Learning Effective Representations for Retrieval Using Self-Distillation with Adaptive Relevance Margins

Lukas Gienapp
Leipzig University & ScaDS.AI

Niklas Deckers
University of Kassel & ScaDS.AI & hessian.AI

Martin Potthast
University of Kassel & ScaDS.AI & hessian.AI

Harrisen Scells
Leipzig University

ABSTRACT

Representation-based retrieval models, so-called biencoders, estimate the relevance of a document to a query by calculating the similarity of their respective embeddings. Current state-of-the-art biencoders are trained using an expensive training regime involving knowledge distillation from a teacher model and batch-sampling. Instead of relying on a teacher model, we contribute a novel parameter-free loss function for self-supervision that exploits the pre-trained language modeling capabilities of the encoder model as a training signal, eliminating the need for batch sampling by performing implicit hard negative mining. We investigate the capabilities of our proposed approach through extensive ablation studies, demonstrating that self-distillation can match the effectiveness of teacher distillation using only 13.5% of the data, while offering a speedup in training time between 3x and 15x compared to parametrized losses. Code and data is made openly available.[1]

1 INTRODUCTION

In information retrieval, transformer-based bi-encoders are used as effective retrieval models. They estimate a document's relevance to a query by computing the similarity of their embeddings. Queries and documents are represented independently, which allows for pre-computing and indexing all document representations offline and computing only the query representation at retrieval time. bi-encoders are thus appealing in practice since they can be easily scaled. In contrast, transformer-based cross-encoders are a type of model that estimates relevance jointly by computing a score for each document-query pair at retrieval time, achieving higher effectiveness than bi-encoders at the expense of a higher computational cost and inference latency. They are thus usally reserved for multi-stage reranking, such that only the top-k documents of an initial ranking produced by a more efficient bi-encoder are re-ranked.

However, highly effective bi-encoders come at the cost of a highly expensive training regime based on knowledge distillation. Figure 1 (left) illustrates this process, where the relevance of each training sample is first estimated by a cross-encoder teacher model. The estimated relevance scores are first used for a computationally intensive batch sampling process, where an optimal distribution of relevance scores within each batch is ensured according to the teacher model, which then provides supervision for training a bi-encoder student model with a margin-based loss function. While this knowledge distillation setup yields highly effective bi-encoder ranking models, we highlight three drawbacks: **(1)** A teacher model must be available for the desired dataset and domain. cross-encoder models are usually chosen as teacher models to maximize effectiveness [13]. Training a cross-encoder model for knowledge distillation is computationally expensive and may not be possible for domains with little ground-truth ranking data. **(2)** Teacher scores must be inferred for every training sample, requiring a forward pass. This adds inference cost onto a process that is already prohibitively expensive. Improving training efficiency is crucial for information retrieval, as the majority of energy spent over the lifetime of a model, especially models that are efficient at retrieval time, goes into training [25]. **(3)** Batch sampling prohibits continuous learning on new data. The current best batch sampling procedures rely on clustering the training data set [13]. A static training dataset is thus required, rendering it challenging to adapt such approaches in settings where training data is collected continuously, such as online learning-to-rank.

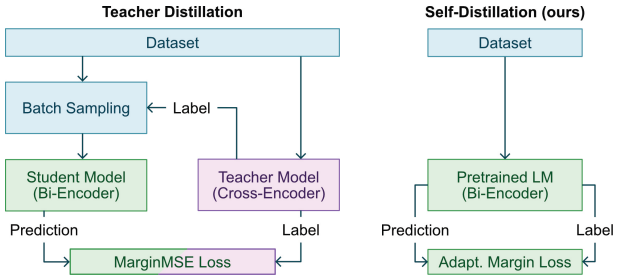

Figure 1: Conceptual overview of a typical bi-encoder teacher distillation training regime (left) and our proposed self-distillation approach (right). Blue is the data, green is the bi-encoder and purple is the cross-encoder model.

In this paper, we propose a novel training regime and loss function for bi-encoders (Figure 1, right) that eliminates the complexities of previous approaches while maintaining competitive effectiveness. Instead of using a teacher model, we leverage the pre-trained text similarity capabilities of the encoder model to provide a supervision signal in a self-distillation setup. This eliminates the need for expensive exhaustive inference over the training dataset, does not rely on batch sampling techniques, and is highly data efficient. We then conduct systematic ablation studies which demonstrate that self-distillation can match the effectiveness of teacher-distillation based training regimes, with nDCG@10 and Recall@1000 scores of no significant difference, while using only 13.5% of the data. Finally, our proposed loss function is hyperparameter free, eliminating the need for expensive hyperparameter tuning.

[1] github.com/webis-de/adaptive-relevance-margin-loss

2 PRELIMINARIES & RELATED WORK

We define Q as a set of queries representing user information needs and D as a set of documents potentially containing relevant information. The objective of a ranking function is to order the documents $d \in D$ based on their relevance to each query q, accomplished with a relevance model $\rho : Q \times D \rightarrow \mathbb{R}$, which assigns relevance scores that establish an order. We summarize related work on operationalizing ρ with representation learning, covering encoder models (Section 2.1), knowledge distillation (Section 2.2), loss functions (Section 2.3), and negative mining (Section 2.4).

