ConTextual Masked Auto-Encoder for Dense Passage Retrieval

Xing Wu[1,2,3 *], **Guangyuan Ma**[1,2 *], **Meng Lin**[1,2], **Zijia Lin**[3], **Zhongyuan Wang**[3], **Songlin Hu**[1,2†]

[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Kuaishou Technology
{wuxing, maguangyuan, linmeng, husonglin}@iie.ac.cn, linzijia07@tsinghua.org.cn, wangzhongyuan@kuaishou.com

Abstract

Dense passage retrieval aims to retrieve the relevant passages of a query from a large corpus based on dense representations (i.e., vectors) of the query and the passages. Recent studies have explored improving pre-trained language models to boost dense retrieval performance. This paper proposes CoT-MAE (ConTextual Masked Auto-Encoder), a simple yet effective generative pre-training method for dense passage retrieval. CoT-MAE employs an asymmetric encoder-decoder architecture that learns to compress the sentence semantics into a dense vector through self-supervised and context-supervised masked auto-encoding. Precisely, self-supervised masked auto-encoding learns to model the semantics of the tokens inside a text span, and context-supervised masked auto-encoding learns to model the semantical correlation between the text spans. We conduct experiments on large-scale passage retrieval benchmarks and show considerable improvements over strong baselines, demonstrating the high efficiency of CoT-MAE. Our code is available at https://github.com/caskcsg/ir/tree/main/cotmae.

Introduction

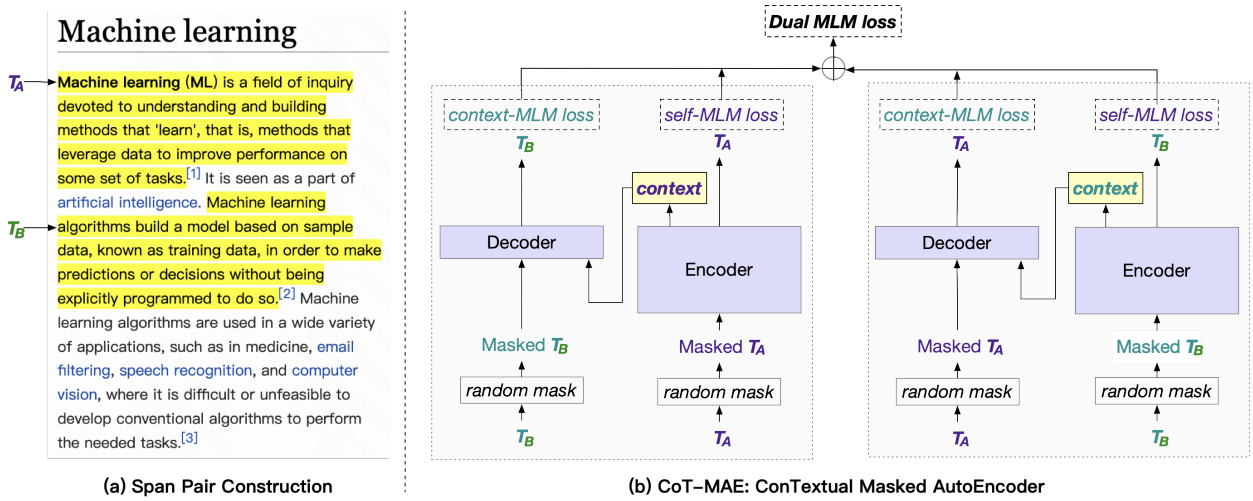
Passage retrieval aims to retrieve the relevant passages of a query from a large corpus, which benefits many downstream applications, such as web search (Fan et al. 2021; Guo et al. 2022; Lin, Nogueira, and Yates 2021), question answering (Karpukhin et al. 2020; Lee et al. 2020; Zhu et al. 2021) and dialogue systems (Gao et al. 2022a; Yu et al. 2021).

For a long time, sparse retrieval represented by BM25 (Robertson, Zaragoza et al. 2009) was the dominant retrieval method. Recently, dense retrieval has received increasing attention with the development of pre-trained language models (PLM) (Devlin et al. 2018; Liu et al. 2019). Dense retrieval models are generally based on pre-trained language models with a siamese or dual-encoder architecture to encode queries and documents into low-dimensional vector space for effective search (Hofstätter et al. 2021; Humeau et al. 2019; Xiong et al. 2020; Zhan et al. 2021, 2020). The relevances between queries and documents are calculated with cosine similarity or dot-product function in the vector space. Therefore, high-quality text representation based on PLM is crucial for dense passage retrieval.

