

Text Embeddings by Weakly-Supervised Contrastive Pre-training

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao
 Linjun Yang, Daxin Jiang, Rangan Majumder, Furu Wei
 Microsoft Corporation
<https://github.com/microsoft/unilm>

Abstract

This paper presents E5¹, a family of state-of-the-art text embeddings that transfer well to a wide range of tasks. The model is trained in a contrastive manner with weak supervision signals from our curated large-scale text pair dataset (called CCPairs). E5 can be readily used as a general-purpose embedding model for any tasks requiring a single-vector representation of texts such as retrieval, clustering, and classification, achieving strong performance in both zero-shot and fine-tuned settings. We conduct extensive evaluations on 56 datasets from the BEIR and MTEB benchmarks. For zero-shot settings, E5 is the first model that outperforms the strong BM25 baseline on the BEIR retrieval benchmark without using any labeled data. When fine-tuned, E5 obtains the best results on the MTEB benchmark, beating existing embedding models with 40× more parameters.

1 Introduction

Text embeddings are low-dimensional vector representations for arbitrary-length texts and play key roles in many NLP tasks such as large-scale retrieval. Compared to the high-dimensional and sparse representations like TF-IDF, text embeddings have the potential to overcome the lexical mismatch issue and facilitate efficient retrieval and matching between texts. It also offers a versatile interface easily consumable by downstream applications.

While pre-trained language models such as BERT [17] and GPT [7] can produce transferrable text representations, they are not ideal for tasks such as retrieval and text matching where a single-vector embedding of texts is more desired due to its efficiency and versatility. To obtain better text embeddings, contrastive learning is often the go-to framework to enhance the sequence-level representations from text pairs. Along this line of research, some works are geared towards learning task-specific embeddings. For example, GTR [43] and Sentence-T5 [44] fine-tune pre-trained models with supervised datasets to learn embeddings customized for passage retrieval and semantic textual similarity, respectively. Other works learn unsupervised embeddings from automatically constructed text pairs. Typical methods to construct text pairs include Inverse Close Task (ICT) [9], random cropping [28] and neighboring text spans [41], etc. While such synthetic data are of unlimited quantity, they are often poor in quality and the resulted embeddings fail to match the performance of the classic BM25 baseline without further fine-tuning [40].

In this work, we learn a high-quality general-purpose text embedding termed E5, **EmbEddings** from **bidirectional Encoder rEpresentations**. E5 aims to provide strong off-the-shelf text embeddings suitable for any tasks requiring single-vector representations in both zero-shot or fine-tuned settings. To achieve this goal, instead of relying on limited labeled data or low-quality synthetic text pairs, we contrastively train E5 embeddings from CCPairs, a curated web-scale text pair dataset containing

¹E5: **EmbEddings** from **bidirectional Encoder rEpresentations**

heterogeneous training signals. We construct the CCPairs dataset by combining various semi-structured data sources such as CommunityQA, Common Crawl and Scientific papers, and perform aggressive filtering with a consistency-based filter [15] to improve data quality. We choose a simple contrastive learning recipe using in-batch negatives with a large batch-size to train our model. Extensive experiments on both BEIR and MTEB benchmarks demonstrate the effectiveness of the proposed method. On the BEIR zero-shot retrieval benchmark [53], E5 is the first model to outperform the strong BM25 baseline without using any labeled data. When fine-tuned on labeled datasets, the performance can be further improved. Results on 56 datasets from the recently introduced MTEB benchmark [40] show that our E5_{base} is competitive against GTR_{xxl} and Sentence-T5_{xxl}, which have 40× more parameters.

2 Related Work

There have been long-lasting interests in transforming texts into low-dimensional dense embeddings. Early works include Latent Semantic Indexing (LSA) [16] and Latent Dirichlet Allocation (LDA) [3]. LSA utilizes the decomposition of a word-document co-occurrence matrix to generate document embeddings, while LDA adopts probabilistic graphical models to learn topic distributions. Arora et al. show that a simple weighted average of word vectors [38] can be a strong baseline for sentence embeddings.

With the development of pre-trained language models [17, 35, 48] and large-scale labeled datasets such as SNLI [6] and MS-MARCO [8], methods like Sentence-BERT [49], SimCSE [22], Sentence-T5 [44] and SGPT [39] directly fine-tune language models to output continuous embeddings. Most research focuses on short texts and thus uses the term "sentence embeddings". For long documents, it remains an open research question whether fixed-length embeddings can encode all the information. Contrastive loss popularized by SimCLR [10] turns out to be more effective than classification-based losses [49, 14] for embeddings. LaBSE [20], LASER [2] and CLIP [47] further extend to multilingual and multi-modal scenarios using parallel sentences and image-text pairs.

Another direction is to design self-supervised pre-training tasks for text matching and retrieval. [9] proposes the well-known inverse cloze task (ICT), where a random sentence within a passage is chosen as a pseudo-query and the rest is treated as a positive sample. However, Contriever [28] shows that random cropping with data augmentation is more effective than ICT on a range of zero-shot information retrieval tasks. OpenAI text embeddings [41] use neighboring texts as positives and scale up the model size to 175B. Oguz et al. [45] performs domain-matched pre-training to improve in-domain results. SPAR [11] trains a dense retriever by treating BM25 as a teacher model. Although the aforementioned approaches can easily obtain abundant supervision signals, such synthetic data tend to be of low quality. Results on the BEIR benchmark [53] show they struggle to match the performance of BM25 if not further fine-tuned on labeled datasets.

Evaluation and interpretation of text embeddings are also non-trivial. Most benchmarks measure the embedding quality through downstream task performances. For example, SentEval [13] uses linear probing and a collection of semantic textual similarity (STS) datasets, while the BEIR benchmark [53] focuses on zero-shot information retrieval scenarios. The recently introduced MTEB benchmark [40] combines 56 datasets spanning across 8 tasks and 112 languages. Experiments show no model can achieve state-of-the-art results on all embedding tasks yet. In this paper, we do not use the SentEval toolkit since its linear probing setup depends on the optimization hyperparameters.

