RetroMAE-2: Duplex Masked Auto-Encoder For Pre-Training Retrieval-Oriented Language Models

Zheng Liu[1†*], **Shitao Xiao**[2†], **Yingxia Shao**[2*], **Zhao Cao**[1]

1: Huawei Technologies Ltd. Co. 2: Beijing University of Posts and Telecommunications

zhengliu1026@gmail.com, {stxiao,shaoyx}@bupt.edu.cn,

caozhao1@huawei.com

Abstract

To better support information retrieval tasks such as web search and open-domain question answering, growing effort is made to develop retrieval-oriented language models, e.g., RetroMAE (Xiao et al., 2022b) and many others (Gao and Callan, 2021; Wang et al., 2021a). Most of the existing works focus on improving the semantic representation capability for the contextualized embedding of the [CLS] token. However, recent study shows that the ordinary tokens besides [CLS] may provide extra information, which help to produce a better representation effect (Lin et al., 2022). As such, it's necessary to extend the current methods where all contextualized embeddings can be jointly pre-trained for the retrieval tasks.

In this work, we propose a novel pre-training method called Duplex Masked Auto-Encoder, *a.k.a.* DupMAE. It is designed to improve the quality of semantic representation where all contextualized embeddings of the pre-trained model can be leveraged. It takes advantage of two complementary auto-encoding tasks: one reconstructs the input sentence with the [CLS] embedding; the other one predicts the bag-of-words feature of the input sentence with the ordinary tokens' embeddings. The two tasks are jointly conducted to train a unified encoder, where the whole contextualized embeddings are aggregated in a compact way to produce the final semantic representation. DupMAE is simple but empirically competitive: it substantially improves the pre-trained model's representation capability and transferability, where superior retrieval performances can be achieved on popular benchmarks, like MS MARCO and BEIR. Our code is released at: https://github.com/staoxiao/RetroMAE.

1 Introduction

Neural retrieval is important to many real-world scenarios, such as web search, question answering, and conversational system (Huang et al., 2013; Karpukhin et al., 2020; Komeili et al., 2021; Izacard et al., 2022; Zhu et al., 2021; Dong et al., 2022). In recent years, pre-trained language models, e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2019), are widely adopted as the retrievers' backbone networks. The generic pre-trained language models are not directly applicable to retrieval tasks. Thus, it calls for complex fine-tuning strategies, such as sophisticated negative sampling (Xiong et al., 2020; Qu et al., 2020), knowledge distillation (Hofstätter et al., 2021; Lu et al., 2022), and the joint optimization of retriever and ranker (Ren et al., 2021; Zhang et al., 2021). To reduce this effort and bring in better retrieval quality, there are growing interests in developing retrieval-oriented language models. One common practice is to leverage self-contrastive learning (Chang et al., 2020; Guu et al., 2020), where the language models are learned to discriminate heuristically acquired positive and negative samples in the embedding space. Later on, auto-encoding is found to be more effective (Wang et al., 2021a; Lu et al., 2021), where the language models are learned to reconstruct the input based on the generated embeddings. Recent works (Xiao et al., 2022b; Wang et al., 2022) further extend the auto-encoding methods by introducing sophisticated encoding and decoding mechanisms, which brings about remarkable improvements of retrieval quality on a wide variety of benchmarks.

The existing retrieval-oriented pre-trained models mainly rely on the contextualized embedding from the special token, i.e., [CLS], to represent the semantic about input (Gao and Callan, 2021; Lu et al., 2021; Xiao et al., 2022b; Wang et al., 2022). However, recent study finds that other ordinary tokens may provide extra information and help to generate better semantic representations (Lin et al., 2022). Such a statement is consistent with previous research (Luan et al., 2021; Santhanam et al., 2021),

[†]. Equal contribution and designated as co-first authors.

[*]. Co-corresponding authors

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2635–2648
July 9-14, 2023 ©2023 Association for Computational Linguistics

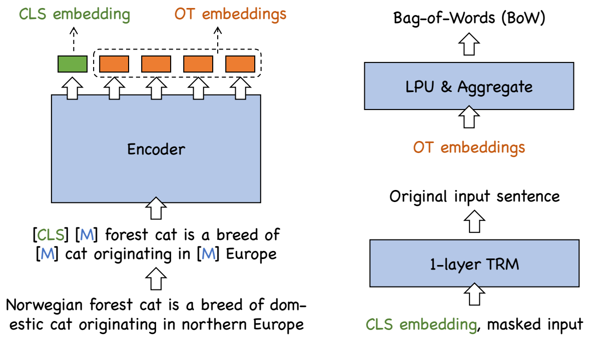

CLS embedding
OT embeddings
Encoder
[CLS] [M] forest cat is a breed of [M] cat originating in [M] Europe
Norwegian forest cat is a breed of domestic cat originating in northern Europe
Bag-of-Words (BoW)
LPU & Aggregate
OT embeddings
Original input sentence
1-layer TRM
CLS embedding, masked input

Figure 1: DupMAE. Encoder: the sentence is masked and encoded as the contextualized embeddings for [CLS] and ordinary tokens. Decoder: the CLS embedding is joined with the masked input, where the original input is recovered by an 1-layer transformer; OT embeddings are mapped into vocabulary space via LPU and aggregated to predict the BoW feature.

as multi-vector or token-granularity representations may give higher discriminative power than those based on one single vector. As a result, it is necessary to extend the previous works, such that the representation capability can be jointly pre-trained for both [CLS] and ordinary tokens.

To this end, we propose a novel auto-encoding framework called Duplex Masked Auto-Encoder, *a.k.a.* **DupMAE** (Figure 1). It employs two differentiated decoders working collaboratively, which aim to 1) improve each embedding's individual capacity, as well as 2) contribute to the quality of the joint representation derived from all embeddings.

- **Workflow**. DupMAE contains an unified encoder, which produces the contextualized embeddings for both [CLS] and ordinary tokens. The generated embeddings are used for two decoding tasks. On one hand, the [CLS] embedding, joined with the masked input, is used to recover the input sentence from an one-layer transformer. On the other hand, the ordinary tokens' embeddings are transformed into the vocabulary space (V), i.e, $|V|$-dim vectors, with a linear projection unit (LPU). The transformation results are aggregated into a $|V|$-dim vector by max-pooling, where the bag-of-words feature about the input is predicted.

