever, the document index is parameterized and embedded into a large-scale model, implementing a model-based IR system. We analyze the reasons why our new method achieves better results exactly correspond to its advantages: a) DynamicRetriever uses a consolidated model to optimize the indexing and retrieval stage in an end-to-end way; b) it keeps a dynamic index of all documents that can be updated during model training. DSI-Semantic is also a model-based information retrieval approach, and its effect highly depends on the representation of document identifiers.

2) Observing the results of the D-Vanilla model, we find that this intuitive strategy can achieve great performance in small subsets. Specifically, on the Top 100 K subset, the D-Vanilla model improves BERT by 6.04% on the evaluation metric Recall@20, and 33.01% on MRR. However, as the data scale increases, the performance of the D-Vanilla model drops sharply, especially for the subsets with random 200 K and 300 K documents. We analyze the possible reason is that different from the previous dense retrieval framework, and our docid-level DynamicRetriever system regards each document independently. Thus the corresponding training data for each document is much less than that for each token. With the increase of document corpus size, the difficulty of distinguishing between documents increases; thus the retrieval effect naturally decreases.

3) Compared to the D-Vanilla model, the performance of our improved D-OverDense model is much better. Whether on the data subset of 100 K documents or the later expanded 200 K and 300 K subsets (Top or Random) and the NQ dataset, the performance of the OverDense model is consistently better than all baselines, showing strong scalability. Especially in subsets with 200 K and 300 K documents, the D-OverDense's results are significantly improved compared to the D-Vanilla model. Unlike the D-Vanilla model, which uses the pre-training tasks, we design to learn the parameters for indexing documents, the OverDense model uses fine-tuned two-tower BERT model to generate the document representations for initializing this part of the parameters. Then, a further fine-tuning task based on Q-D pairs is conducted to fully use supervised information. Therefore, the performance of D-OverDense can be greatly improved.

In summary, the experimental results prove that our proposed DynamicRetriever, which considers using model parameters as dynamic document indexes and capturing docid-level information, is helpful in improving document retrieval results.

5.2 Ablation study

In our DynamicRetriever, there are several pre-training tasks and a fine-tuning task to parameterize the index of documents. We conducted ablation studies to analyze the effect of each task and display the results in Table 4.

Table 4 Ablation study of our models. "w/o fine-tune" is to test the model performance on zero-shot learning

Model	Top 100 K			
	Recall@20		MRR	
D-Vanilla	0.878	–	0.564	–
w/o pre-train	0.010	–98.9%	0.002	–99.7%
w/o fine-tune	0.532	–39.4%	0.290	–48.5%
D-OverDense	0.886	–	0.573	–
w/o fine-tune	0.828	–6.6%	0.424	–26.0%

We considered dropping several tasks listed as follows.

D-Vanilla w/o pre-train. As for the D-Vanilla model, we train it with passages segmented from each document, sampled terms, and n-grams to embed document information into model parameters. We discard this pre-training task for verification.

D-Vanilla w/o fine-tune. We skip the fine-tuning stage with Query-Document pairs and directly evaluate the D-Vanilla model after pre-training.

D-OverDense w/o fine-tune. This variant drops the fine-tuning stage and tests the D-OverDense model initialized with the doc representation generated by BERT dual encoder.

Observing the results in Table 4, we find that removing pre-training tasks will significantly damage the results on all evaluation metrics. This indicates that the pre-training tasks indeed embed the token-level and contextual information of the documents into the model parameters. Passages tend to contain more contextual information, and sampled terms focus on important tokens. In addition, discarding the fine-tuning task also greatly impacts the two models' performance. This result confirms the effectiveness of fine-tuning with the Q-D pairs. In our DynamicRetriever, all documents are separated, so that the fine-tuning task mainly captures the docid-level information, which proved important for document retrieval and ranking.

5.3 Performance on different queries/

documents

To verify the model in a more fine-grained way, we split the queries and documents into several subsets according to their characteristics and evaluated the model performance on them.

Overlap queries versus non-overlap queries. As stated in Section 4.1, only a part of the test queries corresponds to documents with training data, according to which we split all test queries from the random 100 K subset into "overlap query" and "non-overlap query". The evaluation results of all models on the two query sets are shown in Fig. 5. The BM25 model and BERT dual encoder show stable performance on both query subsets, whereas our D-Vanilla and D-OverDense show much more significant improvements on overlap queries. Especially for the D-Vanilla model, it performs worse than BERT on non-overlap queries but achieves better results on the overlap query set. As a model with both strong abilities of generalization and feature extraction, D-OverDense shows the best results on both query subsets.

