BERT-QE: Contextualized Query Expansion for Document Re-ranking

Zhi Zheng[1,3], **Kai Hui**[2]*, **Ben He**[1,3]✉, **Xianpei Han**[3], **Le Sun**[3]✉, **Andrew Yates**[4]
[1] University of Chinese Academy of Sciences, Beijing, China
[2] Amazon Alexa, Berlin, Germany
[3] Institute of Software, Chinese Academy of Sciences, Beijing, China
[4] Max Planck Institute for Informatics, Saarbrücken, Germany
zhengzhi18@mails.ucas.ac.cn, kaihuibj@amazon.com
benhe@ucas.ac.cn, {xianpei, sunle}@iscas.ac.cn, ayates@mpi-inf.mpg.de

Abstract

Query expansion aims to mitigate the mismatch between the language used in a query and in a document. However, query expansion methods can suffer from introducing non-relevant information when expanding the query. To bridge this gap, inspired by recent advances in applying contextualized models like BERT to the document retrieval task, this paper proposes a novel query expansion model that leverages the strength of the BERT model to select relevant document chunks for expansion. In evaluation on the standard TREC Robust04 and GOV2 test collections, the proposed BERT-QE model significantly outperforms BERT-Large models.

1 Introduction

In information retrieval, the language used in a query and in a document differs in terms of verbosity, formality, and even the format (e.g., the use of keywords in a query versus the use of natural language in an article from Wikipedia). In order to reduce this gap, different query expansion methods have been proposed and have enjoyed success in improving document rankings. Such methods commonly take a pseudo relevance feedback (PRF) approach in which the query is expanded using top-ranked documents and then the expanded query is used to rank the search results (Rocchio, 1971; Lavrenko and Croft, 2001; Amati, 2003; Metzler and Croft, 2007) .

Due to their reliance on pseudo relevance information, such expansion methods suffer from any non-relevant information in the feedback documents, which could pollute the query after expansion. Thus, selecting and re-weighting the information pieces from PRF according to their relevance before re-ranking are crucial for the effectiveness of the query expansions. Existing works identify expansion tokens according to the language model on top of feedback documents, as in RM3 (Lavrenko and Croft, 2001), extract the topical terms from feedback documents that diverge most from the corpus language model (Amati, 2003), or extract concepts for expansion (Metzler and Croft, 2007). In the context of neural approaches, the recent neural PRF architecture (Li et al., 2018) uses feedback documents directly for expansion. All these methods, however, are under-equipped to accurately evaluate the relevance of information pieces used for expansion. This can be caused by the mixing of relevant and non-relevant information in the expansion, like the tokens in RM3 (Lavrenko and Croft, 2001) and the documents in NPRF (Li et al., 2018); or by the facts that the models used for selecting and re-weighting the expansion information are not powerful enough, as they are essentially scalars based on counting.

Inspired by the recent advances of pre-trained contextualized models like BERT on the ranking task (Yilmaz et al., 2019; Nogueira et al., 2020), this work attempts to develop query expansion models based on BERT with the goal of more effectively using the relevant information from PRF. In addition, as indicated in previous studies (Qiao et al., 2019; Dai and Callan, 2019), the (pre-)trained BERT-based ranking models have a strong ability to identify highly relevant chunks within documents. This actually provides advantages in choosing text chunks for expansion by providing more flexibility in terms of the granularity for expansions, as compared with using tokens (Lavrenko and Croft, 2001), concepts with one or two words (Metzler and Croft, 2007), or documents (Li et al., 2018).

Given a query and a list of feedback documents from an initial ranking (e.g., from BM25), we propose to re-rank the documents in three sequential

* This work has been done before joining Amazon.

phases. In phase one, the documents are re-ranked with a fine-tuned BERT model and the top-ranked documents are used as PRF documents; in phase two, these PRF documents are decomposed into text chunks with fixed length (e.g., 10), and the relevance of individual chunks are evaluated; finally, to assess the relevance of a given document, the selected chunks and original query are used to score the document together. To this end, a novel query expansion model, coined as BERT-QE, based on the contextualized model is developed.

Contributions of this work are threefold. 1) A novel query expansion model is proposed to exploit the strength of contextualized model BERT in identifying relevant information from feedback documents; 2) Evaluation on two standard TREC test collections, namely, Robust04 and GOV2, demonstrates that the proposed BERT-QE-LLL could advance the performance of BERT-Large significantly on both shallow and deep pool, when using BERT-Large in all three phases; 3) We further trade-off the efficiency and effectiveness, by replacing BERT-Large with smaller BERT architectures and demonstrate that, with a smaller variant of BERT-QE, e.g., BERT-QE-LMT, one could outperform BERT-Large significantly on shallow pool with as least as an extra 3% computational cost; meanwhile, a larger variant, e.g., BERT-QE-LLS, could significantly outperform BERT-Large on both shallow and deep pool with 30% more computations.

2 Method

In this section we describe BERT-QE, which takes a ranked list of documents as input (e.g., from an unsupervised ranking model) and outputs a re-ranked list based on the expanded query.

2.1 Overview

There are three phases in the proposed BERT-QE. Namely, **phase one**: the first-round re-ranking of the documents using a BERT model; **phase two**: chunk selection for query expansion from the top-ranked documents; and **phase three**: the final re-ranking using the selected expansion chunks. The essential parts of the proposed BERT-QE are the second and third phases, which are introduced in detail in Sections 2.2 and 2.3. Without loss of generality, a fine-tuned BERT model serves as the backbone of the proposed BERT-QE model and is used in all three phrases. We describe the fine-tuning process and phase one before describing phases two and three in more detail.

