Text and Code Embeddings by Contrastive Pre-Training

Arvind Neelakantan [*1] **Tao Xu** [*1] **Raul Puri** [1] **Alec Radford** [1] **Jesse Michael Han** [1] **Jerry Tworek** [1]
Qiming Yuan [1] **Nikolas Tezak** [1] **Jong Wook Kim** [1] **Chris Hallacy** [1] **Johannes Heidecke** [1] **Pranav Shyam** [1]
Boris Power [1] **Tyna Eloundou Nekoul** [1] **Girish Sastry** [1] **Gretchen Krueger** [1] **David Schnurr** [1]
Felipe Petroski Such [1] **Kenny Hsu** [1] **Madeleine Thompson** [1] **Tabarak Khan** [1] **Toki Sherbakov** [1] **Joanne Jang** [1]
Peter Welinder [1] **Lilian Weng** [1]

Abstract

Text embeddings are useful features in many applications such as semantic search and computing text similarity. Previous work typically trains models customized for different use cases, varying in dataset choice, training objective and model architecture. In this work, we show that contrastive pre-training on unsupervised data at scale leads to high quality vector representations of text and code. The same unsupervised text embeddings that achieve new state-of-the-art results in linear-probe classification also display impressive semantic search capabilities and sometimes even perform competitively with fine-tuned models. On linear-probe classification accuracy averaging over 7 tasks, our best unsupervised model achieves a relative improvement of 4% and 1.8% over previous best unsupervised and supervised text embedding models respectively. The same text embeddings when evaluated on large-scale semantic search attains a relative improvement of 23.4%, 14.7%, and 10.6% over previous best unsupervised methods on MSMARCO, Natural Questions and TriviaQA benchmarks, respectively. Similarly to text embeddings, we train code embedding models on (text, code) pairs, obtaining a 20.8% relative improvement over prior best work on code search.

1. Introduction

Deep unsupervised learning with generative and embedding models has seen dramatic success in the past few years. Generative models (Peters et al., 2018; Raffel et al., 2019; van den Oord et al., 2016; Ramesh et al., 2021; Brown et al., 2020; Chen et al., 2021) are trained to max-
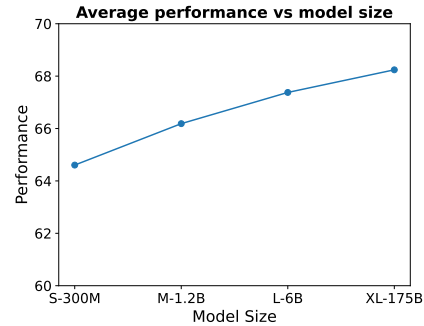
Figure 1. Average performance of unsupervised `cpt-text` models of different sizes across 22 tasks consisting of linear-probe classification, text search, and sentence similarity tasks.

imize the likelihood of observed data while embedding models are trained to distinguish observed data from noise (Sohn, 2016; van den Oord et al., 2018; Radford et al., 2021; Jia et al., 2021; Gao et al., 2021; Izacard et al., 2021). Generative models have been shown to produce realistic content and benefit many downstream applications, reducing the need for labeled training datasets. In generative models, the information about the input is typically distributed over multiple hidden states of the model. While some generative models (Kingma & Welling, 2014; Kiros et al., 2015) can learn a single representation of the input, most autoregressive Transformer (Vaswani et al., 2017) models do not (Raffel et al., 2019; Brown et al., 2020; Chen et al., 2021; Ramesh et al., 2021). However, learning such a representation (or embedding) is necessary for many tasks. Systems that search over millions or billions of items require each entry to be embedded as a dense representation and build an index in advance to save computational costs at query time. These embeddings are useful features for classification tasks and can also enable data visualization applications via techniques such as clustering. Embedding models are explicitly optimized to learn a low dimensional representation that captures the semantic meaning of the input (Radford et al., 2021; Jia et al., 2021; Giorgi et al., 2020; Gao et al., 2021; Izacard et al., 2021).

In this work, we train embedding models using a contrastive learning objective with in-batch negatives (Sohn, 2016; Yih et al., 2011) on unlabeled data. The input is encoded with a Transformer encoder (Vaswani et al., 2017) and we leverage naturally occurring paired data to construct training data with no explicit labels. Text embedding models are trained on paired text data where we consider neighboring pieces of text on the Internet as positive pairs. Code embedding models treat the top-level docstring in a function along with its implementation as a (text, code) pair. The training signal of the contrastive objective on its own is not sufficient to learn useful representations and we overcome this by initializing our model with other pretrained models (Brown et al., 2020; Chen et al., 2021). Finally, we find that it is critical to use a sufficiently large batch to achieve the optimal performance. We show that this simple recipe combining pre-trained model initialization, large-batch contrastive learning and training at scale, can produce text and code embeddings that possess a broad range of capabilities.

We train a series of unsupervised text embedding models (cpt-text) of different sizes, ranging from 300M to 175B parameters, and observe a consistent performance improvement with increasing model sizes (Figure 1). On classification accuracy averaging across 7 linear-probe classification tasks in SentEval (Conneau & Kiela, 2018), our largest unsupervised model achieves new state-of-the-art results with a relative improvement of 4% and 1.8% over the previous best unsupervised (Giorgi et al., 2020) and supervised (Gao et al., 2021) text embedding models, respectively.