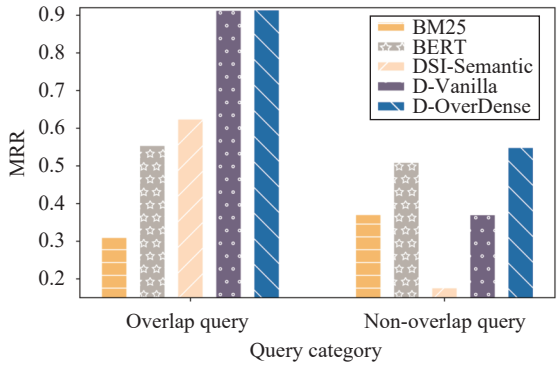

Fig. 5 Results on different query subsets

Additive documents. In practical application scenarios, there are continuously new documents and webpages. In order to evaluate the cold-start ability of our model to deal with such additive documents, we conduct experiments on how models trained with the top 100 K dataset perform on another 10 K documents. The comparison results are presented in Table 5. The DSI-Semantic model and D-Vanilla model can not be directly extended to new documents without initialization and the pre-training stage, because the model is not aware of these document identifiers. Benefiting from the generalization, BM25, BERT dual encoder, and D-OverDense perform well on these additive documents, where D-OverDense is the best. The result confirms the advantage of D-OverDense in combining both strong generalizability and multi-level feature extraction.

5.4 Exploration of distributed model

Our proposed DynamicRetriever replaces the traditional index of documents with model parameters, which naturally needs to consider the problem of how to deal with massive web documents. In Section 3.5 of this paper, we briefly discuss this challenge and provide two potential solutions. Here, we attempt to explore the method of the distributed model. The experiments are conducted on the whole 3.2 million document collection of MS MARCO. We randomly divide all these documents into 32 groups, where each group corresponds to 100 K documents, and each group owns an individual D-OverDense model for indexing and retrieval. When testing the distributed model, given a query, the D-OverDense model of each group will retrieve and return the top 100 documents, and then these documents are merged into a ranking list according to their relevance score and returned as the final ranking result. The experimental results are shown in Table 6. We analyze the potential and existing challenges of this method as follows.

Table 5 Results for additive documents (cold-start problem)

Model	Recall@20		MRR	
BM25	0.419 4	−42.0%	0.196 4	−55.9%
BERT	0.806 4	–	0.445 2	–
DSI-Semantic	–	–	–	–
D-Vanilla	–	–	–	–
D-OverDense	0.825 9	+2.42%	0.468 2	+5.17%

Table 6 Exploration of distributed model for massive documents

Model	MS MARCO		
	Recall@1	Recall@20	MRR
Each group	0.523 2	0.842 3	0.644 5
BERT	0.166 5	0.632 1	0.281 7
Distributed model	0.101 1	0.472 4	0.189 5

From Table 6, we can find that an individual D-OverDense model performs well on any document group, which indicates that our approach is promising to locate relevant documents accurately. However, the ranking results after merging decrease sharply and are significantly worse than the results of the classical BERT model. We analyzed it, and it is because the scale of document scores between D-OverDense models trained independently is not consistent. Thus directly merging documents from various groups according to their scores would generate poor ranking results. We further compare the distribution of the document scores between different groups. The comparison results of the score distribution illustrated in Fig. 6 also confirm our analysis and conjecture.