DPR(Karpukhin et al. 2020) successfully shows that dense retrieval models can outperform BM25 methods. Since then, some works have emerged to boost dense retrieval performance by improving the pre-training process tailored for dense retrieval. (Lu et al. 2021; Gao and Callan 2021a; Liu and Shao 2022) encourage the encoder to improve the text representation modeling ability through auxiliary self-supervised reconstruction tasks. Auxiliary tasks usually utilize a weak decoder to reconstruct the masked text with the assistance of the text's vector from the encoder, which forces the encoder to provide better text representations. Although these works have been shown to be very effective and achieved some improvements in dense retrieval, *they mainly focus on single-text internal modeling without considering contextual information.* (Chang et al. 2020; Gao and Callan 2021b; Ma et al. 2022) proposes multi-source and multi-granularity contrastive span prediction tasks to model the semantic similarity between relevant text spans during pre-training. These discriminative tasks introduced in the pre-training improve dense retrieval and the trained retrievers achieve state-of-the-art performance. Meanwhile, in multimodal representation learning (Li et al. 2021, 2022), generative pre-training tasks, such as image-based text decoding, have been proven effective, *leaving us with a question of how well span-level context-grounded generative auxiliary task will boost the dense retrieval.*

To improve text representation modeling for dense passage retrieval, this paper proposes a novel pre-training method by further introducing semantic correlations between text spans in a generative manner. When modeling, we jointly consider the semantics of the tokens inside a text span and the semantical correlation between the text spans. As shown in Figure 1-(a), we select two neighboring text spans $\mathbf{T_A}$ and $\mathbf{T_B}$ from a document with a sampling strategy to form a span pair. Then, as shown in Figure 1-(b), we use the masked auto-encoding to model the whole process. First, we consider self-supervised masked auto-encoding, which reconstructs a masked text span only considering the unmasked tokens in the span on the encoder side. As shown in the purple flow in Figure 1-(b), we apply masking operations to $\mathbf{T_A}$ and $\mathbf{T_B}$ respectively, feeding the masked text to the encoder to reconstruct the original

*These authors contributed equally.
†Corresponding author.



(a) Span Pair Construction

(b) CoT–MAE: ConTextual Masked AutoEncoder

Figure 1: CoT-MAE. (a) The process of span pair construction. We select two neighboring text spans $\mathbf{T_A}$ and $\mathbf{T_B}$ from a document with a sampling strategy to form a span pair. The two spans in a pair are each other's context. (b) The model design for CoT-MAE. We use an asymmetric encoder-decoder structure, with a deep encoder having enough parameters to learn good text representations modeling ability and a shallow decoder to assist the encoder in achieving this goal.

text with (*self*-)masked language modeling (MLM) objective. Next, we consider context-supervised auto-encoding, which reconstructs a masked text span jointly considering the unmasked tokens in the span and its neighboring span embedding, i.e., its context, on the decoder side. As shown in the green flow in Figure 1-(b), on the decoder side, let's take $\mathbf{T_B}$ as an example. We apply a mask operation to $\mathbf{T_B}$ and provide the masked $\mathbf{T_B}$ with its context, i.e., the encoded embedding of $\mathbf{T_A}$, to the decoder, jointly reconstructing the $\mathbf{T_B}$ with *context*-MLM objective. We call the whole process ConTextual Masked AutoEncoding (CoT-MAE) and uniformly optimize with the MLM loss. Note that we only use the decoder in pre-training. During the dense retrieval fine-tuning, the decoder is no longer needed.

There are several benefits to our design. First, CoT-MAE uses self-supervised and context-supervised masked auto-encoding tasks. The two tasks are linked by context representation and jointly optimized to learn better text representation modeling capabilities. Second, the asymmetric encoder-decoder structure is very flexible. We use a deep encoder with adequate parameters to learn a good text representation modeling ability. We use a shallow decoder without enough parameters to recover the masked tokens well, which results in a strong reliance on the context from the encoder and forces the encoder to learn to provide better text representation. Finally, we can use asymmetric masking operations, that is, using different masking rates on the encoder and decoder sides. Inspired by (He et al. 2022; Wettig et al. 2022) that a large mask rate may benefit pre-training, we adopt up to 30% and 45% mask rate in CoT-MAE's encoder and decoder, respectively. A larger mask rate on the decoder side further increases the difficulty of context-supervised masked auto-encoding and forces the encoder side to acquire more powerful text encoding capabilities.

To verify the effectiveness of our proposed CoT-MAE, we conduct experiments on large-scale web search and open-domain QA benchmarks: MS-MARCO Passage Ranking (Nguyen et al. 2016), TREC Deep Learning (DL) Track 2020 (Craswell et al. 2020), and Natural Questions (NQ) (Kwiatkowski et al. 2019). Experimental results show that CoT-MAE achieves considerable gains over competing baseline retrievers. In addition, we perform extensive ablation experiments to illustrate the soundness of the CoT-MAE design.

Our contributions can be summarized as follows:

- We propose a novel generative pre-training method CoT-MAE tailored for dense retrieval.
- We design a data construction method and an asymmetric encoder-decoder structure for efficient pre-training.
- Experiments show that CoT-MAE achieves considerable gains over competing retrievers on benchmarks.

Related Works

DPR (Karpukhin et al. 2020) outperforms BM25 methods with dense retrieval models. Since then, many works have emerged to boost dense retrieval performance, including techniques to improve pre-training and fine-tuning.

Pre-training tasks tailored for dense retrieval One category (Lu et al. 2021; Gao and Callan 2021a; Liu and Shao 2022) forces the encoder to provide better text representations with auxiliary self-supervised auto-encoding tasks. For example, (Lu et al. 2021; Gao and Callan 2021a) proposes to perform the auto-encoding using a weak decoder, with restricted capacity and attention flexibility to push the encoder to provide better text representation. (Liu and Shao 2022) proposes to apply asymmetric masking ratios to the encoder and the weak decoder. A sentence embedding from the encoder is combined with its aggressively masked version to reconstruct the original sentence by the decoder. Similar to

these works, our method adopts an asymmetric encoder and weak decoder architecture. Differently, we propose context-supervised auto-encoding, in which a masked version of a given text and its neighboring text's embeddings, i.e., its context, from the encoder are jointly fed into the decoder to reconstruct the given text.

