A Unified Pretraining Framework for Passage Ranking and Expansion

Ming Yan, Chenliang Li, Bin Bi, Wei Wang, Songfang Huang
Alibaba Group
{ym119608, lcl193798, b.bi, hebian.ww, songfang.hsf}@alibaba-inc.com

Abstract

Pretrained language models have recently advanced a wide range of natural language processing tasks. Nowadays, the application of pretrained language models to IR tasks has also achieved impressive results. Typical methods either directly apply a pretrained model to improve the re-ranking stage, or use it to conduct passage expansion and term weighting for first-stage retrieval. We observe that the passage ranking and passage expansion tasks share certain inherent relations, and can benefit from each other. Therefore, in this paper, we propose a general pretraining framework to enhance both tasks with Unified Encoder-Decoder networks (UED). The overall ranking framework consists of two parts in a cascade manner: (1) passage expansion with a pretraining-based query generation method; (2) re-ranking of passage candidates from a traditional retrieval method with a pretrained transformer encoder. Both the two parts are based on the same pretrained UED model, where we jointly train the passage ranking and query generation tasks for further improving the full ranking pipeline. An extensive set of experiments have been conducted on two large-scale passage retrieval datasets to demonstrate the state-of-the-art results of the proposed framework in both the first-stage retrieval and the final re-ranking. In addition, we successfully deploy the framework to our online production system, which can stably serve industrial applications with a request volume of up to 100 QPS in less than 300ms.

Introduction

Pretrained language models have advanced the state-of-the-art in a variety of NLP tasks ranging from text classification, language inference to question answering (Wang et al. 2018; Rajpurkar et al. 2016). The same trend is witnessed in the IR community, where several recent works have been proposed to apply these models to IR tasks and achieve promising results such as document retrieval (Yilmaz et al. 2019; MacAvaney et al. 2019) and passage ranking (Nogueira and Cho 2019; Dai and Callan 2019b).

Nowadays, in ad-hoc retrieval, the prevalent approach either directly deploys the pretrained models like BERT as re-rankers over an initial list of candidate passages retrieved from a traditional retrieval method (Nogueira and Cho 2019), or uses the deep pretrained models to conduct passage expansion (Nogueira et al. 2019b) and term weighting (Dai and Callan 2019a) for enhancing first-stage retrieval. The association between the retrieval stage and the re-ranking stage is not fully exploited, where different pretrained models usually work separately. With the same goal of relevance matching, we hypothesize that different stages in a full ranking pipeline are related, and can benefit from each other. Therefore, in this paper, we propose a unified pretraining framework for enhancing the full ranking pipeline with both stages, where task relationships can be better exploited. A brief illustration of the proposed framework is shown in Figure 1.

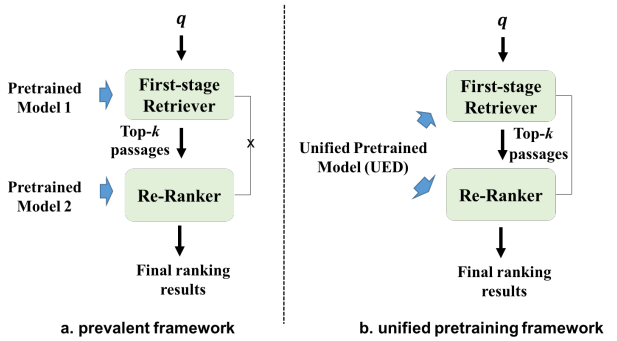

Figure 1: Comparison of the unified pretraining framework with the previous prevalent framework for full ranking.

Based on the unified pretrained encoder-decoder networks (UED), we enhance the overall framework from two aspects: (1) to tackle the vocabulary mismatch problem in the first stage retrieval, we leverage passage expansion technique as in (Nogueira et al. 2019b) to improve the term-based retriever with a pretraining-based query generation model, and (2) we exploit the language understanding ability of pretrained model for enhancing the neural re-ranker, and jointly train both the ranking and generation tasks to better capture the task relationships. In literature, passage expansion (Tao et al. 2006; Efron, Organisciak, and Fenlon 2012) proves to be helpful for improving ranking performance. Take Figure 2 as an example, the predicted query "the characteristics of dyslexia" of the passage can help the model better understand the main point of the passage, which can in turn help re-rank the passage in terms of the actual query "What are characteristics of dyslexia?". The passage ranking task of knowing the relevant passage to a given query and query

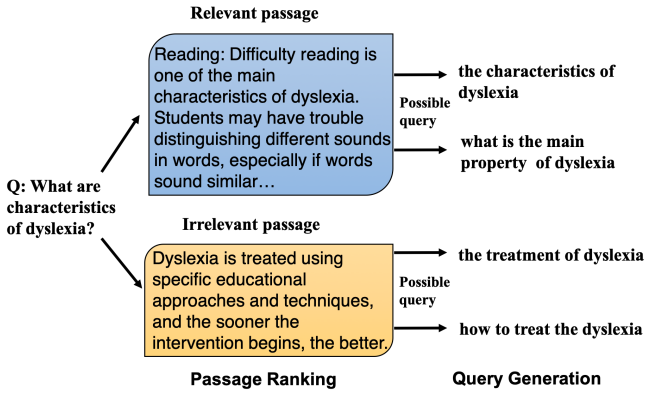

Figure 2: Illustration example of the relation between passage ranking and query generation in passage expansion.

generation task of predicting possible queries for a given passage can be related. Therefore, different from the previous methods that treat the two tasks independently, we propose to leverage the same pretrained UED model under a general framework for supporting both the query generation and neural re-ranking models, based on which different tasks can be jointly optimized.