2.1 Relevance Estimation with Encoder Models

Text encoder models like BERT [5] operationalize ρ in two main ways: bi-encoders and cross-encoders [33]. Bi-encoders independently embed the query q and each document d using an encoder model f_θ, typically by using the representation of the special `[CLS]` token (optionally processed by a linear layer) [2, 7], and express relevance in terms of a similarity function ϕ as $\rho = \phi(f_\theta(q), f_\theta(d))$ [2, 8, 17]. In contrast, cross-encoders process both query and document jointly, operationalizing ρ as a relevance classification of the joint representation [22]: $\rho = c(f_\theta(q \oplus d))$, where $\oplus$ is string concatenation and $c : \mathbb{R}^D \rightarrow [0, 1]$ is a binary classifier assigning a probability. While bi-encoders are more efficient [33], cross-encoders capture complex interactions between query and document [22], yielding higher effectiveness albeit with higher latency at query time, as document representations cannot be precomputed. We focus our investigation on training bi-encoders with comparable effectiveness to cross-encoders.

2.2 Knowledge Distillation

Knowledge distillation [12] aims to train a student model to reproduce the output of a teacher model [15]. It is commonly employed for model compression, i.e., to develop smaller and faster models capable of maintaining effectiveness at reduced computational effort [9]. In retrieval tasks, distillation, as seen in works by Hofstätter et al. [13, 14], trains bi-encoder models using precomputed relevance scores from a cross-encoder teacher model. Although distillation can successfully impart ranking effectiveness to bi-encoders, the use of cross-encoder teachers imposes constraints on the training process due to the computational cost of pre-computing relevance scores for a large set of query-document pairs. Thus, while distillation produces highly efficient student models, it incurs high computational costs for training.

Self-distillation is a subfield of knowledge distillation, where the student and teacher models are identical [11], i.e., leveraging a model's own output to provide a training signal for a refined version of itself. The success of this approach is attributed to the rich output distribution, capturing additional information about training examples [20]. While self-distillation is predominantly applied when teacher and student are trained towards the same task, it can also be used in a transfer setting [34]. Here, the pretrained capabilities of the model on a general task are exploited to train a more refined version of itself on a specialized target task. Self-distillation reduces the required computational effort, as no additional teacher inference is needed. We take inspiration from prior work in the literature about self-distillation to apply to the task of ranking.

2.3 Bi-Encoder Loss Functions

In the typical training setup for bi-encoder models, data comes in the form of triplets (q, d^+, d^-), comprising a query, a positive document, and a negative document. Various loss functions have been proposed within this setup: for instance, triplet loss aims to adjust the representation space so that the positive document d^+ is closer to the query q than the negative document d^- by a certain margin ε [26]. Karpukhin et al. [17] minimize the negative log-likelihood between a single positive example and multiple negative examples. Xiong et al. [31] minimize a classification loss function like binary cross-entropy, hinge loss, or negative log-likelihood between scores predicted for pairs of positive and negative examples. All aim to rearrange representations to prioritize relevance, i.e., learn a representation space in which for a query, a relevant passage is closer than a non-relevant passage by a margin ε.

In knowledge distillation contexts, teacher scores provide a source of supervision. The MarginMSE loss [13] adopts the concept of optimizing the mean squared error between the relevance difference, i.e., $\phi(q, d^+) - \phi(q, d^-)$, and a desired target ε. The value of ε is dictated individually for each triplet by its predicted teacher model output. This notion of introducing an 'adaptive' target value for each instance also has broader applications beyond distillation. For instance, Li et al. [18] use class distances in an image classification problem as adaptive target, Ha and Blanz [10] use differences in ground-truth labels as adaptive targets for learning to rank, and Sharma et al. [27] use the similarity between augmented views of the same sample. We apply self-distillation to predict an adaptive target for each training triplet using the bi-encoder model itself.

2.4 In-Batch Negatives & Hard Negative Mining

To increase the efficiency of triplet-based losses, additional training instances can be formed by considering documents from other triplets in the same batch as supplementary negative examples for a given query-positive pair [32]. Yet, a major caveat is that f_θ rapidly learns to correctly project trivial triplets, i.e., triplets where d^+ and d^- are easy to distinguish w.r.t. their relevancy. Consequently, identifying 'hard' triplets, where distinguishing between positive and negative documents is challenging, becomes crucial for learning a robust latent space [32]. Yet, in-batch negatives are likely topically unrelated to a given query, and thus have limited learning potential.

To alleviate this issue, batch sampling techniques have been devised that select batches of triplets such that the positive and negative documents are hard to distinguish in relevance, both within each pair and across pairs of in-batch negatives [1]. For retrieval, Hofstätter et al. [14] introduce balanced topic-aware sampling (TAS-B), where triplets are chosen to ensure that the relevance differences of query-positive and query-negative pairs across all training pairs in a batch adhere to a certain distribution. All triplets in the training dataset are put into bins based on their margin, and then batches are constructed by uniformly selecting triplets from all bins. A reproduction study by Wang and Zuccon [29] found that further effectiveness is possible by discarding bins with high relevance differences, removing 'easy' negatives from consideration. By predicting adaptive targets, we propose that self-distillation can be used to perform 'implicit' negative sampling based on the magnitude of the target without explicitly sampling batches.

3 SELF-DISTILLATION WITH ADAPTIVE RELEVANCE MARGINS

We formulate three criteria to simplify the training process and increase the training efficiency of bi-encoders for retrieval and address the drawbacks of teacher-distillation approaches outlined before: **(1)** no teacher model for target estimation: the loss function should only rely on the information contained in provided training triplets, **(2)** no batch sampling procedure: the loss function should be effective with randomly ordered data, **(3)** no hyperparameters: the loss function should be applicable without computationally intensive hyperparameter tuning. In this section, we introduce *adaptive relevance margins*, a novel self-distillation loss function for retrieval fine-tuning. We motivate it starting from how a margin loss is traditionally calculated, and deriving several variants for the inner term of the loss incrementally.