The other category's works (Chang et al. 2020; Gao and Callan 2021b; Ma et al. 2022) propose multi-source and multi-granularity contrastive span prediction tasks to resemble passage retrieval in pre-training. (Chang et al. 2020) proposes three tasks: inverse cloze task (ICT), body first selection (BFS), and wiki link prediction (WLP). The three tasks exploit the document structure of Wikipedia pages to automatically generate contrastive pairs and pull closer relevant pairs while pushing away irrelevant ones. Similarly, (Gao and Callan 2021b) introduces a corpus-level contrastive span prediction loss to the pre-training process, with the hypothesis that spans from the same document are closer than those from different documents. (Ma et al. 2022) generalizes the contrastive span prediction task to several levels of granularity, i.e., word-level, phrase-level, sentence-level and paragraph-level. Differently, we introduce a generative non-contrastive contextual masked auto-encoding task via the decoder-side MLM reconstruction assisted by contextual embedding, which is more challenging but proved to be more effective.

Fine-tuning Many attempts have explored to improve fine-tuning peformance, such as mining hard negatives (Xiong et al. 2020; Zhan et al. 2021), late interaction (Khattab and Zaharia 2020), distill knowledge from a strong teacher (Lin, Yang, and Lin 2021; Santhanam et al. 2021), query clustering (Hofstätter et al. 2021), data augmentation (Qu et al. 2020) and jointly optimize retriever and re-ranker (Ren et al. 2021b; Zhang et al. 2022, 2021) .

For example, (Xiong et al. 2020) proposes to construct hard negatives by searching the corpus with a periodic updated Approximate Nearest Neighbor (ANN) index, which has been proved very effective and adopted by the following methods. (Zhan et al. 2021) further utilizes fine-tuned dense retriever to improve the quality of mined hard negatives. (Khattab and Zaharia 2020) proposed a late interaction that applies a MaxSim operation on the last hidden states of the encoder to model the fine-grained similarity between queries and documents. (Lin, Yang, and Lin 2021) distills from ColBERT's MaxSim operator into a retriever, meanwhile (Santhanam et al. 2021) proposes to distill from a stronger re-ranker to the ColBERT. (Hofstätter et al. 2021) introduce an efficient topic-aware query and balanced margin sampling technique to improve the fine-tuning efficiency. (Qu et al. 2020) combines three effective strategies to achieve good performance, i.e., cross-batch negatives, denoised hard negatives, and data augmentation. (Ren et al. 2021b) introduce the dynamic listwise distillation by designing a unified listwise training approach to improve both the retriever and the re-ranker adaptively. (Zhang et al. 2022) designs Hybrid List Aware Transformer Reranking (HLATR) as a subsequent reranking module to incorporate retrieval and reranking stage features. (Zhang et al. 2021) present adversarial retriever-ranker, which consists of a dual-encoder retriever and a cross-encoder ranker to be jointly optimized according to a minimax adversarial objective. As we focus on the improvement brought by pre-training, following (Gao and Callan 2021a,b; Ma et al. 2022), we reuse the open source fine-tuning pipeline Tevatron (Gao et al. 2022b) to evaluate the effectiveness of our pre-training method.

Approach

In this section, we first introduce the masked auto-encoder structure as preliminary knowledge. Then we introduce the data construction and model structure of CoT-MAE.

Preliminary: Masked Auto-Encoding

Textual Masked Auto-Encoding, i.e., BERT's MLM task, is trained on unlabeled data without requiring additional manual labeling. Formally, given a text $\mathbf{T}$ with n consequent tokens, we prepend a special token [CLS] to the beginning of the text as

$$\mathbf{T} = \{t_0, t_1, ..., t_n\} \tag{1}$$

, in which the t_0 denotes the [CLS] token. We randomly select a certain percentage, i.e., mask rate, of tokens and replace them with special token [MASK]. Inspired by (He et al. 2022; Wettig et al. 2022) that a large mask rate may be beneficial for pre-training, we employ a mask rate greater than the BERT's 15% setting. We denote the tokens replaced by [MASK] as $m(\mathbf{T})$ and the rest tokens as $\mathbf{T}\backslash m(\mathbf{T})$. $\mathbf{T}\backslash m(\mathbf{T})$ is then passed through the encoder to recover $m(\mathbf{T})$ with the masked language model(MLM) loss. We formulate this process as:

$$\mathcal{L}_{mlm} = -\sum_{t \in m(\mathbf{T})} \log p(t \mid \mathbf{T}\backslash m(\mathbf{T})) \tag{2}$$

For the l-th transformer layer in the encoder or decoder, its outputs are the hidden states of the layer

$$\mathbf{h}^l = \{h_0^l, h_1^l, ..., h_n^l\} \tag{3}$$

. Usually, the hidden states of the [CLS] position in the last layer of the encoder, i.e., h_0^{last}, will be used as the embedding representation of $\mathbf{T}$.

