

# Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction

Xinyu Ma, Jiafeng Guo\*, Ruqing Zhang, Yixing Fan and Xueqi Cheng  
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,  
Chinese Academy of Sciences, Beijing, China  
University of Chinese Academy of Sciences, Beijing, China  
{maxinyu17g, guojiafeng, zhangruqing, fanyixing, cxq}@ict.ac.cn

## ABSTRACT

Dense retrieval has shown promising results in many information retrieval (IR) related tasks, whose foundation is high-quality text representation learning for effective search. Some recent studies have shown that autoencoder-based language models are able to boost the dense retrieval performance using a weak decoder. However, we argue that 1) it is not discriminative to decode all the input texts and, 2) even a weak decoder has the bypass effect on the encoder. Therefore, in this work, we introduce a novel contrastive span prediction task to pre-train the encoder alone, but still retain the bottleneck ability of the autoencoder. The key idea is to force the encoder to generate the text representation close to its own random spans while far away from others using a group-wise contrastive loss. In this way, we can 1) learn discriminative text representations efficiently with the group-wise contrastive learning over spans and, 2) avoid the bypass effect of the decoder thoroughly. Comprehensive experiments over publicly available retrieval benchmark datasets show that our approach can outperform existing pre-training methods for dense retrieval significantly. Code and pre-trained models will be available at the URL <sup>1</sup>.

## CCS CONCEPTS

• Information systems → Information retrieval.

## KEYWORDS

Dense Retrieval, Pre-training for IR, Discriminative Representation

### ACM Reference Format:

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. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*, July 11 15, 2022, Madrid, Spain. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3477495.3531772>

\* Jiafeng Guo is the corresponding author.

<sup>1</sup><https://github.com/Albert-Ma/COSTA>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGIR '22, July 11 15, 2022, Madrid, Spain

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-8732-3/22/07...\$15.00

<https://doi.org/10.1145/3477495.3531772>

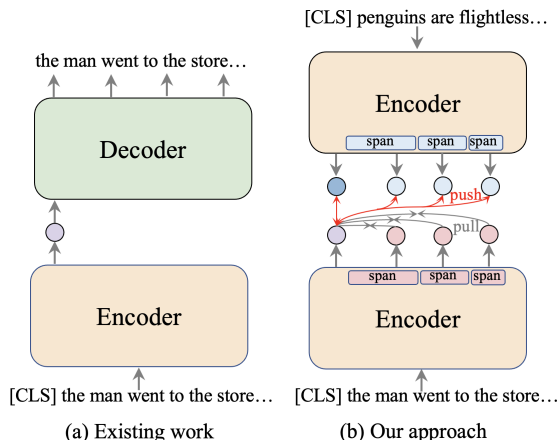
## 1 INTRODUCTION

Dense retrieval is receiving increasing interest in recent years from both industrial and academic communities due to its benefits to many IR related tasks, e.g., Web search [9, 17, 26], question answering [20, 23, 43] and conversational systems [10, 39]. Without loss of generality, dense retrieval usually utilizes a Siamese or bi-encoder architecture to encode queries and documents into low-dimensional representations to abstract their semantic information [18, 19, 21, 38, 40, 41]. With the learned representations, a dot-product or cosine function is conducted to measure the similarity between queries and documents. In essence, high-quality text representation is the foundation of dense retrieval to support effective search in the representation space.

Taking the pre-trained representation models like BERT [8] and RoBERTa [28] as the text encoders have become a popular choice [21, 38, 40] in dense retrieval. Beyond these direct applications, there have been some works on the pre-training objectives tailored for dense retrieval [2, 24]. For example, Chang et al. [2] presented three pre-training tasks that emphasize different aspects of semantics between queries and documents, including Inverse Cloze Task (ICT), Body First Selection (BFS), and Wiki Link Prediction (WLP). As we can see, some tasks even depend on certain special document structures, e.g., hyperlinks. When applying such pre-trained models to dense retrieval, marginal benefit could be observed on typical benchmark datasets as shown in Section 5.1.

To boost the dense retrieval performance, recent studies begin to focus on the autoencoder-based language models, which are inspired by the information bottleneck [37] to force the encoder to provide better text representations [25, 29]. As shown in Figure 1 (a), these methods pair a decoder on top of the encoder and then train the decoder to reconstruct the input texts solely from the representations given by the encoder. When generating text in the autoregressive fashion, the model takes not only the encoder's encodings but also the previous tokens as input. Through mathematical analysis, Lu et al. [29] showed that the decoder can bypass the dependency on the encoder via its access to previous predicted tokens, especially when the text is long and the decoder is strong. That is, a low reconstruction loss does not mean good text representations. To address this issue, they proposed to pre-train the autoencoder using a weak decoder, with restricted capacity (i.e., a shallower Transformer) and attention flexibility (i.e., restricting its access to the previous context).

Despite the great success on dense retrieval, the autoencoder-based language approach has several shortcomings: (1) It assumes that each token is equally important to the input text. A large proportion of words in the text are common words like the, of, etc,



**Figure 1: COSTA pre-trains the encoder alone with the contrastive span prediction task while still retaining the bottleneck ability by forcing the encoder to generate the text representation close to its own random spans. Moreover, it enhances the discriminative ability of the encoder by pushing the text representation away from others.**

since they are useful for composing the natural language. Thus, in the auto-regressive decoder, the model tends to decode common words to achieve a low reconstruction loss. As a result, the discriminative power of representations may be decreased, especially when the text is long with much noisy information. (2) The decoder, which is useless in the inference stage, is not a necessary part in the training stage either. Regardless of how weak the decoder is, it still has access to previous tokens to exploit the natural language patterns. Therefore, the bypass effect of the decoder still remains which may largely limit the encoding power of the encoder.

Therefore, in this paper, we propose to drop out the decoder and enforce the information bottleneck by the encoder itself to provide better text representations. The key idea is to enhance the consistency between semantic representations of the given text and that of its own random spans (i.e., a group) using a group-wise contrastive loss. To learn both the bottleneck and the discriminative ability, we introduce the contrastive span prediction task for dense retrieval. Specifically, as shown in Figure 1 (b), the contrastive span prediction task aims to force the encoder to learn the representation of the given text by pulling it towards that of its own multiple random spans in the semantic space, while pushing it far away from the representations of all the instances in other groups. We pre-train the Transformer-based encoder with the contrastive span prediction task and the Masked Language Model (MLM) task. The pre-trained model, namely COSTA for short, could then be fine-tuned on a variety of downstream dense retrieval tasks.

