Query-as-context Pre-training for Dense Passage Retrieval

Xing Wu Guangyuan Ma Wanhui Qian Zijia Lin
Songlin Hu

Abstract

Recently, methods have been developed to improve the performance of dense passage retrieval by using context-supervised pre-training. These methods simply consider two passages from the same document to be relevant, without taking into account the potential negative impacts of weakly correlated pairs. Thus, this paper proposes query-as-context pre-training, a simple yet effective pre-training technique to alleviate the issue. Query-as-context pre-training assumes that the query derived from a passage is more likely to be relevant to that passage and forms a passage-query pair. These passage-query pairs are then used in contrastive or generative context-supervised pre-training. The pre-trained models are evaluated on large-scale passage retrieval benchmarks and out-of-domain zero-shot benchmarks. Experimental results show that query-as-context pre-training brings considerable gains for retrieval performances, demonstrating its effectiveness and efficiency.

1 Introduction

Passage retrieval is the process of retrieving relevant passages from a large corpus in response to a query, which is useful in a variety of downstream applications such as web search (Fan et al., 2021; Guo et al., 2022; Lin et al., 2021a), question answering (Karpukhin et al., 2020; Lee et al., 2020; Zhu et al., 2021) and dialogue systems (Gao et al., 2022a; Yu et al., 2021). The success of pre-trained language models (PLMs) (Devlin et al., 2018; Liu et al., 2019) has led to the development of more powerful PLM-based dense and sparse passage retrieval approaches.

PLM-based dense retrieval methods (Xiong et al., 2020; Lu et al., 2021; Hofst"atter et al., 2021; Gao and Callan, 2021b; Ren et al., 2021b; Ma et al., 2022; Liu and Shao, 2022; Wu et al., 2022; Wang et al., 2022)

use PLMs to encode queries and passages into a shared semantic space. The semantic relationships between query and passage representations are then measured by dot product or cosine similarities. Pre-training and fine-tuning techniques have been developed to improve the performance of dense retrieval models. Pre-training processes for dense retrieval aim to improve the text representation modeling ability of the encoder through auxiliary self-supervised or context-supervised tasks.

Context-supervised pre-training (Gao and Callan, 2021b; Wu et al., 2022) assumes that two passages within the same document are contextual or related to each other and can therefore be used for contrastive learning or contextual decoding. However, context-supervised pre-training ignores the fact that the passages within a document may be weakly related or even irrelevant in many cases. As shown in Figure 1, two passages within a document from the MS-MARCO corpus (Nguyen et al., 2016) are not directly related in content. According to our statistic results via human annotation, only 35.5

In contrast to dense retrieval, sparse retrieval is based on the "bag-of-words" assumption and represents passages and queries as sparse term-based vectors. PLM-based sparse retrieval (Nogueira and Lin, 2019; Dai and Callan, 2019; Mao et al., 2020; Bai et al., 2020; Formal et al., 2021b,a; Mallia et al., 2021; Shen et al., 2022) uses PLM to improve sparse vectors. One representative technique is Query Prediction (Nogueira and Lin, 2019), which predicts a set of relevant queries to enrich the passage's content and thus alleviates the mismatch problem. Query prediction has been shown to be effective in sparse retrieval, but has not yet been explored in the context of dense retrieval, especially in the pre-training process. This raises the question of whether the query prediction technique can benefit the pre-training process tailored for dense retrieval.

The observation that predicted queries align better with the content of a passage in our statistical analyses (see Appendix A) suggests that query prediction could be a promising way to alleviate the issue of weakly correlated passages for context-supervised pre-training. Thus, this paper focuses on exploring query prediction techniques to improve context-supervised pre-training methods for dense retrieval. Our proposed method, termed query-as-context pre-training, assumes that a query derived from a passage (using a generative model like T5) is more likely to be a relevant context to the passage. In contrast to the previous context-supervised methods that create a training pair using two randomly selected passages from a document, the query-as-context method generates a training pair by combining a passage with a predicted query, as illustrated in Figure 2.