Static Targets. The traditional margin loss for a triplet (q_i, d_i^+, d_i^-) of query, positive (relevant) document, and negative (non-relevant) document contrasts the query relevance margin, i.e., the difference between the cosine similarity $\phi(q_i, d_i^+)$ of a query and a positive document and the cosine similarity $\phi(q_i, d_i^-)$ between query and a negative document, with a target value ε [14]:[2]

$$l_i = \underbrace{\phi\left(q_i, d_i^+\right) - \phi\left(q_i, d_i^-\right)}_{\text{Relevance margin}} \underbrace{- \varepsilon}_{\text{Target}} \tag{1}$$

This variant optimizes for a latent space where positive and negative documents are separated by a value of at least ε w.r.t. their query similarity. However, ε must be tuned for the best effectiveness.

Adaptive Targets. Rather than using the same ε value as a target across all training instances, knowledge distillation allows one to derive per-document target scores using a teacher model [14]. We propose using the bi-encoder model itself as a source for this teacher signal in a self-distillation setup. As BERT-style encoders are pretrained with a masked token or sentence prediction objective, they exhibit strong capabilities for measuring text similarity [5, 19]. We can, therefore, replace the ε with the similarity between positive and negative documents for a query, scaled to the 0-1 range:

$$l_i = \underbrace{\phi\left(q_i, d_i^+\right) - \phi\left(q_i, d_i^-\right)}_{\text{Relevance margin}} - \underbrace{\left(\frac{1 + \phi(d_i^+, d_i^-)}{2}\right)}_{\text{Scaled doc. sim.}} \tag{2}$$

Rescaling the target is warranted since we desire a latent space where positive and negative documents are orthogonal. This adaptive margin may initially seem counterintuitive as the model learns to predict a larger difference in relevance for similar texts. However, this can be framed from the perspective of hard negatives: if the positive and negative document are similar, i.e., $(1 + \phi(d^+, d^-))/2 \approx 1$, they are a 'hard' training instance. This larger margin leads to an increased contribution to the overall loss and, thus, higher impact on the model parameters compared to a dissimilar positive and negative document pair, i.e., $(1+\phi(d^+, d^-))/2 \approx 0$. Adaptive targets thus implicitly perform hard negative mining without hyperparameters.

[2]We explicitly omit writing the representation function f_θ in our notation for brevity, therefore q, d already signify query/document representations.

Distributed Targets. So far, the loss calculation has used only a single (q_i, d_i^+, d_i^-) triple. While naïve in-batch negative variants l_{ij} of the static and adaptive targets losses can be derived by replacing d_i^- with multiple $d_j^- \in D^-$, this yields sub-par results if not combined with batch sampling techniques. Instead, we extend the adaptive targets approach to approximate this sampling by using multiple different target values for each individual training triple.

$$l_{ij} = \underbrace{\phi\left(q_i, d_i^+\right) - \phi\left(q_i, d_i^-\right)}_{\text{Relevance margin}} - \underbrace{\left(\frac{1 + \phi(d_i^+, d_j^-)}{2}\right)}_{\text{Scal. doc. sim. distrib.}} \tag{3}$$

The intuition behind this approach is that the target now forms a distribution over the negative documents in the batch. Instead of comparing each triplet in the batch with a single target, we repeatedly compare the margin to the original negative with targets from all of the negatives in a batch. In other words, these repeated comparisons estimate the average scaled similarity of a positive document to (a sample of) the population of negative documents. If this similarity is high ($\approx$ 1), the sample contains documents that are hard to distinguish w.r.t. relevancy and thus contains rich information to establish a positive-negative class separation. If this similarity is low ($\approx$ 0), the sample contains documents that are easy to distinguish w.r.t relevancy. A high in-batch negative document similarity translates to a large margin to maximize the impact on the loss, and vice versa.

While adaptive targets opt to use in-batch information to increase the count of total available training triples, distributed targets instead exploit the in-batch distribution to make a more accurate estimation of the target for each original triple. One can also consider the distributed targets approach as optimizing for the implicit mean document distance. The reason optimizing it implicitly works and optimizing it explicitly, i.e., by taking the mean scaled document similarity within the loss as target, does not work is because optimizing it explicitly repositions only d_i^- in latent space rather than all $d_j^- \in D^-$ as the implicit mean does.

Error Function. To form a proper loss function, the previously derived inner terms l_{ij} are optimized using mean squared error (MSE), which calculates the average squared difference of the relevance margin and target over all samples in a batch. A batch consists of a set of queries Q, a set of associated positive (relevant) documents D^+, and a set of associated negative (non-relevant) documents D^-, with batch size $B = |Q| = |D^+| = |D^-|$. For all variants l_i not relying on in-batch information, the error is calculated as

$$\mathcal{L}_{MSE}\left(Q, D^+, D^-\right) = \frac{1}{B}\sum_{i=1}^{B}\left(l_i\right)^2 \tag{4}$$

For variants l_{ij} using in-batch information it is calculated as

$$\mathcal{L}_{MSE}\left(Q, D^+, D^-\right) = \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B}\left(l_{ij}\right)^2 \tag{5}$$

Summary. Figure 2 illustrates the behaviour of all three loss variants. The position of a query, a positive document, and negative documents are visualized in cosine similarity space. Green elements

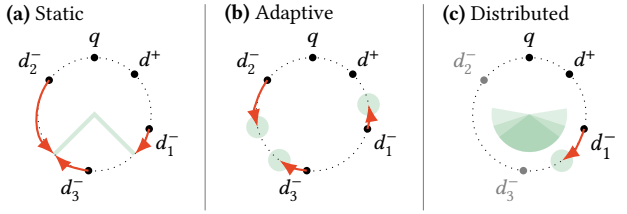

Figure 2: Conceptual illustration of target variants in cosine similarity space. Black dots indicate location on the unit sphere. Green elements indicate target values; red arrows indicate induced shifts in latent space for negative documents.

indicate the target values and red arrows indicate the repositioning in latent space induced by the respective loss. We only illustrate repositioning of negative documents for simplicity; in practice, all three, query, positive, and negative could be simultaneously affected. For **(a)** static targets, a relevance margin of ε is forced (green lines); for **(b)** adaptive targets, individual relevance margins for each negative are estimated based on document similarity (green dots); and for **(c)** distributed targets, for a single negative, the distribution of the document similarities of all negatives (green area) is used as margin (green dot).