COSTA enables us to mitigate the aforementioned technical issues since: (1) COSTA offers incentives for representations of instances in a group sharing the same semantics to be similar, while penalizing the representations of groups expressing different semantics to be distinguished from each other. The group-wise supervision enables the encoder to look beyond the local structures of input texts and become more aware of the semantics of

other groups. This contributes to learning discriminative text representations. (2) Without an explicit decoder, our method is able to enforce an information bottleneck on the text representations. That is, pre-training the encoder alone can avoid the bypass effect of the decoder thoroughly.

We pre-train COSTA based on the English Wikipedia which contains millions of well-formed wiki-articles. We then fine-tune COSTA on four representative downstream dense retrieval datasets, including MS MARCO passage ranking task, MS MARCO document ranking task, and two TREC 2019 Deep Learning tracks. The empirical experimental results show that COSTA can achieve significant improvements over the state-of-the-art baselines. We also visualize the query and document representations to illustrate how COSTA generates discriminative text presentations to improve the retrieval performance. Under the low-resource setting, we show that good retrieval performance can be achieved across different datasets by fine-tuning the COSTA with very little supervision.

## 2 RELATED WORK

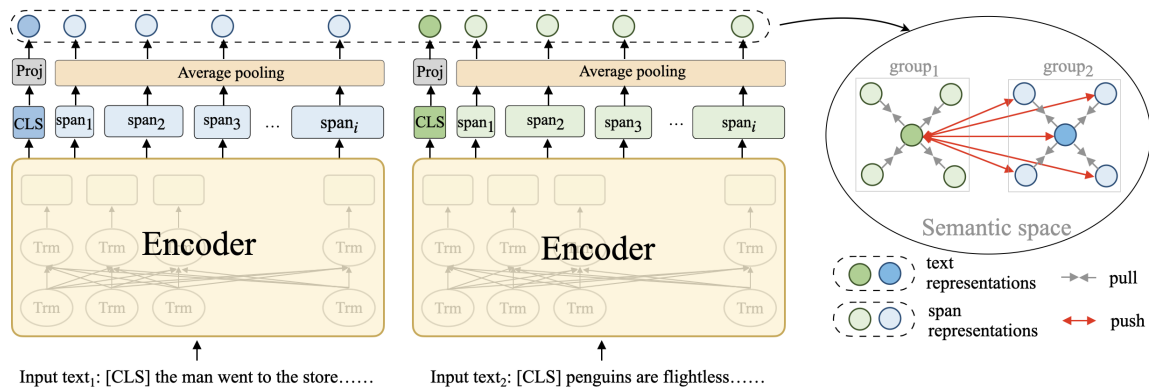
In this section, we briefly review the most related topics to our work, including dense retrieval and the pre-training method for IR.

### 2.1 Dense Retrieval

Dense retrieval models generally adopt a bi-encoder architecture to encode queries and documents separately for effective search. The relevance is computed by the simple similarity function like cosine or dot-product. Karpukhin et al. [20] showed that by utilizing in-batch negatives, dense retrieval models perform much better than BM25. After that, researchers began to explore various fine-tuning techniques to enhance dense retrieval models, such as mining hard negatives [38, 40], late interaction [21], distill knowledge from a strong teacher [27], query clustering [18], and data augmentation [33]. For example, Xiong et al. [38] proposed to construct hard negatives from an Approximate Nearest Neighbor (ANN) index of the corpus and period refresh the index during training. Zhan et al. [40] presented a dynamic hard negatives mining method applied upon fine-tuned dense retrieval models. Khattab and Zaharia [21] introduced a MaxSim late interaction operation to model the fine-grained similarity between queries and documents. Lin et al. [27] proposed to distill from ColBERT’s MaxSim operator into a simple dot product to enable a single-step ANN search. Qu et al. [33] introduced three training strategies, i.e., cross-batch negatives, denoised hard negatives, and data augmentation. Although these methods greatly improve the performance of dense retrieval models, they usually need more computational or storage cost that may limit their use.

### 2.2 Pre-training for IR

Researchers in the IR community have also designed pre-training objectives tailored for IR. For example, Chang et al. [2] proposed three tasks that closely resemble passage retrieval in question answering (QA): (1) ICT: The query is a sentence randomly drawn from the passage and the document is the rest of the sentences; (2) BFS: The query is a random sentence in the first section of a Wikipedia page, and the document is a random passage from the same page; and (3) WLP: The query is a random sentence in



**Figure 2: Contrastive Span Prediction Task.** A special token [CLS] is added in front of each input text to represent the whole text. The encoder maps the whole text and its own multi-granularity spans into semantic representations. Note a text and its spans form a group. Then, a projector network and an average pooling layer are applied on top of the text representation and the span representation, respectively. Finally, we train the Transformer-based encoder via a group-wise contrastive loss to force the representation of the text close to that of its spans in a group, while far away from other groups.

the first section of a Wikipedia page, and the document is a passage from another page where there is a hyperlink between the two pages. When applying these tasks in ad-hoc retrieval, we observed marginal improvements on typical benchmark datasets. Ma et al. [30, 31] proposed to sample representation words from the document according to a unigram document language model or a contrastive term distribution, and then pre-train the Transformer model to predict the pairwise preference between the two sampled word sets jointly with MLM. This task is designed for the re-ranking stage, when applied in the retrieval stage, improvements are limited. [11, 12] modify the Transformer architecture to establish structural readiness by doing LM pre-training actively condition on dense representations. The most related work with ours is SEED-Encoder [29] that pre-trains an autoencoder with a weak decoder to learn document representations for dense retrieval. Concretely, they use a three-layer Transformer as the decoder and also restrict its span attention to 2. But this method has two major issues as we discuss in Intro 1.

### 3 OUR APPROACH

In this section, we describe our proposed pre-training method for dense retrieval in detail.