There are several advantages to using the query-as-context setting. Firstly, the query is more likely to be related to the passage because it is generated from the passage. Additionally, the use of passage-query pairs in supervised downstream retrieval training is consistent with using passage-query pairs in pre-training, which helps to bridge the gap between the two processes. Finally, since the passage-query pairs are generally shorter than the previously used passage-passage pairs, it speeds up the pre-training process and reduces the training overhead.

To verify the effectiveness of our proposed query-as-context pre-training, we conduct experiments on large-scale web search benchmarks: MS-MARCO Passage Ranking (Nguyen et al., 2016), TREC Deep Learning (DL) Track 2019 (Craswell et al., 2020a) and Track 2020 (Craswell et al., 2020b). We also evaluate query-as-context pre-trained models on the BEIR (Thakur et al., 2021) benchmark with a large set of out-of-domain datasets. Experimental results show that query-as-context achieves considerable gains over competing baselines.

Our contributions can be summarized as follows:

- We reveal the previously ignored issue of weakly correlated passage pairs during context-supervised pre-training.
- We propose query-as-context pre-training, a simple yet effective pre-training technique to alleviate the issue above.
- Experiments show that query-as-context pre-training brings considerable gains and meanwhile speeds up pre-training.

2 Preliminary: Context-supervised Pre-training

In this section, we begin by providing an overview of the pre-training corpus. Subsequently, we describe the masked language modeling task, which serves as a foundational task of pre-training. Finally, we present two representative contrastive and generative context-supervised pre-training methods, on which our proposed query-as-context will be applied.

Pre-training Corpus Given a set of documents, we randomly extract pairs of passages from each document, which forms a training corpus as follows: $[\ \mathrm{x}_0, y_0, \ldots, x_n, y_n] where \mathrm{x}_a, y_a$ is a pair of passages from the same document.

Masked Language Modeling (MLM) Formally, given a passage x with n tokens, a special token [CLS] is added to the beginning of the passage, resulting in [x : $r_0, r_1, \ldots, r_n]where r_0$ represents the [CLS] token. Then, a certain percentage of positions are randomly selected as "mask positions" (maskpos) and are replaced with a special token [MASK] or a random token. The masked passage is then passed through a text encoder, which commonly consists of L layers of transformer blocks. For the l-th transformer layer in the encoder, its outputs are the hidden states of the layer [h_l : $h_l^0, h_l^1, \ldots, h_l^n]The output of the last layer is then used to calculate the MLM' s target loss [L_{\text{mlm}} := \sum_{i \in maskpos} \text{CE}(g(h_i), r_i)]where CE is short for cross entropy function and g is a projection of the correspon$ to a vocabulary distribution.

2.1 coCondenser

coCondenser (Gao and Callan, 2021b) is a representative contrastive context-supervised method. For coCondenser, two passages from a document are considered relevant and form a positive pair, while two passages from different documents are considered as irrelevant and form a negative pair. These pairs constitute mini-batches for contrastive learning. A common approach for generating an embedding representation of a passage is to use the hidden states of the [CLS] position in the last layer of the encoder, i.e., h_L^0. Thus, the embedding representations of passages x and y are $h_L^0(x)$ and $h_L^0(y)$, simplified as h_x and h_y. Then, for a mini-batch B, the contrastive learning objective w.r.t. x is formulated as: [$\text{L}_{\text{co}} =$

$* \log \frac{\exp(\text{sim}(h_x, h_y)/\tau)}{\sum_{h' \in B} \exp(\text{sim}(h_x, h')/\tau)}]where \tau$ is a temperature hyper-parameter and $\text{sim}(\cdot, \cdot)$ is the dot product similarity function.