CoT-MAE: ConTextual Masked Auto-Encoder

CoT-MAE jointly learns to model the semantics of the tokens inside a text span and the semantical correlation between the text spans.

We first describe how to build training data from unlabeled documents, as shown in Figure 1-(a). Given a document, we use tools like NLTK to split it into text spans that do not exceed a maximum length. Then we select two neighboring text spans $\mathbf{T_A}$ and $\mathbf{T_B}$ from a document with a sampling strategy to form a span pair. The two spans in a pair are each other's neighbor or context.

Sampling Strategies We use three sampling strategies, termed Near, Olap and Rand. The Near strategy samples two adjacent spans without overlapping to form a pair. The Olap

strategy samples two adjacent spans with partially overlapping segments to form a pair. The Rand strategy randomly samples two non-overlapping spans to form a pair.

Then, we introduce the model design for CoT-MAE, as shown in Figure 1-(b). We use an asymmetric encoder-decoder structure, with a strong deep encoder and a weak shallow decoder. The deep encoder has enough parameters to learn good text representation modeling ability, and the shallow decoder is set to assist the encoder in achieving this goal. As the CoT-MAE adopts a dual modeling process for a span pair, we will only introduce the left half of Figure 1-(b) in detail. That is, $\mathbf{T_A}$ is on the encoder side, and $\mathbf{T_B}$ is on the decoder side. We adopt asymmetric masking operations, using different mask rates on the encoder and decoder sides. We apply random mask operation to $\mathbf{T_A}$ on the encoder side. We denote the tokens replaced by [MASK] as $m_{enc}(\mathbf{T_A})$ and the rest tokens as $\mathbf{T_A} \backslash m_{enc}(\mathbf{T_A})$. Similarly, we apply another random mask operation to $\mathbf{T_B}$ on the decoder side. We denote the tokens replaced by [MASK] as $m_{dec}(\mathbf{T_B})$ and the rest tokens as $\mathbf{T_B} \backslash m_{dec}(\mathbf{T_B})$.

Self-supervised Pre-training On the encoder side, we reconstruct a text span only considering the unmasked tokens in the span. The unmasked tokens $\mathbf{T_A} \backslash m_{enc}(\mathbf{T_A})$ is passed through the encoder to recover $m_{enc}(\mathbf{T_A})$ with the self-supervised masked language model(self-MLM) loss:

$$\mathcal{L}^{A}_{smlm} = -\sum_{t \in m(\mathbf{T_A})} \log p(t \mid \mathbf{T_A} \backslash m(\mathbf{T_A})) \quad (4)$$

. The subscript "smlm" denotes the process is self-supervised, superscript A denotes the self-supervised pre-training is applied on T_A.

Context-supervised Pre-training On the decoder side, we reconstruct the other text span in the pair considering its unmasked tokens and its neighboring span embedding, i.e., its context embedding. Specifically, for $\mathbf{T_B}$, its context embedding is the [CLS] hidden state of $\mathbf{T_A}$ from the encoder's last layer, also denoted as h_0^{last}. We jointly feed $\mathbf{T_B} \backslash m_{dec}(\mathbf{T_B})$ and h_0^{last} into the decoder to recover $m(\mathbf{T_B})$ using the context-supervised masked language model loss:

$$\mathcal{L}^{AB}_{cmlm} = -\sum_{t \in m(\mathbf{T_B})} \log p(t \mid [h_0^{last}, \mathbf{T_B} \backslash m(\mathbf{T_B})]) \quad (5)$$

. The subscript "cmlm" denotes the process is context-supervised, the superscript "AB" denotes that $\mathbf{T_B}$ uses $\mathbf{T_A}$ as context, and "[]" denotes concatenation operation. Then, we add the losses from both the encoder and the decoder to get a summed loss:

$$\mathcal{L}^{AB} = \mathcal{L}^{A}_{smlm} + \mathcal{L}^{AB}_{cmlm} \quad (6)$$

At the same time, there is also a dual case, $\mathbf{T_B}$ is on the encoder side and $\mathbf{T_A}$ is on the encoder side. The summed loss is:

$$\mathcal{L}^{BA} = \mathcal{L}^{B}_{smlm} + \mathcal{L}^{BA}_{cmlm} \quad (7)$$

Finally, the total loss of our proposed CoT-MAE is:

$$\mathcal{L} = \mathcal{L}^{AB} + \mathcal{L}^{BA} \quad (8)$$

Fine-tuning on Dense Passage Retrieval

At the end of CoT-MAE pre-training, we only keep the encoder and discard the decoder. The encoder weights are used to initialize a query encoder f_q and a passage encoder f_p for dense retrieval. The query(or passage) encoder use the last layer's [CLS] embedding as the query(or passage) representation, denoted as $f_q(q)$(or $f_p(p)$). The similarity of a query-passage pair $< q, p >$ is defined as an inner product:

$$s(q, p) = f_q(q) \cdot f_p(p)$$

Query and passage encoders are fine-tuned on the retrieval task's training corpus with a contrastive loss:

$$\mathcal{L} = -\log \frac{\exp\left(s\left(q, p^{+}\right)\right)}{\exp\left(s\left(q, p^{+}\right)\right) + \sum_{l} \exp\left(s\left(q, p_{l}^{-}\right)\right)}$$

, where p^+ denotes a positive passage and $\{p_l^-\}$ denotes a set of negative passages. In practice, we reuse a widely adopted evaluation pipeline, i.e., Tevatron (Gao et al. 2022b). The pipeline trains a first-stage-retriever using BM25 negatives, then trains a second-stage-retriever using BM25 negatives and hard negatives mined by the first-stage-retriever. The second-stage-retriever is used as the final retriever for evaluation.