#### 3.1 Motivation

Existing work has demonstrated that autoencoder-based language models are able to learn high-quality text representations for effective search. These models typically pair a decoder on top of the encoder and train the decoder to reconstruct the input texts solely from the encoder’s encodings. However, as observed in previous works [29], the decoder in autoencoder-based language models can exploit natural language patterns via its access to previous tokens and bypass the dependency on the encoder, especially when the sequence is long and the decoder is strong. Although Lu et al. [29] has proposed to pre-train the autoencoder using a weak decoder, the bypass effect of the decoder still remains. Besides, decoding all the input texts equally may decrease the discriminative power

of the representations, since the decoder favors the generation of common words with high frequency.

To address these issues, we propose to discard the decoder and enforce the information bottleneck on the encoder itself for better representation quality. We introduce a novel contrastive span prediction task to pre-train the encoder alone. Specifically, given an input text, we sample a set of spans to build a group. Then, we encourage the representation of the text and that of its own spans in the semantic space to be closer while keeping other groups away. In this way, it not only eliminates the bypass effect of the decoder thoroughly, but also aids the pre-trained model in better capturing discriminative semantics of the text via more effective group-level contrastive supervision. COSTA is pre-trained with contrastive span prediction task and Masked Language Model (MLM) task. The overall model architecture is shown in Figure 2. The pseudo pre-training algorithm is described in Algorithm 1. We introduce the contrastive span prediction task in detail next.

#### 3.2 Contrastive Span Prediction Task

Given a mini-batch of input texts, we first sample a set of multi-granularity spans for each text and encode the whole texts and spans into semantic representations. A group-wise contrastive loss is then applied to force the whole text representation close to its own random spans while far away from other groups in the same mini-batch. The detailed pre-training procedures are as follows.

**Multi-granularity Span Sampling.** Different granularities of span capture different properties of the input text. For example, fine-grained spans can capture specific words or entities mentioned in the text. Coarser-grained spans can instead capture more abstract properties of the text. In this work, to better capture the semantic information, we explicitly sample a set of spans at several levels of granularity for each input text, including word-level, phrase-level, sentence-level and paragraph-level. Given a tokenized document with  $n$  words  $d = (x_1, x_2, \dots, x_n)$ , the detailed span sampling process is shown as follows.

- (1) For each level of granularity, we first sample the span length from a beta distribution following [15], i.e.,

$$p_{span} \sim \text{Beta}(\alpha, \beta),$$

$$\ell_{span} = p_{span} * (\ell_{max} - \ell_{min}) + \ell_{min}, \quad (1)$$

where  $\ell_{min}$  and  $\ell_{max}$  denote the minimum length and maximum span length of each level of granularity.  $\alpha$  and  $\beta$  are two hyperparameters.

- (2) We then randomly sample the starting position  $start$  with respect to

$$start \sim U(1, n - \ell_{span}). \quad (2)$$

The span is finally determined by the start position  $start$  and span length  $\ell_{span}$ :

$$end = start + \ell_{span},$$

$$span = [x_{start}, \dots, x_{end-1}]. \quad (3)$$

We only store the starting and end positions for each span, not the raw span text. Note that for the word-level span, we sample a whole word (without tokenized) and filter out the stopwords.

We sample  $T$  spans for each level of granularity and repeat the above process for all the granularity to obtain the final  $4T$  spans with respect to each input text.

**Text Encoding.** To obtain the representations of the whole text and its multiple spans, we adopt the multi-layer Transformer encoder architecture as the text encoder.

Specifically, a special token  $x_0 = [\text{CLS}]$  is added in front of the input to represent the whole text like in BERT: ( $d = x_0, x_1, x_2, \dots, x_n$ ). The Transformer encoder maps each word in the  $d$  to low-dimensional dense representations, i.e.,

$$\mathbf{h}_0, \mathbf{h}_1, \dots, \mathbf{h}_n = \text{Transformer}(x_0, x_1, \dots, x_n), h_i \in \mathbb{R}^H, \quad (4)$$

where  $H$  is the hidden size.

Previous works [3, 4, 16] have suggested that the additional nonlinear projection is critical to prevent representation collapse for contrastive learning. Therefore, for the whole text representation, we feed it into a projector network  $projector(\cdot)$ , which is a feed-forward neural network (FFN) with a non-linear activation, i.e.,

$$projector(\cdot) = \text{Tanh}(\text{FFN}(\cdot)). \quad (5)$$

Therefore, the whole text representation is transformed into  $\mathbf{z}_0$  by,

$$\mathbf{z}_0 = projector(\mathbf{h}_0). \quad (6)$$

For the spans in the text  $d$ , we conduct the average pooling operation over the output word representations to obtain the span representations  $\{\mathbf{z}_i\}_{i=1}^{4T}$ . For example, if the start position and the end position of a span are  $start$  and  $end$  respectively, the span representation is obtained as follows:

$$\mathbf{z}_i = \text{AvgPooling}([\mathbf{h}_{start}, \dots, \mathbf{h}_{end-1}]). \quad (7)$$

**Group-wise Contrastive Loss.** Based on the representations of each text and its multi-granularity span representations, we apply a group-wise contrastive loss to pre-train the text encoder. The group-wise contrastive loss aims to pull the text representation close to the instances in its group, and push it away from the representations from other groups. The classical contrastive loss function is incapable of handling such a case where multiple examples instead of one example are denoted as positive to one instance [22].

---

### Algorithm 1: Pre-training COSTA

---

```

Initialize COSTA model parameters  $\Theta$ .
The pre-training corpus  $\mathcal{D}$ .
Set the number of spans sampled per granularity:  $T$ .
Set the max number of epoch:  $epoch_{max}$ .

// multi-granularity span sampling
for  $d$  in  $\mathcal{D}$  do
  Sample  $4T$  spans for  $d$ , w.r.t Eq.(1),Eq.(2),Eq.(3)
  Pack the span indexes (start, end) into  $\mathcal{D}_s$ .
end

// model pre-training
for epoch in  $1, 2, \dots, epoch_{max}$  do
  for mini-batch  $b$  in  $\mathcal{D}$  do
    // encode texts into dense representations
     $\{\mathbf{h}_0, \mathbf{h}_1, \dots\}_0^{|\mathcal{b}|} = \text{Eq.}(4)$ 
    // obtain the whole text representations
     $\{\mathbf{z}_0\}_0^{|\mathcal{b}|} = \text{Eq.}(6)$ 
    // obtain the span representations
     $\{(start, end) * 4T\}_0^{|\mathcal{b}|} = \mathcal{D}_s(b)$ 
     $\{\mathbf{z}_i * 4T\}_0^{|\mathcal{b}|} = \text{Eq.}(7)$ 
    // Compute loss:  $L(\Theta)$ 
     $L(\Theta) = \text{Eq. } 8$  for contrastive span prediction task
     $L(\Theta) = \text{Eq. } 9$  for masked language modeling task
    Update model parameters  $\Theta$ 
  end
end