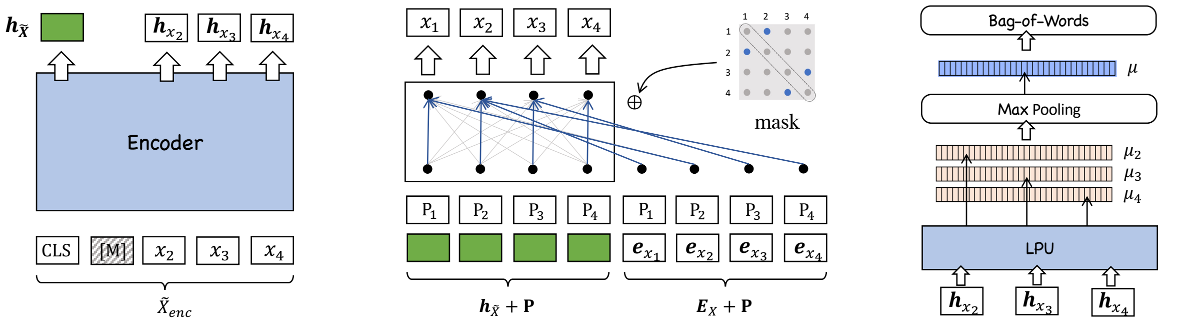
- **Merits**. The above workflow is highlighted by its simplicity: an one-layer transformer to recover the input, and a linear projection unit to preserve the BoW feature. Therefore, the pre-training is `Cost-Effective` given all decoding takes operate at a low cost. More importantly, the pre-training task is made highly `Demanding` on embedding quality: since the decoders are extremely simplified, it forces the encoder to fully extract the input information so that high-fidelity reconstruction can be made. Finally, the differentiated tasks may help the embeddings learn `Complementary` information: the [CLS] embedding focuses more on semantic information; while the OT embeddings, which directly preserve the BoW features, may incorporate more lexical information.

- **Representation**. The contextualized embeddings from [CLS] and ordinary tokens are aggregated in a straightforward way to generate the representation of the input. The [CLS] embedding is reduced to a lower dimension by linear projection. The ordinary tokens' embeddings, after transformed into the vocabulary space and aggregated by max-pooling, are sparsified by selecting the top-N elements. The two results are concatenated as one vector. With a proper configuration of linear projection and sparsification, it may preserve the same memory footprint and cost of inner-product computation as the conventional methods.

Our proposed method is simple but empirically competitive. We perform DupMAE on common pre-training corpus where a BERT-based scale encoder is produced. Our pre-trained model achieves superior performances in various downstream tasks. For supervised evaluations on **MS MARCO**, it reaches a MRR@10 of 42.6 in passage retrieval and a MRR@100 of 45.1 in document retrieval. For zero-shot evaluations on **BEIR**, it achieves an average NDCG@10 of 49.1 on all 18 datasets. It even notably outperforms strong baselines with more sophisticated fine-tuning approaches or much bigger model sizes. Therefore, it validates that the representation capability and transferability of the pre-trained model can be substantially improved thanks to DupMAE.

2 Related Works

Neural retrieval is critical for many real-world applications, such as web search, question answering, advertising and recommender systems (Karpukhin et al., 2020; Zhang et al., 2022; Xiao et al., 2022c, 2021, 2022a). It maps the query and document into embeddings within the same latent space, making their semantic relationship to be measured by the embedding similarity. In recent years, the pre-trained language models have been widely applied to deep semantic retrieval such that discriminative representations can be generated for the queries and documents. Despite the preliminary progress achieved by early pre-trained models, like BERT (Devlin et al., 2019), it is noticed that the more



(A) Encoding

(B) [CLS] decoding

(C) OT decoding

Figure 2: Framework of DupMAE. The unified encoder generates the contextualized embeddings for the [CLS] and ordinary tokens (OT). The [CLS] decoding reconstructs the original sentence leveraging an one-layer transformer; the OT decoding predicts the BoW feature of the input on top of the linear projection unit (LPU) and max-pooling.

advanced models bring little benefit to the retrieval quality, and it's believed that the conventional pre-training algorithms are not compatible with the purpose of deep semantic retrieval (Gao and Callan, 2021; Lu et al., 2021; Wang et al., 2022).

To mitigate the above problem, people become increasingly interested in developing retrieval oriented pre-trained models. For example, it is proposed to leverage self-contrastive learning (SCL) where the language models are pre-trained to discriminate positive samples generated by data augmentation and in-batch negative samples (Chang et al., 2020; Guu et al., 2020; Izacard et al., 2021). The SCL based algorithms are limited by many factors, like the quality of data augmentation and the requirement of huge amounts of negative samples. Later on, the auto-encoding based algorithms receive growing interests: the input sentences are encoded into embeddings, based on which the original sentences are reconstructed (Lu et al., 2021; Wang et al., 2021a). The recently proposed methods, such as SimLM (Wang et al., 2022) and Retro-MAE (Xiao et al., 2022b), extend the previous auto-encoding framework by upgrading the encoding and decoding mechanisms, which substantially improves the quality of deep semantic retrieval.

The existing retrieval-oriented pre-training methods target on improving the semantic representation capacity for the contextualized embedding from the [CLS] token. However, it is noticed that the ordinary tokens may provide additional information besides [CLS], especially when dealing with long and semantic-rich documents (Luan et al., 2021; Humeau et al., 2019; Lin et al., 2022). As a result, it is necessary to extend the current works, where the representation capability can be enhanced for both types of contextualized embeddings.

3 Methodology

We start with an overview of DupMAE in this section. The framework of DupMAE is shown as Figure 2. There is an unified encoder (A), where the masked input is encoded into its contextualized embeddings. There are two decoders working collaboratively. One decoder is applied for [CLS] decoding (B): it employs a single-layer transformer, which reconstructs the original sentence based on the [CLS] embedding. The other one is used for OT decoding (C): it utilizes a linear projection unit (LPU), which transforms the ordinary token embeddings into the vocabulary space. The transformed results are aggregated by max-pooling, where the BoW feature of the input is predicted. The two decoding tasks are jointly conducted to train the encoder. The [CLS] and OT embeddings are aggregated for the final representation of the input. With proper dimension reduction, it may preserve the same computation cost of inner-product and memory footprint as one single dense vector.