4 EXPERIMENTAL SETUP

Data. Both training and evaluation use the MSMARCO-Passage collection [21]. For training, we use the official set of 39 780 811 training triplets, each comprising a query, positive, and negative text passage. For evaluation, we employ relevance annotations from the TREC-DL19 [4] and TREC-DL20 [3] datasets, which contain 43 and 54 densely judged queries for the MSMARCO-Passage collection, respectively. Since TREC annotations include graded relevance (ranging from 0 for non-relevant to 3 for perfect), we use the recommended binarization level of $r > 1$ [4] for all binary metrics. Text is tokenized with a maximum sequence length of 30 for queries and 200 for passages [14].

Implementation and Hyperparameters. We implement our experiment pipeline in PyTorch [23] using the Lightning [6] and Hugging Face [30] ecosystem. We fine-tune several pretrained BERT-style encoder models, which vary in parameter count, pretraining regime, and embedding kind. The full list is provided in Table 1. Text representations are computed using the `[CLS]` representation of each document, optionally processed by a linear layer. All models produce 768-dimensional embeddings. We train using the ADAM optimizer with a learning rate of $5e^{-6}/B$, a weight decay rate of $1e^{-6}$, and batch sizes $B \in [64, 128]$. Training uses a single Nvidia A100 40GB GPU. To improve training efficiency, we use mixed-precision weights and quantized optimizer states. We provide fully reproducible code and supplementary run files for trained models. [3]

Evaluation. We study two evaluation setups: reranking and full ranking. In the reranking setup, which is used for evaluating ablation runs with low overhead, we rerank the top-1000 documents from the official TREC-DL19/20 BM25 baseline runs for each query. For the full ranking setup, replicating real-world use cases, we embed and index the entire MSMARCO-Passage collection using a FAISS [16] HNSW index with optimized 32-byte product quantization, and perform exact re-ranking of the top-k results. In both setups, we use cosine similarity between query and document representations for relevance estimation. Similar to prior work, in the full ranking setup we report the measures nDCG@10, representing ranking-oriented tasks, and Recall@1000, representing retrieval-oriented tasks. However, in the reranking setup, the BM25 baseline run artificially restricts the recall obtainable. Therefore, we use Hits@100 as a suitable surrogate for recall that does not rely on all relevant documents being retrievable for each topic.

[3]Code: github.com/webis-de/adaptive-relevance-margin-loss
Run files: zenodo.org/records/11197962

Table 1: Overview of pretrained models used for finetuning, with № parameters, embedding kind and № dimensions.

Model		№ Params	Embedding	№ Dim.
RoBERTa	🤗 [19]	124M	`[CLS]` + Linear	768
MPNet	🤗 [28]	109M	`[CLS]` + Linear	768
distilBERT	🤗 [24]	66M	`[CLS]`	768

Baselines. To contextualize the results of our proposed loss functions, we compare against three different baseline models: **(1)** BM25 given its nature as standard retrieval baseline and reranking input in our reranked evaluation setting; **(2)** a BERT-based bi-encoder fine-tuned with a static margin of $\varepsilon = 1$ [14] **(3)** a BERT-based bi-encoder finetuned with TAS-B batch sampling and teacher margins [14, 29].

5 RESULTS & DISCUSSION

We conduct four experiments to study the benefits of using adaptive and distributed margins for fine-tuning bi-encoder retrieval models. First, we conduct an ablation study on static margins to see how adjusting the parameter ε affects effectiveness, with and without in-batch negatives (Section 5.1). Second, we study the effectiveness of adaptive margins, including ablation studies on batch size and use of in-batch negatives (Section 5.2). Third, we repeat this analysis for distributed margins (Section 5.3). Finally, we compare the optimal parameter settings of each loss variant found in the previous analyses (Section 5.4), and compare them to other state-of-the-art and baseline models (Section 5.5). This comparison provides insights into the overall effectiveness of our proposed approach.

5.1 Static Targets

In the first ablation experiment, we conduct an exhaustive grid search to find the optimal static target hyperparameter ε. We fine-tune each of the three pretrained encoders using the static target loss variant, while varying ε in the range of 0 to 1 in steps of 0.1. To make this exhaustive grid search feasible in terms of computation time, we train each configuration on the same randomly sampled subset of MSMARCO-Passage, containing 2.56 M triplets. Evaluation scores are calculated in the reranking setting. The maximum batch size possible across all models on the available hardware was used ($B = 128$). Experiments with smaller batch size ($B = 64$) showed the same trends and relative differences.

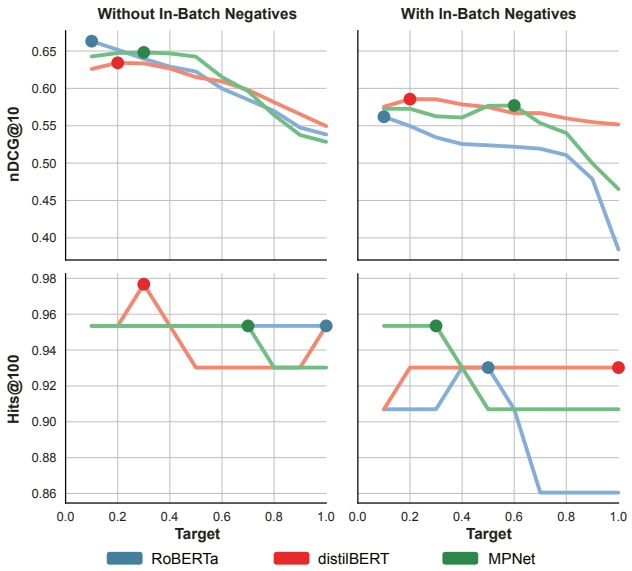

Figure 3: nDCG@10 (upper row) and Hits@100 scores (lower row) on TREC-DL19 for different static target values, without (left column) and with (right column) using in-batch negatives. Dots mark maxima.