Most closely related to our work is a series of community efforts by *sentence-transformers*² to train embeddings with a collection of labeled and automatically collected datasets. In this paper, we show that it is possible to train high-quality embeddings using self-supervised pre-training only. In terms of benchmark results, our model can achieve superior performance when fine-tuned on less labeled data.

3 CCPairs: A Large Collection of Text Pair Dataset

The quality and diversity of the data is crucial for training general-purpose text embeddings. In this work, we mine and assemble CCPairs, a large high-quality text pair dataset from web sources which provide diverse training signals transferring well to a wide range of tasks.

²<https://github.com/UKPLab/sentence-transformers>

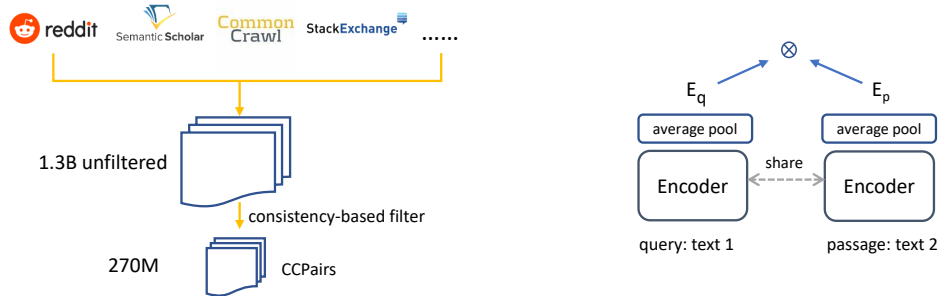


Figure 1: Overview of our data curation pipeline and model architecture.

Harvesting semi-structured data sources Large-scale high-quality datasets like C4 [48] and CCMatrix [51] are vital for the success of language model pre-training and machine translation. For learning text embeddings, existing works either utilize small-scale human-annotated data such as NLI [22] and MS-MARCO [8] or adopt heuristics such as random cropping [28] to obtain large-scale but very noisy supervision signals.

Instead, we curate a text pair dataset CCPairs (**Colossal Clean text Pairs**) by harvesting heterogeneous semi-structured data sources. Let (q, p) denote a text pair consisting of a query q and a passage p . Here we use “*passage*” to denote word sequences of arbitrary length, which can be a short sentence, a paragraph, or a long document. Our dataset includes (post, comment) pairs from Reddit³, (question, upvoted answer) pairs from Stackexchange⁴, (entity name + section title, passage) pairs from English Wikipedia, (title, abstract) and citation pairs from Scientific papers [36], and (title, passage) pairs from Common Crawl⁵ web pages and various News sources.

We only include data sources that can be automatically mined, and some subsets are directly reused from existing datasets. Simple heuristic rules are applied to filter data from Reddit and Common Crawl. For example, we remove Reddit comments that are either too long (> 4096 characters) or receive score less than 1, and remove passages from web pages with high perplexity [60]. After preliminary filtering, we end up with ~ 1.3 billion text pairs, most of which come from Reddit and Common Crawl. For more details and examples, please refer to Appendix A.

Consistency-based filter To further improve data quality and make training costs manageable, we propose a consistency-based data filtering technique: a model is first trained on the 1.3B noisy text pairs, and then used to rank each pair against a pool of 1 million random passages. A text pair is kept only if it falls in the top- k ranked lists. In other words, the model’s prediction should be consistent with the training labels. Here we set $k = 2$ based on manual inspection of data quality. After this step, we end up with ~ 270 M text pairs for contrastive pre-training.

The intuition for this technique comes from the memorization behaviors of neural networks [19]: when trained on noisy datasets, neural networks tend to memorize the clean labels first and then gradually overfit the noisy labels. Similar techniques [42, 15, 23] have been widely used for removing dataset noises. It is also possible to apply this filter iteratively, we will leave it for future work.

4 Method

Our embeddings can be trained with only unlabeled text pairs from CCPairs with contrastive pre-training. A second-stage fine-tuning on small, high-quality labeled datasets can be performed to further boost the quality of the resulted embeddings. See Figure 1 for an overview.

³<https://files.pushshift.io/reddit/>

⁴<https://archive.org/details/stackexchange>

⁵<https://commoncrawl.org/>

4.1 Contrastive Pre-training with Unlabeled Data

Contrastive pre-training aims to distinguish the relevant text pairs from other irrelevant or negative pairs. Given a collection of text pairs $\{(q_i, p_i)\}_{i=1}^n$, we assign a list of negative passages $\{p_{ij}^-\}_{j=1}^m$ for the i -th example. Then the InfoNCE contrastive loss [10] is as follows:

$$\min L_{\text{cont}} = -\frac{1}{n} \sum_i \log \frac{e^{s_{\theta}(q_i, p_i)}}{e^{s_{\theta}(q_i, p_i)} + \sum_j e^{s_{\theta}(q_i, p_{ij}^-)}} \quad (1)$$

where $s_{\theta}(q, p)$ is a scoring function between query q and passage p parameterized by θ . Following the popular biencoder architecture, we use a pre-trained Transformer encoder and average pooling over the output layer to get fixed-size text embeddings \mathbf{E}_q and \mathbf{E}_p . The score is the cosine similarity scaled by a temperature hyperparameter τ :

$$s_{\theta}(p, q) = \cos(\mathbf{E}_q, \mathbf{E}_p) / \tau \quad (2)$$

Where τ is set to 0.01 in our experiments by default. We use a shared encoder for all input texts and break the symmetry by adding two prefix identifiers “*query*:” and “*passage*:” to q and p respectively. For some data sources such as citation pairs, it is not obvious which side should be the query, we randomly choose one for simplicity. Such an asymmetric design turns out to be important for some retrieval tasks where there exist paraphrases of the query in the target corpus.