3.1 Encoding

The input sentence X is sampled and masked as $\tilde{X}_{enc}$ by randomly replacing some of its tokens with the special token [M]. A moderate masking ratio is applied during the encoding stage (30%); as a result, the majority of the input information will be preserved by encoding result. The encoding network $\Phi^{enc}(\cdot)$ is used to transform the masked sentence into the contextualized embeddings for [CLS] ($\mathbf{h}_{\tilde{X}}$) and ordinary tokens ($\mathbf{H}_{\tilde{X}_{enc}}$):

$$\mathbf{h}_{\tilde{X}}, \mathbf{H}_{\tilde{X}_{enc}} \leftarrow \Phi_{enc}(\tilde{X}_{enc}). \tag{1}$$

In order to capture the in-depth semantics about the sentence, a full-scale BERT-like encoding network is used to generate to the contextualized embeddings. The masked tokens for the encoder are predicted following the typical form of masked language modeling (MLM) (Devlin et al., 2019). The training loss of MLM is denoted as $\mathcal{L}_{mlm}$.

3.2 [CLS] Decoding

The [CLS] embedding joins with the masked input (re-generated) to decode the original sentence. Following the recent auto-encoding based pre-training methods (Xiao et al., 2022b; Wang et al., 2022), the decoding is performed with a simplified network and an aggressive masking ratio. These settings will force the embedding to fully capture the input information where high-fidelity reconstruction can be made. Particularly, the input X is masked as $\tilde{X}_{dec}$, with half of its tokens selected for masking. An one-layer transformer is utilized for decoding, and two hidden-state streams: $\mathbf{H}_1$ (query stream), $\mathbf{H}_2$ (context stream), are used as the input:

$$\begin{aligned}\mathbf{H}_1 &\leftarrow [\mathbf{h}_{\tilde{X}} + \mathbf{p}_0, ..., \mathbf{h}_{\tilde{X}} + \mathbf{p}_N],\\ \mathbf{H}_2 &\leftarrow [\mathbf{h}_{\tilde{X}}, \mathbf{e}_{x_1} + \mathbf{p}_1, ..., \mathbf{e}_{x_N} + \mathbf{p}_N].\end{aligned} \tag{2}$$

Here, $\mathbf{h}_{\tilde{X}}$ is the [CLS] embedding from encoder, $\mathbf{e}_{x_i}$ is the i-th token embedding, $\mathbf{p}_i$ is the i-th position embedding. Given the above input, it performs self-attention w.r.t. the mask matrix $\mathbf{M} \in \mathbb{R}^{L \times L}$:

$$\begin{aligned}&\mathbf{Q} = \mathbf{H}_1\mathbf{W}^Q, \mathbf{K} = \mathbf{H}_2\mathbf{W}^K, \mathbf{V} = \mathbf{H}_2\mathbf{W}^V;\\ &\mathbf{M}_{ij} = \begin{cases} 0, & \text{can be attended,} \\ -\infty, & \text{masked;} \end{cases}\\ &\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}^T\mathbf{K}}{\sqrt{d}} + \mathbf{M})\mathbf{V}.\end{aligned} \tag{3}$$

The output $\mathbf{A}$, together with $\mathbf{H}_1$ (from the residual connection) are used to predict the original input. Finally, the following objective is optimized:

$$\mathcal{L}_{dec} = \sum\nolimits_{x_i \in X} \text{CE}(x_i | \mathbf{A}, \mathbf{H}_1). \tag{4}$$

As the decoder only contains one transformer layer, each token x_i is reconstructed based on the unique context which are visible to the i-th row of $\mathbf{M}$. The mask matrix is generated by the following rules:

$$\mathbf{M}_{ij} = \begin{cases} 0, & x_j \in s(X_{\neq i}), \text{ or } j_{|i \neq 0} = 0 \\ -\infty, & \text{otherwise.} \end{cases} \tag{5}$$

In the i-th row, the sampled positions $s(X_{\neq i})$ and the first position are set to 0, meaning that they will be made visible to the i-th token during self-attention. Meanwhile, the non-sampled positions and the diagonal position (indicating the position of the i-th token itself) will be $-\infty$, which will keep them masked during self-attention.

3.3 OT Decoding and Training Objective

The decoding task for OT embeddings are designed based on two considerations. On one hand, it will follow the same spirit as the [CLS] decoding task, where the decoding network is designed to be simplified. On the other hand, it will take a differentiated objective with the [CLS] decoding; therefore, it may facilitate the two types of embeddings to capture complementary information. In this place, we proposed the following decoding task for OT embeddings.

First of all, the OT embeddings (with masked tokens excluded) $\mathbf{H}_{\tilde{X}_{enc}}$: $\{\mathbf{h}_{x_1}, ..., \mathbf{h}_{x_N}\}$ are linearly transformed into the vocabulary space:

$$\boldsymbol{\mu}_{x_i} \leftarrow \mathbf{h}_{x_i}^T \mathbf{W}^O, \; x_i \in \tilde{X}_{enc}, \tag{6}$$

($\mathbf{W}^O \in \mathbb{R}^{d \times |V|}$, d: embedding dimension, $|V|$: vocabulary size.) The transformed results are aggregated through token-wise max-pooling:

$$\boldsymbol{\mu}_{\tilde{X}_{enc}} \leftarrow token.\text{Max}(\{\boldsymbol{\mu}_{x_i} | \tilde{X}_{enc}\}), \tag{7}$$

where the largest activation values of all tokens in $\tilde{X}_{enc}$ will be preserved for each vocabulary.