Specifically, the overall framework consists of a term-based retriever and a neural re-ranker both based on the same pretrained UED model, as shown in Figure 1. For the term-based retriever, we use an off-the-shelf search engine to index the full collection of passages and adopt the simple and effective BM25 (Robertson, Zaragoza et al. 2009) method for the first-stage retrieval. Prior to indexing, we use the pretrained UED network to generate possible queries for each passage, where the passage is further expanded with the generated queries. Moreover, we also extract certain keywords from the passage and use them to guide the generation process, where we can generate unseen query words and properly adjust term weights for better term-level matching. The index is built on the expanded passages for effective passage expansion. For the re-ranker, we build the ranking model on the encoder of UED model and fine-tune it with a point-wise ranking objective. To reserve both the language understanding and language generation abilities and better associate the two different tasks, we adopt a two-stage pretraining protocol, where the encoder is first pretrained with autoencoding objectives as in BERT and the decoder is then additionally pretrained in a standard autoregressive way as in GPT. Based on the pretrained UED network, we finally jointly train both the passage ranking task and query generation task towards a more accurate full ranking goal.

The main contributions of this paper can be summarized as follow:

- We propose a unified pretraining framework for ad-hoc text retrieval, by taking full advantages of term-matching signals from passage expansion and relevance matching signals from passage ranking.
- To the best of our knowledge, we are the first to explore the potential of using a unified encoder-decoder network for enhancing both the query generation and text re-ranking under a general two-stage ranking framework, where the task relationships can be better exploited.
- Extensive experiments on two large-scale passage retrieval datasets demonstrate the effectiveness of the proposed method, where we achieve new state-of-the-art results on both the MS MARCO passage retrieval task and TREC 2019 Deep Learning Track.

Related Work

Term-based Retrieval Term-based retrieval methods such as BM25 (Robertson, Zaragoza et al. 2009) have been widely used for fast retrieval from a large-scale text corpus. Despite the efficiency, the vocabulary mismatch problem remains one of the central challenges in term-based retrieval. Typical methods to tackle this problem include relevance feedback (Salton and Buckley 1990; Lv and Zhai 2009) and query expansion (Voorhees 1994; Xu and Croft 2000; Fang and Zhai 2006). These methods mainly focus on enhancing query representations to better match documents. On the other hand, some other works adopt the document expansion method which improve retrieval by enriching document representations (Tao et al. 2006; Efron, Organisciak, and Fenlon 2012). In (Tao et al. 2006), Tao et al. construct a probabilistic neighborhood for each document, and expand the document with its neighborhood information. More recently, several works have been proposed to improve the term-based retrieval by leveraging the contextual neural models (Nogueira et al. 2019b; Dai and Callan 2019a). Nogueira et al. (Nogueira et al. 2019b) conduct document expansion by generating queries from documents using neural machine translation. In (Dai and Callan 2019a), the authors propose to map BERT's contextualized text representations to context-aware term weights of passages for improved term-based retrieval. In our first stage retrieval, we follow the line of generating possible queries for each passage to conduct document expansion. Different from the previous methods, we propose to associate the first stage retrieval and subsequent re-ranking stage with a unified pretrained model, where different tasks in two stages can benefit from each other for more effective ranking.

Pretrained Language Models Recently, we have seen rapid progress in both text understanding and text generation with the introduction of pretrained language models such as BERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019b), GPT (Radford et al. 2018) and T5 (Raffel et al. 2019). Nowadays, language model pretraining has also been successfully applied to IR tasks such as document retrieval (Yilmaz et al. 2019; MacAvaney et al. 2019) and passage ranking (Nogueira and Cho 2019; Dai and Callan 2019b). Nogueira et al. (Nogueira and Cho 2019) give one of the first successful applications of BERT to passage re-ranking, which acts as a starting point for BERT re-ranking. In (Nogueira et al. 2019a), the authors propose a multi-stage ranking architecture by applying BERT to both pointwise ranking and pairwise ranking in a cascade fashion. Boualili et al. (Boualili, Moreno, and Boughanem 2020) propose to integrate the exact term-matching signal to pretrained language models by marking the exact match tokens between

query and passage. Furthermore, recent works such as (Khattab and Zaharia 2020) and (Humeau et al. 2019) have begun to study more efficient BERT ranking architectures by separately encoding the query and passage. To better optimize the end-to-end ranking performance, we propose a two-stage pre-training protocol to train a unified encoder-decoder network which incorporates both the language understanding and language generation abilities, and use the same UED model for supporting different ranking stages in our framework.