Another critical issue for contrastive training is how to select the negative samples. Here we choose to use the in-batch negatives [10], where the passages from other pairs in a batch serve as negative samples. We find that this simple strategy enables more stable training and outperforms methods such as MoCo [25] when the batch size is sufficiently large.

4.2 Fine-tuning with Labeled Data

While contrastive pre-training on the CCPairs provides a solid foundation for general-purpose embeddings, further training on labeled data can inject human knowledge into the model to boost the performance. Although these datasets are small, existing works [43, 44] have shown that supervised fine-tuning leads to consistent performance gains. In this paper, we choose to further train with a combination of 3 datasets: NLI⁶ (Natural Language Inference), MS-MARCO passage ranking dataset [8], and NQ (Natural Questions) dataset [30, 32]. Empirically, tasks like STS (Semantic Textual Similarity) and linear probing benefit from NLI data, while MS-MARCO and NQ datasets transfer well to retrieval tasks.

Building on the practices of training state-of-the-art dense retrievers [50, 58], we use mined hard negatives and knowledge distillation from a cross-encoder (CE) teacher model for the MS-MARCO and NQ datasets. For the NLI dataset, contradiction sentences are regarded as hard negatives. The loss function is a linear interpolation between contrastive loss L_{cont} for hard labels and KL divergence D_{KL} for distilling soft labels from the teacher model.

$$\min D_{\text{KL}}(p_{\text{ce}}, p_{\text{stu}}) + \alpha L_{\text{cont}} \quad (3)$$

Where p_{ce} and p_{stu} are the probabilities from the cross-encoder teacher model and our student model. α is a hyperparameter to balance the two loss functions. L_{cont} is the same as in Equation 1.

4.3 Applications to Text Embedding Tasks

After the above two steps, we obtain high-quality text embeddings transferring well to a wide range of tasks without fine-tuning the model parameters. Combined with techniques like approximate nearest neighbor search, embeddings provide a scalable and efficient solution for applications like web search. Here we briefly illustrate several use cases of our text embeddings.

Zero-shot Retrieval First, the passage embeddings for the target corpus are computed and indexed offline. Then for each query, we compute its query embedding and return the top- k ranked lists from the corpus based on cosine similarity.

Few-shot Text Classification A linear classifier is trained on top of the frozen embeddings with a few labeled examples. Different tasks only need to train and save the parameters of the classification heads. It can be seen as a particular form of parameter-efficient learning [27].

⁶The version released by SimCSE.

Zero-shot Text Classification The input and label texts are converted to sentences based on manually written prompt templates. The predicted label is the one closest to the input text in the embedding space. Take the sentiment classification of movie reviews as an example, with the original input “*I enjoy watching it*”, the label text is “*it is an example of terrible/great movie review*” and the input text becomes “*movie review: I enjoy watching it*”.

Semantic Textual Similarity Given two text embeddings, we use the cosine function to measure their semantic similarity. Since the absolute similarity scores do not enable an easy interpretation, the evaluation is usually based on rank correlation coefficients.

Text Clustering Standard clustering algorithms such as k-means can be applied straightforwardly. Texts belonging to the same category are expected to be close in the embedding space.

For tasks other than zero-shot text classification and retrieval, we use the query embeddings by default.

5 Experiments

5.1 Pre-training and Fine-tuning Configurations

Pre-training We pre-train on our proposed text pair dataset for three model sizes: $E5_{\text{small}}$, $E5_{\text{base}}$ and $E5_{\text{large}}$ initialized from MiniLM [59], *bert-base-uncased*, and *bert-large-uncased-whole-word-masking* respectively. The batch size is set to a large value of 32,768 to increase the number of negatives. The learning rate is $\{3, 2, 1\} \times 10^{-4}$ for the {small, base, large} models, with linear decay and the first 1,000 steps for warmup. We pre-train for 20k steps in total with AdamW optimizer, which is approximately 2.5 epochs over the dataset. It takes {16, 32, 64} V100 GPUs and {1, 1, 2} days for the {small, base, large} models. To improve training efficiency and reduce GPU memory usage, we adopt mixed precision training and gradient checkpointing.

Fine-tuning is performed on the concatenation of 3 datasets: MS-MARCO passage ranking [8], NQ [32, 30], and NLI [22] datasets. We reuse the mined hard negatives and re-ranker scores from SimLM [58] for the first two datasets. Models are fine-tuned for 3 epochs with batch size 256 on 8 GPUs. Learning rate is $\{3, 2, 1\} \times 10^{-5}$ for the {small, base, large} models with 400 steps warmup. For each example, we use 7 hard negatives. Since the NLI dataset only has 1 hard negative for each example, 6 sentences are randomly sampled from the entire corpus.

We use E5-PT to denote models with contrastive pre-training only. More implementation details can be found in Appendix B.

5.2 Evaluation Datasets

BEIR Benchmark [53] is a collection of 19 information retrieval datasets, ranging across ad-hoc web search, question answering, fact verification and duplicate question retrieval, etc. We evaluate the 15 datasets that provide public downloads. The main metric is nDCG@10.

MTEB Benchmark [40] is recently proposed for benchmarking massive text embedding tasks. Though MTEB is multilingual due to the inclusion of bitext mining datasets, most datasets are still only available in English. In this paper, we evaluate the English subsets, which have 56 datasets spanning across 6 categories: Classification (Class.), Clustering (Clust.), Pair Classification (PairClass.), Rerank, Retrieval (Retr.), STS, and Summarization (Summ.). The evaluation metrics are accuracy, v-measure, average precision, MAP, nDCG@10, and Spearman coefficients, respectively. Please refer to the MTEB paper for details.

5.3 Results on BEIR benchmark

Results with Unsupervised Methods In Table 1, we show model results that do not use any labeled data. When averaged over all 15 datasets, $E5\text{-PT}_{\text{base}}$ outperforms the classic BM25 algorithm by 1.2 points. To the best of our knowledge, this is the first reported result that an unsupervised model can beat BM25 on the BEIR benchmark. When scaling up to $E5\text{-PT}_{\text{large}}$, we see further benefits from 42.9 to 44.2.