Secondly, we propose the following objective where the BoW feature of the input is recovered. As a result, the lexical information can be better encoded by the OT embeddings.

$$\min. - \sum_{x \in set(X)} \log \frac{\exp(\boldsymbol{\mu}_{\tilde{X}_{enc}}[x])}{\sum_{x' \in V} \exp(\boldsymbol{\mu}_{\tilde{X}_{enc}}[x'])}, \tag{8}$$

where $x \in set(X)$ is a unique token of the input X, V is the whole vocabulary. The encoder's loss, the decoding losses from [CLS] (Eq. 4) and OT (Eq. 8) are added up as our training objective:

$$\min. \mathcal{L}_{mlm} + \mathcal{L}_{dec} + \mathcal{L}_{BoW}. \tag{9}$$

3.4 Representation

A remaining problem of DupMAE is how to generate the semantic representation for the input. It's expected that the [CLS] and OT embeddings can be collaborated, where a stronger representation can be produced. Besides, it has to be compact, such that the retrieval process can be efficient in terms of computation cost and memory consumption. To these ends, we propose the following aggregation method. Firstly, the [CLS] embedding $\mathbf{h}_X$ is linearly transformed to a lower dimension (d'):

$$\hat{\mathbf{h}}_X \leftarrow \mathbf{h}_X^T \mathbf{W}^{cls},\ \mathbf{W}^{cls} \in \mathbb{R}^{d \times d'}. \qquad (10)$$

Secondly, knowing that the OT embeddings are aggregated into a high-dim vector $\boldsymbol{\mu}_X$, we directly reduce its dimension via sparsification:

$$\hat{\boldsymbol{\mu}}_X \leftarrow \{i : \boldsymbol{\mu}_X[i] \mid i \in I_X\}. \qquad (11)$$

Here, I_X stands for the indexes where $\boldsymbol{\mu}_X[i] \in$ Top-k($\boldsymbol{\mu}_X$), k is the number of elements to be preserved for $\boldsymbol{\mu}_X$. For each document, we concatenate the dim-reduction results of [CLS] and OT embeddings as its semantic representation: $[\hat{\mathbf{h}}_X; \hat{\boldsymbol{\mu}}_X]$. For each query, we measure its relevance to a document based on the following form of inner-product:

$$\langle q, d \rangle = \hat{\mathbf{h}}_q^T \hat{\mathbf{h}}_d + \sum\nolimits_{I_d} \boldsymbol{\mu}_q[i] \boldsymbol{\mu}_d[i]. \qquad (12)$$

With proper configurations, the computation cost of inner product and memory footprint will be same as working conventional dense embeddings.

Fine-Tuning. The pre-trained encoder is fine-tuned with three steps. Firstly, the contrastive learning is conducted for the in-batch negatives (IB):

$$\min. -\sum_q \log \frac{\exp(\langle q, d^+ \rangle)}{\sum_{d \in \{d^+, \text{IB}\}} \exp(\langle q, d \rangle)}. \qquad (13)$$

Secondly, we get the ANN hard negatives for each query based on the first-stage encoder D^- (Xiong et al., 2020), and continue to perform contrastive learning with both hard and in-batch negatives:

$$\min. -\sum_q \log \frac{\exp(\langle q, d^+ \rangle)}{\sum_{d \in \{d^+, D^-, \text{IB}\}} \exp(\langle q, d \rangle)}. \qquad (14)$$

Thirdly, we perform knowledge distillation: a cross-encoder is trained to discriminate the positives (d^+) from negatives (d^-) for each query. Then, the soft labeled cross-entropy is minimized:

$$\min. -\sum_q \sigma_q^d \log \frac{\exp(\langle q, d^+ \rangle)}{\sum_{d \in \{d^+, D^-\}} \exp(\langle q, d \rangle)} \qquad (15)$$

where σ_q^d is the softmax activation of the cross-encoder's prediction of q and d's relevance.

The first two fine-tuning steps are cost effective, as they only involve low-cost operations. The third step will bring a much larger cost due to the training and scoring of the cross-encoder. Nevertheless, it also helps to fine-tune the model for a better precision. In our experiments, comprehensive analysis is made for DupMAE's impact on different stages.

4 Experiment

The empirical studies are conducted to explore the following research questions.

- **RQ 1.** Whether DupMAE produces better semantic representations, compared with the existing competitive pre-training baselines?
- **RQ 2.** Whether DupMAE is able to maintain its advantages throughout different situations?
- **RQ 3.** Whether DupMAE benefits from the joint utilization of both [CLS] and OT embeddings, and what's the individual contribution from each embedding?
- **RQ 4.** Whether the pre-training tasks contribute to both [CLS] and OT embeddings?

Benchmarks. The experiments are conducted for both supervised and zero-shot settings. We choose the **passage** and **document** retrieval task of **MS MARCO** benchmark (Nguyen et al., 2016) for supervised evaluations. It contains queries from Bing Search, where ground-truth answers to the queries need to be retrieved from 8.8 million passages and 3 million documents, respectively. The queries from the dev set and TREC Deep Learning track in 2019 (DL'19) (Craswell et al., 2020) are used for evaluation. We leverage **BEIR** benchmark (Thakur et al., 2021) for zero-shot evaluations. It contains a total of 18 datasets, which covers diverse types of retrieval tasks, such as question answering, duplication detection, and fact verification, etc. Following the official evaluation script, the pre-trained models are fine-tuned with MS MARCO queries, and evaluated for their out-of-domain retrieval performances on each of the 18 datasets.

Baselines. We consider the following baselines for supervised evaluations according to their fine-tuning strategies. The first one only leverage **hard or in-batch negatives**, including ANCE (Xiong et al., 2020), SEED (Lu et al., 2021), ADORE (Zhan et al., 2021), COSTA (Ma et al., 2022), PROP (Ma et al., 2021a), B-PROP (Ma et al., 2021b), Aggretriever (Lin et al., 2022), and co-Condener (Gao and Callan, 2022). The second type leverage **sophisticated fine-tuning** strategies like knowledge distillation, including RocketQAv2 (Ren et al., 2021), AR2 (Zhang et al., 2021), AR2+SimANS (Zhou et al., 2022), SPLADEv2 (Formal et al., 2021), ColBERTv2 (Santhanam et al., 2021), ERNIE-Search (Lu et al., 2022),

	Passage Dev		DL'19
Methods	**MRR@10**	**R@1000**	**NDCG@10**
ANCE	0.330	0.959	0.648
SEED	0.339	0.961	–
coCondenser	0.382	0.717	0.684
Aggretriver	0.363	0.973	0.678
RocketQAv2	0.388	0.981	–
AR2	0.395	0.986	–
AR2+SimANS	0.409	0.987	–
SPLADEv2	0.368	0.979	0.729
ColBERTv2	0.397	0.984	–
ERNIE-Search	0.401	0.982	–
SimLM	0.411	0.987	0.714
RetroMAE (stage 3)	0.416	0.988	0.681
DupMAE (stage 2)	0.410	0.987	0.713
DupMAE (stage 3)	**0.426**	**0.989**	**0.751**

Table 1: MS MARCO passage retrieval.