Unified Pretraining Framework

Unsupervised UED Pretraining

We choose the Transformer encoder-decoder from (Vaswani et al. 2017) as our base architecture. The encoder is pre-trained to support the text re-ranking task, while the decoder is additionally pretrained for query generation task.

Typically, there are two different ways to pretrain the language models with either autoencoding objectives or autoregressive objectives. Autoencoding-based methods usually leverage a bidirectional architecture to reconstruct the original text from corrupted input, which demonstrate strong ability for the language understanding tasks such as BERT (Devlin et al. 2018) and RoBERTa (Liu et al. 2019b). However, they cannot address the generation task well due to the data leakage problem brought by bidirectional context. On the other hand, autoregressive-based methods such as GPT (Radford et al. 2018) use the standard language model objective to maximize the data probability with unidirectional context, which is more natural for the generation task.

To incorporate both the language understanding and language generation abilities for associating the two different tasks, we adopt a two-stage pre-training protocol, where the encoder is first pre-trained with autoencoding objectives (for passage ranking task) and the decoder is then additionally pre-trained in a standard autoregressive way (for query generation task) when keeping the encoder fixed. To better cope with the downstream ranking and generation tasks, the encoder is pretrained with a slightly different sentence prediction task as in Wang et al. (2019), while the decoder is pretrained to predict the next sentence.The overview of the pre-training details is illustrated as in the upper part of Figure 3, which we will detail as follows.

Encoder Pre-training BERT (Devlin et al. 2018) introduces two pre-training objectives – masked language modeling (MLM) and next sentence prediction (NSP), where one is to predict the right tokens when masking part of them, and the other is to predict whether the next sentence is the right sentence or just a random segment. Compared with the MLM objective, the NSP objective is usually too weak and cannot model complex sentence relationships as indicated in (Wang et al. 2019; Lan et al. 2019). In document ranking, one of the keys lies in understanding the document content or more precisely sentences in the document given a query. It is helpful to equip the pre-trained model with the ability to understand the complex sentence relationships. Therefore, we change the original NSP task in BERT to a new sentence relation prediction (SRP) task as in StructBERT (Wang et al. 2019), to equally predict whether the two segment are next sentence relation, previous sentence relation or no relation [1]. The other settings are kept the same as BERT and the pre-training objective is to minimize the joint loss of MLM task and SRP task:

$$\mathcal{L}_{enc} = \mathcal{L}_{MLM} + \mathcal{L}_{SRP}$$

Decoder Pre-training For the query generation task, the input is usually given as long passage, and the model is asked to generate a shorter piece of query text based on comprehension of the whole passage content. To minimize the discrepancy between the self-supervised pre-training and supervised fine-tuning, we use an unbalanced next sentence generation task (NSG) to pretrain the decoder as in (Bi et al. 2020; Alberti et al. 2019), where the target is to predict the consecutive span tokens of the next sentence with a standard language modeling objective. Specifically, given a contiguous text segment of length L (e.g. 400) from an unlabeled corpus, we use all the sentences in this segment as context input to the encoder, and use the next one sentence as text output to be generated by the decoder. The decoder is pretrained to autoregressively generate text output out of the contextual representations from the encoder. The pre-training loss for the decoder is defined as:

$$\mathcal{L}_{dec} = -\sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log \prod_{t=1}^{n} P(y_t|y_{<t}, x) \qquad (1)$$

where $\mathcal{X}$ represents the set of input context, $\mathcal{Y}$ represents the set of text to be generated and n is the length of tokens in output text y. In this way, we expect that the decoder can be pretrained by making the most use of more context to predict a consecutive short span, which can better match the query generation task.

Two-stage Ranking with UED

We first introduce how to enhance the passage expansion and passage re-ranking stages with UED, then a joint training strategy is used to further improve the full ranking performance by sharing common pretrained networks. A brief overview can be found in the lower part of Figure 3.

First Stage Retrieval with Passage Expansion Given a user query $\mathbf{q}$ and a full collection of corpus $\mathbf{D}$, a term-based BM25 retriever is used to retrieve top-k candidate passages $\mathbf{D}_1$ from the whole corpus $\mathbf{D}$. To overcome the vocabulary mismatch problem, we expand each passage by generating possible queries from the passage based on UED.

For each passage $\mathbf{d} = \{p_1, \cdots, p_M\} \in \mathbf{D}$, the task is to predict a set of queries $\mathcal{Q}^{gen} = \{\mathbf{q}_1^{gen}, \cdots, \mathbf{q}_L^{gen}\}$ for which that passage will be relevant, where $\mathbf{q}^{gen} = \{q_1, \cdots, q_N\}$ and N is the total number of generated query words. We first extract a total collection of query-relevant passage pairs from the labeled training corpus, with the relevant passage as input context and user query as groundtruth output, to fine-tune

[1] Details can be found in the original paper.