```

---

Specifically, given a mini-batch with  $N$  input texts ( $d_1, \dots, d_N$ ), we can obtain  $N$  whole text representations and  $N * 4T$  span representations, which form a total of  $N * (4T + 1)$  representations. Let  $S(i)$  be the index set of spans from  $d_i$ , and the size of  $S(i)$  equals  $4T$ . The group-wise contrastive loss function  $\mathcal{L}_{GWC}$  is defined as,

$$\mathcal{L}_{GWC} = \sum_{i=1}^N -\frac{1}{4T} \sum_{p \in S(i)} \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_p)/\tau)}{\sum_{j=1}^{N*(4T+1)} \mathbb{1}_{[i \neq j]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}, \quad (8)$$

where  $\text{sim}(\cdot)$  is the dot-product function and  $\tau$  is the temperature hyperparameter.

### 3.3 Masked Language Modeling

The MLM [36] task first randomly masks out some tokens from the input and then trains the model to predict the masked tokens based on the rest of the tokens. As discussed in previous works [30], the MLM objective could in general contribute to building good contextual text representations for IR, thus it would help COSTA to learn good span representations. Therefore, similar to BERT, we also adopt MLM as one of the pre-training objectives besides the contrastive span prediction objective.

Specifically, the MLM loss  $\mathcal{L}_{MLM}$  is defined as:

$$\mathcal{L}_{MLM} = - \sum_{\hat{x} \in m(\mathbf{x})} \log p(\hat{x} | \mathbf{x}_{\setminus m(\mathbf{x})}), \quad (9)$$

where  $\mathbf{x}$  denotes the input sentences,  $m(\mathbf{x})$  and  $\mathbf{x}_{\setminus m(\mathbf{x})}$  denotes the masked words and the rest words from  $\mathbf{x}$ , respectively.

We then pre-train the Transformer-based encoder using the group-wise contrastive loss in Eq. (8) jointly with the MLM loss in Eq. (9), as follows,

$$\mathcal{L}_{total} = \lambda \mathcal{L}_{GWC} + \mathcal{L}_{MLM},$$

where  $\lambda$  is the hyperparameter.

## 4 EXPERIMENTAL SETTINGS

We first introduce our experimental settings, including pre-training corpus, downstream tasks, baseline methods, evaluation metrics, and implementation details.

### 4.1 Research Questions

We conduct experiments to verify the effectiveness of COSTA. Specifically, we target the following research questions:

- **RQ1:** How does COSTA perform compared with baselines using the same fine-tuning strategy?
- **RQ2:** How does COSTA perform compared with advanced dense retrieval models using complicated fine-tuning strategies?
- **RQ3:** How do different components of COSTA affect the retrieval performance?
- **RQ4:** How about the discriminative ability of COSTA?
- **RQ5:** How does COSTA perform under the low-resource setting?

### 4.2 Pre-training Corpus and Downstream Tasks

We use the English Wikipedia as our pre-training corpus following the existing work [8, 29–31].

- **Wikipedia** is a widely used corpus that contains millions of well-formed Wiki-articles.

We fine-tune COSTA on several standard dense retrieval benchmarks.

- **MS MARCO Passage Ranking (MARCO Dev Passage)** [32] is a large-scale benchmark dataset for web passage retrieval, with about 0.5 million training queries and 8.8 million passages.
- **MS MARCO Document Ranking (MARCO Dev Doc)** [32] is another large-scale benchmark dataset for web document retrieval, with about 0.4 million training queries and 3 million documents.
- **TREC 2019 Passage Ranking (TREC2019 Passage)** [5] replaces the test set in the MS MARCO passage ranking task with a novel set produced by TREC with more comprehensive labeling.
- **TREC 2019 Document Ranking (TREC2019 Document)** [5] replaces the test set in the MS MARCO document ranking task with a novel set produced by TREC with more comprehensive labeling.

### 4.3 Baselines

We adopt three types of baselines for comparison, including sparse retrieve models, other pre-trained models, and advanced dense retrieval models that fine-tune the existing pre-trained models using complicated techniques.

*4.3.1 Traditional Sparse Retrieval Models.* For traditional retrieval models, we consider taking BM25 and DeepCT as the baselines,

and we also report several representative results according to the TREC overview paper [5].

- **BM25** [34] is a highly effective strong baseline model that represents the traditional sparse retrieval models.
- **DeepCT** [6, 7] is a neural term weighting framework that learns to map BERT’s contextualized text representations to context-aware term weights for sentences and passages.

*4.3.2 Other Pre-training Models.* Other pre-training methods include the general-purpose pre-trained language model BERT and other pre-trained models tailored for IR:

- **BERT** [8] is the dominant pre-trained model which achieves great success on various language understanding tasks. BERT is pre-trained with MLM and Next Sentence Prediction(NSP) tasks using the Transformer encoder.
- **ICT** [2] is specifically designed for passage retrieval in QA scenario. ICT randomly samples a sentence from a passage in a Wiki page and takes the rest sentences in the passage as its positive context.
- **PROP** [30] is designed for ad-hoc retrieval which pre-trains the Transformer model with representative words prediction task(ROP) and MLM. ROP predicts the pairwise preference for the sampled pairs of word sets from the unigram language model.
- **B-PROP** [31] improves PROP by replacing the unigram language model with a contrastive term distribution obtained from the [CLS]-token attention of BERT.
- **SEED** [29] pre-trains the autoencoder using a weak decoder for dense retrieval. SEED is pre-trained with the reconstruction task and MLM on English Wikipedia.