Table 1: Unsupervised methods on the BEIR benchmark (nDCG@10). For SimCSE, we report results with BERT_{base}. cpt_{300M} [41] is only available through paid API and evaluation results on some datasets are missing in the original paper. The highest number for each dataset is in bold, and the second highest is underlined. †: we report the LaPraDor [62] results without ensembling with BM25. *: reproduction with the released checkpoint.

	BM25	SimCSE	LaPraDor [†]	Contriever	cpt _{300M}	E5-PT _{small}	E5-PT _{base}	E5-PT _{large}
MS MARCO	22.8	9.4	16.9*	20.6	19.9	25.4	<u>26.0</u>	26.2
Trec-Covid	65.6	26.2	22.7	27.4	52.9	52.0	61.0	<u>61.8</u>
NFCorpus	32.5	9.9	31.1	31.7	32.0	29.3	35.8	<u>33.7</u>
NQ	32.9	11.7	18.1	25.4	-	37.3	<u>39.0</u>	41.7
HotpotQA	60.3	19.8	30.3	48.1	51.5	46.0	<u>52.4</u>	52.2
FiQA	23.6	9.8	20.3	24.5	34.1	38.3	<u>40.0</u>	43.2
ArguAna	31.5	38.3	45.9	37.9	38.7	42.5	42.2	<u>44.4</u>
Touche-2020	36.7	8.9	9.4	19.3	<u>21.0</u>	19.9	16.9	<u>19.8</u>
CQADupStack	29.9	13.2	22.0	28.4	-	35.0	<u>35.4</u>	38.9
Quora	78.9	78.0	78.7	83.5	68.1	85.8	85.7	86.1
DBPedia	31.3	15.0	25.0	29.2	27.2	34.5	<u>35.4</u>	37.1
Scidocs	15.8	5.5	13.3	14.9	-	19.9	<u>21.1</u>	21.8
Fever	75.3	21.1	36.8	68.2	57.1	62.5	63.4	<u>68.6</u>
Climate-Fever	21.3	11.8	13.8	15.5	<u>15.8</u>	14.5	15.4	15.7
Scifact	66.5	25.7	55.5	64.9	65.4	68.5	73.7	<u>72.3</u>
Average	41.7	20.3	29.3	36.0	-	40.8	<u>42.9</u>	44.2
Best on	<u>5</u>	0	1	0	0	0	2	7

In terms of pre-training tasks, Contriever adopts random cropping, while LaPraDor combines ICT and dropout-as-positive-instance from SimCSE. The methods can easily obtain large-scale training data, while our approach requires more effort in dataset curation. Such efforts pay off with better results. Recent studies [34, 60, 21] also show that improving data quality is a vital step for training large language models.

Table 2: Supervised fine-tuning results on the BEIR benchmark. Results for ANCE [61], ColBERT [31] and Contriever come from Izacard et al. [28]. The best result is in bold, and the second best is underlined.

	ANCE	GTR _{base}	ColBERT	Contriever	cpt _{300M}	GTR _{large}	E5 _{small}	E5 _{base}	E5 _{large}
MS MARCO	38.8	42.0	40.1	40.7	-	43.0	42.3	<u>43.1</u>	44.1
Trec-Covid	65.4	53.9	67.7	59.6	67.9	55.7	76.8	79.6	<u>78.3</u>
NFCorpus	23.7	30.8	30.5	32.8	33.2	32.9	33.9	36.6	<u>36.1</u>
NQ	44.6	49.5	52.4	49.8	-	54.7	58.7	<u>60.0</u>	62.9
HotpotQA	45.6	53.5	59.3	63.8	59.4	57.9	56.3	62.2	<u>63.3</u>
FiQA	29.5	34.9	31.7	32.9	38.4	42.4	34.8	36.4	<u>38.6</u>
ArguAna	41.5	51.1	23.3	44.6	47.0	52.5	46.7	<u>51.4</u>	49.4
Touche-2020	24.0	20.5	20.2	23.0	28.5	21.9	26.8	<u>28.3</u>	27.2
CQADupStack	29.6	35.7	35.0	34.5	-	38.4	36.1	<u>38.9</u>	39.4
Quora	85.2	88.1	85.4	86.5	70.6	89.0	87.7	87.9	<u>88.2</u>
DBPedia	28.1	34.7	39.2	<u>41.3</u>	36.2	39.1	38.6	41.0	42.4
Scidocs	12.2	14.9	14.5	16.5	-	15.8	16.4	<u>19.0</u>	20.1
Fever	66.9	66.0	77.1	<u>75.8</u>	72.1	71.2	53.5	58.2	65.0
Climate-Fever	19.8	<u>24.1</u>	18.4	23.7	18.5	26.2	15.8	15.4	22.4
Scifact	50.7	60.0	67.1	67.7	67.2	63.9	65.6	73.1	<u>72.6</u>
Average	40.5	44.0	44.4	46.6	-	47.0	46.0	<u>48.7</u>	50.0
Best on	0	0	1	1	1	<u>4</u>	0	3	5

Results with Supervised Fine-tuning In Table 2, we fine-tune our models on supervised datasets and then transfer them to the BEIR benchmark. Since our fine-tuning datasets include MS-MARCO and NQ, the corresponding numbers are in-domain results. For other datasets, these are zero-shot transfer results. Our E5_{base} model achieves an average nDCG@10 of 48.7, already surpassing existing methods with more parameters such as GTR_{large} [43]. Most datasets benefit from supervised fine-tuning, but there are also a few exceptions such as FiQA, Scidocs, and Fever, etc. This is likely due to the lack of enough domain diversity for the fine-tuning datasets.