Experiments

In this section, we first introduce the details of pre-training and fine-tuning. We then introduce the experimental results.

Pre-training

We initialize CoT-MAE's encoder from the pre-trained 12-layer BERT-*base* and decoder from scratch. Following (Gao and Callan 2021b), the pre-training dataset is constructed from the MS-MARCO passages corpus[1] with 3.2M documents. We use NLTK to split each document into sentences and group consecutive sentences into spans without exceeding the maximum span length equals 128. We use a uniform mixture of the three sampling strategies introduced, and dynamically select two spans in different epochs to form a span pair during the pre-training process. We pre-train up to 1200k steps using AdamW optimizer, with a learning rate of 1e-4, and a linear schedule with warmup ratio 0.1. We train for 4 days with a global batch size of 1024 on 8 Tesla A100 GPUs. Due to the high compute budget in pre-training, we do not tune these hyperparameters, but leave that to future work. After pre-training, we discard the decoder, only leaving the encoder for fine-tuning.

Fine-tuning

We fine-tune the pre-trained CoT-MAE on MS-MARCO passage ranking (Nguyen et al. 2016), Natural Question (Kwiatkowski et al. 2019) and TREC Deep Learning (DL) Track 2020 (Craswell et al. 2020) tasks for evaluation. Following coCondenser(Gao and Callan 2021b), we use the

[1]https://msmarco.blob.core.windows.net/msmarcoranking/msmarco-docs.tsv.gz

Model	MS-MARCO			NQ			TREC DL 20
	MRR@10	R@50	R@1k	R@5	R@20	R@100	nDCG@10
Sparse retrieval							
BM25	18.7	59.2	85.7	-	59.1	73.7	47.9†
docT5query (Nogueira and Lin 2019)	21.5	64.4	89.1	-	-	-	-
DeepCT (Dai and Callan 2019)	24.3	69.0	91.0	-	-	-	-
GAR (Mao et al. 2020)	-	-	-	60.9	74.4	85.3	-
Dense retrieval							
ANCE (Xiong et al. 2020)	33.0	-	95.9	-	81.9	87.5	-
SEED (Lu et al. 2021)	33.9	-	96.1	-	83.1	88.7	-
TAS-B (Hofstätter et al. 2021)	34.0	-	97.5	-	-	-	69.3
COIL (Gao, Dai, and Callan 2021)	35.5	-	96.3	-	-	-	-
ColBERT (Khattab and Zaharia 2020)	36.0	82.9	96.8	-	-	-	-
NPRINC (Lu et al. 2020)	31.1	-	97.7	73.3	82.8	88.4	-
Condenser (Gao and Callan 2021a)	36.6	-	97.4	-	83.2	88.4	66.5†
RocketQA (Qu et al. 2020)	37.0	85.5	97.9	74.0	82.7	88.5	-
PAIR (Ren et al. 2021a)	37.9	86.4	98.2	74.9	83.5	89.1	-
coCondenser (Gao and Callan 2021b)	38.2	86.5	98.4	**75.8**	**84.3**	89.0	68.0†
COSTA (Ma et al. 2022)	36.6	84.1	97.3	-	-	-	67.8†
RetroMAE (Liu and Shao 2022)⋆	39.3	-	98.5	-	-	-	-
CoT-MAE	**39.4**	**87.0**	**98.7**	75.5	**84.3**	**89.3**	**70.4**

Table 1: Main results on the MS-MARCO passage ranking and Natural Questions (NQ) datasets. The best score on a given dataset is marked in bold, and the second best is underlined. † denotes our reproduction with public checkpoints. ⋆ is a contemporaneous work.

MS-MARCO corpus released in (Qu et al. 2020), following RocketQA(Qu et al. 2020), we use the NQ version created by DPR(Karpukhin et al. 2020). We reuse a widely adopt evaluation pipeline, i.e., Tevatron (Gao et al. 2022b), with a common fixed 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 or multi-vector representation, though they can lead to further improvements. Following (Gao and Callan 2021b; Hofstätter et al. 2021), for evaluation metrics, we use MRR@10, Recall@50, and Recall@1000 for MS-MARCO, Recall@5, Recall@20, Recall@100 for the NQ and nDCG@10 for TREC DL.

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 (Hofstätter et al. 2021), RetroMAE (Liu and Shao 2022), and so on.