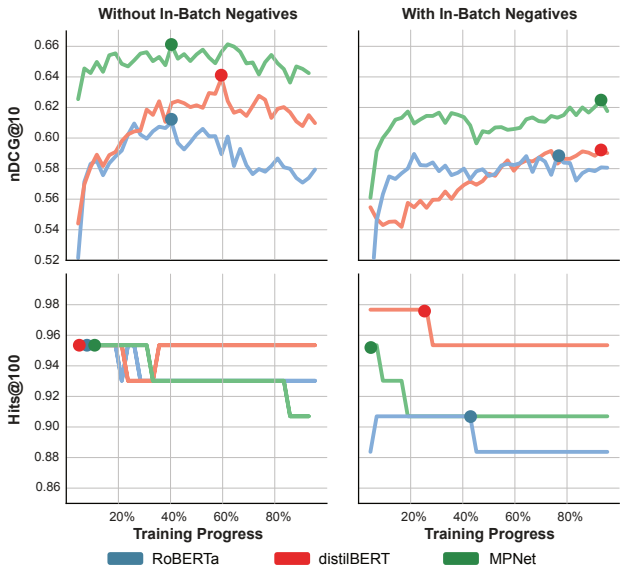

Figure 4: Evolution of nDCG@10 (upper row) and Hits@100 scores (lower row) for adaptive targets on TREC-DL19 over 1 epoch using with and without in-batch negatives (columns). Dots mark maxima

Figure 3 shows evaluation scores for each combination of ε and model, with and without using in-batch negatives. Four effects are apparent: **(1)** Tuning ε as a hyperparameter yields improvements across all configurations compared to the default choice of $\varepsilon = 1$. Moreover, the improvement between the default and optimal configuration is distinct, surpassing 0.1 nDCG@10 in almost all cases. Consequently, margin optimization is necessecary to extract maximum effectiveness in each setup. **(2)** The optimal value of ε varies depending on the model; each model achieves its highest evaluation score at different points. This indicates that hyperparameter optimization needs to be repeated for each pretrained model, and there is no universal choice for retrieval finetuning as a whole. **(3)** The optimal margin varies depending on the task; all tested models show a different ε-value at which the respective evaluation measure, i.e., nDCG@10 or Hits@100, is maximized. **(4)** Using in-batch negatives does not yield an improvement in effectiveness. For all three tested models and both measures, the effectiveness without in-batch negatives is the same or higher than with in-batch negatives. This suggests that the models do not benefit from the additional training data, likely because it is noisy and provides little information, as negatives in randomly sampled batches are 'easy'.

The effectiveness of the static target loss variant, whether with or without in-batch negatives, is highly susceptible to the training setup used. Factors such as model choice, value of ε, and measure result in varying levels of effectiveness and exhibit complex interdependencies. At the same time, substantial gains in effectiveness can be achieved through hyperparameter tuning. However, negative mining techniques appear necessary to effectively use in-batch information.

5.2 Adaptive Targets

In the second ablation experiment, we examine various configurations of the adaptive target approach. Figure 4 shows the evaluation scores computed at every 1000^{th} step during the training process. The training uses the entire MSMARCO-Passage collection. Evaluation scores are calculated in the reranking setting with a batch size of $B = 128$; similar trends were observed for $B = 64$.

Two key observations can be made: **(1)** Using adaptive targets is data efficient. The maximum nDCG@10 score is attained after processing approximately half of the data. However, effectiveness declines thereafter, suggesting a divergence from the target task. This issue could potentially be addressed through learning rate scheduling or by terminating training early. **(2)** The use of in–batch negatives produces mixed results in terms of effectiveness. Regarding nDCG@10, across all configurations, the achieved score is lower when using in-batch negatives, and data efficiency is reduced. The maximum score is reached much later in the training process. However, for Hits@100, only RoBERTa does not benefit from the additional information, and the negative impact on data efficiency is diminished. Adaptive targets can be effectively employed as a hyperparameter-free replacement of the static target variant, yielding comparable evaluation scores. It demonstrates more retrieval-favored characteristics (Hits@100) than the static variant, albeit at the expense of ranking accuracy (nDCG@10).

5.3 Distributed Targets

In the third ablation experiment, we examine the distributed targets variant. Figure 5 shows evaluation scores calculated in the reranked setting for all three pretrained models. It explicitly includes ablation regarding batch size, i.e., negative sample count.