Table 3: Results on the MTEB benchmark [40] (56 datasets in English subset). Here we only report averaged numbers on each task category for space reasons, please check out Appendix B for a detailed version. BERT-FT_{base} uses the same fine-tuning data as E5 but initializes from BERT_{base}.

# of datasets →	Class. 12	Clust. 11	PairClass. 3	Rerank 4	Retr. 15	STS 10	Summ. 1	Avg 56
<i>Unsupervised models</i>								
Glove	57.3	27.7	70.9	43.3	21.6	61.9	28.9	42.0
BERT	61.7	30.1	56.3	43.4	10.6	54.4	29.8	38.3
SimCSE-BERT-unsup	62.5	29.0	70.3	46.5	20.3	74.3	31.2	45.5
E5-PT _{small}	67.0	41.7	78.2	53.1	40.8	68.8	25.2	54.2
E5-PT _{base}	67.9	43.4	79.2	53.5	42.9	69.5	24.3	55.5
E5-PT _{large}	69.0	44.3	80.3	54.4	44.2	69.9	24.8	56.4
<i>Supervised models</i>								
SimCSE-BERT-sup	67.3	33.4	73.7	47.5	21.8	79.1	23.3	48.7
BERT-FT _{base}	68.7	33.9	82.6	50.5	41.5	79.2	29.0	55.2
Contriever	66.7	41.1	82.5	53.1	41.9	76.5	30.4	56.0
GTR _{large}	67.1	41.6	<u>85.3</u>	55.4	47.4	78.2	29.5	58.3
Sentence-T5 _{large}	72.3	41.7	85.0	54.0	36.7	81.8	29.6	57.1
E5 _{small}	71.7	39.5	85.1	54.5	46.0	80.9	24.9	58.8
E5 _{base}	<u>72.6</u>	42.1	85.1	<u>55.7</u>	<u>48.7</u>	81.0	26.0	<u>60.3</u>
E5 _{large}	73.1	43.3	85.9	56.5	50.0	82.1	25.2	61.3
<i>Larger models</i>								
GTR _{xxl}	67.4	42.4	86.1	56.7	48.5	78.4	30.6	59.0
Sentence-T5 _{xxl}	73.4	43.7	85.1	56.4	42.2	82.6	30.1	59.5

5.4 Results on MTEB benchmark

In Table 3, E5 models not only substantially outperform existing ones with similar sizes, but also match the results of much larger models. The top-2 models on MTEB leaderboard ⁷ GTR_{xxl} and Sentence-T5_{xxl} have 4.8B parameters, while our E5_{large} model is more than 10× smaller with 300M parameters. We expect that our model will benefit from continual scaling up.

Since the difference between BERT-FT_{base} and E5_{base} is that BERT-FT_{base} only has fine-tuning stage, their performance gap demonstrates the usefulness of contrastive pre-training on our proposed CCPairs dataset. For most task categories except Clustering, performance improves after supervised fine-tuning. Consistent with prior works [43, 44], this once again demonstrates the importance of incorporating human knowledge for learning better text embeddings. It remains an open question whether state-of-the-art embeddings can be obtained in a purely self-supervised manner.

Table 4: Zero-shot text classification results. “Majority” always predicts the majority class label. Zero-shot BERT_{base} uses the average pooling of the last layer as text embeddings.

	Zero-shot					Full Fine-tune	
	Majority	BERT _{base}	E5 _{small}	E5 _{base}	E5 _{large}	BERT _{base}	BERT _{large}
SST-2 [52]	50.9	58.9	79.7	81.3	85.3	93.5	94.9

Table 4 shows the zero-shot text classification results on the dev set of the SST-2 dataset [52]. By formulating text classification as embedding matching between input and label texts, our model can be much better than the “majority” baseline in a zero-shot setting. We use the prompt template from Section 4.3.

5.5 Analysis

In this section, we conduct a series of analyses to examine various design choices. All the numbers in this section are from base-size models. For the BEIR benchmark, we choose 6 datasets with more stable results across different runs. Some negative results are also listed in Appendix C.

Impacts of Batch Size Since we use in-batch negatives for contrastive pre-training, larger batch size will provide more negatives and therefore improve the quality of the learned text embeddings. In

⁷<https://huggingface.co/spaces/mteb/leaderboard>, as of November 22, 2022

Table 5: Impacts of different batch sizes for contrastive pre-training.

batch size	NFCorpus	NQ	FiQA	Quora	DBPedia	Scifact	Avg
32k	35.8	39.0	40.0	85.7	35.4	73.7	51.6
8k	33.3	38.5	37.6	85.7	34.0	71.8	50.2
1k	28.2	33.1	30.4	84.0	30.1	69.1	45.8

Table 5, increasing batch size from 1K to 32K leads to consistent gains across all 6 datasets. It is also possible to train with smaller batch sizes by adding hard negatives [50]. However, the engineering efforts of mining hard negatives for large datasets (>100M) are non-trivial.

Table 6: Fine-tuning with different combinations of labeled data.

Fine-tuned on	Retrieval	STS	Classification	Summ.	MTEB Avg
No fine-tuning	42.9	69.5	67.9	24.3	55.5
MS-MARCO + NQ	50.3	78.3	68.3	25.8	58.9
NLI	38.3	81.1	72.6	26.5	57.2
All above	48.7	81.0	73.1	26.0	60.3

Fine-tuning Datasets GTR models are fine-tuned with “MS-MARCO + NQ”, while Sentence-T5 models use NLI instead. In Table 6, we can see that the “MS-MARCO + NQ” setting performs best on retrieval tasks, and the NLI data is beneficial for STS and linear probing classification. Similar observations are also made by Muennighoff et al. [40]. Combining all of them leads to the best overall scores on the MTEB benchmark. This also illustrates the importance of dataset diversity for learning text embeddings.

Table 7: Data filtering. For the top 2 rows, we train with 1M random text pairs.