An additional auxiliary decoder is also appended to the encoder, which consists of M layers of transformers. The auxiliary decoder takes the concatenation of the [CLS] representation from the L-th layer, i.e., h_L^0, and the token representations from the encoder's M-th (e.g. $L/2$-th) layer, i.e., $h_M^1, \ldots, h_M^n$, as inputs. Similar to MLM, the output of the auxiliary decoder's last layer is then used to perform an auxiliary MLM pre-training: [$\text{L}_{\text{aux-mlm}} := \sum_{i \in maskpos} \text{CE}(g(d_i), r_i)]Finally, the total all loss of coCondenser is : [L = L_{\text{mlm}} + L_{\text{aux-mlm}} + L_{\text{co}}]For more details, please refer to (Gao and Callan, 2021b).$

2.2 CoT-MAE

CoT-MAE (Wu et al., 2022) is a representative generative context-supervised method that uses an asymmetric encoder-decoder structure, with a deep encoder of L layers and a shallow decoder of M layers. It performs MLM

training on both the encoder and the decoder simultaneously. For a pair of passages x, y, suppose x is fed into the encoder side and y is fed into the decoder side.

On the encoder side, x is reconstructed using only the unmasked tokens in the passage, similar to BERT's MLM process, but with a higher mask rate (e.g. 30[$\mathrm{d}_0 := h_x, a_1, \ldots, a_n]$*Theconcatenation*$\mathrm{d}_0$ is then passed through the M layers of Transformer blocks, and the hidden states of k layer is formulated as: [$\mathrm{d}_k : d_k^0, d_k^1, \ldots, d_k^n$]*Theoutputsofthelastlayerindecoderarethenusedfor*LM*pre*$-$ *training*, *withthelossdefinedas* : $[L_{\text{ctx-mlm}} := \sum_{i \in maskpos} \mathrm{CE}(g(d_i), s_i)]$*Thesubscriptctxdenotestheproc supervised.Then*, *weaddthelossesfromboththeencoderandthedecodertogetthefinalloss* : $[L = L_{\text{mlm}} + L_{\text{ctx-mlm}}]$*Formoredetails*, *pleaserefer to*(*Wuetal.*, 2022).

3 Query-as-context Pre-training

In this section, we first introduce the details of query-as-context pre-training, and then introduce the fine-tuning process of the pre-trained models on the retrieval tasks.

3.1 Pre-training

Pre-training is conducted on a large scale of documents without annotations. For each document D, we extract a set of passages with a maximum length, $x_0, x_1, \ldots$. Following (Nogueira and Lin, 2019), for each passage x_i, we use a fine-tuned T5 model for generating queries. We apply nucleus sampling with top-p=0.95 and top-k=25 to produce multiple queries for promoting diversity.

Specially, each passage x_i will be fed into the fine-tuned T5 model, and generate C candidate queries, $q_i^j * j = 1^C$. During training, we will randomly select one of the candidate queries to use as the context for the passage: [$\mathrm{y}_i =$ sample$(q_i^j * j = 1^C)$]*Thepassageandsampledqueryformatrainingpair* x_i, y_i, which can be used to replace the original pair used in Equation 1. Specifically, the passage-query pair are directly used for contrastive pre-training of coCondenser. For CoT-MAE, we fed the passage into the encoder, and query into the decoder for generative pre-training. Model implementations for coCondenser and CoT-MAE have been introduced in Section 2.1 and 2.2.

3.2 Fine-tuning

We fine-tune on the downstream retrieval tasks to verify the effectiveness of pre-training. Following (Gao and Callan, 2021b; Wu et al., 2022), the

fine-tuning process on MS-MARCO is based on a two-stage pipeline with hard negative mining (Gao et al., 2022b), as depicted in Figure 3. This process involves two stages of training, bi-encoder retriever 1 and bi-encoder retriever 2, which are both initialized with the query-as-context pre-trained models. The retrievers are trained with contrastive learning on the manually annotated passage-query pairs. For a manually annotated passage-query pair (p^+, q^+), the representations of the passage and the query form a positive example (h_{p^+}, h_{q^+}).