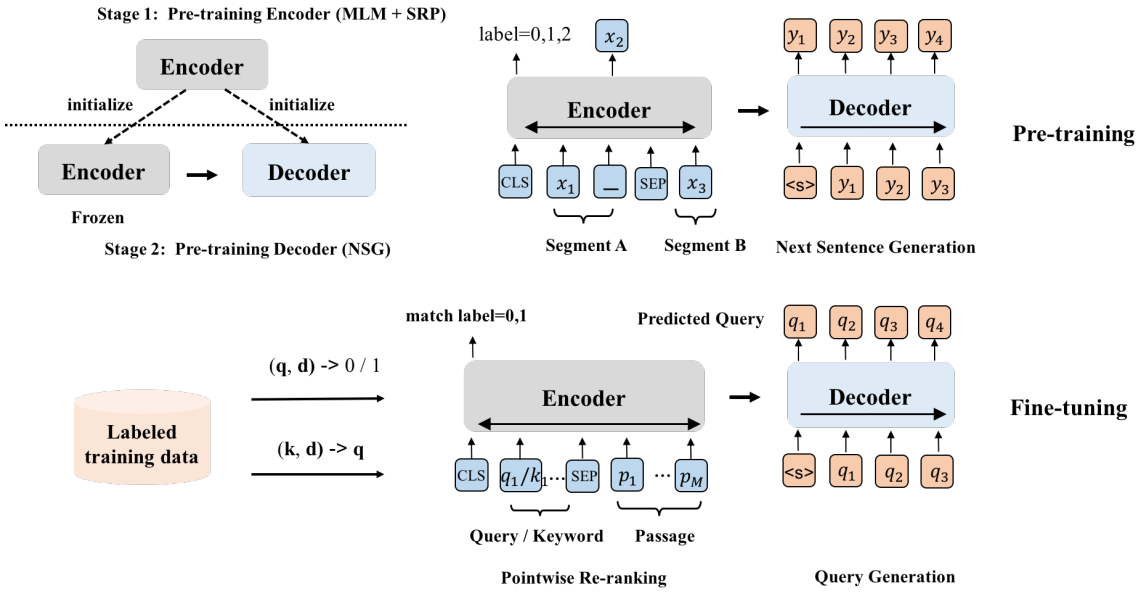

Figure 3: Overview of the pre-training and fine-tuning procedures of the encoder-decoder networks.

the UED model for query generation. Due to the large context of the passage, the generated query may lose certain key information. Therefore, we also extract a set of keywords $\mathbf{k} = \{k_1, \cdots, k_K\}$ from the given passage by using the unsupervised keyword extraction algorithm RAKE (Rose et al. 2010), based on which to guide the process of query generation. Then, we treat the extracted keywords as a virtual "input query" and feed them together with the passage context into the pretrained encoder network as "[CLS] **k** [SEP] **d** [SEP]". At decoding time, the decoder will sequentially generate query words by attending on the output hidden states of the encoder network. During training, we minimize a maximum-likelihood loss as in Equ. 1, which is most widely used in generation tasks.

After the query generation model is trained, for each passage $\mathbf{d} \in \mathbf{D}$, we generate the top-$L$ queries $\mathcal{Q}^{gen}$ using top-k sampling decoder (Fan, Lewis, and Dauphin 2018). Then we append the generated queries to the original passage in the corpus. The search index is built on the expanded passages for more effective term-based retrieval of the top-k candidate passages $\mathbf{D}_1$.

Passage Re-ranking Given the user query **q** and candidate passage set $\mathbf{D}_1$, the aim of the neural re-ranker is to estimate a score s of how relevant a candidate passage $\mathbf{d} \in \mathbf{D}_1$ is to a query **q** with the labeled query-passage data. We build our re-ranking model on top of the encoder of UED network. For the input, we use the same notation used by BERT, where the query **q** is fed as segment A and the passage text **d** as segment B. The input token sequence is packed as "[CLS] **q** [SEP] **d** [SEP]". The pointwise ranking objective is used for training and we use the [CLS] vector in the final layer of UED encoder to compute a score s for each passage. The final list of passages are ranked by the score s.

We start training from the UED encoder, and fine-tune it with the objective as:

$$\mathcal{L}_{rnk} = -\sum_{i=1}^{N} l_i \cdot log(s_i) + (1 - l_i) \cdot log(1 - s_i)$$

$$s_i = \sigma(\mathbf{w} \cdot \mathbf{h}_{CLS}^{L})$$

where $l_i \in \{0, 1\}$ is the ground-truth label of the query-passage pair, $\mathbf{w}$ is a trainable parameter, $\mathbf{h}_{CLS}$ is the hidden state of [CLS] token in the final layer of the pretrained encoder network and σ is the sigmoid function.

Joint Training To better exploit the task relationship, we use a simple mini-batch based stochastic gradient descent (SGD) to learn the parameters of our model as in (Liu et al. 2019a). In each epoch, a mini-batch b_t is selected with equal probability from the training data of passage ranking and query generation tasks, and the model is updated each time according to the ranking loss or generation loss for the corresponding task. In this way, we aim to optimize the sum of both tasks towards full ranking framework.