# of pairs		NFCorpus	NQ	FiQA	Quora	DBPedia	Scifact	Avg
1M	w/o filter	23.0	15.1	18.5	83.1	18.2	51.4	34.9
	w/ filter	26.8	22.7	24.5	85.0	27.5	57.5	40.7
All	w/o filter	34.5	35.4	39.1	85.7	32.9	72.5	50.0
	w/ filter	35.8	39.0	40.0	85.7	35.4	73.7	51.6

Data Filtering One crucial step in our dataset curation pipeline is filtering out low-quality text pairs. In Table 7, when training with 1M pairs, using filtered data has a nearly 6 points advantage. When all the text pairs are used, the “w/o filter” setting has about $4\times$ more data but is still behind by 1.6 points. Though recent studies [29, 47] show that deep learning models are quite robust to dataset noises, data filtering still has benefits in improving training efficiency and model quality.

Negative Sampling We explore two alternative methods to enlarge the number of negatives: Pre-batch negatives [33] reuse embeddings from previous batches as additional negatives, while MoCo [25] introduces a momentum encoder and uses a FIFO queue to store negatives. For both approaches, the negative size can be easily scaled up without incurring much GPU memory overhead. The downside is that most negatives are produced by an older version of model parameters. In Table 8, in-batch negatives still perform favorably. Empirically, we find that MoCo is more sensitive to certain hyperparameters such as temperature, better results are possible with more tuning.

BM25 vs Dense Retrieval With the rapid development of dense retrieval models, can we replace the long-standing BM25 algorithm from now on? The answer is likely “*not yet*”. BM25 still holds obvious advantages in terms of simplicity, efficiency, and interpretability. For long-tail domains such as Trec-Covid [55] and retrieval tasks that involve long documents (Touche-2020) [4] or rely heavily on exact lexical match (Fever) [54], further research efforts are still necessary to improve current dense retrievers.

6 Conclusion

In this work, we train a general-purpose text embedding model E5 from weak supervision signals. We adopt a simple contrastive training framework with in-batch negatives and learn from a large-scale text pair dataset we harvest from heterogeneous data sources across the web. E5 offers strong off-the-shelf

Table 8: Comparison of different negative sampling strategies.

	# negatives	NFCorpus	NQ	FiQA	Quora	DBPedia	Scifact	Avg
In batch	32k	35.8	39.0	40.0	85.7	35.4	73.7	51.6
+ pre-batch	64k	29.4	27.2	29.4	84.6	25.0	64.3	43.3
MoCo	130k	29.7	36.1	32.0	81.6	29.9	63.6	45.5

performance for a wide range of tasks requiring single-vector text representations such as retrieval, semantic textual similarity, and text matching. When further customized for downstream tasks, E5 achieves superior fine-tuned performance compared to existing embedding models with $40\times$ more parameters on the large, 56-task MTEB benchmark datasets.

References

- [1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=SyK00v5xx>.
- [2] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019. doi: 10.1162/tacl_a_00288. URL <https://aclanthology.org/Q19-1038>.
- [3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608. MIT Press, 2001. URL <https://proceedings.neurips.cc/paper/2001/hash/296472c9542ad4d4788d543508116cbc-Abstract.html>.
- [4] Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Shahbaz Syed, Timon Gurcke, Meriem Beloucif, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, et al. Overview of touché 2022: argument retrieval. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 311–336. Springer, 2022.
- [5] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*, pages 716–722. Springer, 2016.
- [6] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL <https://aclanthology.org/D15-1075>.
- [7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>.
- [8] Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268, 2016.

- [9] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 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, 2020. URL <https://openreview.net/forum?id=rkg-mA4FDr>.
- [10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 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*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020. URL <http://proceedings.mlr.press/v119/chen20j.html>.
- [11] Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *arXiv preprint arXiv:2110.06918*, 2021.
- [12] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. Specter: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, 2020.
- [13] Alexis Conneau and Douwe Kiela. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018. European Language Resources Association (ELRA). URL <https://aclanthology.org/L18-1269>.
- [14] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1070. URL <https://aclanthology.org/D17-1070>.
- [15] Zhuyun Dai, Vincent Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. *ArXiv*, abs/2209.11755, 2022.
- [16] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.
- [17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- [18] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. Climate-fever: A dataset for verification of real-world climate claims. *arXiv preprint arXiv:2012.00614*, 2020.
- [19] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/1e14bfe2714193e7af5abc64ecbd6b46-Abstract.html>.
- [20] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, 2022.
- [21] Leo Gao, Stella Rose Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv*, abs/2101.00027, 2021.

- [22] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL <https://aclanthology.org/2021.emnlp-main.552>.
- [23] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/a19744e268754fb0148b017647355b7b-Abstract.html>.
- [24] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. Dbpedia-entity v2: a test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1265–1268, 2017.
- [25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00975. URL <https://doi.org/10.1109/CVPR42600.2020.00975>.
- [26] Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian document computing symposium*, pages 1–8, 2015.
- [27] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019. URL <http://proceedings.mlr.press/v97/houlsby19a.html>.
- [28] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Towards unsupervised dense information retrieval with contrastive learning. *ArXiv*, abs/2112.09118, 2021.
- [29] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 2021. URL <http://proceedings.mlr.press/v139/jia21b.html>.
- [30] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL <https://aclanthology.org/2020.emnlp-main.550>.
- [31] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM, 2020. doi: 10.1145/3397271.3401075. URL <https://doi.org/10.1145/3397271.3401075>.

- [32] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tac1_a_00276. URL <https://aclanthology.org/Q19-1026>.
- [33] Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. 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)*, pages 6634–6647, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.518. URL <https://aclanthology.org/2021.acl-long.518>.
- [34] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *ACL*, 2022.
- [35] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.
- [36] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.447. URL <https://aclanthology.org/2020.acl-main.447>.
- [37] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. Wwv’18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942, 2018.
- [38] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.
- [39] Niklas Muennighoff. Sgpt: Gpt sentence embeddings for semantic search. *ArXiv*, abs/2202.08904, 2022.
- [40] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *ArXiv*, abs/2210.07316, 2022.
- [41] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas A. Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David P. Schnurr, Felipe Petroski Such, Kenny Sai-Kin Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. *ArXiv*, abs/2201.10005, 2022.
- [42] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi-Phuong-Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. SELF: learning to filter noisy labels with self-ensembling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL <https://openreview.net/forum?id=HkgsPhNYPS>.
- [43] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hern’andez ’Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. *ArXiv*, abs/2112.07899, 2021.
- [44] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, 2022.