When training retriever 1, for query q^+, the negative samples p^- include in-batch negative passages and BM25 mined hard negative passages. When training retriever 2, hard negatives are also mined using a well-trained retriever 1 and combined with the other negative passages to create the negative samples p^-. Both stages are optimized using the InfoNCE loss: [L = -log $\frac{\exp(\text{sim}(h_{p^+}, h_{q^+})/\tau)}{\sum_{p \in p^+, p^-} \exp(\text{sim}(h_p, h_{q^+})/\tau)}$]$where\tau$ is a temperature hyper-parameter fixed to 1 and $\text{sim}(\cdot, \cdot)$ is dot product similarity function. Following (Thakur et al., 2021), we train the retriever with MS-MARCO negatives for the out-of-domain evaluation on BEIR benchmarks.

4 Experiment

In this section, we provide details on the pre-training and fine-tuning processes. Then we present the experimental results.

4.1 Pre-training

Query-as-context Dataset Following (Gao and Callan, 2021b; Wu et al., 2022), the pre-training dataset is collected from the MS-MARCO passages corpus, which contains 3.2 million documents. We use NLTK to split each document into sentences, and group these sentences into passages of no more than 144 consecutive tokens. For each passage, we generate candidate queries via a public T5 model[1]](https://huggingface.co/doc2query/all-with-prefix-t5-base-v1). During pre-training, we select a batch of passages at each step and randomly choose a candidate query as context for each passage to form a relevant pair.

Model Implementation Following (Wu et al., 2022), the encoder for CoT-MAE is initialized with a pre-trained 12-layer BERT-base model, while

[1] [https://huggingface.co/doc2query/all-with-prefix-t5-base-v1

the decoder is initialized from scratch. We pre-train the model using the AdamW optimizer for a maximum of 50k steps, with a learning rate of 4e-4, a batch size of 16k, and a linear schedule with a warmup ratio of 0.1. We use 16 Tesla V100 GPUs to train the model for 60 hours, and then discard the decoder, leaving only the encoder for fine-tuning.

Following (Gao and Callan, 2021b), the encoder for coCondenser is initialized from a pre-trained 12-layer Condenser (Gao and Callan, 2021a) model. The training is conducted on 8 Tesla V100 GPUs for 120,000 steps over 90 hours using the AdamW optimizer with a learning rate of 1e-4, a global batch size of 2k, and a weight decay of 0.01. Once the pre-training is finished, the Condenser head is discarded, resulting in a model with the same architecture as $\text{BERT}_{\text{base}}$ for fine-tuning.

4.2 Fine-tuning

Datasets and Evaluation We fine-tune the pre-trained coCondenser and CoT-MAE on MS-MARCO passage ranking (Nguyen et al., 2016), TREC Deep Learning (DL) Track 2019 (Craswell et al., 2020a) and 2020 (Craswell et al., 2020b) tasks for evaluation.

MS-MARCO (Nguyen et al., 2016) is a benchmark dataset that contains real user queries collected from Bing search and web pages, and includes approximately 8.8 million passages in total. The training set consists of around 500,000 annotated query-document pairs, while the dev set contains 6,980 annotated queries. Since the test set is not publicly available, the dev set is used for evaluation following previous work (Gao and Callan, 2021b; Wu et al., 2022). We evaluate our performance on MS-MARCO using MRR@10, Recall@50, and Recall@1K.

TREC Deep Learning (DL) (Craswell et al., 2020a,b) tracks provide test sets with more elaborate annotations to evaluate the real capacity of ranking models. We evaluate the 2019 and 2020 test sets. The 2019 test set contains 43 annotated queries and the 2020 test set contains 54 annotated queries. We evaluate our performance on TREC with NDCG@10.

Implementation We reuse a widely adopted evaluation pipeline (Gao and Callan, 2021b; Wu et al., 2022; Gao et al., 2022b), with a common random seed (42) to support reproducibility. Note that, as we focus on improving the pre-training technique, we do NOT use any enhanced methods, such as distillation from a strong re-ranker (Ren et al., 2021b; Santhanam et al., 2021) or multi-vector representation (Khattab and Zaharia, 2020), which can lead to further improvements. The fine-tuning is only trained on the

MS-MARCO dataset and evaluated on the dev set and TREC DL 2019/2020 test sets. It's trained on 8 Tesla V100 GPUs using the AdamW optimizer with a learning rate of 2e-5, a global batch size of 64, and a weight decay of 0.01. The passage length is also set to 144, the negative depth is set to 200 and the number of negative passages for one query in the fine-tuning iteration is 15.