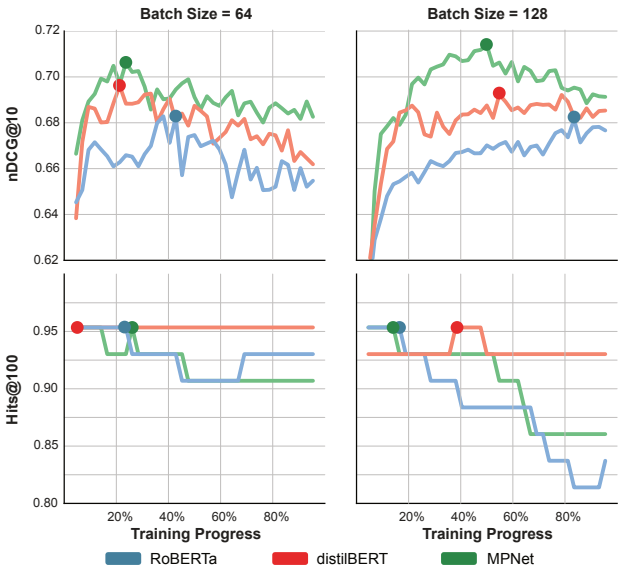

Figure 5: Evolution of nDCG@10 (upper row) and Hits@100 scores (lower row) for distributed targets per batch size (columns) on TREC-DL19 over 1 epoch. Dots mark maxima.

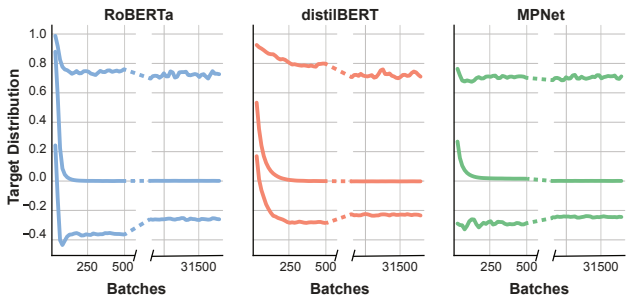

Figure 6: Evolution of mean (middle line), max (upper line) and min (lower line) of target distribution throughout the training process for 1 epoch. Values smoothed by rolling mean over 32 batches. First and last 500 batches shown.

Three key observations can be made for the distributed targets variant: **(1)** it converges towards the maximum nDCG@10 score even quicker than the adaptive targets variant (see Figure 4). Batch size influences convergence, with smaller batches yielding a more data-efficient result. **(2)** The use of in-batch information leads to improved results compared to previous in-batch static or adaptive variants. Specifically, nDCG@10 is highest in the distributed setup. **(3)** Absolute maximum scores are only marginally affected by batch size, with larger batch sizes yielding slightly higher nDCG@10 scores in two out of the three fine-tuned models. Distributed targets also shows to be a robust hyperparameter-free replacement for static targets. They exhibit complementary behavior to adaptive targets, showcasing better ranking accuracy (nDCG@10) while maintaining similar retrieval characteristics (Hits@100).

Table 2: Example of training instance with query (q), positive (d^+) and negative passages with highest (d^-_{max}), closest to mean (d^-_{mid}), and lowest (d^-_{min}) respective target score (t).Predicted relevance (ρ) and resulting instance loss (l^2_{ij}) given.

	Text	ρ	t	l^2_{ij}
q	Where is wild rice grown in USA?	—	—	—
d^+	Wild rice (Z. aquatica), also an annual, grows in the Saint Lawrence River […]	0.79	—	—
d^-_{max}	TIvan L. Sander. Northern red oak (Quercus rubra), also known as common […]	0.34	0.69	0.06
d^-_{mid}	Your answer will appear in the acres field. How to convert acres to ft2 […]	0.06	0.51	0.05
d^-_{min}	Colorado Springs Municipal (Colorado Springs, CO) Right now, 14 airlines […]	0.12	0.46	0.04

To contextualize these findings, Figure 6 illustrates the training characteristics of the distributed target variant. It plots the mean, minimum, and maximum values of the $D^+ \times D^-$ similarity matrix throughout the training process. **(1)** The in-batch target distribution rapidly settles to a mean of 0 (rescaled to 0.5 for loss computation), suggesting that non-relevant documents are, on average, orthogonal in latent space to relevant documents. This indicates an ideal class separation in cosine similarity space. **(2)** The distribution features outliers in the positive spectrum, i.e., the maximum value (upper line) is further apart from the mean (middle line) than the minimum value (lower line). Consequently, documents highly similar to the relevant one of a query are less common than dissimilar ones, approaching the expected imbalanced ratio of relevant to irrelevant documents for a query. **(3)** While the initial pretrained similarity distribution, i.e., the first 100 steps, differs significantly among the three pretrained models, their finetuned distributions are largely identical. This suggests that a robust latent space for retrieval can be generalized independently of pretraining regime and architecture.

Table 2 presents an exemplary excerpt of a training batch, comprising a query with its associated positive and three negative passages (maximum, minimum, and closest to the mean predicted target). Additionally, the table provides the predicted relevance (ρ), scaled target (t), and instance loss (l^2_{ij}) calculated with the fine-tuned DistilBERT model. The theorized loss properties can be observed. The relevance (ρ) of documents aligns with their target ordering, i.e., semantic document similarity, showcasing successful self-distillation. Moreover, the less related a negative document is to the positive document, the lower its contribution to the overall loss, showcasing implicit negative mining.

5.4 Comparison of Target Variants

To compare all three variants of the proposed loss function, we train a model for each using parameter choices informed by the previous ablation studies. We use a batch size of $B = 128$ and employ exponential learning rate decay ($\gamma = 0.99999$). To prevent the effectiveness degradation observed for adaptive and distributed variants, we implement an early stopping criterion. If no improvement in nDCG@10 score on TREC-DL19 calculated every 500 batches is achieved for 16 consecutive checks training is halted.

Table 3: Comparison of effectiveness per pretrained model (M), target variant (T), in-batch negatives (IB), and total training time. Maximum per column and model marked bold. † marks runs with statistically equivalent effectiveness compared to maximum.