Text embedding in previous work was studied under different domains, varying in data, training objective and model architecture. Precisely, sentence embedding (Reimers & Gurevych, 2019; Gao et al., 2021; Giorgi et al., 2020) and neural information retrieval (Lee et al.; Guu et al., 2020; Karpukhin et al., 2020a; Sachan et al., 2021; Izacard et al., 2021) have remained different research topics evaluated on distinct benchmarks, even though both aim to learn high-quality text representation. However, we find the same model that achieves good performance on sentence embedding benchmarks, as discussed above, is also able to obtain impressive results on large-scale information retrieval. When evaluated on the MSMARCO passage ranking task (Nguyen et al., 2016) to search over 4M passages, cpt-text gets a relative improvement of 23.4% over previous best unsupervised methods (Robertson, 2009). On the task of searching on 21M documents from Wikipedia, cpt-text obtains a relative improvement of 14.7%, and 10.6% over previous unsupervised methods (Izacard et al., 2021) for Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), respectively. On TriviaQA, our unsupervised method is even competitive with fine-tuned models.

Next, we train code embedding models (cpt-code) using the same recipe. Our models learn via (text, code) pairs, extracted from open source code. We evaluate our model on CodeSearchNet (Husain et al., 2020), a commonly used code search benchmark, where the task is to find the most relevant code snippet given a natural language query. Our models achieve new state-of-the-art results with a 20.8% relative improvement over the previous best result (Guo et al., 2021). Unlike text embedding models, we observe no performance improvement on code search when increasing the number of parameters of cpt-code from 300M to 1.2B.

Finally, we experiment with fine-tuning our models on several supervised datasets and study the transfer learning performance. When fine-tuned on NLI (Natural Language Inference) datasets, we see a further boost in linear-probe classification, outperforming the previous best transfer method (Gao et al., 2021) by 2.2%. On SST-2 sentiment classification (Socher et al., 2013), we find that our representations are sufficiently descriptive that even a simple k-NN classifier achieves results comparable to a linear-probe classifier. Interestingly, zero-shot performance with our embeddings outperforms the supervised neural network models introduced along with the release of the SST-2 dataset. We also fine-tune the unsupervised model on MS-MARCO and evaluate it on a suite of zero-shot search tasks in the BEIR benchmark (Thakur et al., 2021). In the transfer setting, our models achieve a 5.2% relative improvement over previous methods (Izacard et al., 2021) and is comparable even with methods (Santhanam et al., 2021; Formal et al., 2021; Wang et al., 2020) that demand substantially more computation at test time.

2. Approach

Our models are trained with a contrastive objective on paired data. In this section, we present more details on the model architecture and the training objective. The training set consists of paired samples, $\{(x_i, y_i)\}_{i=1}^{N}$, where (x_i, y_i) corresponds to a positive example pair, indicating that x_i and y_i are semantically similar or contextually relevant.

2.1. Model

Given a training pair (x, y), a Transformer (Vaswani et al., 2017) encoder E is used to process x and y independently. The encoder maps the input to a dense vector representation or embedding (Figure 2). We insert two special token delimiters, [SOS] and [EOS], to the start and end of the input sequence respectively. The hidden state from the last layer corresponding to the special token [EOS] is considered as the embedding of the input sequence.
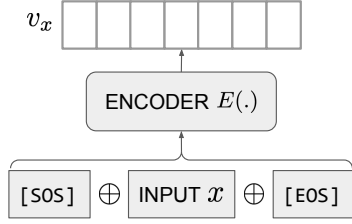
Figure 2. The encoder E maps input x to embedding v_x. Special tokens, `[SOS]` and `[EOS]`, are appended to the start and end of the input sequence respectively. The last layer hidden state corresponding to the token `[EOS]` is extracted as the embedding of the input sequence.
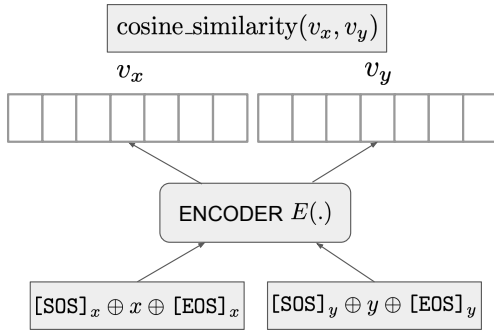

Figure 3. The encoder E maps inputs x and y, to embeddings, v_x and v_y independently. The similarity score between x and y is defined as the cosine similarity between these two embedding vectors.

The Transformer encoder maps the input, x and y, to embeddings, v_x and v_y respectively and the similarity between two inputs is quantified by the cosine similarity between their embeddings, v_x and v_y (Figure 3).

$$v_x = E(\texttt{[SOS]}_x \oplus x \oplus \texttt{[EOS]}_x)$$
$$v_y = E(\texttt{[SOS]}_y \oplus y \oplus \texttt{[EOS]}_y)$$
$$\text{sim}(x, y) = \frac{v_x \cdot v_y}{\|v_x\| \cdot \|v_y\|}$$

where $\oplus$ is an operation to concatenate two strings together. We found that using different delimiters leads to more stable training. For x, we use '[' as $\texttt{[SOS]}_x$ and ']' as $\texttt{[EOS]}_x$, while we use '{' and '}' as $\texttt{[SOS]}_y$ and $\texttt{[EOS]}_y$ respectively for y.