4.3 Baselines

Our baseline methods include the sparse retrieval method and the dense retrieval method, as shown in Table 1. Results of sparse retrieval baselines are mainly from (Qu et al., 2020), including BM25, docT5query (Nogueira and Lin, 2019), DeepCT (Dai and Callan, 2019) and GAR (Mao et al., 2020). Results of dense retrieval baselines are mainly from (Gao and Callan, 2021b; Liu and Shao, 2022; Ren et al., 2021b; Ma et al., 2022), including ANCE (Xiong et al., 2020), SEED (Lu et al., 2021), TAS-B (Hofst"atter et al., 2021), RetroMAE (Liu and Shao, 2022), SimLM (Wang et al., 2022) and etc. We compare the query-as-context performances with their baselines on both retriever 1 and retriever 2.

4.4 Main Results

As shown in Table 1, the results demonstrate that query-as-context pre-training leads to improved performance.

coCondenser When reproducing coCondenser, the pre-training steps extend to 120k steps. The main evaluation metric, MRR@10 on the MS-MARCO passage ranking dataset, of retriever 2 improves by 0.6pp compared to the original paper (Gao and Callan, 2021b). When query-as-context pre-training is used, there is a further improvement of 0.6pp on MRR@10. On both TREC DL 19 and 20 test sets, there are improvements of 2pp on DL 19 and 3.4pp on DL 20. In addition, query-as-context pre-training also improves the MRR@10 and R@50 scores of retriever 1.

CoT-MAE When reproducing CoT-MAE, for efficiency, we adopt a much larger batch size than in (Wu et al., 2022), which allows us to reduce the number of training steps from 1200k to 50k. This results in faster training, but somehow lower performance on the MS-MARCO MRR@10 metric compared to the original paper. However, when query-as-context pre-training is applied, there is an obvious improvement of 1.4pp on MRR@10, reaching 40.2. Even

compared to the 1200k model's performance in the original paper, we still achieve a non-trivial improvement of 0.8pp. To the best of our knowledge, this is the new state-of-the-art result for a single vector pre-trained (not a reranker-distilled) dense retriever. On both TREC DL 19 and 20 test sets, there are improvements of 0.8pp on DL 19 and 3pp on DL 20. In addition, query-as-context pre-training also improves the MRR@10, R@50, and R@1K scores of retriever 1.

Overall, the query-as-context pre-training approach is effective, improving both contrastive and generative context-supervised pre-training. This is due to two main reasons: (1) pre-trained models can provide better parameters initialization for both retriever 1 and retriever 2; (2) a better retriever 1 can be used to mine more effective hard negatives, which further improves the training of retriever 2.

4.5 Out-of-domain Evaluation

We evaluate the out-of-domain performance of query-as-context pre-trained models on the zero-shot benchmark BEIR (Thakur et al., 2021). BEIR benchmark contains 9 different open-domain information retrieval tasks from 18 different datasets. We evaluate the models on the 14 publicly available datasets[2]. As shown in the table, both the coCondenser and the CoT-MAE results show non-trivial improvements on most datasets when using query-as-context pre-training. Specifically, using query-as-context pre-training improves the performance of the coCondenser model on 9 different datasets. The improvement in CoT-MAE is more significant, with notable gains observed on 13 datasets.

5 Analyses

In this section, we examine the efficiency advantage and analyze the impact of different settings on query-as-context pre-training.

5.1 Impact of Generated Query Number

During pre-training, using multiple candidate queries leads to better diversity as each passage is paired with a different candidate query in each epoch.

[2]The current state-of-the-art models on the BEIR benchmark reach higher scores as they are pre-trained on the WIKI dataset. Due to the high cost of pre-training, we directly evaluate the models pre-trained on the MS-MARCO dataset and leave the exploration on the WIKI dataset in future work.