	Document Dev		DL'19
Methods	**MRR@100**	**R@100**	**NDCG@10**
BM25	0.277	0.807	0.519
BERT	0.389	0.877	0.594
ICT	0.396	0.882	0.605
PROP	0.394	0.884	0.596
B-PROP	0.395	0.883	0.601
COIL	0.397	–	0.636
ANCE (first-p)	0.377	0.893	0.615
ANCE (max-p)	0.384	0.906	0.628
STAR	0.390	0.913	0.605
Adore	0.405	0.919	0.628
SEED	0.396	0.902	0.605
COSTA	0.422	0.919	0.626
RetroMAE (stage 2)	0.432	0.935	0.593
DupMAE (stage 2)	**0.451**	**0.950**	**0.667**

Table 2: MS MARCO document retrieval.

SimLM (Wang et al., 2022), RetroMAE (Xiao et al., 2022b). We emphasize two methods for zero-shot evaluations. One is BM25, which is a common sparse retrieval method and a strong baseline in zero-shot settings. The other type are the large-scale pre-trained retrievers based on contrastive learning: Contriever (Izacard et al., 2021) and the family of GTR-* (Ni et al., 2021). Among them, GTR-XXL is a super large model with 4.8B parameters (over 40× larger than BERT base).

Implementation details. DupMAE utilizes a bi-directional transformer network as its encoder, with 12 layers, 768 hidden-dim, and a vocabulary of 30522 tokens (same as BERT base). The decoder is an one-layer transformer. The [CLS] embedding and OT embedding are reduced to dim-384 by default. As a result, it will preserve the same computation cost of inner-product as the baselines which use dim-768 embeddings. We also explore other configurations of dimensions in our experiments. The masking ratio is set to 0.3 for encoder and 0.5 for decoder. We leverage three commonly used corpora for pre-training: Wikipedia, BookCorpus (Devlin et al., 2019), and MS MARCO (Nguyen et al., 2016). The pre-training and fine-tuning take place on machines with 8× Nvidia V100 (32GB) GPUs. The models are implemented with PyTorch 1.8 and HuggingFace transformers 4.16.

4.1 Main Results

The **supervised evaluations** are shown as Table 1 and 2, where the following observations can be made. Firstly, DupMAE achieves superior performances on both tasks of MS MARCO. For passage retrieval, it reaches a MRR@10 of 0.426, outperforming the previous SOTA pre-trained models, like SimLM and RetroMAE, by +1% absolute point. For document retrieval, it achieves a MRR@100 of 0.451, leading to +1.9% absolute improvements. Such observations indicate that the pre-trained model's representation quality is substantially improved with DupMAE. Note that DupMAE's performances are much higher than baselines like ColBERTv2, SPLADE, and COIL. These methods utilize multi-vector for semantic representation, which is more expensive in terms of memory and computation. Besides, even with DupMAE (stage 2), which simply takes one-round of hard-negative sampling, we may outperform many of the baselines relying on sophisticated fine-tuning strategies, like knowledge distillation (ColBERTv2, ERNIE-Search) and joint learning of retriever and ranker (AR2, AR2+SimANS).

To summary, the above observations reflect DupMAE's two-fold merits to real-world applications: **1.** it improves the best performance where neural retrievers may get, **2.** it helps to produce strong retrieval quality in a cost-effective way.

For **zero-shot settings**, we report the retrieval performance on every single dataset, and measure the overall performance by taking the average of all 18 datasets (Table 3). Firstly, DupMAE achieves remarkable performance on BEIR, reaching an average NDCG@10 of 0.477 in all 18 datasets. It outperforms its close peer RetroMAE on 13 out of 18 datasets, and by +2.5% absolute point in total average. Secondly, it is known that BM25 is a strong baseline for zero-shot retrieval, which outperforms many of the existing pre-trained models on BEIR benchmark. Even for the massive-scale GTR-XXL,

Datasets	**BM25**	**BERT**	**SEED**	**Condenser**	**Contriever**	**GTR-base**	**GTR-XXL**	**RetroMAE**	**DupMAE**	**DupMAE†**
TREC-COVID	0.656	0.615	0.627	0.750	0.596	0.539	0.501	**0.772**	0.728	0.770$^{\uparrow}$
BioASQ	0.465	0.253	0.308	0.322	0.383	0.271	0.324	0.421	0.508	**0.514**$^{\uparrow}$
NFCorpus	0.325	0.260	0.278	0.277	0.328	0.308	0.342	0.308	0.346	**0.366**$^{\uparrow}$
NQ	0.329	0.467	0.446	0.486	0.498	0.495	0.568	0.518	0.570	**0.578**$^{\uparrow}$
HotpotQA	0.603	0.488	0.541	0.538	0.638	0.535	0.599	0.635	0.681	**0.683**$^{\uparrow}$
FiQA-2018	0.236	0.252	0.259	0.259	0.329	0.349	**0.467**	0.316	0.345	0.375$^{\uparrow}$
Signal-1M(RT)	**0.330**	0.204	0.256	0.261	0.199	0.261	0.273	0.265	0.213	0.237$^{\uparrow}$
TREC-NEWS	0.398	0.362	0.358	0.376	0.428	0.337	0.346	0.428	0.427	**0.433**$^{\uparrow}$
Robust04	0.408	0.351	0.365	0.349	0.476	0.437	**0.506**	0.447	0.479	0.503$^{\uparrow}$
ArguAna	0.315	0.265	0.389	0.298	0.446	0.511	**0.540**	0.433	0.474	0.465$^{\downarrow}$
Touche-2020	0.367	0.259	0.225	0.248	0.204	0.205	0.256	0.237	0.343	**0.382**$^{\uparrow}$
CQADupStack	0.299	0.282	0.290	0.347	0.345	0.357	**0.399**	0.317	0.320	0.336$^{\uparrow}$
Quora	0.789	0.787	0.852	0.853	0.865	0.881	**0.892**	0.847	0.845	0.853$^{\uparrow}$
DBPedia	0.313	0.314	0.330	0.339	0.413	0.347	0.408	0.390	0.418	**0.419**$^{\uparrow}$
SCIDOCS	0.158	0.113	0.124	0.133	**0.165**	0.149	0.161	0.150	0.153	**0.165**$^{\uparrow}$
FEVER	0.753	0.682	0.641	0.691	0.758	0.660	0.740	0.774	0.800	**0.817**$^{\uparrow}$
Climate-FEVER	0.213	0.187	0.176	0.211	0.237	0.241	**0.267**	0.232	0.232	0.219$^{\downarrow}$
SciFact	0.665	0.533	0.575	0.593	0.677	0.600	0.662	0.653	0.699	**0.725**$^{\uparrow}$
AVERAGE	0.423	0.371	0.391	0.407	0.448	0.416	0.458	0.452	0.477	**0.491**$^{\uparrow}$