*4.3.3 Advanced Dense Retrieval Models that Fine-tunes with Complicated Techniques.* As many works have studied various complicated fine-tuning methods to enhance dense retrieval models, we thus also compare COSTA fine-tuned using simple strategies with those models. These models include:

- **ANCE**[38] investigated the hard negative mining problem where they periodically refresh its corpus index to retrieve the hard negatives produced by the current model.
- **TCT-ColBERT**[27] distills from ColBERT’s MaxSim operator into a simple dot-product operation for computing relevance scores.
- **TAS-B**[18] proposes to train dense retrieval models with a query topic-aware and balanced margin of passage pairs sampling strategy.
- **ADORE+STAR**[40] is built on the STAR model which has mined one-round hard negatives besides BM25 negatives to train a strong document encoder. ADORE then retrieves the hard negatives from the pre-build STAR document index in real-time and only optimizes the query encoder.
- **RocketQA**[33] utilizes three training strategies, namely cross-batch negatives, denoised hard negatives, and data augmentation, to improve the dense retrieval model.

### 4.4 Evaluation Metrics

We use the official metrics of these four benchmarks. For the MS MARCO passage ranking task, we report the Mean Reciprocal Rank

at 10 (MRR@10) and recall at 1000 (R@1000). For the MS Document ranking task, we report the MRR@100 and R@100. For two TREC tasks, we report normalized discounted cumulative gain at 10 (NDCG@10), and R@100 and R@1000 for passage ranking and document ranking, respectively.

## 4.5 Implementation Details

Here, we describe the implementation of pre-training procedures, and fine-tuning procedures in detail.

**4.5.1 Pre-training Procedures.** In this section, we describe the multi-granularity span sampling and model pre-training in detail.

**Multi-granularity Span Sampling.** In the span sampling phase, for phrase-level span,  $\ell_{min}$  and  $\ell_{max}$  are set to 4 and 16, respectively. For sentence-level span,  $\ell_{min}$  and  $\ell_{max}$  are set to 16 and 64, respectively. For paragraph-level span,  $\ell_{min}$  and  $\ell_{max}$  are set to 64 and 128, respectively. The  $\alpha$  and  $\beta$  hyperparameter in beta distribution is set to 4 and 2, respectively, which skews sampling towards longer spans. We sample 5 spans for each granularity per input text.

**Model Pre-training.** Our text encoder uses the same model architecture as BERT [8]. Considering the large computational cost of pre-training from scratch, we initialize the encoder from the official BERT’s checkpoint and only learn the projector from scratch. We use a learning rate of  $5e-5$  and Adam optimizer with a linear warm-up technique over the first 10% steps. We pre-train on Wikipedia for 6 epochs. The long input documents are truncated to several chunks with a maximum length of 512. The hyper-parameter of  $\tau$  in Eq. 8 is set to 0.1. The hyper-parameter of  $\lambda$  in Eq. 10 is set to 0.1. We pre-train COSTA on up to four NVIDIA Tesla V100 32GB GPUs.

For BERT and SEED-Encoder, we directly use the open-sourced models as both of them are pre-trained on English Wikipedia and BookCorpus [42]. For ICT, we use the same pre-training procedures as COSTA and pre-train it with MLM and ICT tasks. We pre-train PROP and B-PROP using a bi-encoder architecture for a fair comparison.

**4.5.2 Fine-tuning Procedures.** For the passage ranking datasets and the document ranking datasets, we use a similar but not identical fine-tuning strategy. Following STAR [40], we use a two-stage strategy to fine-tune downstream dense retrieval tasks with Tevatron toolkit [13].

**Fine-tuning Passage Ranking Datasets.** For the two passage ranking datasets including MARCO Dev Passage and TREC2019 Passage, we train COSTA with official BM25 negatives first for 3 epochs, and then mine hard negatives of the BM25 warm-up model to continue training 2-3 epochs. Note that we only mine the hard negatives from the BM25 warm-up model once. The query length and the passage length are set to 32 and 128 respectively. We use a learning rate of  $5e-6$  and a batch size of 64.

**Fine-tuning Document Ranking Datasets.** For the two document ranking datasets on MARCO Dev Doc and TREC2019 Doc, following existing works [29, 38, 40], we use the model fine-tuned on the passage ranking task as the starting point. Since the fine-tuned COSTA is too strong, continuing fine-tuning this model with official BM25 negatives decreases the performance greatly on the document ranking datasets. We thus iteratively mine the static hard

**Table 1: Comparisons between COSTA and the baselines on the two passage ranking datasets. Two-tailed t-tests demonstrate the improvements of COSTA to the baselines are statistically significant ( $p \leq 0.05$ ). \* indicate significant improvements over BERT. † indicate significant improvements over ICT, PROP, and B-PROP. And ‡ indicate significant improvements over SEED. Results not available or not applicable are marked as ‘-’.**

Model	MARCO Dev Passage		TREC2019 Passage	
	MRR@10	R@1000	NDCG@10	R@1000
Sparse retrieval models				
BM25	0.187	0.857	0.501	0.745
DeepCT[6]	0.243	0.905	0.551	-
Best TREC Trad[5]	-	-	0.554	-
Fine-tuning with official BM25 negatives				
BERT	0.316	0.941	0.616	0.704
ICT	0.324	0.938	0.618	0.705
PROP	0.320	0.948	0.586	0.709
B-PROP	0.321	0.945	0.603	0.705
SEED[29]	0.329	0.953	-	-
SEED(ours)	0.331*	0.950*	0.625*	0.733*†
COSTA	<b>0.342*†‡</b>	<b>0.959*†</b>	<b>0.635*†‡</b>	<b>0.773*†‡</b>
Fine-tuning with static hard negatives				
BERT	0.335	0.957	0.661	0.769
ICT	0.339	0.955	0.670	0.775
PROP	0.337	0.951	0.673	0.771
B-PROP	0.339	0.952	0.672	0.774
SEED	0.342*	0.963	0.679*	0.782*†
COSTA	<b>0.366*†‡</b>	<b>0.971*†</b>	<b>0.704*†‡</b>	<b>0.816*†‡</b>

negatives twice and only fine-tune 1 epoch on the static hard negatives for each iteration. We use a learning rate of  $5e-6$  and a batch size of 64. The document length is truncated to the first 512 tokens.

We pair each positive example with 7 negative examples for all these 4 datasets. All fine-tuning procedures use Adam optimizer with a linear warm-up technique over the first 10% steps.

## 5 EXPERIMENT RESULTS

In this section, we analyze the experimental results to demonstrate the effectiveness of the proposed COSTA method.