Fine-tuning BERT model. Similar to (Yilmaz et al., 2019), a BERT model (e.g., BERT-large) is first initialized using a checkpoint that has been trained on MS MARCO (Bajaj et al., 2018). The model is subsequently fine-tuned on a target dataset (e.g., Robust04). This choice is to enable comparison with the best-performing BERT model, such as a fine-tuned BERT-Large (Yilmaz et al., 2019). Before fine-tuning the BERT model on a target dataset, we first use the aforementioned model trained on MS MARCO to identify the top-ranked passages in this dataset. These selected query-passage pairs are then used to fine-tune BERT using the loss function as in Equation (1).

$$\mathcal{L} = -\sum_{i \in I_{pos}} \log(p_i) - \sum_{i \in I_{neg}} \log(1 - p_i) \quad (1)$$

Therein, I_{pos} and I_{neg} are sets of indexes of the relevant and non-relevant documents, respectively, and p_i is the probability of the document d_i being relevant to the query. This configuration is similar to Dai and Callan (2019), with the difference that we use only passages with the highest scores instead of all passages. In our pilot experiments, this leads to comparable effectiveness but with a shorter training time.

Phase one. Using the fine-tuned BERT model, we re-rank a list of documents from an unsupervised ranking model for use in the second phase. As shown in Equation (2), given a query q and a document d, $rel(q, d)$ assigns d a relevance score by modeling the concatenation of the query and the document using the fine-tuned BERT. The ranked list is obtained by ranking the documents with respect to these relevance scores. We refer the reader to prior works describing BERT and ranking with BERT for further details (Devlin et al., 2019; Nogueira and Cho, 2019).

$$rel(q, d) = \text{BERT}(q, d) \quad (2)$$

2.2 Selecting Chunks for Query Expansion

In the **second phase**, the top-k_d documents from the first phase are employed as feedback documents and k_c chunks of relevant text are extracted from them. This phase is illustrated in Figure 1. In more detail, a sliding window spanning m words is used to decompose each feedback document into overlapping chunks where two neighboring chunks

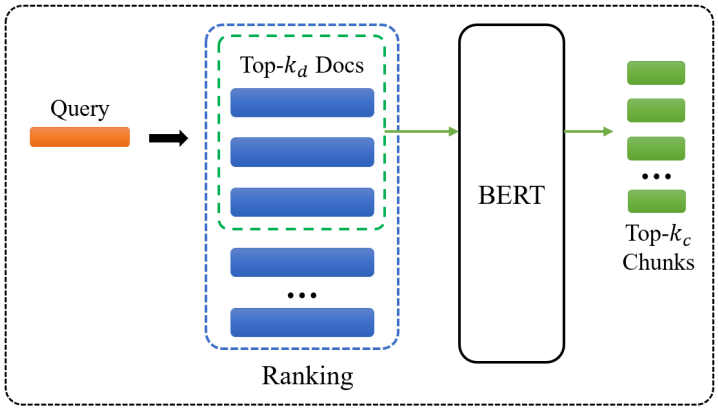

Figure 1: Chunk selection for query expansion in phase two.

are overlapped by up to $m/2$ words. The i-th chunk is denoted as c_i. As expected, these chunks are a mixture of relevant and non-relevant text pieces due to the lack of supervision signals. Therefore, the fine-tuned BERT model from Section 2.1 is used to score each individual chunk c_i, as indicated in Equation (3). The top-k_c chunks with the highest scores are selected. These k_c chunks, which are the output from phase two, serve as a distillation of the feedback information in the feedback documents from phase one. We denote the chunks as $\mathcal{C} = [c_0, c_1, \cdots, c_{k_c-1}]$.

$$rel(q, c_i) = \text{BERT}(q, c_i) \tag{3}$$

2.3 Final Re-ranking using Selected Chunks

In **phase three**, the chunks selected from phase two are used in combination with the original query to compute a final re-ranking. This process is illustrated in Figure 2.

Evaluating the relevance of a document using the selected feedback chunks. For each individual document d, the k_c chunks selected in phase two are used to assess its relevance separately, and the k_c evaluations are thereafter aggregated to generate the document's relevance score. As described in Equation (4), the fine-tuned BERT model from Section 2.1 is used to compute $rel(c_i, d)$, which are further aggregated into a relevance score $rel(\mathcal{C}, d)$. Akin to (Li et al., 2018), the relevance of individual chunks are incorporated as weights by using the softmax function $\text{softmax}_{c_i \in \mathcal{C}}(.)$ among all chunks in $\mathcal{C}$ on top of the $rel(q, c_i)$.

$$rel(\mathcal{C}, d) = \sum_{c_i \in \mathcal{C}} \text{softmax}_{c_i \in \mathcal{C}}(rel(q, c_i)) \cdot rel(c_i, d) \tag{4}$$

Combining $rel(\mathcal{C}, d)$ with $rel(q, d)$. To generate the ultimate relevance score $rel(q, \mathcal{C}, d)$ for d, akin to the established PRF models like RM3 (Lavrenko and Croft, 2001) and NPRF (Li et al., 2018), the relevance scores based on the feedback and the original query are combined as in Equation (5). α is a hyper-parameter, governing the relative importance of the two parts.

$$rel(q, \mathcal{C}, d) = (1-\alpha) \cdot rel(q, d) + \alpha \cdot rel(\mathcal{C}, d) \tag{5}$$

We note that the same fine-tuned BERT model does not necessarily need to be used in each phase. In our experiments, we consider the impact of using different BERT variants from Table 1 in each phase. For example, phases one and three might use the BERT-Large variant, while phase two uses the BERT-Small variant with fewer parameters.