Main Results

We present the main results in Table 1, which shows that CoT-MAE achieves considerable gains compared to competing baselines on three datasets. On the MS-MARCO passage ranking dataset, CoT-MAE exceeds the previous state-of-the-art pre-training method coCondenser 1.2% on MRR@10, which is a clear lead. This suggests that the generative pre-training method is more effective than the contrastive learning method in leveraging context to enhance semantic representation modeling capacities. Compared with methods like RocketQA that improve the fine-tuning stage or methods like ColBERT that adopt multi-vector, CoT-MAE also performs better. On the NQ dataset, CoT-MAE also achieves comparable performance to coCondenser, outperforming the remaining methods. While coCondenser performs a little better on R@5, CoT-MAE outperforms it on R@100. On the TREC DL dataset, CoT-MAE clearly outperforms the evaluated baselines and reaches 70 on nDCG@10. In general, comparing CoT-MAE with the previous effective methods on the most commonly used benchmark datasets shows that the CoT-MAE pre-training process can effectively improve dense retrieval. The improvement derives from two aspects. On the one hand, the pre-training method considers both the semantics of the tokens inside a text span and the semantical correlation between neighboring text spans. On the other hand, the mixed data construction strategies and the asymmetric encoder-decoder structure with asymmetric masking strategies together contribute to efficient pre-training.

Analysis

We first compare CoT-MAE with the state-of-the-art distilled retrievers. Then we seek to understand the impact of different settings on CoT-MAE performance. Unless otherwise stated, our analyses are based on the MS-MARCO passage ranking task and pre-trained for 800k (not the 1200k in the main experiment) steps due to the high compute budget.

Model	Distilled	MRR@10	R@50	R@1k
RocketQAv2	✓	38.8	86.2	98.1
ColBERTv2	✓	39.7	86.8	98.4
AR2	✓	39.5	87.8	98.6
CoT-MAE	✗	39.2	87.2	**98.7**
CoT-MAE	✓	**40.4**	**88.5**	**98.7**

Table 2: Comparison with distilled retrievers.

Enc	Dec	MRR@10	Enc	Dec	MRR@10
0%	15%	37.8	45%	45%	39.0
0%	30%	38.4	45%	60%	38.8
15%	15%	38.7	30%	15%	38.8
15%	30%	38.8	30%	30%	38.9
15%	45%	38.9	30%	45%	**39.2**

Table 3: Impact of mask rate. "Enc" denotes encoder, "Dec" denotes decoder.

Comparison with Distilled Retrievers

As CoT-MAE focuses on improving pre-training to boost text representation in retrieval, comparison with retrievers distilled from re-rankers is optional and left here. The typical passage ranking process involves a retrieval then re-ranking pipeline. Re-ranker is a cross-encoder that models the full interaction between the query and the passage, which is strong but too computationally intensive to be applied in the retrieval. (Ren et al. 2021b; Santhanam et al. 2021; Zhang et al. 2021) employ a re-ranker as a teacher to distill the retriever, trying to transfer the ability from the re-ranker to the retriever.

To further verify the effectiveness of CoT-MAE pre-training, we first compare the fine-tuned CoT-MAE retriever (without distillation) with the state-of-the-art distilled retrievers. Then we train a CoT-MAE reranker that reaches 43.3 on MRR@10 and use the re-ranker to distill the CoT-MAE retriever.

As shown in Table 2, it is impressive that the fine-tuned CoT-MAE retriever (without distillation) achieves performance close to the distilled retrievers, demonstrating the effectiveness of CoT-MAE pre-training. The fine-tuned CoT-MAE retriever slightly outperforms all distilled retrievers on R@1k, is only inferior to AR2 on R@50, and surpasses RocketQAv2 on MRR@10. After applying distillation, the CoT-MAE retriever clearly exceeds all previously state-of-the-art distilled retrievers on MRR@10, R@50 and R@1k, showing that the CoT-MAE re-ranker can further improve the CoT-MAE retriever.

Impact of Mask Rate

(Wettig et al. 2022) finds that a much larger mask ratio can outperform the 15% baseline from BERT (Devlin et al. 2018). Therefore, we explore the effect of different mask rates for encoder and decoder. As shown in Table 3, in our experiments, when the encoder mask rate equals 30%, and the decoder mask rate equals 45%, CoT-MAE achieves the best performance. When the encoder mask rate does not exceed 30%, the performance of CoT-MAE improves as the decoder mask rate increases. When the encoder mask rate is as high as 45%, the performance of CoT-MAE decreases slightly. We believe this is due to the insufficient context from the encoder when its mask rate is too large. In general, CoT-MAE is quite robust to a wide range of mask rates, and an appropriate large mask rate can achieve relatively better performance, which is similar to (Wettig et al. 2022)'s finding in BERT pre-training.

Near	Olap	Rand	MRR@10
✓			38.7
	✓		38.7
		✓	38.8
✓	✓		39.1
✓		✓	38.8
	✓	✓	39.1
✓	✓	✓	**39.2**

Table 4: Impact of sampling strategies. ✓indicates that this sample strategy is selected.

Layers	1	2	3	4	6
MRR@10	38.6	**39.2**	38.8	38.9	39.0

Table 5: Impact of decoder layers.

Impact of Sampling Strategies

In CoT-MAE's data construction process, we use three span sampling strategies, **Near**, **Olap** and **Rand**. We further explore the impact of different sampling strategies. We experiment with combinations of different strategies, covering using only a single strategy, mixing two strategies, and mixing three strategies. As shown in Table 4, the best performance is achieved when mixing three strategies, with a slight drop when mixing two strategies and a larger drop when only a single strategy. The trend shows that the diversity of data composition can benefit pre-training. However, even with only a single strategy, CoT-MAE can also achieve a good result, further illustrating the effectiveness of the CoT-MAE model design.