### 5.1 Baseline Comparison with the Same Fine-tuning Strategy

To answer RQ1, we compare COSTA with various baselines on four benchmark datasets. For a fair comparison, different pre-trained models leverage the same fine-tuning strategy. The performance comparisons are shown in Table 1 and Table 2.

**Results on the two passage ranking datasets.** As shown in Table 1, we have the following observations: (1) The relative order of different models fine-tuned with BM25 negatives is quite

**Table 2: Comparisons between COSTA and the baselines on two document ranking datasets. Two-tailed t-tests demonstrate the improvements of COSTA to the baselines are statistically significant ( $p \leq 0.05$ ). \* indicates significant improvements over BERT. † indicates significant improvements over ICT, PROP, and B-PROP. ‡ indicates significant improvements over SEED. Results not available or not applicable are marked as ‘-’.**

Model	MARCO Dev Doc		TREC2019 Doc	
	MRR@100	R@100	NDCG@10	R@100
Sparse retrieval models				
BM25	0.277	0.808	0.519	0.395
DeepCT[6]	0.320	-	0.544	-
Best TREC Trad[5]	-	-	0.549	-
1st iteration: Fine-tuning with static hard negatives				
BERT	0.358	0.869	0.563	0.266
ICT	0.364	0.873	0.566	0.273
PROP	0.361	0.871	0.565	0.269
B-PROP	0.365	0.871	0.567	0.268
SEED	0.372*	0.879*	0.573*	0.272
COSTA	<b>0.395*†‡</b>	<b>0.894*†‡</b>	<b>0.582*†‡</b>	<b>0.278*</b>
2nd iteration: Fine-tuning with static hard negatives				
BERT	0.389	0.877	0.594	0.301
ICT	0.396	0.882	0.605	0.303
PROP	0.394	0.884	0.596	0.298
B-PROP	0.395	0.883	0.601	0.305
SEED	0.396	0.902*	0.605*	0.307
COSTA	<b>0.422*†‡</b>	<b>0.919*†‡</b>	<b>0.626*†‡</b>	<b>0.320*†‡</b>

consistent with that fine-tuned with static hard negatives. (2) The dense retrieval models outperform all the traditional sparse retrieval models by a large margin. It indicates that by capturing the semantics meanings of queries and documents, dense retrieval models can overcome the vocabulary mismatch problem caused by sparse retrieval models for better ranking performance. (3) PROP and B-PROP show slight improvements over BERT, indicating that the ROP task is not as effective in the retrieval stage as that in the re-ranking stage. The reason might be that the ROP task only considers the inside information of a document (i.e., it samples words from the corresponding document language model), leading to the unawareness of the outside information (i.e., other documents). (4) Although ICT utilizes the in-batch negatives, taking a random sentence as the pseudo query can only learn topic-level relevance which is insufficient for dense retrieval tasks. (5) SEED performs better than BERT, PROP, B-PROP and ICT significantly indicating that by enforcing a bottleneck on the data, SEED can provide better semantic representations for dense retrieval.

**Table 3: Comparison between COSTA and advanced dense retrieval models using complicated fine-tuning strategies on the MARCO Dev Passage. Best results are marked bold.**

Model	MRR@10	R@1000
ANCE[40]	0.330	0.959
TCT-ColBERT[27]	0.335	0.964
TAS-B[18]	0.343	<b>0.976</b>
ADORE+STAR[40]	0.347	-
RoctetQA w/o Data Aug [33]	0.364	-
COSTA	<b>0.366</b>	0.971

When we look at COSTA, we find that COSTA achieves significant improvements over all the baselines. (1) COSTA can outperform SEED significantly, demonstrating dropping the decoder and utilizing the contrastive strategy help it to produce better text representations. (2) Compared with ICT, PROP and B-PROP, COSTA is a unified task since the group-wise contrastive span prediction task teaches the model not only to learn better representations from the input itself but also to be distinguishable from other representations.

**Results on the two document ranking datasets.** As shown in Table 2, the performance trend on the document ranking datasets is consistent with the passage ranking datasets except for the TREC2019 Doc dataset. For the TREC2019 Doc, dense retrieval models significantly underperform BM25 in terms of R@1000. This phenomenon is also observed in other works [5, 29, 38, 40]. A possible reason is that MS MARCO has many unlabeled relevant documents in the corpus, but the model often treats these documents as negatives. So this problem would bias the process of model training and thus hurt its performance. Finally, we can see that COSTA achieves more improvement in terms of ranking metrics like MRR and NDCG than the recall metric. For example, COSTA improves SEED’s top-ranking performance by 7% on both the MS MARCO passage ranking dataset and document ranking dataset in terms of MRR metric. It shows that COSTA has a better discriminative ability that can rank the positive document higher.

## 5.2 Baseline Comparison with Different Fine-tuning Strategies

To answer RQ2, we further compare COSTA with advanced dense retrieval models that fine-tune existing pre-trained models with complicated strategies. As shown in Table 3, we can observe that surprisingly, fine-tuning with simple strategies COSTA performs better than these advanced dense retrieval models with complicated fine-tuning strategies. Some dense retrieval models are also based on more powerful pre-trained models like RoBERTa [34] and ERNIE [35]. The better results indicates that the performance improvement of COSTA mainly comes from our pre-training stage. This demonstrates the effectiveness of our proposed contrastive span prediction task. Undoubtedly, these fine-tuning methods can also be applied to our COSTA, and we leave this for further study.

**Table 4: The performance of COSTA with different span granularities. Best results are marked bold.**

Method	MRR@10	R@1000
Base	<b>0.335</b>	<b>0.952</b>
w/o word-level	0.334	0.952
w/o phrase-level	0.331	0.953
w/o sentence-level	0.331	0.947
w/o paragraph-level	0.326	0.940

### 5.3 Breakdown Analysis

To answer RQ3, in this section, we ablate several components of the span sampling procedure, the projector network in COSTA, and the temperature hyperparameter in the loss function. All reported results in this section are based on the MS MARCO passage ranking development set trained with only BM25 negatives. The default experiment setting is that we sample 3 spans for each level of granularity, use a projector to transform the whole text representations and the temperature is set to 1. We pre-train the model using the default settings on Wikipedia for 3 epochs (half of our full pre-training schedule).