3 Experimental Setup

In this section, we describe our experiment configurations. Source code, data partitions for cross-validation, result files of initial rankings, and the trained models are available online[1].

3.1 Dataset and Metrics

Akin to (Guo et al., 2016; Yilmaz et al., 2019), we use the standard Robust04 (Voorhees, 2004) and GOV2 (Clarke et al., 2004) test collections. Robust04 consists of 528,155 documents and GOV2 consists of 25,205,179 documents. We employ 249 TREC keyword queries for Robust04 and 150 keyword queries for GOV2. Akin to (Yilmaz et al., 2019), in this work, all the rankings from BERT-based models, including the proposed models and

[1] https://github.com/zh-zheng/BERT-QE

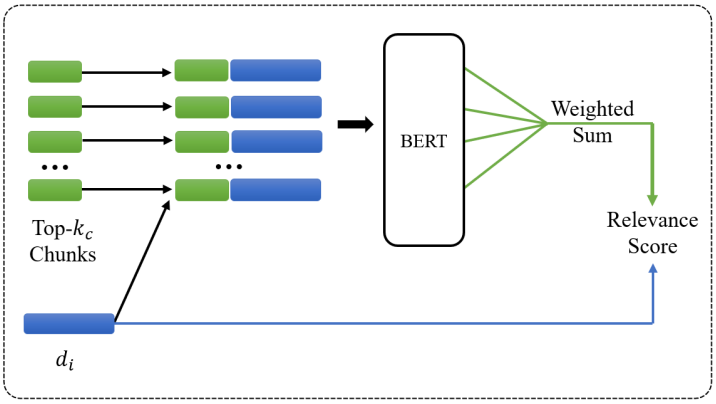

Figure 2: Re-rank documents using selected chunks in phase three.

the baselines, have been interpolated with the initial ranking scores (DPH+KL in this work) in the same way wherein the hyper-parameters are tuned in cross-validation[2]. We report P@20, NDCG@20 to enable the comparisons on the shallow pool; and MAP@100, MAP@1000 are reported for deep pool. In addition, statistical significance for paired two-tailed t-test is reported, where the superscripts $***$, $**$ and $*$ denote the significant level at 0.01, 0.05, and 0.1, respectively.

3.2 Initial Ranking

DPH+KL is used as the ranking model to generate the initial ranking. DPH is an unsupervised retrieval model (Amati et al., 2007) derived from the divergence-from-randomness framework. DPH+KL ranks the documents with DPH after expanding the original queries with Rocchio's query expansion using Kullback-Leibler divergence (Amati, 2003; Rocchio, 1971), as implemented in the Terrier toolkit (Macdonald et al., 2012). Its results are also listed for comparison.

3.3 Models in Comparisons

Unsupervised query expansion models, like Rocchio's query expansion (Rocchio, 1971) with the KL divergence model (Amati, 2003), and RM3 (Lavrenko and Croft, 2001), are employed as a group of baseline models, wherein the query is expanded by selecting terms from top-ranked documents from the initial ranking.

- **BM25+RM3** is also used as a baseline model, which follows the experimental settings from (Yilmaz et al., 2019), and the implementation from Anserini (Lin et al., 2016) with default settings is used.

- **QL+RM3** is the query likelihood language model with RM3 for PRF (Lavrenko and Croft, 2001), for which the Anserini's (Lin et al., 2016) implementation with default settings is used.

Neural ranking models. We also include different neural ranking models for comparisons.

- **SNRM** (Zamani et al., 2018) is a standalone neural ranking model by introducing a sparsity property to learn a latent sparse representation for each query and document. The best-performing version of SNRM with PRF is included for comparison.

- **NPRF** (Li et al., 2018) is an end-to-end neural PRF framework that can be used with existing neural IR models, such as DRMM (Guo et al., 2016). The best-performing variant NPRF$_{ds}$-DRMM is included for comparison.

- **CEDR** (MacAvaney et al., 2019) incorporates the classification vector of BERT into existing neural models. The best-performing variant CEDR-KNRM is included for comparison.

- **Birch** (Yilmaz et al., 2019) is a re-ranking approach by fine-tuning BERT successively on the MS MARCO and MicroBlog (MB) datasets. The best-performing version 3S: BERT(MS MARCO→MB), denoted as Birch(MS→MB) for brevity, is included for comparison.

- **BERT-Large and BERT-Base** in the MaxP configuration are fine-tuned on the training sets with cross-validation as described in Section 2.1.

3.4 Variants of BERT

Different variants of BERT models with different configurations are employed. We list the key hyper-parameters of each variant in Table 1, namely, the

[2]The details of the interpolation for BERT-QE are included in Appendix.