- [45] Barlas Oguz, Kushal Lakhota, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Scott Yih, Sonal Gupta, and Yashar Mehdad. Domain-matched pre-training tasks for dense retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1524–1534. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-naacl.114. URL <https://doi.org/10.18653/v1/2022.findings-naacl.114>.
- [46] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktaschel, and Sebastian Riedel. Kilt: a benchmark for knowledge intensive language tasks. In *North American Chapter of the Association for Computational Linguistics*, 2020.
- [47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. URL <http://proceedings.mlr.press/v139/radford21a.html>.
- [48] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.
- [49] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL <https://aclanthology.org/D19-1410>.
- [50] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.224. URL <https://aclanthology.org/2021.emnlp-main.224>.
- [51] Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. CCMatrix: Mining billions of high-quality parallel sentences on the web. 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)*, pages 6490–6500, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.507. URL <https://aclanthology.org/2021.acl-long.507>.
- [52] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, 2013.
- [53] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [54] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL <https://aclanthology.org/N18-1074>.
- [55] Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. Trec-covid: constructing a pandemic

- information retrieval test collection. In *ACM SIGIR Forum*, volume 54, pages 1–12. ACM New York, NY, USA, 2021.
- [56] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, 2018.
 - [57] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, 2020.
 - [58] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *ArXiv*, abs/2207.02578, 2022.
 - [59] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, 2021.
 - [60] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France, 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://aclanthology.org/2020.lrec-1.494>.
 - [61] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=zeFrfgYzln>.
 - [62] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3557–3569, 2022.
 - [63] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

A Dataset Details

For Common Crawl, we download the 2022-33 snapshot and cc_net⁸ is used for preprocessing including language identification, de-duplication, language model filtering, etc. Web pages from the MS-MARCO document ranking corpus are also included. For the data filtering step, we examine each pair of passages within a web page instead of just using the title as a query. For Wikipedia, we use the version released by Petroni et al. [46]. To avoid possible data contamination, we remove text pairs that occur in the evaluation datasets based on exact string match.

Reddit data is collected from the year 2018 to August 2022. For the S2ORC data, we use a sample weight of 0.3 during training to avoid over-fitting the scientific domains.

For the BEIR benchmark, we use the 15 datasets that provide public downloads: MS MARCO [8], Trec-Covid [55], NFCorpus [5], NQ [32], HotpotQA [63], FiQA [37], ArguAna [56], Touche-2020 [4], CQADupStack [26], Quora, DBPedia [24], Scidocs [12], Fever [54], Climate-Fever [18], and Scifact [57].

⁸https://github.com/facebookresearch/cc_net

Table 9: Details for each data source after filtering. The “Others” category includes “SimpleWiki”, “GooAQ”, “WikiHow”, “Yahoo Answers” from <https://huggingface.co/datasets/sentence-transformers/embedding-training-data>.

data source	type of text pairs	random example	# of pairs
Wikipedia	(entity+section title, passage)	q: Lexden History p: The site on which Lexden now stands was crossed by the fortifications of iron age Colchester...	24M
Reddit	(post, upvoted comment)	q: What makes a client good quality to you? I’m putting together my ideal client ... p: Respectful of schedules. And pays on time...	60M
Common Crawl	(title, passage)	q: Central Intake Unit Broome County p: Caseworkers from Central Intake assess the household and risk of placement. If eligible...	69M
Stackexchange	(title, answer) (title+description, answer)	q: Will killing Python made problems for Apache p: Python and Apache aren’t related, unless your app is making use of Python. ...	19M
S2ORC	(title, abstract) (title, citation title) (abstract, citation abstract)	q: Constructive Dual DP for Reservoir Optimization p: Dynamic programming (DP) is a well established technique for optimization of reservoir manage...	90M
News	(title, passage) (highlight, passage)	q: LG Display reports Q1 operating loss as... p: April 25 (Reuters) - South Korea’s LG Display Co Ltd reported its first quarterly operating loss...	3M
Others	misc.	misc.	6M
All above	-	-	~ 270M

B Implementation Details

We list the hyperparameters in Table 11. Since some evaluation datasets have long texts, we freeze the position embeddings during both pre-training and fine-tuning and set the maximum text length to 512 for evaluation.

For the Quora duplicate retrieval task in the BEIR benchmark, we add prefix “*query:* ” to all the questions. For other retrieval tasks, we use “*query:* ” and “*passage:* ” prefixes correspondingly.

The MS-MARCO results in Table 12 use document titles provided by RocketQA [50]. This evaluation setup is consistent with most state-of-the-art dense retrievers. However, the MS-MARCO data from the BEIR benchmark does not have titles, so the results are expected to be lower.

Table 10: Model configurations.

	# layers	hidden size	# params
E5 _{small}	12	384	33M
E5 _{base}	12	768	110M
E5 _{large}	24	1024	330M

Table 11: Hyperparameters for contrastive pre-training and fine-tuning.

	pre-training			fine-tuning		
	E5-PT _{small}	E5-PT _{base}	E5-PT _{large}	E5 _{small}	E5 _{base}	E5 _{large}
learning rate	3×10^{-4}	2×10^{-4}	10^{-4}	3×10^{-5}	2×10^{-5}	10^{-5}
GPUs	16	32	64	8	8	8
warmup steps	1000	1000	1000	400	400	400
batch size	32K	32K	32K	256	256	256
max steps	20K	20K	20K	n.a.	n.a.	n.a.
max length	128	128	128	192	192	192
epochs	n.a.	n.a.	n.a.	3	3	3
τ	0.01	0.01	0.01	0.01	0.01	0.01
α	n.a.	n.a.	n.a.	0.2	0.2	0.2
weight decay	0.01	0.01	0.01	0.01	0.01	0.01
hard negatives	0	0	0	7	7	7

In-domain Evaluation We report results for in-domain datasets in Table 12. These results can help illustrate the benefits brought by contrastive pre-training when abundant in-domain labeled data are

available. For MS-MARCO passage ranking, MRR@10 and Recall@1k are reported. For the NQ dataset, Recall@20 and Recall@100 are the main metrics.