2.2. Training Objective

The paired samples in the training set are contrasted against in-batch negatives (Yih et al., 2011; Sohn, 2016). Contrastive learning with in-batch negatives has been widely

Model	Parameters	Embed Dimensions	Batch size
S	300M	1024	12288
M	1.2B	2048	6912
L	6B	4096	5896
XL	175B	12288	4976

Table 1. Batch size used to train the models of different sizes.

used for unsupervised representation learning in prior work (Radford et al., 2021; Jia et al., 2021; Chen et al., 2020; Izacard et al., 2021). For each example in a mini-batch of M examples, the other $(M - 1)$ in the batch are used as negative examples. The usage of in-batch negatives enables re-use of computation both in the forward and the backward pass making training highly efficient. The `logits` for one batch is a $M \times M$ matrix, where each entry $\text{logit}(x_i, y_j)$ is given by,

$$\text{logit}(x_i, y_j) = \text{sim}(x_i, y_j) \cdot \exp(\tau),$$
$$\forall (i, j), i, j \in \{1, 2, \dots, M\}$$

where τ is a trainable temperature parameter.

Only entries on the diagonal of the matrix are considered positive examples. The final training loss is the sum of the cross entropy losses on the row and the column direction, as described in the following numpy style pseudo code.

```
labels = np.arange(M)
l_r = cross_entropy(logits, labels, axis=0)
l_c = cross_entropy(logits, labels, axis=1)
loss = (l_r + l_c) / 2
```

We initialize our models with pre-trained generative language models. `cpt-text` is initialized with GPT models (Brown et al., 2020) and `cpt-code` is initialized with Codex models (Chen et al., 2021). When fine-tuning our models (Section 3), the supervised training data like NLI datasets contain explicit negative examples and they are used along with the in-batch negatives.

3. Results

Our models are trained on naturally occurring paired data. `cpt-text` models are trained on Internet data with neighboring pieces of text as positive pairs for the contrastive objective. The code embedding `cpt-code` models use (text, code) pairs extracted from open source code. As discussed in Section 3.4.1, sufficiently large batch size is crucial to achieve good performance with our setup. Table 1 lists the batch sizes used to train the models of different sizes.

We evaluate our text embedding models on a broad range of tasks: linear-probe classification, sentence similarity, and

semantic search. While sentence embedding (Reimers & Gurevych, 2019; Gao et al., 2021; Giorgi et al., 2020) methods report results only on embedding benchmarks and neural information retrieval methods (Lee et al.; Guu et al., 2020; Karpukhin et al., 2020a; Sachan et al., 2021; Izacard et al., 2021) report results only on search benchmarks, we use the *same* unsupervised model across all these tasks.

3.1. Text Embedding

The SentEval benchmark (Conneau & Kiela, 2018) is widely adopted to assess the quality of sentence embeddings, consisting of a broad collection of tasks in the categories of linear-probe classification and sentence similarity, and we use the same to evaluate ours.

3.1.1. LINEAR PROBE CLASSIFICATION

When evaluated on linear-probe classification, the embeddings are used as features to train a linear classifier to solve a variety of downstream tasks. The results in Table 2 demonstrate a clear advantage of larger model sizes producing better features for improved classification performance. In transfer learning setup, we fine-tune unsupervised `cpt-text` models on SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) datasets using entailment pairs as positive examples and contradiction pairs as negative examples. On both unsupervised learning and transfer learning settings, we achieve state-of-the-art results.

3.1.2. ZERO-SHOT AND k-NN CLASSIFICATION

In this section, we discuss results using zero-shot classification and k-nearest neighbor classification on the SST-2 binary sentiment classification task (Socher et al., 2013). We experiment with 6B (L) `cpt-text` model fine-tuned on NLI data for this study. In the first zero-shot experiment, each input text is assigned with one of the two labels ('positive', 'negative') based on which label has its embedding closest to the input text embedding. The performance can be further improved by prompting, where we use a simple label description, 'this is an example of a positive/negative movie review.', instead of a single word. This zero-shot usage of embeddings is novel compared to prior work on embeddings and it is interesting to note that our zero-shot results are better than the supervised neural network results reported along with the release of the dataset (Socher et al., 2013). In the k-NN classification experiment, given an input text, the prediction is the majority label among 256 training examples closest to the test input in the embedding space. As shown in Table 3, the k-NN classifier without any task-specific tuning of trainable parameters achieves results comparable to a linear classifier.

3.1.3. SENTENCE SIMILARITY

On sentence similarity tasks in SentEval, we find that our models perform worse than previous SOTA methods (Table 4). Sentence similarity is not a completely well-defined downstream task (e.g. are the sentences, 'Jack loves Jill' and 'Mary loves chocolates', similar?).[1,2] For example, Goodman (1972) argue that two objects can be infinitely similar or dissimilar (Vervaeke et al., 2012). A possible explanation for why our models perform better than prior work on search and classification but not on these tasks is that our models might not be optimized for the specific definition used by these sentence similarity benchmarks. It is important to note that previous embedding search methods do not report performance on sentence similarity tasks (Karpukhin et al., 2020a; Sachan et al., 2021; Izacard et al., 2021). More discussion on this phenomenon is presented in Section 3.4.2.

3.2. Text Search

Previous work on training embedding methods for search typically requires fine-tuning on a particular text search dataset (Karpukhin et al., 2020a; Sachan et al., 2021; Qu et al., 2021). It is also common to have a multi-step setup where fine-tuned models rely on an expensive query and document cross-attention encoder in the final step (Qu et al., 2021; Wang et al., 2020). In contrast, we push the limits of using a *single* embedding model for large-scale semantic search.