**5.3.1 Impact of the Span Granularity.** We first study the impact of span granularity in the span sampling procedure. In order to control the total number of spans sampled per input text unchanged, we set it to 12. That is, when we sample 3 of 4 span granularity, 4 spans are sampled for each of the other three granularity.

Table 4 shows the impact of the different span granularity. We can see that all the granularity contributes to the whole text representation learning. But with the increase of span length, i.e., from word-level to paragraph-level, the performance decreases more and more. For example, without the word-level span, it leads to a slight decrease in terms of MRR@10 while without the paragraph-level span, it leads to a significant decrease. This may be because longer spans contain more information of the input and thus contribute more to the text representation learning. Shorter spans like phrase-level and sentence-level are also helpful as they are useful for capturing fine-grained information of the input texts.

**Table 5: Performance comparison of COSTA with different span numbers. Best results are marked bold.**

Span Number	3	5	10	20
MRR@10	0.335	<b>0.339</b>	0.332	0.320
R@1000	0.952	<b>0.953</b>	0.949	0.946

**5.3.2 Impact of the Span Number.** We then study the impact of the number of spans sample per input text. Given a mini-batch containing  $N$  input texts, we sample  $T$  spans for each granularity which results in  $N * 4T$  spans.

As shown in Table 5, we varies  $T$  from 3 to 20. We can see that with more spans sampled from the input texts, the performance doesn't improve consistently. When  $T = 5$ , it achieves the best performance across other settings. We hypothesize this is because

**Table 6: Performance comparison of COSTA with different projector architectures in the pre-training and fine-tuning phase. Best results are marked bold.**

Setting	Pre-training		Fine-tuning	
	MRR@10	R@1000	MRR@10	R@1000
w/ nonlinear Proj	0.335	0.952	0.335	0.951
w/ linear Proj	0.332	0.950	0.334	0.952
w/o	0.327	0.944	0.335	0.952

the difficulty of the group-wise contrastive objective increases especially at the beginning of training. Recall that the loss function in Eq.(8) forces the whole text representations close to their random spans, when  $T = 20$ , COSTA needs to align the whole text representations with  $4 * T = 80$  span representations. So the model may be hard to learn as the text representations are distributed randomly at the beginning. The difficulty is also observed from the loss curve where the loss decreases very slowly. Other learning techniques like curriculum learning [1] are left for further studies to alleviate this problem. The best span number sampled per granularity in our experiment is  $T = 5$ .

**Table 7: Performance comparison of COSTA with different temperatures  $\tau$  on MS MARCO Dev Passage. Best results are marked bold.**

$\tau$	10	1	0.1	0.01
MRR@10	0.274	0.329	0.335	0.267

**5.3.3 Impact of the Projector Architecture.** Existing work in contrastive learning shows that introducing a learnable nonlinear transformation between the representation and the contrastive loss substantially improves the quality of the learned representations [3, 4, 16]. We thus study three different architectures for the projector in the pre-training phase and the fine-tuning phase respectively: nonlinear MLP, linear MLP, and identity mapping (i.e., without MLP). When studying the pre-training stage, we discard the projector in the fine-tuning stage. When studying the fine-tuning stage, we use the model pre-trained with a nonlinear projector network.

The performance comparison is shown in Table 6. We can observe that in the pre-training phase, using a nonlinear projection perform better than using a linear projection, and much better than no projection. One possible reason is that introducing a nonlinear projection ensures that the whole text representation is not equal to any span representation since they are output from the same layer, and also makes it easy to align with multiple span representations simultaneously. We also studied the impact of the projector in the fine-tuning stage. For the fine-tuning stage, there is no big difference when fine-tuning the model with or without the projector network. It indicates that the pre-trained model is strong enough to learn high-quality text representations given the task labels for dense retrieval.



5.3.4 *Impact of the Temperature in Loss Function.* The temperature  $\tau$  in the contrastive loss function Eq. 8 is used to control the smoothness of the distribution normalized by softmax operation and thus influences the gradients when backpropagation. A large temperature smooths the distribution while a small temperature sharpens the distribution. As shown in Table 7, we find the performance is sensitive to the temperature. Either too small or too large temperature will make our model perform badly. It might be that a smaller value would make the model too hard to converge while a larger one leads to a trivial solution. We select 0.1 as the temperature in our COSTA pre-training.

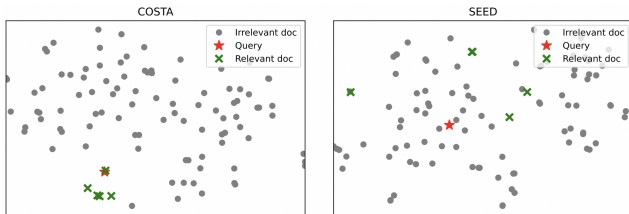


Figure 3: The t-SNE plot of query and document representations for SEED and COSTA. The QID is 47923 and is from TREC2019 Passage test set.

## 5.4 Visual Analysis

To answer RQ4, we visualize the query and text representations using t-SNE to see their distributions in the semantic space. We conduct this analysis on the MARCO Dev Passage dataset.

In detail, we first fine-tune COSTA and SEED with a very limited number of queries, i.e., 1000, on MS MARCO Dev Passage dataset. We then plot a t-SNE example using the sampled query and its top-100 candidate documents. Results in Figure 3 show that COSTA maps the relevant document in the semantic space closer while far away from others. For SEED, the distribution of relevant documents in the latent space is relatively random. This is because that SEED only reconstructs the input texts and treats all tokens equally which is unable to learn discriminative representations. This demonstrates that by pre-training with the contrastive span prediction task, COSTA can generate discriminative dense representations compared with SEED.

## 5.5 Low-Resource Setting

To answer RQ5, we simulated a low-resource setting on the MS MARCO passage ranking dataset and TREC2019 passage ranking dataset. We randomly sample different fixed limited numbers of queries from the original training set following the existing work [14]. Specifically, we choose to sample 500, 1000, 2000, and 5000 queries for this experiment. And to improve the stability, we sample 3 different splits for each setting and report the average performance for each setting. We fine-tune COSTA and SEED with batch size as three different values (i.e., 4, 8, 16), learning rate as two different values (i.e.,  $5e-6$ ,  $1e-5$ ), and pick the last checkpoint to evaluate the original large development set.