Size	Configuration
Tiny (T)	$L = 2, H = 128, A = 2$
Small (S)	$L = 4, H = 256, A = 4$
Medium (M)	$L = 8, H = 512, A = 8$
Base (B)	$L = 12, H = 768, A = 12$
Large (L)	$L = 24, H = 1024, A = 16$

Table 1: Configurations of different BERT variants.

number of hidden layers, the hidden embedding size, and the number of attention heads, which are denoted as L, H and A, respectively[3]. The details of these models can be found in (Turc et al., 2019). We indicate the configurations used for individual phases with the model's suffix. For example, BERT-QE-**LLS** indicates that a fine-tuned BERT-**L**arge is used in phases one and two, and in phase three a fine-tuned BERT-**S**mall is used.

3.5 Implementation of BERT-QE

Individual documents are decomposed into overlapped passages with 100 words using a sliding window, wherein the stride is 50. For the proposed BERT-QE, in phase two, $k_d = 10$ top-ranked documents from the search results of phase one are used, from which $k_c = 10$ chunks are selected for expansion, and chunk length $m = 10$ is used. In phase one and phase three, the BERT model is used to re-rank the top-1000 documents. In Section 5, we also examine the use of different k_c and m, namely, $k_c = [5, 10, 20]$ and $m = [5, 10, 20]$, investigating the impacts of different configurations.

3.6 Training

To feed individual query-document pairs into the model, the query q and the document[4] d for training are concatenated and the maximum sequence length is set to 384. We train BERT using cross-entropy loss for 2 epochs with a batch size of 32 on a TPU v3. The Adam optimizer (Kingma and Ba, 2015) is used with the learning rate schedule from (Nogueira and Cho, 2019) with an initial learning rate of 1e-6. We conduct a standard five-fold cross-validation. Namely, queries are split into five equal-sized partitions. The query partition on Robust04 follows the settings from (Dai and Callan, 2019). On GOV2, queries are partitioned by the order of TREC query id in a round-robin manner. In each fold, three partitions are used for training, one is for validation, and the remaining one is for testing. In each fold, we tune the hyper-parameters on the validation set and report the performance on test set based on the configurations with the highest NDCG@20 on the validation set[5]. The ultimate performance is the average among all folds.

3.7 Computation of FLOPs

Akin to literature (Liu et al., 2020), we report FLOPs (floating point operations) which measures the computational complexity of models. Similar to (Khattab and Zaharia, 2020), we report FLOPs that includes all computations in the three phases of BERT-QE.

4 Results

In this section, we report results for the proposed BERT-QE model and compare them to the baseline models. First, in Section 4.1, we use BERT-Large models for all three phases of BERT-QE. In Section 4.2, we evaluate the impact of using smaller BERT models (Table 1) for the second and third phases in order to improve the efficiency of the proposed model.

4.1 Results for BERT-QE-LLL

In this section, we examine the performance of the proposed BERT-QE by comparing it with a range of unsupervised ranking models, neural IR models, and re-ranking models based on BERT-Base and BERT-Large. We aim at advancing the state-of-the-art ranking performance of BERT-Large, and start with using BERT-Large for all three phases in BERT-QE. We denote this variant as BERT-QE-LLL, where the suffix LLL indicates the use of the same fine-tuned BERT-Large in all three phases[6].

The effectiveness of BERT-QE-LLL. To put our results in context, we first compare BERT-QE-LLL with the reported effectiveness for different neural IR models from literature. Due to the fact that results for GOV2 have not been reported in these works, only the comparisons on Robust04 are included in Table 2. In comparison with the state-of-the-art results of a fine-tuned BERT-Large, namely, Birch(MS→MB) (Yilmaz et al.,

[3]Note that the BERT-Small corresponds to BERT-Mini in https://github.com/google-research/bert, for the sake of convenient descriptions.

[4]As described in Section 2.1, we actually use the most relevant passage.

[5]Results on validation sets can be found in Appendix.

[6]Empirically, the BERT trained on MS MARCO is directly used in phase two, which performs on par with using the fine-tuned BERT according to pilot experiments.

Model	P@20	NDCG@20	MAP@1K
SNRM with PRF	0.3948	0.4391	0.2971
NPRF	0.4064	0.4576	0.2904
CEDR	0.4667	0.5381	-
Birch(MS→MB)	0.4669	0.5325	0.3691
BERT-Large	0.4769*	0.5397	0.3743
BERT-QE-LLL	**0.4888*****	**0.5533*****	**0.3865*****

Table 2: Compare the effectiveness of BERT-QE-LLL with neural IR models and neural PRF model on Robust04 when using title queries. Statistical significance relative to Birch(MS→MB) (Yilmaz et al., 2019) at $\texttt{p-value} < 0.01$, 0.05, and 0.1 are denoted as $***$, $**$, and $*$, respectively.

2019), it can be seen that the fine-tuned BERT-Large in this work achieves comparable results. In addition, BERT-QE-LLL significantly outperforms Birch(MS→MB) at the 0.01 level. The significance tests relative to other models are omitted because their result rankings are not available.

As summarized in Table 3, we further compare BERT-QE-LLL with BERT-Base and BERT-Large on both Robust04 and GOV2. We also include several unsupervised baselines for reference. As can be seen, BERT-Large significantly outperforms all non-BERT baselines by a big margin, regardless of whether query expansion is used. Thus, only significance tests relative to BERT-Large are shown. From Table 3, on Robust04, in comparison with BERT-Large, BERT-QE-LLL could significantly improve the search results on both shallow and deep pool at 0.01 significant level, achieving a 2.5% improvement in terms of NDCG@20 and a 3.3% improvement for MAP@1K. On GOV2, we have similar observations that BERT-QE-LLL could significantly improve BERT-Large on all reported metrics.