6 Conclusions

In this paper, we propose a novel pre-training framework DynamicRetriever for document retrieval, which re-

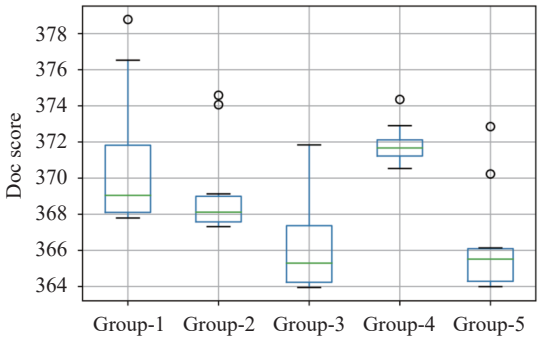

Fig. 6 Score distribution of different groups

gards the document identifiers as tokens and trains the relations from text to them. In such a framework, the semantic information of each document is stored in the model as parameters, and there is no need to build an index when retrieving documents for a given query. We implement the model with two training strategies: training from scratch and training over dense vectors. Experiments on the MS MARCO document ranking dataset show that our model-based IR system can improve the retrieval quality significantly, and the OverDense model demonstrates strong generalization and robustness when scaling up the corpus size. In the future, there are still many challenges, such as how to deal with the massive amount of documents at the Internet level. Therefore, model compression, multitasking, and other directions will be explored.

Acknowledegements

This work was supported by National Natural Science Foundation of China (Nos. 61872370 and 61832017), Beijing Outstanding Young Scientist Program (No. BJJWZYJH012019100020098), Beijing Academy of Artificial Intelligence (BAAI), the Outstanding Innovative Talents Cultivation Funded Programs 2021 of Renmin University of China, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

References

[1] S. Robertson, H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009. DOI: 10.1561/1500000019.

[2] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient estimation of word representations in vector space. [Online], Available: https://arxiv.org/abs/1301.3781, 2013.

[3] C. Y. Xiong, Z. Y. Dai, J. Callan, Z. Y. Liu, R. Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tokyo, Japan, pp. 55–64, 2017. DOI: 10.1145/3077136.3080809.

[4] Z. Y. Dai, C. Y. Xiong, J. Callan, Z. Y. Liu. Convolutional neural networks for soft-matching N-grams in Ad-Hoc search. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, Marina Del Rey, USA, pp. 126–134, 2018. DOI: 10.1145/3159652.3159659.

[5] J. T. Zhan, J. X. Mao, Y. Q. Liu, J. F. Guo, M. Zhang, S. P. Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1503–1512, 2021. DOI: 10.1145/3404835.3462880.

[6] L. Xiong, C. Y. Xiong, Y. Li, K. F. Tang, J. L. Liu, P. N. Bennett, J. Ahmed, A. Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[7] L. Y. Gao, Z. Y. Dai, T. F. Chen, Z. Fan, B. Van Durme, J. Callan. Complement lexical retrieval model with semantic residual embeddings. In *Proceedings of the 43rd European Conference on Information Retrieval*, Springer, pp. 146–160, 2021. DOI: 10.1007/978-3-030-72113-8_10.

[8] K. Guu, K. Lee, Z. Tung, P. Pasupat, M. W. Chang. REALM: Retrieval-augmented language model pre-training. [Online], Available: https://arxiv.org/abs/2002.08909, 2020.

[9] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, USA, pp. 4171–4186, 2019. DOI: 10.18653/v1/N19-1423.

[10] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever. Improving language understanding by generative pre-training. [Online], Available: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf, 2018.

[11] K. Clark, M. T. Luong, Q. V. Le, C. D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[12] R. Nogueira, W. Yang, K. Cho, J. Lin. Multi-stage document ranking with BERT. [Online], Available: https://arxiv.org/abs/1910.14424, 2019.

[13] W. C. Chang, F. X. Yu, Y. W. Chang, Y. M. Yang, S. Kumar. Pre-training tasks for embedding-based large-scale retrieval. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[14] X. Y. Ma, J. F. Guo, R. Q. Zhang, Y. X. Fan, X. Ji, X. Q. Cheng. PROP: Pre-training with representative words prediction for Ad-Hoc retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 283–291, 2021. DOI: 10.1145/3437963.3441777.

[15] W. Yang, H. T. Zhang, J. Lin. Simple applications of BERT for ad hoc document retrieval. [Online], Available: https://arxiv.org/abs/1903.10972, 2019.

[16] R. Nogueira, K. Cho. Passage re-ranking with BERT. [Online], Available: https://arxiv.org/abs/1901.04085, 2019

[17] L. Y. Gao, Z. Y. Dai, J. Callan. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Proceedings of the 43rd European Conference on Information Retrieval*, Springer, pp. 280–286, 2021. DOI: 10.1007/978-3-030-72240-1_26.