Table 3: Zero-shot retrieval (NDCG@10) on BEIR. DupMAE† is the extended DupMAE via domain-adaptation, where ↑ indicates the improvement over DupMAE. The highest values w./w.o. DupMAE† are marked in **bold** and underlined, respectively.

which uses as much as 4.8 billion parameters and huge amounts of pre-training data, it still loses to BM25 on 8 out 18 datasets. However, with DupMAE, we may outperform BM25 on 15 out of 18 datasets, leading to as much as +5.4% absolute improvement in total average. The above performances are impressive considering that DupMAE is merely based on a BERT-base scale encoder and uses much less pre-training data compared with other strong baselines, like Contriever and GTR.

Recently, it becomes popular to leverage domain-adaptation to improve neural retrievers' zero-shot performances (Xin et al., 2021; Wang et al., 2021b). In this place, we adopt a straightforward approach for domain adaptation: we continually perform DupMAE pre-training on BEIR unlabeled corpus before fine-tuning with the source domain training queries (denoted as DupMAE†). Despite simplicity, this approach is surprisingly effective, as performances are improved on 16 out of 18 datasets, leading to an average NDCG@10 of 0.491.

Given the analysis about the main experiment results in Table 1, 2 and 3, we may draw the following conclusions in response to **RQ 1** and **2**:

- **Con 1**. DupMAE makes large improvements over the baselines, verifying that it substantially contributes to the pre-trained model's representation capacity and transferability.
- **Con 2**. DupMAE is able to maintain superior retrieval performances across different evaluation tasks on both supervised and zero-shot scenarios, which indicates DupMAE's strong usability in real-world applications.

4.2 Ablation Studies

After verifying DupMAE's overall effectiveness, it remains to figure out which factors contribute to its improvements. Thus, we perform ablation studies as Table 4. We use MS MARCO dataset for our exploration, and fine-tune the pre-trained models with hard negative samples (stage 2).

We conduct the following two sets of experiments. Firstly, we explore **the impact from pre-training**, whose results are shown in the upper part of Table 4. Remember that DupMAE includes two decoding tasks as discussed in Section 3.3: CLS decoding and OT decoding, we make evaluations for three alternative forms accordingly. 1) CLS decoding only, where only the [CLS] embedding is pre-trained 2) OT decoding only, where only the OT embeddings are pre-trained, 3) CLS and OT decoding, which is exactly the pre-training method used by DupMAE. We also introduce RetroMAE for comparison. Although RetroMAE and "CLS decoding only" share the same pre-training task, their representations are generated differently, as DupMAE jointly uses [CLS] and OT embeddings.

		MS MARCO (Passage) Dev				
	Methods	**MRR@10**	**MRR@100**	**R@10**	**R@100**	**R@1000**
1.	RetroMAE	0.3928	0.4032	0.6749	0.9178	0.9849
	CLS decoding only	0.4008	0.4099	0.6906	0.9229	0.9840
	OT decoding only	0.4002	0.4092	0.6890	0.9213	0.9831
	CLS and OT decoding	**0.4102**	**0.4202**	**0.7049**	**0.9280**	**0.9874**
2.	CLS:768	0.3941	0.4040	0.6865	0.9174	0.9871
	OT:768	0.4019	0.4114	0.6934	0.9095	0.9814
	CLS:384, OT:384	**0.4102**	**0.4202**	**0.7049**	0.9280	0.9874
	CLS:384, OT:260	0.4071	0.4171	0.7037	**0.9293**	**0.9882**

Table 4: Ablation studies: 1. impact from pre-training, 2. impact from embedding dimensions.

We may get the following observations from the experiment results. Firstly, the joint utilization of the two pre-training tasks leads to the optimal retrieval quality, where the MRR@10 grows beyond "CLS only" and "OT only" by almost +1% absolute point. As a result, the effectiveness of jointly performing both pre-training tasks can be verified. Secondly, RetroMAE's performance is inferior to other methods, especially "CLS pre-train only" which share the pre-training task with it. Such an observation reveals the different capacity between the two semantic representations: DupMAE relies on the contextualized embeddings from both [CLS] and ordinary tokens, while RetroMAE only leverages the [CLS] token's embedding.

We further explore **the impact from different semantic representations** in the lower part of Table 4). As introduced in Section 3.3, DupMAE's default semantic representation (dim-768) consists of two parts: half of its elements come from the linear projection of [CLS] embedding, while the other half come from the sparsification of OT embeddings (denoted as "CLS:384, OT:384"). In this place, we consider two variational formulations: (1) "CLS:768", which directly uses the [CLS] embedding, and (2) "OT:768", where the top 768 elements of the OT embeddings are used for the representation of the input. According to the experiment results, the performance of "OT:768" is slightly better than "CLS:768". At the same time, "CLS:384, OT:384" (the default setting of DupMAE) gives rise to a better performance than both variational formulations. The above observations indicate that the contextualized embeddings from [CLS] and ordinary tokens may provide complementary information about the input data. As a result, the joint utilization of both types of embeddings is able to generate a more powerful semantic representation.

Note that although "CLS:384, OT:384" preserves the same computation cost of inner-product as "CLS:768", it's memory cost is slightly higher than "CLS:768", as extra space is needed to save the indexes of OT embeddings' sparsification results. Particularly, each index will take about 15 extra bits for index storage knowing that the vocabulary space is 30522 . In this place, we introduce another variational formulation "CLS:384, OT:260" by further reducing the dimension of OT embeddings. As a result, it may take the same memory footprint as "CLS:768". It can be observed that the new combination "CLS:384, OT:260" still outperforms the first two variations, and maintains a similar performance as "CLS:384, OT:384".