The efficiency of BERT-QE. Beyond the effectiveness, we are also interested in the efficiency of BERT-QE-LLL, for which the FLOPs is reported. The FLOPs per query for BERT-Large is 232.6T, meanwhile BERT-QE-LLL is 2603T. This means BERT-QE-LLL requires 11.19x more computations than BERT-Large. This is mostly due to the use of BERT-Large models for all three phases as described in Section 2. Note that, one may be able to reduce the time consumption during inference by parallelizing the individual phases of BERT-QE. In the following, the efficiency of a model is reported in terms of its relative comparison to BERT-Large, namely, in the form of the times of BERT-Large's computational cost.

4.2 Employing Smaller BERT Variants in BERT-QE

According to Section 4.1, although with competitive effectiveness, BERT-QE-LLL is very expensive for computation due to the use of BERT-Large in all three phases. In this section, we further explore whether it is possible to replace the BERT-Large components with smaller BERT variants from Table 1 in the second and third phases, in order to further improve the efficiency of the proposed BERT-QE model. Given that our goal is to improve on BERT-Large, in this work, we always start with BERT-Large for the first-round ranking.

Smaller BERT variants for chunk selector. As described in Section 2.2, in the second phase, a BERT model is used to select text chunks of a fixed length (i.e., $m = 10$) by evaluating individual text chunks from the top-k_d documents and selecting the most relevant chunks using a BERT model. Intuitively, compared with ranking a document, evaluating the relevance of a short piece of text is a relatively simple task. Thus, we examine the use of smaller BERT variants as summarized in the second section (namely, BERT-QE-LXL, where X is T, S, M, or B) in Table 4. As shown, compared with using BERT-Large in phase two, on Robust04, all four BERT-QE variants can outperform BERT-Large significantly at the 0.01 level. Furthermore, BERT-QE-LML can even achieve slightly higher results than BERT-QE-LLL. On GOV2, on the other hand, the uses of BERT-Tiny, BERT-Small, and BERT-Medium could still outperform BERT-Large significantly at the 0.05 or 0.1 level, but with decreasing metrics in most cases. Overall, for phase two, BERT-Large is a good choice but the smaller BERT variants are also viable. The uses of BERT-Tiny, BERT-Small, and BERT-Medium in phase two can outperform BERT-Large significantly with lower FLOPs.

Smaller BERT variants for final re-ranker.

Model	Robust04				GOV2			
	P@20	NDCG@20	MAP@100	MAP@1K	P@20	NDCG@20	MAP@100	MAP@1K
DPH	0.3616	0.4220	0.2150	0.2512	0.5295	0.4760	0.1731	0.3012
BM25+RM3	0.3821	0.4407	0.2451	0.2903	0.5634	0.4851	0.2022	0.3350
QL+RM3	0.3723	0.4269	0.2314	0.2747	0.5359	0.4568	0.1837	0.3143
DPH+KL	0.3924	0.4397	0.2528	0.3046	0.5896	0.5122	0.2182	0.3605
BERT-Base	0.4653	0.5278	0.3153	0.3652	0.6591	0.5851	0.2535	0.3971
BERT-Large	0.4769	0.5397	0.3238	0.3743	0.6638	0.5932	0.2612	0.4082
BERT-QE-LLL	**0.4888*****	**0.5533*****	**0.3363*****	**0.3865*****	**0.6748*****	**0.6037*****	**0.2681*****	**0.4143*****

Table 3: Effectiveness of BERT-QE-LLL. Statistical significance relative to BERT-Large at $\texttt{p-value} < 0.01$, 0.05, and 0.1 are denoted as $***$, $**$, and $*$, respectively.

Model	FLOPs	Robust04				GOV2			
		P@20	NDCG@20	MAP@100	MAP@1K	P@20	NDCG@20	MAP@100	MAP@1K
BERT-Base	0.28x	0.4653	0.5278	0.3153	0.3652	0.6591	0.5851	0.2535	0.3971
BERT-Large	1.00x	0.4769	0.5397	0.3238	0.3743	0.6638	0.5932	0.2612	0.4082
BERT-QE-LLL	11.19x	0.4888***	0.5533***	0.3363***	0.3865***	**0.6748*****	**0.6037*****	0.2681***	**0.4143*****
BERT-QE-L**T**L	11.00x	0.4855***	0.5500***	0.3318***	0.3821***	0.6691**	0.5986*	0.2663***	0.4138***
BERT-QE-L**S**L	11.00x	0.4861***	0.5504***	0.3325***	0.3828***	0.6732***	0.6011**	**0.2685*****	0.4142***
BERT-QE-L**M**L	11.01x	**0.4932*****	**0.5592*****	**0.3368*****	**0.3870*****	0.6715**	0.6013*	0.2675*	0.4136*
BERT-QE-L**B**L	11.05x	0.4839**	0.5503***	0.3339***	0.3843***	0.6725**	0.6004	0.2639	0.4103
BERT-QE-LM**T**	1.03x	0.4839***	0.5483***	0.3276*	0.3765	0.6698**	0.5994**	0.2642	0.4098
BERT-QE-LM**S**	1.12x	0.4910***	0.5563***	0.3315***	0.3810**	0.6658	0.5945	0.2654***	0.4115***
BERT-QE-LM**M**	1.85x	0.4888***	0.5569***	0.3335***	0.3829***	0.6732***	0.6002*	0.2668***	0.4131***
BERT-QE-LM**B**	3.83x	0.4906***	0.5580***	0.3367***	0.3858***	0.6728***	0.6011**	0.2649	0.4128**
BERT-QE-LL**T**	1.20x	0.4841***	0.5466**	0.3287**	0.3771	0.6695**	0.6009**	0.2650**	0.4110*
BERT-QE-LL**S**	1.30x	0.4869***	0.5501**	0.3304**	0.3798*	0.6688*	0.5998**	0.2657***	0.4115***
BERT-QE-LL**M**	2.03x	0.4811	0.5470	0.3320**	0.3815**	0.6728***	0.6013***	0.2651**	0.4107
BERT-QE-LL**B**	4.01x	0.4865***	0.5507***	0.3337***	0.3834***	0.6678	0.5984	0.2665**	0.4127**