3.2.1. LARGE-SCALE SEARCH

First, we evaluate our models on several large-scale text search benchmarks. MSMARCO (Nguyen et al., 2016) requires the model to search over 4M documents while Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) involve searching over 21M Wikipedia documents. We use the FAISS library (Johnson et al., 2019) to build the vector indices for approximate k-nearest neighbor search. The *same* unsupervised model discussed previously achieves impressive performance on semantic search. Table 5 demonstrates that `cpt-text` outperforms prior unsupervised approaches by a big margin and larger model sizes consistently lead to improved performance. Surprisingly, on TriviaQA, our model is even competitive with fine-tuned models.

---

[1] https://twitter.com/yoavgo/status/1431299645570011142
[2] https://twitter.com/yoavgo/status/1483565266575540225?s=20

	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Unsupervised								
BERT (Devlin et al., 2019)	78.7	86.2	94.4	88.7	84.4	92.8	69.4	84.9
SimCSE (Gao et al., 2021)	84.7	88.6	95.4	87.5	89.5	95.0	72.4	87.6
DECLUTR (Giorgi et al., 2020)	85.2	90.7	95.8	88.5	90.0	93.2	**74.6**	88.3
`cpt-text` S	87.1	90.1	94.9	88.3	91.8	95.2	71.6	88.4
`cpt-text` M	89.0	90.9	96.7	89.6	93.9	96.6	73.6	89.9
`cpt-text` L	90.6	92.6	97.0	90.6	95.3	97.0	73.6	90.9
`cpt-text` XL	**92.2**	**93.5**	**97.4**	**91.5**	**96.2**	**97.4**	74.1	**91.8**
Transfer from NLI data								
SBERT (Reimers & Gurevych, 2019)	84.9	90.1	94.5	90.3	90.7	87.4	75.9	87.7
SimCSE (Gao et al., 2021)	88.4	92.5	95.2	90.1	93.3	93.8	77.7	90.2
`cpt-text` S	87.3	91.0	94.6	90.5	91.4	95.0	75.6	89.3
`cpt-text` M	89.8	92.7	95.7	91.3	95.3	96.6	76.5	91.1
`cpt-text` L	90.8	93.5	96.2	91.2	95.7	96.0	76.9	91.5
`cpt-text` XL	**92.4**	**93.9**	**97.0**	**91.8**	95.8	96.4	**78.1**	**92.2**

Table 2. `cpt-text` models of different sizes, ranging from 300M (S) to 175B (XL), are compared to previous work on linear-probe classification tasks in SentEval. We report performance of unsupervised models, as well as those fine-tuned on NLI data.

Method	Accuracy
Zero-shot	88.1
Zero-shot with prompting	89.1
k-NN	93.3
Linear-probe	95.7
Full fine-tuned SOTA	97.5

Table 3. Comparison of different classification strategies using the 6B `cpt-text` model fine-tuned on NLI data for SST-2 binary sentiment task (Socher et al., 2013). Our zero-shot results are better than the 85.4% accuracy obtained by supervised neural networks reported along with the release of the dataset (Socher et al., 2013).

STS	-12	-13	-14	-15	-16	Avg
Unsupervised						
SimCSE (Gao et al., 2021)	**72.9**	**84.0**	**75.6**	**84.8**	**81.8**	**79.8**
`cpt-text` S	62.1	60.0	62.0	71.8	73.7	65.9
`cpt-text` M	62.7	62.8	64.6	73.9	75.3	67.9
`cpt-text` L	62.4	66.4	67.6	76.0	77.5	70.0
`cpt-text` XL	64.1	67.5	68.4	76.7	78.7	71.1
Transfer from NLI						
SimCSE (Gao et al., 2021)	**77.5**	**87.3**	**82.4**	**86.7**	83.9	**83.6**
`cpt-text` S	72.8	80.6	78.7	84.7	82.0	79.8
`cpt-text` M	73.7	80.2	78.9	85.0	82.8	80.1
`cpt-text` L	71.8	79.7	79.0	85.8	84.0	80.1
`cpt-text` XL	72.3	80.3	78.9	85.1	**85.1**	80.3

Table 4. `cpt-text` performs worse than the previous best sentence embedding method on sentence similarity tasks. We investigate this result in more detail in Section 3.4.2.

	MSMARCO	NQ	TriviaQA
Fine-tuned SOTA	44.3	84.8, 89.8	84.1, 87.8
Unsupervised			
BM25	18.4	62.9, 78.3	76.4, 83.2
ICT	-	50.9, 66.8	57.5, 73.6
MSS	-	59.8, 74.9	68.2, 79.4
Contriever	-	67.2, 81.3	74.2, 83.2
`cpt-text` S	19.9	65.5, 77.2	75.1, 81.7
`cpt-text` M	20.6	68.7, 79.6	78.0, 83.8
`cpt-text` L	21.5	73.0, 83.4	80.0, 86.8
`cpt-text` XL	**22.7**	**78.8, 86.8**	**82.1, 86.9**

Table 5. Evaluation of unsupervised `cpt-text` models of different sizes on several large-scale text search benchmarks. We report MRR@10 on MSMARCO and Recall@20, Recall@100 for NQ and TriviaQA as done in prior work. Results for training with Inverse Cloze Task (ICT) and masked salient spans (MSS) objectives are taken from Sachan et al. (2021). `cpt-text` achieves the best results among unsupervised methods, surpassing keyword search methods on MSMARCO (Robertson, 2009) and embedding based methods (Izacard et al., 2021) on NQ and TriviaQA.