Given the above analysis, we may come to the following conclusions in response to **RQ 3** and **4**:

- **Con 3**. The collaboration of [CLS] and OT embeddings brings stronger semantic representations, indicating that encoded information from the two types of embeddings are complementary to each other.
- **Con 4**. Both tasks: [CLS] and OT decoding, contribute to DupMAE; the joint conduct of both tasks leads to the optimal performance.

5 Conclusion

This paper presents DupMAE, a new approach for retrieval-oriented pre-training, where the semantic representation capacities can be jointly enhanced for all contextualized embeddings of the language model. It employs two complementary tasks: one reconstructs the original input from the [CLS]'s embedding, the other one predicts the BoW features based on the OT embeddings. The two tasks are jointly conducted to learn an unified encoder. The two types of embeddings, with reduced dimensions, are aggregated to be a joint semantic representation. The effectiveness of our proposed method is empirically verified, where remarkable performances are achieved on MS MARCO and BEIR benchmarks throughout different situations.

Limitations

Although DupMAE is to learn representation instead of generative models, it performs pre-training on open web data. Therefore, it is also subject to potential ethical and social risks, like bias, discrimination, and toxicity. Besides, DupMAE is pre-trained with comparatively limited amount of data due to the constraint on computation resources. Despite that it already achieves a promising retrieval performance at present, it remains to explore whether the performance can be further improved with the scaling up of pre-training data, by leveraging more high-quality datasets like C4 and OpenWebText.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Nos. 62272054, 62192784) and Xiaomi Young Talents Program.

References

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*, pages 4171–4186. Association for Computational Linguistics.

Qian Dong, Shuzi Niu, Tao Yuan, and Yucheng Li. 2022. Disentangled graph recurrent network for document ranking. *Data Science and Engineering*, pages 30–43.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR*, pages 113–122.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*.

Sheng-Chieh Lin, Minghan Li, and Jimmy Lin. 2022. Aggretriever: A simple approach to aggregate textual representation for robust dense passage retrieval. *arXiv preprint arXiv:2208.00511*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 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*, pages 2780–2791.

Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, et al. 2022. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *arXiv preprint arXiv:2205.09153*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a discriminative text encoder for dense retrieval via contrastive span prediction. *arXiv preprint arXiv:2204.10641*.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021a. Prop: pre-training with representative words prediction for ad-hoc retrieval. In *WSDM*, pages 283–291.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021b. B-prop: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *SIGIR*, pages 1513–1522.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

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. *arXiv preprint arXiv:1910.10683*.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *arXiv preprint arXiv:2110.07367*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021a. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. *arXiv preprint arXiv:2104.06979*.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021b. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *arXiv preprint arXiv:2112.07577*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *arXiv preprint arXiv:2207.02578*.

Shitao Xiao, Zheng Liu, Weihao Han, Jianjin Zhang, Yingxia Shao, Defu Lian, Chaozhuo Li, Hao Sun, Denvy Deng, Liangjie Zhang, et al. 2022a. Progressively optimized bi-granular document representation for scalable embedding based retrieval. In *WWW*, pages 286–296.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022b. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. *arXiv preprint arXiv:2205.12035*.

Shitao Xiao, Zheng Liu, Yingxia Shao, Tao Di, Bhuvan Middha, Fangzhao Wu, and Xing Xie. 2022c. Training large-scale news recommenders with pretrained language models in the loop. In *SIGKDD*, pages 4215–4225.

Shitao Xiao, Zheng Liu, Yingxia Shao, Defu Lian, and Xing Xie. 2021. Matching-oriented product quantization for ad-hoc retrieval. *arXiv preprint arXiv:2104.07858*.

Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul N Bennett. 2021. Zero-shot dense retrieval with momentum adversarial domain invariant representations. *arXiv preprint arXiv:2110.07581*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR*, pages 1503–1512.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Adversarial retriever-ranker for dense text retrieval. *arXiv preprint arXiv:2110.03611*.

Jianjin Zhang, Zheng Liu, Weihao Han, Shitao Xiao, Ruicheng Zheng, Yingxia Shao, Hao Sun, Hanqing Zhu, Premkumar Srinivasan, Weiwei Deng, et al. 2022. Uni-retriever: Towards learning the unified embedding based retriever in bing sponsored search. In *SIGKDD*, pages 4493–4501.

Kun Zhou, Yeyun Gong, Xiao Liu, Wayne Xin Zhao, Yelong Shen, Anlei Dong, Jingwen Lu, Rangan Majumder, Ji-Rong Wen, Nan Duan, et al. 2022. Simans: Simple ambiguous negatives sampling for dense text retrieval. *arXiv preprint arXiv:2210.11773*.

Mingdong Zhu, Derong Shen, Lixin Xu, and Xianfang Wang. 2021. Scalable multi-grained cross-modal similarity query with interpretability. *Data Science and Engineering*, pages 280–293.

A Appendix

According to our experimental results in Table 4, the [CLS] and OT embeddings may jointly produce a stronger semantic representation to improve the retrieval quality. In this place, we provide a case analysis as Table 5 and 6, which will visualize the benefit introduced by each type of embedding, and help to explain the design of the pre-training tasks.

A.1 Settings

In our exploration, the [CLS] embedding and OT embeddings (aggregated and sparsified in the same way as introduced in Section 3.3) are used independently for the retrieval tasks. That's to say, the query and answer's relationships are measured by the [CLS] embeddings' similarity and OT embeddings' similarity, respectively. We select queries from the evaluation set of MS MARCO for demonstration. For each query, we count it as a successful case w.r.t. a specific type of embeddings, if its ground-truth answer can be retrieved within the Top-10 results. If the ground-truth answer is missed by one type of embeddings, its Top-1 retrieved answer will be posted for comparison.

A.2 Analysis

Given the limitation of space, we select four representative queries for demonstration. The four queries can be partitioned into two sets: in Table 5, the ground-truth answers are retrieved by [CLS] embeddings; while in Table 6, the ground-truth answers are retrieved by OT embeddings.