Impact of Decoder Layer Number

We further explore the impact of different decoder layer numbers on CoT-MAE performance. We experiment on some most common layer settings, as shown in the table 5. With a two-layer decoder, CoT-MAE achieves the best results in our experiment. A weaker or stronger decoder will decrease performance. When there is only one transformer layer in the decoder, the modeling ability of the decoder is too weak to fuse context embedding and unmasked token embeddings fully, resulting in inefficient utilization of information. When the number of layers is large,

Relevant	Model	Rank 1st passage
		Query: what is operating system misconfiguration
✗	BERT$_{base}$	Passage: how to fix system drivers have stopped on my pc errors. windows operating system misconfiguration is the main cause of system drivers have stopped on my pc error codes...
✗	coCondenser	Passage: ne of the issues that users are experiencing on windows 10 is unexpected kernel mode trap error so letas see if we can fix this issue. unexpected kernel mode trap is a blue screen of death error that is caused...
✓	CoT-MAE	Passage: windows operating system misconfiguration is the main cause of systemroot inf error codes. therefore, we strongly suggest using the downloadable systemroot inf repair kit to fix systemroot inf errors rakesh...
		Query: does cream of chicken soup have gluten
✗	BERT$_{base}$	Passage: directions. 1 whisk together the flour or cornstarch with the milk or milk substitute. 2 add the remaining ingredients and heat to a boil while whisking til fully dissolved and combined...
✗	coCondenser	Passage: stephanie1 0. i'm really new to being gluten - free, and i was wondering about campbell's soup. cream of mushroom and cream of chicken soup both contain modified food starch. i know that can be...
✓	CoT-MAE	Passage: campbellas cream of chicken soup is not gluten - free. in fact, if you live in the us, there are no currently no campbellas soups that are gluten - free.

Table 6: Examples of rank 1st passage recalled by different models on the dev set of the MS-MARCO passage ranking dataset.

due to the stronger ability of the decoder, the masked auto-encoding task's dependence on the context embedding decreases, leading to insufficient constraints on encoder training. In general, the performance of CoT-MAE for decoders with different layers is quite robust, and an appropriate deeper decoder can obtain relatively good performance.

Impact of Decoder Weight Initialization

The structure of CoT-MAE's encoder is the same as that of BERT, so the encoder is initialized directly with pre-trained BERT. In comparison, the decoder has only two layers, so different initialization options exist. We explore four different initialization methods: initialized from the first two layers (0,1), the uniform selected layers (5,11), the last two layers (10,11) of pre-trained BERT, and random initialization. We train 800k steps for each and plot their performance curves, as shown in Figure 2. In general, the long pre-training is sufficient for the decoder with limited parameters to converge, so the final results of the decoder are quite good. After being pre-trained for more than 600k, random initialization continues to improve, while other initialization methods tend to fluctuate. It is worth noting that CoT-MAEs with different initializations significantly outperform coCondeser (38.2) when training only 200k, which further demonstrates the efficiency of CoT-MAE pre-training.

Qualitative Analysis

To qualitatively understand the gains from our proposed pre-training method, we show in Table 6 examples for which CoT-MAE can accurately recall relevant rank 1st passages. In the examples, although the rank 1st passage recalled by BERT or coCondenser has token-level overlap with the query, they are not highly relevant in semantics. In contrast,

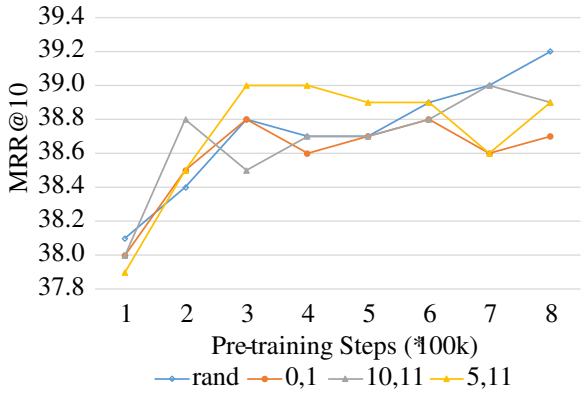


Figure 2: Impact of decoder weight initialization. The y-axis shows the MRR@10 on MS-MARCO passage ranking.

CoT-MAE performs better in semantic understanding due to the joint modeling of token and span context through masked auto-encoding in the pre-training stage. This further demonstrates that the CoT-MAE pre-training method is more effective than the previous pre-training methods.

Conclusions and Future Work

This paper proposes a new generative pre-training method tailored for dense retrieval, considering the semantics of the tokens inside a text span and the semantical correlation between neighboring text spans. Experimental results show that CoT-MAE achieves considerable gains over competing baseline retrievers on benchmarks. In the future, we will further explore the possibility of applying contextual masked auto-encoder in multilingual and multimodal domains.

References

Chang, W.-C.; Yu, F. X.; Chang, Y.-W.; Yang, Y.; and Kumar, S. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*.