3.2.2. BEIR SEARCH

Next, we evaluate our models on 11 zero-shot search tasks in the BEIR evaluation suite (Thakur et al., 2021). First, we observe that our unsupervised model performs competitively even with some previous embedding methods that leverage supervised MSMARCO data (Xiong et al., 2020; Hofstätter et al., 2021). Keyword-based BM25 (Robertson, 2009) achieves the best results in the unsupervised setting while cpt-text achieves the best transfer learning results.

In the transfer setting, our models achieve a 5.2% relative improvement over the previous best embedding method (Izacard et al., 2021). It also outperforms docT5query (Nogueira et al., 2019a) that relies on a fine-tuned T5 model (Raffel et al., 2019) for document expansion. cpt-text results are competitive even with methods that use substantially more compute at test time. BM25+CE (Wang et al., 2020) uses keyword search to select top 100 documents which are then re-ranked by a cross-attention neural network encoder. The ranking encoder network performs computationally expensive joint query and document attention and cannot exploit indexing and approximate nearest neighbor algorithms for fast and efficient search at query time. Several other existing work take this approach of leveraging more computation resources at query time to obtain better search performance. ColBERT v2 (Santhanam et al., 2021) is a multi-vector method that represents the query and the documents as a set of vectors, and employs a multi-step retrieval procedure to obtain relevant documents. Splade v2 (Formal et al., 2021) represents queries and documents as sparse vectors of size equivalent to the vocabulary of the BERT encoder (Devlin et al., 2019). Our cpt-text models compute only one dense embedding per document which are indexed offline and does not depend on any cross-attention re-ranker at query time.

3.3. Code Search

We evaluate our code embedding models on the code search task using the CodeSearchNet benchmark (Husain et al., 2020). Given a natural language query, the model is expected to retrieve the relevant code block among 1K candidates. The models are evaluated on 6 programming languages and our model achieves state-of-the-art results (Table 7). Unlike with text embeddings, we do not see a performance improvement with increased model size for code embeddings.

We also evaluate on a harder setting of finding the relevant code block among 10K candidates instead of 1K. Here, we compare the performance of cpt-text models against cpt-code models (Table 8). It is interesting to see that text embedding performs fairly well in code search especially in Python. We see a drop in performance for code

embedding models with increased distractors and still don't see bigger models giving a boost in search performance.

3.4. Analysis

3.4.1. EFFECT OF BATCH SIZE

Our ablation study highlights the effect of the model's batch size on the final performance. Table 9 compares the performance of S (300M) cpt-text model trained with different batch sizes on the NQ development set. Since we train with in-batch negatives, a larger batch increases the chances of having hard negatives in a batch, resulting in a significant performance boost.

3.4.2. TRAINING BEHAVIOR

We observe that as we train our models for longer, the performance on search and classification tasks increases while the performance on sentence similarity tasks decreases (Figure 4). As discussed previously, sentence similarity is not a well defined task. A hypothesis is that search tasks and sentence similarity tasks might have contradicting definitions. For example, a sentence and its negation could be considered as relevant during search, but not "similar" in sentence similarity tasks. It is also important to note that previous embedding search methods do not report performance on sentence similarity tasks (Karpukhin et al., 2020a; Sachan et al., 2021; Izacard et al., 2021) and previous sentence embedding methods do not evaluate on search tasks (Reimers & Gurevych, 2019; Giorgi et al., 2020; Gao et al., 2021). When deciding the model checkpoints to use for evaluation, we assigned higher importance to search and classification tasks as they are commonly associated with clearly defined real-world applications while sentence similarity tasks are less so.

4. Related Work

The goal of representation learning (Bengio et al., 2012) is to learn an embedding space in which similar examples stay close to each other while dissimilar ones are far apart (Hadsell et al., 2006). In contrastive learning, the learning procedure is formulated as a classification problem given similar and dissimilar candidates (Chopra et al., 2005; Gutmann & Hyvärinen, 2010; Schroff et al., 2015; Sohn, 2016; van den Oord et al., 2018). Recent work relies on contrastive objective to learn representations for images (Wu et al., 2018; He et al., 2020; Chen et al., 2020; Zbontar et al., 2021), text, or both jointly (Lu et al., 2019; Sun et al., 2019; Kim et al., 2021; Radford et al., 2021; Khosla et al., 2020). In self-supervised contrastive learning, positive samples can be collected in various approaches including by creating an augmented version of the original input without modifying the semantic meaning (Gao