Experiments

Datasets

MS MARCO Passage Retrieval [2] is one of the largest passage ranking datasets with about 8.8M passages obtained from the top-10 results retrieved by the Bing search engine from about 1M real user queries. The training set contains about 40M tuples of a query, relevant and non-relevant passages. There are about 500K distinct query-relevant passage pairs in the training set, where each query has one relevant passage on average. The development and test sets contain approximately 6,900 queries each, but relevance labels are made public only for the development set.

[2] https://github.com/microsoft/MSMARCO-Passage-Ranking.

TREC 2019 Deep Learning Track [3] also uses a large human-generated set of training labels, from the MS MARCO dataset. Different from MS MARCO dataset, it uses a different hold-out test set and use relevance judges to evaluate the quality of passage rankings. It has 200 test queries, where the passages are labelled by NIST assessors using multi-graded judgments, allowing to measure NDCG.

Experiment Settings

Pre-training Details UED is based on the Transformer which consists of a 24-layer encoder and a 6-layer decoder. Both the encoder and decoder have the same model settings as in BERT. The pre-training details of the encoder are the same as in BERT, except that the NSP task is replaced by the new SRP task, which is self-trained on BooksCorpus and English Wikipedia corpus. We train it with batch size of 256 and maximum sequence length of 512 for 40 epochs. For the decoder pre-training, we also use the same optimizer and pre-training datasets as in BERT. We use multiple consecutive sentences up to 400 tokens as the source text input to the encoder, and use the subsequent sentence as target text to the decoder. We train it with batch size of 256 and maximum sequence length of 512 for 2 epochs. To better adapt the model to the target corpus, we also continue pre-training the UED on MS MARCO document corpus (0.5B words) with two-stage pretraining protocol for another 100K+100K steps with learning rate of 1e-5.

Retriever & Ranker Settings We use the 500K query-relevant passage pairs for training the query generation model, and 40M labeled query-passage training data for training the neural re-ranker model. We set the size of mini-batch to 24 and learning rate to 5e-6. For the retriever, we truncate the maximum input length to 384, and limit the length of query to 30 tokens when decoding. For each passage, we choose the top-20 generated queries for passage expansion due to the best overall MRR@10 on MS MARCO development set. For effectiveness consideration, we keep the top-1000 passages for subsequent re-ranking. For re-ranking stage, we build each batch by sampling equal amount of relevant and irrelevant passages to avoid biasing towards predicting irrelevant labels. We fine-tune with maximum sequence length of 384 for 500K steps and checkpoint each model at the 50K steps.

Evaluation Metrics For MS MARCO passage retrieval task, since only binary label is given, we use the official evaluation metrics MRR@10 for evaluation. For TREC 2019 task, we use the main official evaluation metrics NDCG@10 and MAP. For the evaluation of the first-stage retrieval, we use the retrieval metric Recall@1000 of the BM25 method on the expanded search index. For the baseline methods, we compare our method with the previous state-of-the-art neural ranking models and top pre-training methods, which are shown in Table 1 and Table 2. DeepCT + TFR-BERT (Han et al. 2020) and StructBERT + P_{gen} (Yan et al. 2019) are the previous state-of-the-art pretraining methods in MS MARCO passage retrieval and TREC 2019 tasks, respectively. DeepCT + TFR-BERT leverages BERT to conduct improved term weighting for first stage retrieval, and then use a ensemble of different pretrained models for final re-ranking. StructBERT + P_{gen} conducts document expansion by leveraging query generation based on pointer-generator model, and then uses StructBERT (Wang et al. 2019) for subsequent re-ranking.

[3] https://microsoft.github.io/TREC-2019-Deep-Learning/

Method	MRR@10 Dev	MRR@10 Eval
Duet	25.4†	25.2
Conv-KNRM	29.0†	27.7
BiLSTM + Co-Attention	29.8†	29.1
MarkedBERT	32.8†	–
BERT Large	36.5†	35.9
ELECTRA Large	37.6†	36.7
BERTter Indexing	37.5†	36.8
BERT Large + T5	38.6†	–
Multi-Stage BERT	39.0†	37.9
DeepCT + TFR-BERT	40.1† / 42.1	– / 40.7
UED (ours)	**42.4 / 43.6**	**– / 42.4**

Table 1: Performance of top published methods on MS MARCO leaderboard as of August 12th, 2020, † means the difference between the baseline model and the proposed model is significant with p-value < 0.05. The compared methods are (Mitra, Diaz, and Craswell 2017), (Dai et al. 2018), (Alaparthi 2019), (Boualili, Moreno, and Boughanem 2020), (Nogueira and Cho 2019), (Clark et al. 2020), (Nogueira et al. 2019b), (Raffel et al. 2019), (Nogueira et al. 2019a) and (Han et al. 2020) in the mentioned order.