[18] J. T. Zhan, J. X. Mao, Y. Q. Liu, M. Zhang, S. P. Ma. RepBERT: Contextualized text embeddings for first-stage retrieval. [Online], Available: https://arxiv.org/abs/2006.15498, 2020.

[19] B. Miutra, N. Craswell. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval*, vol. 13, no. 1, pp. 1–126, 2018. DOI: 10.1561/1500000061.

[20] D. Metzler, Y. Tay, D. Bahri, M. Najork. Rethinking search: Making domain experts out of dilettantes. *ACM SIGIR Forum*, vol. 55, no. 1, Article number 13, 2021. DOI: 10.1145/3476415.3476428.

[21] Y. Tay, V. Q. Tran, M. Dehghani, J. M. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, J. Gupta, T. Schuster, W. W. Cohen, D. Metzler. Transformer memory as a differentiable search index. [Online], Available: https://arxiv.org/abs/2202.06991, 2022.

[22] J. Pennington, R. Socher, C. Manning. GloVe: Global vectors for word representation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1532–1543, 2014. DOI: 10.3115/v1/D14-1162.

[23] G. Q. Zheng, J. Callan. Learning to reweight terms with distributed representations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Santiago, Chile, pp. 575–584, 2015. DOI: 10.1145/2766462.2767700.

[24] J. F. Guo, Y. X. Fan, Q. Y. Ai, W. B. Croft. A deep relevance matching model for Ad-Hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, USA, pp. 55–64, 2016. DOI: 10.1145/2983323.2983769.

[25] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, W. B. Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Shinjuku, Japan, pp. 65–74, 2017. DOI: 10.1145/3077136.3080832.

[26] R. Nogueira, J. Lin. From doc2query to docTTTTTquery. [Online], Available: https://cs.uwaterloo.ca/~jimmylin/publications/Nogueira_Lin_2019_docTTTTTquery-v2.pdf 2019.

[27] R. Nogueira, W. Yang, J. Lin, K. Cho. Document expansion by query prediction. [Online], Available: https://arxiv.org/abs/1904.08375, 2019.

[28] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Q. Zhou, W. Li, P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, vol. 21, no. 1, Article number 140, 2020.

[29] Z. Y. Dai, J. Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. [Online], Available: https://arxiv.org/abs/1910.10687, 2019.

[30] Z. Y. Dai, J. Callan. Context-aware document term weighting for ad-hoc search. In *Proceedings of the Web Conference*, Taiwan, China, pp. 1897–1907, 2020. DOI: 10.1145/3366423.3380258.

[31] J. Johnson, M. Douze, H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021. DOI: 10.1109/TBDATA.2019.2921572.

[32] H. Jégou, M. Douze, C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011. DOI: 10.1109/TPAMI.2010.57.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 6000–6010, 2017.

[34] J. F. Guo, Y. X. Fan, L. Pang, L. Yang, Q. Y. Ai, H. Zamani, C. Wu, W. B. Croft, X. Q. Cheng. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, vol. 57, no. 6, Article number 102067, 2020. DOI: 10.1016/j.ipm.2019.102067.

[35] K. Lee, M. W. Chang, K. Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 6086–6096, 2019. DOI: 10.18653/v1/P19-1612.

[36] J. M. Ni, G. H. Ábrego, N. Constant, J. Ma, K. B. Hall, D. Cer, Y. F. Yang. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Proceedings of the Findings of the Association for Computational Linguistics*, Dublin, Ireland, pp. 1864–1874, 2022. DOI: 10.18653/v1/2022.findings-acl.146.

[37] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Q. Chen, W. T. Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020. DOI: 10.18653/v1/2020.emnlp-main.550.

[38] N. De Cao, G. Izacard, S. Riedel, F. Petroni. Autoregressive entity retrieval. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[39] J. G. Chen, R. Q. Zhang, J. F. Guo, Y. X. Fan, X. Q. Cheng. GERE: Generative evidence retrieval for fact verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid, Spain, pp. 2184–2189, 2022. DOI: 10.1145/3477495.3531827.

[40] J. P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Springer-Verlag, Dublin, Ireland, pp. 302–310, 1994.

[41] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, S. Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019. DOI: 10.1162/tacl_a_00276.