	covid	nfc	fiqa	arg.	touche	quora	scifact	climate	dbp.	hotpot	fever	Avg.
					Unsupervised							
BM25 (Robertson, 2009)	**65.6**	32.5	23.6	31.5	**36.7**	78.9	66.5	**21.3**	**31.3**	**60.3**	**75.3**	**47.6**
Contriever (Izacard et al., 2021)	27.4	31.7	24.5	37.9	19.3	**83.5**	64.9	15.5	29.2	48.1	68.2	40.9
cpt-text S	52.9	32.0	34.1	38.7	21.0	68.1	65.4	15.8	27.2	51.5	57.1	42.2
cpt-text M	44.3	34.5	37.3	**41.2**	23.3	70.3	68.3	15.6	29.6	53.0	58.2	43.2
cpt-text L	42.7	**36.9**	**39.7**	39.2	22.8	68.7	**71.2**	16.1	31.2	54.3	63.8	44.2
					Transfer from MSMARCO							
TAS-B (Hofstätter et al., 2021)	48.1	31.9	30.0	42.9	16.2	83.5	64.3	22.8	38.4	58.4	70.0	46.0
ANCE (Xiong et al., 2020)	65.4	23.7	29.5	41.5	24.0	85.2	50.7	19.8	28.1	45.6	66.9	43.7
Contriever (Izacard et al., 2021)	59.6	32.8	32.9	44.6	23.0	**86.5**	67.7	23.7	41.3	63.8	75.8	50.2
cpt-text S	67.9	33.2	38.4	47.0	28.5	70.6	67.2	18.5	36.2	59.4	72.1	49.0
cpt-text M	58.5	36.7	42.2	**49.2**	29.7	69.7	70.4	19.9	38.6	63.1	77.0	50.5
cpt-text L	56.2	38.0	45.2	46.9	30.9	67.7	74.4	19.4	41.2	64.8	75.6	50.9
cpt-text XL	64.9	**40.7**	**51.2**	43.5	29.1	63.8	**75.4**	22.3	43.2	**68.8**	77.5	**52.8**
docT5query (Nogueira et al., 2019a)	71.3	32.8	29.1	34.9	**34.7**	80.2	67.5	20.1	33.1	58.0	71.4	48.5
BM25+CE (Wang et al., 2020)	**75.7**	35.0	34.7	31.1	27.1	82.5	68.8	**25.3**	39.2	70.7	81.9	52.0
ColBERT v2 (Santhanam et al., 2021)	73.8	33.8	35.6	46.3	26.3	85.2	69.3	17.6	**44.6**	66.7	78.5	52.5
Splade v2 (Formal et al., 2021)	71.0	33.4	33.6	47.9	27.2	83.8	69.3	23.5	43.5	68.4	**78.6**	52.7

Table 6. Comparison of cpt-text to previous methods on 11 zero-shot search tasks in the BEIR evaluation suite (Thakur et al., 2021). Results are reported both in the unsupervised data setting and in the transfer data setting. cpt-text outperforms previous best embedding methods (Xiong et al., 2020; Hofstätter et al., 2021; Izacard et al., 2021) in both the settings. In the unsupervised setting, BM25 (Robertson, 2009) still achieves the best performance while in the transfer setting cpt-text is competitive with methods that use substantially more compute at test time (Wang et al., 2020; Santhanam et al., 2021; Formal et al., 2021).

	Go	Ruby	Python	Java	JS	PHP	Avg.
CodeBERT	69.3	70.6	84.0	86.8	74.8	70.6	76.0
GraphCodeBERT	84.1	73.2	87.9	75.7	71.1	72.5	77.4
cpt-code S	**97.7**	**86.3**	99.8	94.0	86.0	96.7	93.4
cpt-code M	97.5	85.5	**99.9**	**94.4**	**86.5**	**97.2**	**93.5**

Table 7. Comparison of cpt-code on code search across 6 programming languages (Husain et al., 2020) with CodeBERT (Feng et al., 2020) and GraphCodeBERT (Guo et al., 2021). The task requires finding the relevant code block among 1K candidates for a given natural language query. cpt-code performs substantially better than previous methods on all the languages.

	Go	Ruby	Python	Java	JS	PHP	Avg.
cpt-text S	60.6	58.9	92.6	48.4	52.8	47.6	60.1
cpt-text M	65.4	63.1	91.4	47.9	53.5	43.1	60.7
cpt-code S	**90.4**	80.6	98.8	**81.9**	**76.1**	**85.3**	**85.5**
cpt-code M	90.0	**89.1**	**98.9**	81.1	75.6	85.1	85.0

Table 8. Comparison of cpt-code vs cpt-text on large scale code search (Husain et al., 2020). The task is to retrieve the relevant code block among 10K candidates for a given natural language query. It is interesting to note that cpt-text performs quite well on Python code search without explicitly training on (text, code) pairs.

Batch Size	MRR@10
1536	71.4
12288	84.7

Table 9. Performance of the cpt-text 300M model on NQ dev set given different training batch sizes.

et al., 2021), by grouping samples within the same context (Giorgi et al., 2020; Izacard et al., 2021), or by collecting data about the same object from different views (Tian et al., 2019).

Learning word embeddings is a well studied research area (Brown et al., 1992; Gutmann & Hyvärinen, 2010; Mikolov et al., 2013; Pennington et al., 2014). Learning low-dimensional representations of larger text pieces, denser than raw term-based vectors, has been studied extensively as well (Deerwester et al., 1990; Yih et al., 2011). Most of the recent models for learning sentence embeddings rely on supervised NLI datasets, using entailment pairs as positive examples and contradiction pairs as (hard) negatives. SBERT (Reimers & Gurevych, 2019) trained a siamese network to learn a representation where sentence similarity is estimated by the cosine similarity between embeddings. Li et al. (2020) improves the embedding space to be isotropic
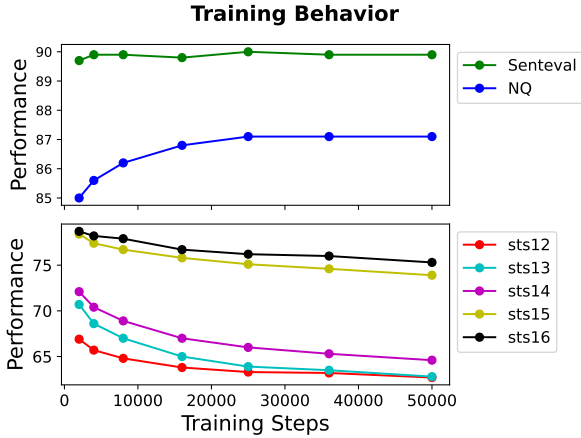
Figure 4. Performance of M (1.2B) `cpt-text` model on classification, search and sentence similarity tasks at different training steps. While the performance on search and classification improves with longer training, the performance on sentence similarity degrades.

via normalizing flows. The whitening operation is another alternative operation to improve the isotropy of the embedding space (Su et al., 2021). It is typical to initialize such models with a pre-trained language model (Devlin et al., 2019) before training on NLI datasets.