Main Results

The official evaluation results on MS MARCO passage retrieval and TREC 2019 Deep Learning Track can be found in Table 1 and Table 2, respectively. At the time of paper submission, the proposed UED model achieves a new state-of-the-art performance on both the MS MARCO leaderboard and TREC 2019 evaluation. From the results, we can see that the pretraining methods can largely improve the performance over the previous state-of-the-art Neural IR methods, which indicates the superiority of the recent pretraining methods for modeling the semantic relevance in text ranking task. Besides, by leveraging a unified encoder-decoder backbone network, UED can outperform all the previous multi-stage ones on both datasets, which demonstrates the effectiveness and generalizability of our approach. It mainly benefits from two points: 1) the superiority of our unified pretraining method for supporting both first stage retrieval and subsequent re-ranking; 2) the possibility of associating different stages in the cascade ranking framework based on the unified pre-trained networks.

Ablation Study

To get better insight into our framework, we conduct an in-depth ablation study on the main components of the framework and the different pre-training settings. The result is shown in Table 3. For the main components, we can see that

Method	NDCG@10	MAP
BM25 + Axiomatic	55.1†	37.4†
Duet	61.4†	34.8†
Conv-KNRM	64.8†	22.9†
BERT Base	66.5†	24.2†
Transformer-Kernel	68.8†	42.0†
BERTter Indexing (+RM3)	74.2†	50.5†
BERT Large + T5	74.9†	49.8†
StructBERT + P_{gen}	75.0†	50.1†
UED (ours)	**77.2**	**52.6**

Table 2: Performance of top methods evaluated on TREC 2019 Deep Learning Track as of August 12th, 2020 (without model ensembling). The compared methods are almost the same as in Table 1, with additional ones of Transformer-Kernel (Hofstätter, Zlabinger, and Hanbury 2019) and StructBERT + P_{gen} (Yan et al. 2019).

Method	MS MARCO	
	MRR@10	Recall@1000
UED (full model)	**42.4**	**94.0**
w/o document expansion	39.5	86.2
w/o final re-ranking	28.7	94.0
w/o two-stage UED pretraining#	39.4	91.2
w/o joint UED finetuning	40.9	93.0
w/o decoder pre-training	41.3	91.8
w/o encoder SRP task*	40.8	93.7
UED (only re-ranking)	39.5	86.2
w/o two-stage UED pretraining#	38.0	86.2
w/o joint UED finetuning	39.0	86.2
BERT Large + BM 25 Baseline	36.5	86.2

Table 3: Ablation study on model components and pre-training strategies. *: we replace with the NSP task, #: we replace with only the BERT large encoder .

the pretrained re-ranker and document expansion play very important roles in the proposed framework, removing each of the component will cause large performance decrease. The document expansion method influences more on the first-stage retrieval (about 8.3% recall decrease without it), while the pretrained re-ranker is the most critical part for the final ranking performance. We also experimented with pairwise and listwise ranking loss but did not see much difference, which is the same as in (Qiao et al. 2019). The two-stage UED pretraining is also critical to our full ranking framework, both the performances of the first stage retrieval and final re-ranking decrease greatly when replacing the UED network with the BERT large model. It shows the effectiveness of the unsupervised UED pretraining for supporting both passage ranking and query generation tasks. Besides, with the shared encoder-decoder network for both the two tasks, a joint training strategy can further improve the overall ranking performance by considering the task relationships. In this way, the encoder can be better trained by leveraging the supervised signals from both tasks.

For different pre-training settings, We can see that both the decoder pre-training and SRP pre-training tasks contribute to the final performance gain. Pre-training the decoder influences more on the first-stage retrieval (Recall@1000), which helps by summarizing and generating more accurate queries for document expansion. For the encoder, using a new SRP task can further improve the ranking performance (MRR@10) by enhancing the pretrained re-ranker with the ability to understand more complex sentence relationship.

Method	MRR@10	Recall
BM25 baseline	18.8	86.2
BM25 baseline + RM3	17.2	86.8
DELM (Tao et al. 2006)	19.4	87.0
doc2query (Nogueira et al. 2019b)	21.8	89.1
DeepCT (Dai and Callan 2019a)	24.3	91.3
T5 (Raffel et al. 2019)	27.8	**94.5**
UED (ours) w/o joint UED finetuning	27.5	93.0
UED (ours)	**28.7**	94.0

Table 4: Performance w.r.t different document expansion methods on MS MARCO development set.

Effectiveness of Passage Expansion

Now we further examine the effectiveness of the proposed query generation method based on UED model for document expansion. Therefore, we remove the re-ranking stage in the framework, and evaluate the effectiveness of the first stage retrieval with BM25 method via different document expansion methods. The result is shown in Table 4. We can see that: (1) by leveraging the supervised question-relevant document signal, UED obtains superior performance compared to the traditional unsupervised document expansion method DELM (Tao et al. 2006) and relevance-based query expansion method RM3 (Abdul-Jaleel et al. 2004). In terms of MRR@10, the query expansion with RM3 may even hurt the performance. This may be due to the fact that the MS MARCO dataset is more precision oriented, with sparse positive labels (1-2 per query) as groundtruth; (2) UED can outperform the previous top-ranked document expansion methods such as doc2query (Nogueira et al. 2019b) and a recent BERT-based term weighting method DeepCT (Dai and Callan 2019a), and also obtain comparable performance to a extremely large T5-based document expansion method (Raffel et al. 2019) which shows the effectiveness of the proposed UED model for enhancing first-stage retrieval. (3) By jointly optimizing both the passage ranking and query generation tasks, the retrieval performance can be further improved, which again demonstrates the advantage of our UED model to capture certain relationships between the two tasks for enhancing first stage retrieval.