As shown in Table 4, we can see that: (1) COSTA performs significantly better than SEED on these two datasets in terms of all metrics, indicating that COSTA is able to learn more discriminative

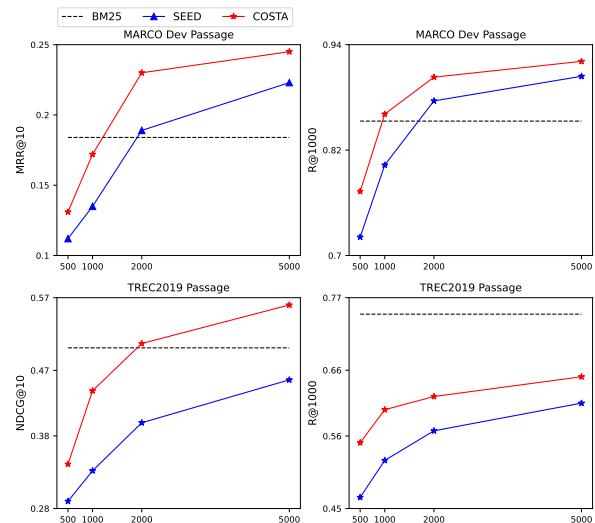


Figure 4: Fine-tuning with limited supervised data. The x-axis indicates the number of training queries.

text representations than SEED. (2) Fine-tuning a limited number of training queries, COSTA can outperform the strong BM25 baseline. For example, COSTA outperforms BM25 with only less than 0.5% queries, i.e., 2000 queries, on MARCO Dev Passage in terms of both MRR@10 and R@1000. (3) COSTA performs worse on the TREC2019 Passage dataset since COSTA is fine-tuned on MARCO Dev Passage which suffers the false-negative problem [33].

## 6 CONCLUSION

In this paper, we proposed a novel contrastive span prediction task to pre-train a discriminative text encoder for dense retrieval. By enhancing the consistency between representations of the original text and its own spans while pushing it away from representations of other groups, COSTA can leverage the merits of both the bottleneck principle and discriminative ability for better representation quality. The bottleneck principle guarantees the representation can represent itself through “reconstruction” and the contrastive strategy can help to make the representation distinguishable across other texts. Through experiments on 4 benchmark dense retrieval datasets, COSTA outperforms several strong baselines. Through visualization analysis and the low-resource setting, we demonstrate that COSTA can produce discriminative representations for dense retrieval. In future work, we would like to apply COSTA to other IR scenarios like open-domain question answering and conversational systems.

## ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62006218, 61902381, and 61872338, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2021100, the Lenovo-CAS Joint Lab Youth Scientist Project, and the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059).

## REFERENCES

- [1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICM09* ACM Press. <https://doi.org/10.1145/1553374.1553380>
- [2] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* OpenReview.net.
- [3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. 119. PMLR, 1597–1607.
- [4] Xinlei Chen and Kaiming He. 2021. Exploring Simple Siamese Representation Learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2021, 15745–15753.
- [5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2020 Deep Learning Track. *ArXiv abs/2102.07662* (2020).
- [6] Zhuyun Dai and Jamie Callan. 2019. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. *ArXiv abs/1910.10687* (2019).
- [7] Zhuyun Dai and Jamie Callan. 2020. Context-Aware Document Term Weighting for Ad-Hoc Search. *Proceedings of The Web Conference* 2020.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. In *Proceedings of the 2019 Conference of the Association for Computational Linguistics*. <https://doi.org/10.18653/v1/n19-1423>
- [9] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, and Yiqun Liu. 2021. Pre-training Methods in Information Retrieval. *CoRR* abs/2111.13853 (2021). [arXiv:2111.13853](https://arxiv.org/abs/2111.13853) <https://arxiv.org/abs/2111.13853>
- [10] Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2022. Neural Approaches to Conversational Information Retrieval. *ArXiv abs/2201.05176* (2022).
- [11] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *EMNLP*
- [12] 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* Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.emnlp-main.75>
- [13] Luyu Gao, Xueguang Ma, Jimmy J. Lin, and Jamie Callan. 2022. Tevatron: An Efficient and Flexible Toolkit for Dense Retrieval. *ArXiv abs/2203.05765* (2022).
- [14] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*
- [15] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.acl-long.72>
- [16] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhao-han Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. *ArXiv abs/2006.07733* (2020).
- [17] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)* (2022), 1–42.
- [18] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy J. Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*
- [19] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *ICLR*
- [20] 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 (EMNLP)* Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-main.550>
- [21] O. Khattab and Matei A. Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).
- [22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*
- [23] Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning Dense Representations of Phrases at Scale. In *ACL/IJCNLP*
- [24] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* Association for Computational Linguistics. <https://doi.org/10.18653/v1/p19-1612>
- [25] Chunyuan Li, Xiang Gao, Yuan Li, Xiujuan Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In *EMNLP*
- [26] Jimmy J. Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (2021).
- [27] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 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)* Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.repl4nlp-1.17>
- [28] 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. *ArXiv abs/1907.11692* (2019).
- [29] 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* Association for Computational Linguistics, 2780–2791.
- [30] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (2021).
- [31] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021. B-PROP: Bootstrapped Pre-training with Representative Words Prediction for Ad-hoc Retrieval. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).
- [32] 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 *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*
- [33] Yinqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.naacl-main.466>
- [34] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr* (2009), 333–389.
- [35] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *ArXiv preprint arXiv:1904.09220* (2019).
- [36] Wilson L. Taylor. 1953. “Cloze Procedure”: A New Tool for Measuring Readability. *Journalism & Mass Communication Quarterly* (1953), 415 – 433.
- [37] Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, 1–5.
- [38] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *ArXiv abs/2007.00808* (2021).
- [39] Shih Yuan Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-Shot Conversational Dense Retrieval. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).
- [40] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).
- [41] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. RepBERT: Contextualized Text Embeddings for First-Stage Retrieval. *ArXiv abs/2006.15498* (2020).
- [42] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. *2015 IEEE International Conference on Computer Vision (ICCV)*, 19–27.
- [43] Yunchang Zhu, Liang Pang, Yanyan Lan, Huawei Shen, and Xueqi Cheng. 2021. Adaptive Information Seeking for Open-Domain Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* Association for Computational Linguistics, 3615–3626.