Table 4: Employ different BERT variants for phase two and three in BERT-QE, wherein BERT-Tiny (T), BERT-Small (S), BERT-Medium (M), and BERT-Base (B) are used. Statistical significance relative to BERT-Large at $\texttt{p-value} < 0.01$, 0.05, and 0.1 are denoted as $***$, $**$, and $*$, respectively.

According to Section 2.3, phase three is the most expensive phase, because a BERT model must compare each document to multiple expansion chunks. Thus, we further explore the possibility of replacing BERT-Large with smaller BERT variants for phase three. Based on the results in the previous section, we consider both BERT-Large and BERT-Medium as the chunk selector, due to the superior effectiveness of BERT-QE-LML. The results are summarized in the third and fourth sections (namely, BERT-QE-LMX and BERT-QE-LLX, where X is T, S, M, or B) of Table 4. On Robust04, the use of smaller BERT variants always leads to decreasing effectiveness. However, when using BERT-Small and BERT-Base for the final re-ranking, the corresponding BERT-QE variants always outperform BERT-Large significantly at the 0.1 level. BERT-QE-LMM, BERT-QE-LMB, and BERT-QE-LLB can even consistently outperform BERT-Large on all four metrics at the 0.01 level. On GOV2, on the other hand, the use of BERT-QE-LMT and BERT-QE-LLM significantly outperforms BERT-Large on shallow metrics, while BERT-QE-LMS and BERT-QE-LLB outperform BERT-Large on deep metrics. In addition, BERT-QE-LMM/LLT/LLS consistently outperform BERT-Large on all metrics at 0.1 level. Overall, considering shallow metrics on both datasets, BERT-QE-LMT can outperform BERT-Large consistently and significantly at the 0.05 level while requiring only 3% more FLOPs. On both shallow and deep metrics, BERT-QE-LLS significantly outperforms BERT-Large with 30% more FLOPs.

5 Analysis

5.1 First-round Re-ranker Ablation Analyses

Intuitively, there are two functions of the first-round ranker: providing the $rel(q, d)$ score in Equation (5) used in the final re-ranking, and providing the top-k_d documents from which the candidate chunks are selected, which are used to compute $rel(\mathcal{C}, d)$ in Equation (4). In this section, we investigate the impact of the first-round re-ranker from these two perspectives. In particular, we conduct

Model	P@20	NDCG@20	MAP@1K
BERT-Large	0.4769	0.5397	0.3743
BERT-QE-LLL	**0.4888*****	**0.5533*****	**0.3865*****
Remove $rel(q, d)$	0.4769	0.5372	0.3767
Chunks from DPH+KL	0.4759	0.5391	0.3766

Table 5: Ablation analyzes for the first-round re-ranker in BERT-QE-LLL, by removing the $rel(q, d)$ from Equation (5) and by replacing the chunks with the ones selected from top-ranked documents of DPH+KL when computing $rel(q, \mathcal{C})$ in Equation (4). Statistical significance relative to BERT-Large at $\texttt{p-value} < 0.01$, 0.05, and 0.1 are denoted as $***$, $**$, and $*$, respectively.

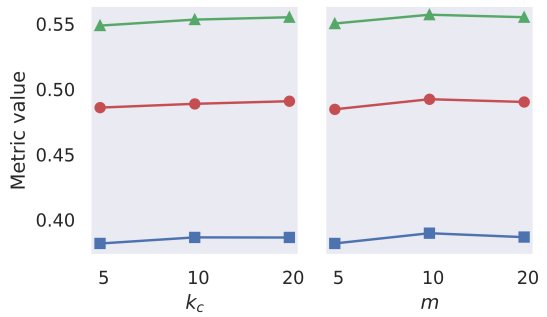

Figure 3: Performance of BERT-QE with different configurations of k_c and m. The $\circ$, $\triangle$, $\square$ correspond to results in terms of P@20, NDCG@20, and MAP@1K, respectively.

two ablation analyses: (1) we remove the $rel(q, d)$ from BERT-Large in Equation (5), but we continue to use the top documents from BERT-Large to select the top-k_c chunks; and (2) we keep the $rel(q, d)$ from BERT-Large in Equation (5), but we select the top-k_c chunks from documents returned by the unsupervised DPH+KL model. The results are summarized in Table 5. For the first ablation, when $rel(q, d)$ from BERT-Large is not used, BERT-QE cannot outperform BERT-Large. Similarly, in the second ablation, selecting chunks from the documents returned by DPH+KL also prevents BERT-QE from outperforming BERT-Large. These results highlight the importance of both functions of the first-round re-ranker. That is, we need a powerful model for the first-round re-ranker to provide ranking score $rel(q, d)$ and the high-quality feedback documents for the chunk selector.