Case Study

To better interpret the advantage of our generation-based passage expansion, we showcase the generation results of three sampled passages by our query generation model, as shown in Table 5. The target query is the actual relevant query to the sampled passage. We can see that our generation model can identify the possible query type of the passage (e.g., "what are", "how many"), and generate words not present in the original passage (e.g., "what", "per day"), which is close to the actual target query. For example, in the first case, we generate almost the same query as the actual relevant query,

Passage	Reading: **Difficulty reading** is one of the main **characteristics** of **dyslexia**. Students may have trouble distinguishing different sounds in words, especially if words sound similar. They may also struggle with the following **reading** skills: 1 Problems rhyming words. 2 Difficulty breaking words...
Target query	what are characteristics of dyslexia
Predicted query	*what are the* **characteristics of dyslexia**
Passage	So just how much **fiber** do you need? The national **fiber** recommendations are 30 to 38 **grams a day** for men and 25 **grams a day** for women between 18 and 50 years old, and 21 **grams a day** if a woman is 51 and older. Another general guideline is to get 14 grams of **fiber** for every 1,000 calories...
Target query	how much fiber in calories per day
Predicted query	*how many* **grams** *of* **fiber** *per day*
Passage	**Definition**. A country that may have a requirement to cooperate with an **international boycott** due to affiliation with certain **international** organization. While I sympathize with those calling for a **boycott** of Stonewall, I personally don't support a **boycott**. However, I don't think ANYONE should see Stonewall. Not because of its politics or revisionism, but because Stonewall is a terrible movie.
Target query	what is international boycott
Predicted query	*what is the* **definition** *of* **international boycott**

Table 5: Case study of sample queries generated by our query generation model. Words in bold are keywords extracted from the passage, and words in italic type are generated from the vocabulary.

which helps properly summarize the main point of the passage and emphasize on the important content. This makes it possible to successfully apply the passage expansion method for term-based retrieval and help address the problem of vocabulary mismatch. Moreover, our generation method tends to generate keywords (e.g., "characteristics", "dyslexia" and "international boycott") extracted from the passage, which can help emphasize on the key information of the passage. In this way, the expanded passage can be better retrieved with larger term weights on key terms. On the other hand, due to the uncertainty of the common generation method, we may also generate some unrelated words, such as "how many grams" and "the definition of". That is also the reason why we still use the original passage content for both term-based retrieval and re-ranking.

On-line Evaluation

On-line Environment We also deploy our framework to the Enterprise Knowledge Assistant system, which is an intelligent search and question answering assistant designed for promoting working experience and efficiency of typical industries. We test our framework in the practical scenario of electrical equipment knowledge search, where we have the largest human-label training data. This scenario is about consulting and searching for the specialized knowledge of electrical equipment and finding solution to typical electrical problem from the unstructured electrical documents. In total, we have more than 30K passages obtained from the electrical documents. There are about 24K distinct query-relevant passage pairs in the training set, where each query has one relevant passage on average. The test set contain approximately 6,400 queries, which are guaranteed to have one relevant passage.

In the practical scenario, real-time responses are expected and large amounts of requests are required to get response simultaneously. Therefore, we adopt the pre-trained distillation method as in (Turc et al. 2019) to further compress our pretrained encoder network while keeping the decoder network as the original size. We distil our UED encoder to a smaller 6-layer UED tiny model with 512 hidden size for efficiency consideration. For the retriever, we select and keep the top-30 passages for its good performance (95.3% recall ratio) and efficiency consideration. The online evaluation result is shown in Table 6. We can see that by using the cascade ranking framewok with the distilled UED tiny model, our system can achieve relatively high ranking effectiveness and efficiency compared to other methods. Besides, with a 6-layer UED tiny model and by properly tradeoff with the amount of recalled passages from first stage retrieval, we can serve the typical scenario within a response time of 300ms, which is acceptable for typical online requirement.

Model	**MRR**	**RECALL@30**	**Latency**
BM25	68.2	88.3	**28ms**
DUET	76.6	88.3	54ms
BERTter Indexing	87.3	92.8	1,315ms
UED-12 layer	**91.8**	**95.3**	612ms
UED-6 layer	90.6	94.8	258ms

Table 6: Performance and latency of different methods in our online practical scenario (we use the Nvidia T40 GPU for serving).

Conclusion

In this paper, we propose a general pretraining framework to enhance both the retrieval and re-ranking stages with a unified encoder-decoder network. The encoder-decoder network is first pretrained in an autoencoding denoising stage, then followed by an autoregressive generation stage, to reserve both the language understanding and generation abilities. Based on the pretrained UED networks, we finally jointly train both the passage ranking and query generation tasks towards a full ranking goal. Experimental results on two large-scale datasets show that the proposed method achieves a new state-of-the-art performance on both the MS MARCO passage retrieval task and TREC 2019 Deep Learning Track. Besides, the proposed method has also been effectively and efficiently deployed in our online system.