Craswell, N.; Mitra, B.; Yilmaz, E.; Campos, D.; and Voorhees, E. M. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

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

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*.

Fan, Y.; Xie, X.; Cai, Y.; Chen, J.; Ma, X.; Li, X.; Zhang, R.; Guo, J.; and Liu, Y. 2021. Pre-training Methods in Information Retrieval. *arXiv preprint arXiv:2111.13853*.

Gao, J.; Xiong, C.; Bennett, P.; and Craswell, N. 2022a. Neural approaches to conversational information retrieval. *arXiv preprint arXiv:2201.05176*.

Gao, L.; and Callan, J. 2021a. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*.

Gao, L.; and Callan, J. 2021b. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Gao, L.; Dai, Z.; and Callan, J. 2021. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186*.

Gao, L.; Ma, X.; Lin, J.; and Callan, J. 2022b. Tevatron: An efficient and flexible toolkit for dense retrieval. *arXiv preprint arXiv:2203.05765*.

Guo, J.; Cai, Y.; Fan, Y.; Sun, F.; Zhang, R.; and Cheng, X. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4): 1–42.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16000–16009.

Hofstätter, S.; Lin, S.-C.; Yang, J.-H.; Lin, J.; and Hanbury, A. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 113–122.

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*.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Khattab, O.; and Zaharia, M. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 39–48.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Lee, J.; Sung, M.; Kang, J.; and Chen, D. 2020. Learning dense representations of phrases at scale. *arXiv preprint arXiv:2012.12624*.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*.

Li, J.; Selvaraju, R.; Gotmare, A.; Joty, S.; Xiong, C.; and Hoi, S. C. H. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34: 9694–9705.

Lin, J.; Nogueira, R.; and Yates, A. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4): 1–325.

Lin, S.-C.; Yang, J.-H.; and Lin, J. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, 163–173.

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

Liu, Z.; and Shao, Y. 2022. RetroMAE: Pre-training Retrieval-oriented Transformers via Masked Auto-Encoder. *arXiv preprint arXiv:2205.12035*.

Lu, J.; Ábrego, G. H.; Ma, J.; Ni, J.; and Yang, Y. 2020. Neural passage retrieval with improved negative contrast. *arXiv preprint arXiv:2010.12523*.

Lu, S.; He, D.; Xiong, C.; Ke, G.; Malik, W.; Dou, Z.; Bennett, P.; Liu, T.-Y.; and Overwijk, A. 2021. Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2780–2791.

Ma, X.; Guo, J.; Zhang, R.; Fan, Y.; and Cheng, X. 2022. Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction. *arXiv preprint arXiv:2204.10641*.

Mao, Y.; He, P.; Liu, X.; Shen, Y.; Gao, J.; Han, J.; and Chen, W. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Nogueira, R.; and Lin, J. 2019. From doc2query to docTTTTTquery. https://www.researchgate.net/publication/

360890853_From_doc2query_to_docTTTTTquery. Accessed: 2019-12-01.

Qu, Y.; Ding, Y.; Liu, J.; Liu, K.; Ren, R.; Zhao, W. X.; Dong, D.; Wu, H.; and Wang, H. 2020. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

Ren, R.; Lv, S.; Qu, Y.; Liu, J.; Zhao, W. X.; She, Q.; Wu, H.; Wang, H.; and Wen, J.-R. 2021a. PAIR: Leveraging passage-centric similarity relation for improving dense passage retrieval. *arXiv preprint arXiv:2108.06027*.

Ren, R.; Qu, Y.; Liu, J.; Zhao, W. X.; She, Q.; Wu, H.; Wang, H.; and Wen, J.-R. 2021b. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *arXiv preprint arXiv:2110.07367*.

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

Santhanam, K.; Khattab, O.; Saad-Falcon, J.; Potts, C.; and Zaharia, M. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

Wettig, A.; Gao, T.; Zhong, Z.; and Chen, D. 2022. Should You Mask 15% in Masked Language Modeling? *arXiv preprint arXiv:2202.08005*.

Xiong, L.; Xiong, C.; Li, Y.; Tang, K.-F.; Liu, J.; Bennett, P.; Ahmed, J.; and Overwijk, A. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Yu, S.; Liu, Z.; Xiong, C.; Feng, T.; and Liu, Z. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 829–838.

Zhan, J.; Mao, J.; Liu, Y.; Guo, J.; Zhang, M.; and Ma, S. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1503–1512.

Zhan, J.; Mao, J.; Liu, Y.; Zhang, M.; and Ma, S. 2020. RepBERT: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.

Zhang, H.; Gong, Y.; Shen, Y.; Lv, J.; Duan, N.; and Chen, W. 2021. Adversarial retriever-ranker for dense text retrieval. *arXiv preprint arXiv:2110.03611*.

Zhang, Y.; Long, D.; Xu, G.; and Xie, P. 2022. HLATR: Enhance Multi-stage Text Retrieval with Hybrid List Aware Transformer Reranking. *arXiv preprint arXiv:2205.10569*.

Zhu, Y.; Pang, L.; Lan, Y.; Shen, H.; and Cheng, X. 2021. Adaptive information seeking for open-domain question answering. *arXiv preprint arXiv:2109.06747*.