Several methods have been studied for unsupervised or self-supervised sentence embedding learning (Logeswaran & Lee, 2018; Zhang et al., 2020; Gao et al., 2021). Common approaches consider sentences within the same context as semantically similar samples (Kiros et al., 2015; Logeswaran & Lee, 2018). To create positive training pairs with augmented samples, a diverse set of text augmentation operations have been explored, including lexicon-based distortion (Wei & Zou, 2019), synonym replacement (Kobayashi, 2018), back-translation (Fang & Xie, 2020), cut-off (Shen et al., 2020) and dropout (Gao et al., 2021). However, unsupervised sentence embedding models still perform notably worse than supervised sentence encoders.

Large-scale text search based on dense embeddings and neural information retrieval (neural IR) have the potential to generalize better than keyword matching in classic IR systems. Neural IR systems encode documents at the indexing stage and then perform nearest neighbor search (Johnson et al., 2019) at query time (Lin et al., 2021). Neural IR models are usually learned by fine-tuning a pre-trained language model on supervised search corpus (Lee et al.; Guu et al., 2020; Karpukhin et al., 2020b; Lewis et al., 2020). Many SOTA search models combine classical IR with neural IR in a staged setup, where the candidates are first narrowed down by BM25 keyword search (Robertson, 2009) and then re-ranked by joint query and document

neural encoders (Nogueira et al., 2019b; Qu et al., 2021). Xiong et al. (2020) proposed ANCE, a contrastive learning framework for learning text representations for dense retrieval using mined hard negatives. Other unsupervised retriever methods use the Inverse Cloze Task or masked salient spans to achieve significant improvement on ODQA tasks (Sachan et al., 2021). In comparison to most prior work, we find that with a large enough batch size, it is possible to achieve good search performance without using supervised data. Finally, the recently published Contriever (Izacard et al., 2021) is most similar to our work on learning text embeddings for text search using contrastive learning on unlabeled data.

Semantic code search refers to the task of retrieving code relevant to a query in natural language. The CodeSearchNet challenge (Husain et al., 2020) presents a set of benchmark code search tasks in different programming languages, as well as a simple baseline model to predict embeddings of query and code via contrastive learning on a dataset of (text, code) pairs. ContraCode (Jain et al., 2021) uses a contrastive learning task of identifying functionally similar programs, where the functionally similar samples are generated via source-to-source compiler transformations. CodeBERT (Feng et al., 2020) learns to predict semantic similarity with a pre-trained language model and GraphCodeBERT (Guo et al., 2021) further improves the performance on the CodeSearchNet benchmark by adding pre-training tasks on code structure.

5. Broader Impacts

Prior research has shown that text representation models encode the biases present in their training data, including those which are discriminatory towards protected groups such as Black people or women (Bolukbasi et al., 2016; Caliskan et al., 2017; May et al., 2019; Zhao et al., 2018; Rudinger et al., 2018). Biases encoded in embedding models may cause representational harms[3] by reinforcing existent societal biases in the text corpus, and further propagating them in downstream tasks of embedding models.

Therefore, we encourage further research on two research agendas: (a) developing robust evaluation methodologies for multiple classes of bias in training data and pre-trained models, and (b) developing and improving methods for mitigating encoded bias, including fine-tuning to reduce bias in pre-trained models (Caliskan et al., 2017; May et al., 2019; Bolukbasi et al., 2016; Liang et al., 2020; Park et al., 2018; Solaiman & Dennison, 2021). Until we have robust evaluation methodology, it is important to restrict and monitor the use of the model in downstream applications. Par-

---

[3]Representational harms occur when systems reinforce the subordination of some groups along the lines of identity, e.g. stereotyping or denigration (Crawford, 2017).

ticularly for those where risk of representational harm is great and those where biased representations may influence the allocation of resources and opportunities to people.

Our embedding models are trained with large batch sizes and require substantial computation resources. While this training regime is environmentally and computationally costly, there are promising paths forward to amortize and offset these costs while allowing users to benefits from the capabilities of these models. For example, safe public access to large pre-trained language models, and efficient training pipelines that leverage improved model architectures and training schemes. We encourage further research and implementation efforts in these areas.

6. Conclusion

We showed that contrastive pre-training on unsupervised data with a sufficiently large batch size can lead to high quality vector representations of text and code. Our models achieved new state-of-the-art results in linear-probe classification, text search and code search. We find that our models underperformed on sentence similarity tasks and observed unexpected training behavior with respect to these tasks. Finally, we discussed the broader impact of our work on society.

References

Bengio, Y., Courville, A. C., and Vincent, P. Representation learning: A review and new perspectives. *Transactions on pattern analysis and machine intelligence*, 35 (8), 2012.

Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., and Kalai, A. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. 29, 2016.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2015.

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., and Mercer, R. L. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480, 1992.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei,

D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Caliskan, A., Bryson, J. J., and Narayanan, A. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML)*, 2020.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.

Conneau, A. and Kiela, D. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018.

Crawford, K. The trouble with bias. Keynote at NeurIPS, 2017.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. ACL, 2019.