References

Abdul-Jaleel, N.; Allan, J.; Croft, W. B.; Diaz, F.; Larkey, L.; Li, X.; Smucker, M. D.; and Wade, C. 2004. UMass at TREC 2004: Novelty and HARD. *Computer Science Department Faculty Publication Series* 189.

Alaparthi, C. S. 2019. Microsoft AI Challenge India 2018: Learning to Rank Passages for Web Question Answering with Deep Attention Networks. *arXiv preprint arXiv:1906.06056* .

Alberti, C.; Andor, D.; Pitler, E.; Devlin, J.; and Collins, M. 2019. Synthetic QA corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416* .

Bi, B.; Li, C.; Wu, C.; Yan, M.; and Wang, W. 2020. PALM: Pre-training an Autoencoding&Autoregressive Language Model for Context-conditioned Generation. *arXiv preprint arXiv:2004.07159* .

Boualili, L.; Moreno, J. G.; and Boughanem, M. 2020. MarkedBERT: Integrating Traditional IR Cues in Pre-trained Language Models for Passage Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1977–1980.

Clark, K.; Luong, M.-T.; Le, Q. V.; and Manning, C. D. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* .

Dai, Z.; and Callan, J. 2019a. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687* .

Dai, Z.; and Callan, J. 2019b. Deeper Text Understanding for IR with Contextual Neural Language Modeling. *arXiv preprint arXiv:1905.09217* .

Dai, Z.; Xiong, C.; Callan, J.; and Liu, Z. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 126–134. ACM.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Efron, M.; Organisciak, P.; and Fenlon, K. 2012. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 911–920. ACM.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833* .

Fang, H.; and Zhai, C. 2006. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 115–122. ACM.

Han, S.; Wang, X.; Bendersky, M.; and Najork, M. 2020. Learning-to-Rank with BERT in TF-Ranking. *arXiv preprint arXiv:2004.08476* .

Hofstätter, S.; Zlabinger, M.; and Hanbury, A. 2019. TU Wien@ TREC Deep Learning'19–Simple Contextualization for Re-ranking. *arXiv preprint arXiv:1912.01385* .

Humeau, S.; Shuster, K.; Lachaux, M.-A.; and Weston, J. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969* .

Khattab, O.; and Zaharia, M. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *arXiv preprint arXiv:2004.12832* .

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* .

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019a. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504* .

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Lv, Y.; and Zhai, C. 2009. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th CIKM*, 255–264. ACM.

MacAvaney, S.; Yates, A.; Cohan, A.; and Goharian, N. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1101–1104.

Mitra, B.; Diaz, F.; and Craswell, N. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, 1291–1299. International World Wide Web Conferences Steering Committee.

Nogueira, R.; and Cho, K. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* .

Nogueira, R.; Lin, J.; and Epistemic, A. 2019. From doc2query to docTTTTTquery. *Online preprint* 1–3.

Nogueira, R.; Yang, W.; Cho, K.; and Lin, J. 2019a. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* .

Nogueira, R.; Yang, W.; Lin, J.; and Cho, K. 2019b. Document Expansion by Query Prediction. *arXiv preprint arXiv:1904.08375* .

Qiao, Y.; Xiong, C.; Liu, Z.; and Liu, Z. 2019. Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv:1904.07531* .

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf* .

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* .

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* .

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3(4): 333–389.

Rose, S.; Engel, D.; Cramer, N.; and Cowley, W. 2010. Automatic Keyword Extraction from Individual Documents. In Berry, M. W.; and Kogan, J., eds., *Text Mining. Applications and Theory*, 1–20. John Wiley and Sons, Ltd. ISBN 9780470689646. doi:10.1002/9780470689646.ch1. URL http://dx.doi.org/10.1002/9780470689646.ch1.

Salton, G.; and Buckley, C. 1990. Improving retrieval performance by relevance feedback. *Journal of the American society for information science* 41(4): 288–297.

Tao, T.; Wang, X.; Mei, Q.; and Zhai, C. 2006. Language model information retrieval with document expansion. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 407–414. Association for Computational Linguistics.

Turc, I.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Voorhees, E. M. 1994. Query expansion using lexical-semantic relations. In *SIGIR94*, 61–69. Springer.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* .

Wang, W.; Bi, B.; Yan, M.; Wu, C.; Bao, Z.; Peng, L.; and Si, L. 2019. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. *arXiv preprint arXiv:1908.04577* .

Xu, J.; and Croft, W. B. 2000. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS)* 18(1): 79–112.

Yan, M.; Li, C.; Wu, C.; Bi, B.; Wang, W.; Xia, J.; and Si, L. 2019. IDST at TREC 2019 Deep Learning Track: Deep Cascade Ranking with Generation-based Document Expansion and Pre-trained Language Modeling .

Yilmaz, Z. A.; Yang, W.; Zhang, H.; and Lin, J. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3481–3487.