• **Good cases by [CLS] embeddings**. In Table 5, the two queries' ground-truth answers are retrieved by the [CLS] embeddings. For both cases, it calls for the pre-trained model to capture fine-grained **semantic relationships** between the query and answer. In particular, the first query is essentially about the car brands which belong to Ford. The [CLS] embedding successfully establish the connection between "build" and "own" (marked in blue). Therefore, the ground-truth answer can be successfully retrieved. Similarly, the second query emphasizes "cncellation" fee. By identifying the relationship between "cncellation" and "Cancel" (marked in blue), the ground-truth answer is successfully retrieved once again. Comparatively, although OT embeddings retrieve answers with close lexical features, e.g., "built", "fee" (marked in red), they appear to be less proficient in capturing the semantic relationships in both cases, where the correct answers are missed from their top-10 results.

• **Good cases by OT embeddings**. In Table 6, the two queries' ground-truth answers are retrieved by the OT embeddings. For both cases, it calls for the pre-trained model to precisely identify the ground-truth answers, which are not only semantically close to the queries, but also contain specific **lexical features**. Particularly, the first query asks about a certain type of material called "copper coated carbon rods". As a result, it is important to retrieve the answer which contain exactly the same term. The [CLS] embedding finds "copper-clad steel" (marked in red). Although similar, it is different from the required term. While with the OT embeddings, the ground-truth answer is successfully retrieved. Note that it's challenging for this case, knowing that the related term "Copper coated carbon electrods" (marked in blue) is wrapped in a long passage. The second query asks about the colour which represents selflessness. Although the [CLS] embedding finds the passage which is relevant to the symbolic meaning of colour (marked in red), it ignores the key term "selflessness" (marked in blue). On top of the OT embeddings, it successfully retrieves the ground-truth answer, which is not only semantically close to the required topic (color symbolism), but also contains the required term (selflessness).

• **Discussions**. It is known that both semantic and lexical features are important to information retrieval problems, such as search engine and question answers. From the above analysis, we may

Query	**Retrieved answer by [CLS] embedding**	**Retrieved answer by OT embeddings**
what cars does ford build?	What car companies does Ford own? Ford owns Jaguar (-30%), Land Rover (-50%), Aston Martin (-%10), Lincoln, Mercury, Volvo (-70%), and Mazda (-40%). I'm not quite sure of those percentages, nor am I sure if Ford owns 100% owns Lincoln and Volvo, but there's the basic gist of what Ford owns now. The above answer is incorrect. Ford has sold Jaguar, Volvo, & Land Rover. (**Ground-Truth. Rank 4th**)	Passenger Cars. The Taurus, Sable and Lincoln are built in Chicago, while many of Ford's engines are assembled in Brook Park, Ohio, with one Dearborn, Michigan, plant dedicated solely to auto parts. (**Rank 1st**)
delta airlines cncellation fee?	How to Cancel Flights on Delta Air Lines. When the credit is used to pay for new flights, the change fee will be assessed. For example, say you bought non-refundable Delta domestic flight tickets for $650, but your plans changed. When you are ready to purchase new flights, the fare has increased to $700. Your credit is $650 âx80x93 $200 change fee = $450, so your out-of-pocket cost to buy the new ticket is $700 - $450 = $250. Make sure to inform Delta before departure that you will not be on the flight and request the travel credit.our credit is $650 âx80x93 $200 change fee = $450, so your out-of-pocket cost to buy the new ticket is $700 - $450 = $250. Make sure to inform Delta before departure that you will not be on the flight and request the travel credit. (**Ground-Truth. Rank 3rd**)	As of publication, Delta charges a minimum fee of $178 for most domestic flights and $250 on flights to Alaska, Hawaii and the Virgin Islands, with additional charges based on the pet and carrier weight. (**Rank 1st**)

Table 5: Cases where the [CLS] embedding helps to retrieve the ground-truth answers.

Query	**Retrieved answer by [CLS] embedding**	**Retrieved answer by OT embeddings**
what are copper coated carbon rods used for?	Copper-clad steel (CCS), also known as copper-covered steel or the trademarked name Copperweld is a bi-metallic product, mainly used in the wire industry that combines the high mechanical resistance of steel with the conductivity and resistance to corrosion of copper. (**Rank 1st**)	Coidan Graphite Products supply Graphite Electrodes primarily used for the secondary production of steel EAF and ladle furnaces. Our graphite electrode stock has additional applications, such as melting products in smelting furnaces, non-ferrous metals, ceramic products and to recycle waste. There are several grades of graphite electrodes, we can match the grade with the application to lower your melting costs. Please click through to see properties of the graphite electrodes we can offer, RP grade, HP grade, SHP grade and UHP graphite electrodes. In addition we supply graphite EDM electrodes for the mould makers together with many other Spark Erosion applications. Copper coated carbon electrodes of many shapes and sizes are used as gouging rods and welding rods in foundry applications. (**Ground-Truth. Rank 8th**)
what color represents selflessness?	But since it is also taken as off-white, it can be the color of degradation or cowardice. Orange. Symbolic of endurance and strength, orange is the color of fire and flame. it represents the red of passion tempered by the yellow of wisdom. It is the symbol of the sun. (**Rank 1st**)	Color Symbolism - The Deeper Meaning of Blue, Blue is on the visual level a calm and peaceful color. We think of it in terms of water, sky and universe. For most of us, sky and water give us a sense of familiarity and consequently of security. For many, the universe represents a larger unity and religion. Therefore, this hue expresses security and spiritual devotion. It is the color that leads to introspection and to our very essence. It represents such ideals as selflessness, sympathy, kindness, compassion and dedication. Blue is assigned to the physical body and, on a larger scale, represents the material aspects of life including the planet earth. (**Ground-Truth. Rank 1st**)

Table 6: Cases where the OT embeddings help to retrieve the ground-truth answers.

observe that the two types of embeddings may have their own advantages: the [CLS] embeddings tend to be more proficient in capturing the semantic closeness, while the OT embeddings may better leverage the lexical similarity. In DupMAE, we design two differentiated auto-encoding tasks for [CLS] and OT embeddings. Although both tasks help to better encode the semantic information with the contextualized embeddings, the OT decoding task emphasizes more of the lexical information, because the BoW feature needs to be directly predicted by the aggregation results of OT embeddings. By having such differentiated tasks, the two types of embeddings may focus on strengthening their unique advantages. Finally, it will help to optimize the quality of the joint representation when both types of embeddings work collaboratively.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.