M	T	IB	TREC-DL19		TREC-DL20		Time	Data Seen
			nDCG@10	R@1000	nDCG@10	R@1000	(hh:mm)	(# Triples)
distilBERT	S	✗	**0.647**	**0.792**	**0.632**	0.821†	15:07*	5.18M
	S	✓	0.574	0.786†	0.569	0.823†	13:30*	2.24M
	A	✗	0.537	0.664	0.542	0.735	2:36	4.93M
	A	✓	0.534	0.790†	0.513	**0.823**	1:04	1.98M
	D	✓	0.639†	0.774†	0.615†	0.809†	3:56	7.30M
RoBERTa	S	✗	**0.651**	0.771†	**0.625**	**0.822**	31:11*	6.53M
	S	✓	0.578	0.767†	0.510	0.820†	31:09*	6.53M
	A	✗	0.527	0.771	0.488	0.698	4:09	3.97M
	A	✓	0.572	**0.773**	0.492	0.812†	5:20	5.12M
	D	✓	0.634†	0.766†	0.594	0.794	6:19	6.02M
MPNet	S	✗	**0.655**	0.791†	**0.646**	0.834†	34:55*	5.63M
	S	✓	0.582	0.791†	0.549	0.835†	33:51*	4.74M
	A	✗	0.561	0.655	0.573	0.733	8:10	6.21M
	A	✓	0.605	**0.817**	0.564	**0.843**	7:29	5.95M
	D	✓	0.635†	0.761	0.622†	0.805	5:37	4.86M

S = static; A = adaptive; D = distributed. *Includes hyperparameter sweep time.

Table 3 lists the obtained evaluation scores in the full ranking setup on both TREC-DL19 and TREC-DL20 for each of the three pretrained models investigated, each of the three loss variants, and (for static and adaptive targets) with and without in-batch negatives. Statistical equivalence was tested among all approaches per model to determine if the difference of each approach to the respective best per measure and model is meaningful (paired TOST, $\epsilon_L = 0.05$, $p < 0.05$, $n = 43$ [TREC-DL19], $n = 54$ [TREC-DL20], Bonferroni correction applied per test group). Equivalence of approaches is indicated in Table 3 using a † symbol. Additionally, the table lists the number of training samples seen and the total training time per approach to compare their data/compute efficiency. For all static target variants, this includes compute time spent on hyperparameter search to identify the optimal target value.

For nDCG@10, the static target approach without in-batch negatives achieves the highest scores across all models and both datasets. However, the distributed targets approach, while scoring slightly lower in all setups is still statistically equivalent, except on TREC-DL20 with RoBERTa as the base model. For Recall@1000, the adaptive target variant with in-batch negatives achieves the highest scores across all models and both datasets, except for distilBERT on TREC-DL19, where the static approach without in-batch negatives scores higher. Additionally, in half of the test cases (distilBERT, RoBERTa on TREC-DL19), the distributed targets are statistically equivalent to the best score. Therefore, adaptive and distributed targets are suitable hyperparameter-free replacements for tuned static targets for retrieval (Recall@1000) and ranking tasks (nDCG@10).

These results also highlight a large difference in training efficiency. Since the static target setup requires hyperparameter tuning, its total training time is between 3x (distilBERT) and 6x (MPNet) higher than the distributed target approach, and between 7x and 15x higher than the adaptive target approach. When comparing efficiency in terms of training triplets seen, two further trends are identified: using in-batch information increases data efficiency, and self-distillation approaches are more data efficient than their static counterparts in all but one case (distilBERT). In-batch information is also detrimental for static variants, suggesting that our approach to implicit negative mining improves effectiveness. The observed gains in compute- and data-efficiency strengthen the applicability of our proposed approach, as it is not only on par in terms of effectiveness but also does not require large amounts of data or compute. Additionally, the choice of pretrained model does not yield meaningful differences in training outcome; across all evaluated setups, the absolute scores show no clear difference between the three pretrained models, and the relative trends concerning the individual loss variants are also stable across all pretrained models.

5.5 Comparison to Baselines

To compare the effectiveness of our proposed approach to established alternatives, Table 4 lists nDCG@10 and Recall@1000 scores in the full ranking setup for each of our three loss variants, as well as scores for state-of-the-art bi-encoder models trained via teacher distillation by Hofstätter et al. [14]. We also include evaluation scores of the reproduction study by Wang and Zuccon [29]. Both studies finetune the distilBERT model exclusively, therefore we restrict our comparison to distilBERT-based runs. The table is grouped into approaches relying on self-distillation (adaptive and distributed targets), teacher distillation (MarginMSE, original [14] and reproduced [29]), and no distillation (static targets and BM25).

To compare our approach to teacher distillation, we obtained run files from the reproduction study by Wang and Zuccon [29], and tested for statistical equivalence of our approaches against their best-performing run (paired TOST, $\epsilon_L = 0.05$, $p < 0.05$, $n = 43$ [TREC-DL19], $n = 54$ [TREC-DL20], Bonferroni correction applied per test group). Statistical equivalence is indicated in Table 4 using a † symbol. No run files were available for the study by Hofstätter et al. [14], hence no statistical testing could be carried out here. To compare data efficiency, the table also includes the number of training triplets seen by each model. The early-stopping criterion of Hofstätter et al. [14] terminates runs after 700-800k steps at a batch size of 32. This puts a conservative estimate of the data seen during training for their models at least at 22.4M training triplets. Wang and Zuccon [29] terminate earlier, at 456 000 steps with the same batch size for their highest-scoring run, i.e., at 14.6M triplets seen. Their other runs use similar amounts of data. Training time is not comparable due to the different software and hardware setups used between our experiments and prior work. Yet, all teacher distillation approaches require an additional cross-encoder forward pass for the full dataset of 40 M triplets for batch sampling.