Fang, H. and Xie, P. CERT: contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020.

Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., and Zhou, M. Codebert: A pre-trained model for programming and natural

languages. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Formal, T., Lassance, C., Piwowarski, B., and Clinchant, S. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021.

Gao, T., Yao, X., and Chen, D. SimCSE: Simple contrastive learning of sentence embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Giorgi, J. M., Nitski, O., Bader, G. D., and Wang, B. Declutr: Deep contrastive learning for unsupervised textual representations. In *Proceedings of ACL/IJCNLP*, 2020.

Goodman, N. *Seven strictures on similarity*. Bobbs Merrill, 1972.

Guo, D., Ren, S., Lu, S., Feng, Z., Tang, D., Liu, S., Zhou, L., Duan, N., Svyatkovskiy, A., Fu, S., Tufano, M., Deng, S. K., Clement, C. B., Drain, D., Sundaresan, N., Yin, J., Jiang, D., and Zhou, M. Graphcodebert: Pre-training code representations with data flow. In *International Conference on Learning Representation (ICLR)*, 2021.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Conference on Artificial Intelligence and Statistics*. PMLR, 2010.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. REALM: retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 1735–1742. IEEE, 2006.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Hofstätter, S., Lin, S., Yang, J., Lin, J., and Hanbury, A. Efficiently teaching an effective dense retriever with balanced topic aware sampling. *arXiv preprint arXiv:2104.06967*, 2021.

Husain, H., Wu, H.-H., Gazit, T., Allamanis, M., and Brockschmidt, M. CodeSearchNet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*, 2020.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Jain, P., Jain, A., Zhang, T., Abbeel, P., Gonzalez, J. E., and Stoica, I. Contrastive code representation learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V., Sung, Y., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Conference of the Association for Computational Linguistics (ACL)*. ACL, 2017.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020a.

Karpukhin, V., Oguz, B., Min, S., Wu, L., Edunov, S., Chen, D., and Yih, W. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020b.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Kim, W., Son, B., and Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *International Conference on Learning Representation (ICLR)*, 2014.

Kiros, J., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. Skip-thought vectors. In *Advances in Neural Information Processing Systems (NeuriPS)*, 2015.

Kobayashi, S. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Lee, K., Chang, M., and Toutanova, K. Latent retrieval for weakly supervised open domain question answering. In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Conference of the Association for Computational Linguistics (ACL)*, pp. 6086–6096. ACL.

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeuriPS)*, 2020.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. On the sentence embeddings from pre-trained language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Liang, P. P., Li, I. M., Zheng, E., Lim, Y. C., Salakhutdinov, R., and Morency, L. Towards debiasing sentence representations. In *Conference of the Association for Computational Linguistics (ACL)*, 2020.

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

Logeswaran, L. and Lee, H. An efficient framework for learning sentence representations. In *International Conference on Learning Representation (ICLR)*, 2018.

Lu, J., Batra, D., Parikh, D., and Lee, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

May, C., Wang, A., Bordia, S., Bowman, S. R., and Rudinger, R. On measuring social biases in sentence encoders. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Nogueira, R., Lin, J., and Epistemic, A. From doc2query to docttttttquery. *Online preprint*, 2019a.

Nogueira, R., Yang, W., Cho, K., and Lin, J. Multistage document ranking with BERT. *arXiv preprint arXiv:1910.14424*, 2019b.

Park, J. H., Shin, J., and Fung, P. Reducing gender bias in abusive language detection. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Pennington, J., Socher, R., and Manning, C. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of NCAAL/IJCNLP*, 2018.

Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, X., Dong, D., Wu, H., and Wang, H. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Conference of the Association for Computational Linguistics (ACL)*, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Robertson, S. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 2009.

Rudinger, R., Naradowsky, J., Leonard, B., and Durme, B. V. Gender bias in coreference resolution. *arXiv preprint arXiv:1804.09301*, 2018.

Sachan, D. S., Patwary, M., Shoeybi, M., Kant, N., Ping, W., Hamilton, W. L., and Catanzaro, B. End-to-end training of neural retrievers for open-domain question answering. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of ACL/IJCNLP*, pp. 6648–6662. ACL, 2021.

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

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

Shen, D., Zheng, M., Shen, Y., Qu, Y., and Chen, W. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*, 2020.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NeuriPS)*, 2016.

Solaiman, I. and Dennison, C. Process for adapting language models to society (PALMS) with values-targeted datasets. *arXiv preprint arXiv:2106.10328*, 2021.

Su, J., Cao, J., Liu, W., and Ou, Y. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. Videobert: A joint model for video and language representation learning. In *International Conference on Computer Vision (ICCV)*, 2019.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems (NeuriPS)*, 2021.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multi-view coding. *European Conference on Computer Vision (ECCV)*, 2019.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeuriPS)*, 2017.

Vervaeke, J., Lillicrap, T. P., and Richards, B. A. Relevance realization and the emerging framework in cognitive science. *Journal of logic and computation*, 22(1):79–99, 2012.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*, 2020.

Wei, J. W. and Zou, K. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. ACL, 2018.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance-level discrimination. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

Yih, W.-t., Toutanova, K., Platt, J. C., and Meek, C. Learning discriminative projections for text similarity measures. In *Conference on Computational Natural Language Learning (CoNLL)*. ACL, 2011.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, 2021.

Zhang, Y., He, R., Liu, Z., Lim, K. H., and Bing, L. An unsupervised sentence embedding method by mutual information maximization. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*, 2018.