Therefore, we explore the effect of the number of generated queries. As shown in Table 3, for coCondenser, increasing the number of queries from 1 to 5 slightly improves performance on the MS-MARCO dataset and leads to a good improvement on the TREC DL 19 and 20 test sets. For CoT-MAE, using 5 queries lead to an increase on the MS-MARCO dataset and TREC DL20 test set, while a slight performance decrease in the TREC DL 19 test set. However, further increasing the number of candidate queries will generally bring about a decline in performance. A proper number of queries retains their correlation to the passages, thus yielding higher performance in query-as-context pre-training.

5.2 Impact of Mixed Context

[ILLEGIBLE]

6 Related Works

Dense Retrieval Different techniques have been developed to improve dense retrieval, both in fine-tuning and pre-training stages. In fine-tuning stage, attempts includes mining hard negatives (Xiong et al., 2020; Zhan et al., 2021), late interaction (Khattab and Zaharia, 2020), query clustering (Hofst"atter et al., 2021), reranker distillation (Lin et al., 2021b; Santhanam et al., 2021), data augmentation (Qu et al., 2020) and jointly learning (Ren et al., 2021b; Zhang et al., [ILLEGIBLE]). In pre-training stage, previous work mainly focuses on improving representation learning with self-supervised or context-supervised objectives, such as Condenser (Gao and Callan, 2021a), coCondenser (Gao and Callan, 2021b), CoT-MAE (Wu et al., 2022), RetroMAE (Liu and Shao, 2022), SimLM (Wang et al., 2022) and LED (Zhang et al., 2022a).

Sparse Retrieval Sparse retrieval methods are based on term matching and inverted indexes. Traditional sparse retrieval models include BM25 and its variants. Recent PLM-based sparse retrieval methods enhance sparse representations with neural models, such as docT5query (Nogueira and Lin, 2019), DeepCT (Dai and Callan, 2019), GAR (Mao et al., 2020), COIL (Gao et al., 2021), uniCOIL (Lin and Ma, [ILLEGIBLE]), SPLADE (Formal et al., 2021a,b), and related approaches. Query prediction and expansion techniques are widely used in sparse retrieval to enrich passage representations, which inspires our use of generated queries as context in dense retrieval pre-training.

7 Conclusions

In this paper, we propose query-as-context pre-training, a simple yet effective pre-training technique for dense passage retrieval. Instead of constructing context pairs with two randomly selected passages from the same document, the proposed method generates a query from a passage and uses the passage-query pair as context-supervised signal. This design alleviates the problem of weakly correlated passage pairs in conventional context-supervised pre-training.

We apply query-as-context pre-training to both contrastive and generative context-supervised pre-training frameworks, including coCondenser and CoT-MAE. Experimental results on MS-MARCO, TREC Deep Learning benchmarks, and BEIR out-of-domain datasets show that query-as-context pre-training consistently brings considerable gains while also improving training efficiency.

[ILLEGIBLE]

8 Limitations

[ILLEGIBLE]

References

[ILLEGIBLE]

A A Statistically Analysis of Weakly Correlated Passages

We randomly select 200 documents from the MS-MARCO dataset and randomly select a passage from each document. Then we construct the contextual pairs in two ways:

1. Random passage-passage pair: Referring to coCondenser (Gao and Callan, 2022), we randomly select another passage within the same document as the context for the passage.

2. Generated passage-query pair: Referring to the out-of-shelve docT5query (Nogueira and Lin, 2019), we use query prediction technology to generate a query as the context for the passage.

We asked the annotators to label whether the random contexts or generated queries are strongly related to the corresponding passages. We manually

annotate the 200 passage-passage pairs and passage-query pairs as high-correlation or low-correlation respectively. To eliminate preference bias, we divide 6 annotators into two groups. One group annotates 100 passage-passage pairs and 100 passage-query pairs, while the other annotates the remaining pairs. The correlation of each pair is voted by the annotation results of three annotators. The statistical results are shown in Table 5.

Only 35.5

Correlation rate [ILLEGIBLE]

Pairs [ILLEGIBLE]

Table 5: Correlation statistics of human annotation results of different contextual pairs, each with 200 pairs.