5.2 Hyper-parameter study

There are two hyper-parameters in the proposed BERT-QE, namely k_c and m. k_c is the number of chunks used in the final-round re-ranking as described in Equation (4). Meanwhile, the chunk size m balances between contextual information and noise. Results for different hyper-parameter settings on Robust04 are shown in Figure 3. For k_c, it can be seen that $k_c = 10, 20$ achieve similar performance, while $k_c = 5$ reduces the results. As the computational cost of phase three is proportional to k_c and the performance gaps between $k_c = 10$ and $k_c = 20$ are actually quite small, $k_c = 10$ is a reasonable and robust configuration. Among different settings of m, $m = 10$ achieves the best performance and therefore is used in the proposed model.

6 Related Work

BERT for IR. Inspired by the success of contextualized models like BERT on NLP tasks, Nogueira and Cho (2019) examine the performance of BERT on the passage re-ranking tasks using MS MARCO and TREC-CAR datasets, and demonstrate superior performances compared with the existing shallow ranking models like Co-PACRR (Hui et al., 2018) and KNRM (Xiong et al., 2017). Thereafter, the application of contextualized BERT model in ranking tasks have attracted many attentions. Dai and Callan (2019) split a document into fixed length passages and use a BERT ranker to predict the relevance of each passage independently. The score of the first passage, the best passage, or the sum of all passage scores is used as the document score. MacAvaney et al. (2019) incorporate BERT's classification vector into existing neural models, including DRMM (Guo et al., 2016), PACRR (Hui et al., 2017), and KNRM (Xiong et al., 2017), demonstrating promising performance boosts. Yilmaz et al. (2019) transfer models across different domains and aggregate sentence-level evidences to rank documents. Nogueira et al. (2019a) propose a multi-stage ranking architecture with BERT that can trade off quality against latency. Wu et al. (2020) propose the context-aware Passage-level Cumulative Gain to aggregate passage relevance representations scores, which is incorporated into a BERT-based model for document ranking. In ad-

dition to these efforts, this work further proposes to exploit the contextualized BERT model to expand the original queries in the proposed BERT-QE framework, boosting the ranking performance by using the pseudo feedback information effectively.

Query expansion has long been applied to make use of the pseudo relevance feedback information (Hui et al., 2011) to tackle the vocabulary mismatch problem. Keyword query expansion methods, such as Rocchio's algorithm (Rocchio, 1971) and the KL query expansion model (Amati, 2003), have been shown to be effective when applied to text retrieval tasks. Moreover, Metzler and Croft (2007) propose to expand beyond unigram keywords by using a Markov random field model. Some query expansion methods use word embeddings to find the relevant terms to the query (Diaz et al., 2016; Zamani and Croft, 2016). Cao et al. (2008) perform query expansion by using classification models to select expansion terms. NPRF (Li et al., 2018) incorporates existing neural ranking models like DRMM (Guo et al., 2016) into an end-to-end neural PRF framework. Rather than expanding the query, Nogueira et al. (2019b) propose a document expansion method named Doc2query, which uses a neural machine translation method to generate queries that each document might answer. Doc2query is further improved by docTTTTTquery (Nogueira and Lin, 2019) which replaces the seq2seq transformer with T5 (Raffel et al., 2019). MacAvaney et al. (2020b) construct query and passage representations and perform passage expansion based on term importance. Despite the promising results of the above document expansion methods for passage retrieval, they are so far only applied to short text retrieval tasks to avoid excessive memory consumption. In comparison with these established expansion models, the proposed BERT-QE aims at better selecting and incorporating the information pieces from feedback, by taking advantages of the BERT model in identifying relevant information.

7 Conclusion

This work proposes a novel expansion model, coined as BERT-QE, to better select relevant information for query expansion. Evaluation on the Robust04 and GOV2 test collections confirms that BERT-QE significantly outperforms BERT-Large with relatively small extra computational cost (up to 30%). In future work, we plan to further improve the efficiency of BERT-QE, by combining the proposed BERT-QE with the pre-computation techniques proposed recently (Khattab and Zaharia, 2020; MacAvaney et al., 2020a), wherein most of the computations could be performed offline.

Acknowledgments

This research work is supported by the National Natural Science Foundation of China under Grants no. U1936207 and 61772505, Beijing Academy of Artificial Intelligence (BAAI2019QN0502), the Youth Innovation Promotion Association CAS (2018141), and University of Chinese Academy of Sciences.

References

Giambattista Amati. 2003. *Probability models for information retrieval based on divergence from randomness*. Ph.D. thesis, University of Glasgow, UK.

Gianni Amati, Edgardo Ambrosi, Marco Bianchi, Carlo Gaibisso, and Giorgio Gambosi. 2007. Fub, IASI-CNR and university of tor vergata at TREC 2007 blog track. In *TREC*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST).

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268v3.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR*, pages 243–250. ACM.

Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. 2004. Overview of the TREC 2004 terabyte track. In *TREC*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST).

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *SIGIR*, pages 985–988. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. In *ACL (1)*. The Association for Computer Linguistics.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM*, pages 55–64. ACM.