Table 12: In-domain results. “target pre-train” refers to intermediate pre-training on the target corpus before supervised fine-tuning. For NQ, we use the passage retrieval setting from DPR [30].

	target pre-train?	MS-MARCO		NQ	
		MRR@10	R@1k	R@20	R@100
ANCE [61]	✗	33.0	95.9	81.9	87.5
RocketQAv2 [50]	✗	38.8	98.1	83.7	89.0
SimLM [58]	✓	41.1	98.7	85.2	89.7
E5 _{small}	✗	37.5	98.1	84.6	89.8
E5 _{base}	✗	38.5	98.5	86.1	90.7
E5 _{large}	✗	39.4	98.7	86.4	90.5

C Negative Results

Here are some attempts that we eventually give up on:

Adding BM25 hard negatives Similar to DPR [30], we add one BM25 hard negative for each positive pair during training. When using 15M data, this strategy improves the overall results by ~ 0.5 points on the BEIR benchmark. However, running the BM25 algorithm over a 250M+ dataset is too time-consuming even with multi-node and multi-process parallelism.

Using RoBERTa instead of BERT for initialization Though RoBERTa shows consistent gains on many NLP tasks, we empirically find that RoBERTa performs worse than BERT initialization on most of the BEIR benchmark datasets.

Auxiliary MLM objective We add a masked language modeling loss for 25% of the training text pairs. The numbers are on par with removing this auxiliary objective, but the training cost goes up.

Table 13: Results for each dataset in the MTEB benchmark [40]. The numbers for the Retrieval category are not included here since the datasets are the same as the BEIR benchmark.

	unsupervised			supervised		
	E5-PT _{small}	E5-PT _{base}	E5-PT _{large}	E5 _{small}	E5 _{base}	E5 _{large}
AmazonCounterfactualClassification	71.7	73.6	70.4	76.2	79.7	77.7
AmazonPolarityClassification	76.1	77.0	83.2	87.5	88.0	90.1
AmazonReviewsClassification	35.0	35.8	37.4	42.6	42.7	43.0
Banking77Classification	82.1	82.9	83.5	81.9	83.3	84.1
EmotionClassification	42.2	44.2	43.5	46.9	49.4	48.1
ImdbClassification	67.9	67.3	77.7	75.6	76.0	82.1
MassiveIntentClassification	70.2	71.1	70.8	72.2	72.3	73.2
MassiveScenarioClassification	74.6	75.4	75.9	75.8	76.8	77.4
MTOPDomainClassification	91.3	92.3	93.2	92.1	93.2	93.9
MTOPIntentClassification	71.9	74.0	74.2	73.2	74.8	76.4
ToxicConversationsClassification	67.0	67.4	66.1	72.8	74.1	70.6
TweetSentimentExtractionClass.	54.4	53.3	52.5	63.3	61.4	61.2
ArxivClusteringP2P	47.9	49.3	49.4	44.1	44.6	46.2
ArxivClusteringS2S	39.9	42.8	43.6	37.1	40.5	41.4
BiorxivClusteringP2P	38.5	38.8	39.2	35.8	36.2	37.6
BiorxivClusteringS2S	35.4	36.5	36.7	31.9	32.7	35.1
MedrxivClusteringP2P	34.4	33.7	33.3	31.3	31.5	32.3
MedrxivClusteringS2S	32.0	32.1	32.2	28.2	28.3	29.7
RedditClustering	46.9	49.3	52.4	42.9	48.2	50.7
RedditClusteringP2P	60.2	64.4	64.6	56.4	62.2	61.4
StackExchangeClustering	57.7	60.2	63.3	59.1	63.9	65.0
StackExchangeClusteringP2P	32.0	34.0	34.7	30.3	32.6	33.6
TwentyNewsgroupsClustering	34.4	36.2	37.9	37.5	42.6	43.8
SprintDuplicateQuestions	91.6	90.8	92.0	95.3	94.9	95.4
TwitterSemEval2015	60.0	62.8	64.7	74.2	74.4	76.1
TwitterURLCorpus	83.2	84.0	84.1	85.8	86.0	86.3
AskUbuntuDupQuestions	57.8	57.6	58.3	59.4	59.7	60.1
MindSmallReranking	29.0	29.6	29.2	29.6	30.1	30.8
SciDocsRR	81.1	82.6	84.3	79.8	82.9	83.9
StackOverflowDupQuestions	44.4	44.2	45.8	49.1	50.1	51.3
BIOSSES	69.2	71.9	69.7	84.2	85.1	84.7
SICK-R	66.6	68.7	69.7	78.9	79.7	80.5
STS12	60.7	57.9	54.7	75.2	74.2	75.9
STS13	71.1	73.5	74.0	81.8	83.3	85.2
STS14	64.2	64.0	65.3	78.5	78.5	80.5
STS15	74.3	75.4	75.8	87.5	88.4	88.8
STS16	76.6	79.8	80.1	84.6	84.2	85.3
STS17	78.3	77.2	76.0	87.9	87.2	89.4
STS22	59.2	56.2	62.8	63.8	62.9	63.0
STSBenchmark	67.7	70.5	70.9	86.4	86.2	87.2
SummEval	25.2	24.3	24.8	24.9	26.0	25.2