Yu-Jia Zhou received the B.Eng. degree in computer science and technology from School of Information, Renmin University of China, China in 2019. He is currently a Ph.D. degree candidate in computer science at School of Information, Renmin University of China. He won the best student paper award in CCIR 2018. He has been invited as a reviewer of international conferences SIGIR, KDD, WSDM.

His research interests include information retrieval, personalized search, deep learning, and data mining.

E-mail: zhouyujia@ruc.edu.cn

ORCID iD: 0000-0002-3530-3787

Jing Yao received the B.Eng. degree in computer science and technology from School of Information, Renmin University of China, China in 2019, and the M.Sc. degree in computer application technology from School of Information, Renmin University of China, Chian in 2022. She has been invited as a reviewer of international conferences SIGIR, WSDM. She is working at Microsoft Research Asia as a researcher now.

Her research interests include information retrieval, personalized search, explainable search/recommendation.

E-mail: jing_yao@ruc.edu.cn

Zhi-Cheng Dou received the B.Sc. and Ph.D. degrees in computer science and technology from Nankai University, China in 2003 and 2008, respectively. He is an associate professor in School of Information, Renmin University of China. He worked at Microsoft Research as a researcher from July 2008 to September 2014. He is a member of the IEEE.

His research interests include information retrieval, data mining, and big data analytics.

E-mail: dou@ruc.edu.cn (Corresponding author)

ORCID iD: 0000-0002-9781-948X

Ledell Wu received the B.Sc. degree in mathematics from Peking University, China in 2009, received the the M.Sc. degree in computer science from and University of Toronto, Canada in 2011. She is currently a research scientist manager at Beijing Academy of Artificial Intelligence (BAAI), China. She worked as a research engineer at Facebook AI Research from 2013–2021. She worked on a couple of research projects that also have boarder impact at Facebook, including general purpose embedding system, large-scale graph embedding system, mono/multilingual entity linking system and dense passage retrieval system. She also studies fairness and biases in machine learning and NLP models.

Her research interests include approximation algorithms, the hardness of approximation, privacy, and machine learning.

E-mail: wuyu@baai.ac.cn

Ji-Rong Wen received the B.Sc. and M.Sc. degrees in computer science from Renmin University of China, China, in 1994 and 1996, and the Ph.D. degree in computer science from Chinese Academy of Sciences, China in 1999. He is a professor at Renmin University of China. He was a senior researcher and research manager with Microsoft Research from 2000 to 2014. He is a senior member of the IEEE.

His research interests include web data management, information retrieval (especially web IR), and data mining.

E-mail: jirong.wen@gmail.com

Machine Intelligence Research

Citation: Y. J. Zhou, J. Yao, Z. C. Dou, L. Wu, J. R. Wen. Dynamicretriever: a pre–trained model–based ir system without an explicit index. *Machine Inteligence Research*. https://doi.org/10.1007/s11633–022–1373–9

Articles may interest you

Design of an executable anfis-based control system to improve the attitude and altitude performances of a quadcopter drone. *Machine Intelligence Research*, vol.18, no.1, pp.124-141, 2021.
DOI: 10.1007/s11633-020-1251-2

Paradigm shift in natural language processing. *Machine Intelligence Research*, vol.19, no.3, pp.169-183, 2022.
DOI: 10.1007/s11633-022-1331-6

Step ap 242 managed model-based 3d engineering: an application towards the automation of fixture planning. *Machine Intelligence Research*, vol.18, no.5, pp.731-746, 2021.
DOI: 10.1007/s11633-020-1272-x

A spatial cognitive model that integrates the effects of endogenous and exogenous information on the hippocampus and striatum. *Machine Intelligence Research*, vol.18, no.4, pp.632-644, 2021.
DOI: 10.1007/s11633-021-1286-z

A novel attention-based global and local information fusion neural network for group recommendation. *Machine Intelligence Research*, vol.19, no.4, pp.331-346, 2022.
DOI: 10.1007/s11633-022-1336-1

Knowing your dog breed: identifying a dog breed with deep learning. *Machine Intelligence Research*, vol.18, no.1, pp.45-54, 2021.
DOI: 10.1007/s11633-020-1261-0

Fault information recognition for on-board equipment of high-speed railway based on multi-neural network collaboration. *Machine Intelligence Research*, vol.18, no.6, pp.935-946, 2021.
DOI: 10.1007/s11633-021-1298-8

WeChat: MIR

Twitter: MIR_Journal