Kai Hui, Ben He, Tiejian Luo, and Bin Wang. 2011. A comparative study of pseudo relevance feedback for ad-hoc retrieval. In *ICTIR*, volume 6931 of *Lecture Notes in Computer Science*, pages 318–322. Springer.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A position-aware neural IR model for relevance matching. In *EMNLP*, pages 1049–1058. Association for Computational Linguistics.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2018. Co-pacrr: A context-aware neural IR model for ad-hoc retrieval. In *WSDM*, pages 279–287. ACM.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*, pages 39–48. ACM.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *SIGIR*, pages 120–127. ACM.

Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, and Jungang Xu. 2018. NPRF: A neural pseudo relevance feedback framework for ad-hoc information retrieval. In *EMNLP*, pages 4482–4491. Association for Computational Linguistics.

Jimmy J. Lin, Matt Crane, Andrew Trotman, Jamie Callan, Ishan Chattopadhyaya, John Foley, Grant Ingersoll, Craig MacDonald, and Sebastiano Vigna. 2016. Toward reproducible baselines: The open-source IR reproducibility challenge. In *ECIR*, volume 9626 of *Lecture Notes in Computer Science*, pages 408–420. Springer.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. Fastbert: a self-distilling BERT with adaptive inference time. In *ACL*, pages 6035–6044. Association for Computational Linguistics.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020a. Efficient document re-ranking for transformers by precomputing term representations. In *SIGIR*, pages 49–58. ACM.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020b. Expansion via prediction of importance with contextualization. In *SIGIR*, pages 1573–1576. ACM.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: contextualized embeddings for document ranking. In *SIGIR*, pages 1101–1104. ACM.

Craig Macdonald, Richard McCreadie, Rodrygo LT Santos, and Iadh Ounis. 2012. From puppy to maturity: Experiences in developing terrier. *Proc. of OSIR at SIGIR*, pages 60–63.

Donald Metzler and W. Bruce Croft. 2007. Latent concept expansion using markov random fields. In *SIGIR*, pages 311–318. ACM.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. *CoRR*, abs/2003.06713.

Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. *Technical report*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019a. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019b. Document expansion by query prediction. *CoRR*, abs/1904.08375.

Yifan Qiao, Chenyan Xiong, Zheng-Hao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of BERT in ranking. *CoRR*, abs/1904.07531.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

J. Rocchio. 1971. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART retrieval system: experiments in automatic document processing*, pages 313–323. Prentice Hall, Englewood, Cliffs, New Jersey.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 robust track. In *TREC*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST).

Zhijing Wu, Jiaxin Mao, Yiqun Liu, Jingtao Zhan, Yukun Zheng, Min Zhang, and Shaoping Ma. 2020. Leveraging passage-level cumulative gain for document ranking. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2421–2431. ACM / IW3C2.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR*, pages 55–64. ACM.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *EMNLP/IJCNLP (1)*, pages 3488–3494. Association for Computational Linguistics.

Hamed Zamani and W. Bruce Croft. 2016. Embedding-based query language models. In *ICTIR*, pages 147–156. ACM.

Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik G. Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *CIKM*, pages 497–506. ACM.

A Appendices

A.1 Interpolation Parameters in BERT-QE

	Robust04					
Fold	P@20	NDCG@20	MAP@100	MAP@1K	α	β
1	0.4730	0.5606	0.3247	0.3765	0.4	0.9
2	0.4900	0.5666	0.3909	0.4362	0.4	0.8
3	0.4740	0.5328	0.2941	0.3471	0.4	0.9
4	0.4684	0.5213	0.2940	0.3440	0.6	0.9
5	0.5400	0.5868	0.3709	0.4233	0.3	0.9
	GOV2					
1	0.6233	0.5728	0.2257	0.3621	0.4	0.9
2	0.7397	0.6675	0.3046	0.4334	0.7	0.9
3	0.7167	0.6177	0.2558	0.4456	0.1	0.7
4	0.6850	0.6027	0.2718	0.4140	0.4	0.8
5	0.6300	0.5731	0.2860	0.4240	0.4	0.8

Table 6: Results on *validation sets*, as well as the chosen interpolation parameters α and β based on *validation sets* for BERT-QE-LLL.

There are two hyper-parameters in BERT-QE, namely α and β, both of which are interpolation coefficients. α is introduced in Equation (5). In addition, akin to (Yilmaz et al., 2019), there is an interpolation with the initial ranking, i.e., DPH+KL, which has been applied to all models, including BERT-QE and baselines, where β is the hyper-parameter. As shown in the following equation, $M(q, d)$ denotes the scores from a re-ranking model, e.g., BERT-QE model. $I(q, d)$ denotes the scores from the initial ranking, namely, DPH+KL. α and β are both tuned on the validation set through grid search on (0,1) with stride 0.1. The models with best nDCG@20 on validation sets are chosen. Different configurations of α and β and the corresponding results are summarized in Table 6.

$$final\_score = \beta \cdot \log(M(q, d)) + (1 - \beta) \cdot I(q, d)$$

Size	# of parameters
Tiny (T)	4M
Small (S)	11M
Medium (M)	41M
Base (B)	109M
Large (L)	335M

Table 7: Number of parameters in BERT variants.

A.2 Number of parameters in BERT variants

We list the number of parameters in different BERT variants used in BERT-QE in Table 7.