For nDCG@10, the best self-distillation approach (distributed) scores 0.07 lower than the best original teacher-distillation approach (TAS-B batch sampling with in-batch negatives). To the scores attained in the reproduction study, a difference of at most 0.04 can be

Table 4: Comparison of our self-distillation approach (first block) with teacher distillation (second & third block) and baseline systems (fourth block). Training setting, loss variant, batch sampling approach, use of in-batch negatives, source, and pretrained model indicated. Best per block and column marked bold. † marks runs with statistically equivalent effectiveness compared to best reproduced TAS-B run ([29], underlined)

	Loss Variant	B. Sampl.	In-B.	Source	TREC-DL19		TREC-DL20		Data Seen
					nDCG@10	Recall@1000	nDCG@10	Recall@1000	(# Triples)
Self-D.	Adaptive	Rand.	✗	Ours	0.53	0.66	0.55	0.74	4.93M
	Adaptive	Rand.	✓	Ours	0.53	**0.79**†	0.51	**0.82**†	1.98M
	Distributed	Rand.	✓	Ours	**0.64**†	0.77†	**0.62**†	0.81†	7.30M
Teacher Distillation	MarginMSE	Rand.	✗	Hofstätter et al. [14]	0.69	0.77	0.65	0.80	≈ 22.4M
		Rand.	✓	Hofstätter et al. [14]	0.70	0.79	0.67	0.84	≈ 22.4M
		TAS-B	✗	Hofstätter et al. [14]	0.69	0.78	0.67	0.83	≈ 22.4M
		TAS-B	✓	Hofstätter et al. [14]	**0.71**	**0.85**	**0.69**	**0.87**	≈ 22.4M
		Rand.	✗	Wang and Zuccon [29]	**0.68**	0.78	0.66	0.82	≈ 14.6M
		Rand.	✓	Wang and Zuccon [29]	0.61	0.66	0.60	0.75	≈ 14.6M
		TAS-B	✗	Wang and Zuccon [29]	0.67	**0.79**	**0.66**	**0.83**	≈ 14.6M
		TAS-B	✓	Wang and Zuccon [29]	0.66	0.75	0.62	0.79	≈ 14.6M
No Dist.	BM25	—	—	—	0.50	0.74	0.48	0.81	—
	Static (Tuned)	Rand.	✗	Ours	**0.65**†	**0.79**†	**0.63**†	**0.82**†	5.2M
	Static (Tuned)	Rand.	✓	Ours	0.56	0.77†	0.53	0.82†	2.2M
	Static ($\epsilon = 1$)	Rand.	✗	Hofstätter et al. [14]	0.60	0.71	0.60	0.76	≈ 22.4M

observed, and the effectiveness is statistically equivalent in all cases. For Recall@1000, a similar trend is observable: compared to the original teacher-distillation approach, ours scores lower by 0.06 at most (adaptive with in-batch negatives), yet still statistically equivalent to the reproduction study. This attests to our approach's competitive effectiveness, matching the scores of teacher-distillation-based approaches. When comparing the static target approach, we can make two observations: first, tuning the target hyperparameter yields a large increase in Recall@1000, but less so for nDCG@10. Second, it also matches the effectiveness of the teacher-distilled models in the reproduction study, further calling into question if the expense of teacher score inference is worth the marginal increase in effectiveness. Yet, the static targets variant necessitates hyperparameter tuning, and thus incurs higher computation cost (see Table 3) while offering similar effectiveness as self-distillation. The data efficiency of the self-distillation approach is between 2x/3x (distributed) and 7x/11x (adaptive) better than the teacher-based approaches.

6 CONCLUSION

In this paper, we investigated self-distillation with distributed relevance margins as a new approach to fine-tuning bi-encoder models for retrieval and ranking tasks. We proposed adaptive relevance margins as a novel self-distillation loss function. In contrast to previous teacher-distillation-based approaches, it eliminates the need for a teacher model and batch sampling procedure, and is hyperparameter-free. This is achieved by replacing the teacher signal with a text-to-text similarity score estimated by the fine-tuned model itself. Additionally, we introduced distributed targets, extending the adaptive approach with a novel approach to using in-batch information. Both variants perform implicit in-batch negative mining.

Our experimental study demonstrated the effectiveness and training efficiency of the self-distillation approach. We have shown that both adaptive and distributed targets achieve competitive effectiveness while requiring substantially less training triplets and training compute time compared to state-of-the-art teacher-distillation methods. At the same time, we have demonstrated that teacher-distillation only offers marginal yields in effectiveness over hyperparameter-tuned static margin losses. Our results underline the viability of self-distillation as an alternative for training bi-encoder models, as it is data-efficient and hyperparameter-free.

Self-distillation could also find applications in information retrieval beyond bi-encoder training. In future work we will investigate if it can be applied to pairwise cross-encoder training, which also uses triplets of a query and a positive/negative document. Further, the implicit negative mining aspect of adaptive and distributed targets can be extended further, e.g., by incorporating different scaling-, weighting-, or cutoff-schemes. Optimizing a single positive-negative pair against a distribution of target margins is also a promising direction for future work that we will investigate.

ACKNOWLEDGMENTS

This publication has been partially supported by the ScaDS.AI Center for Scalable Data Analytics and Artificial Intelligence, funded by the Federal Ministry of Education and Research of Germany and by the Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus, by a Research Fellowship for Harrisen Scells from the Alexander von Humboldt Foundation, and by the OpenWebSearch.eu project, funded by the European Union (GA 101070014). We would also like to thank Shuai Wang for providing us with run files and additional insight about of their reproduction efforts [29].

REFERENCES

[1] Yinqiong Cai, Jiafeng Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. 2022. Hard Negatives or False Negatives: Correcting Pooling Bias in Training Neural Ranking Models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, M. Al Hasan and Li X. (Eds.). ACM, 118–127.

[2] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[3] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*, E. M. Voorhees and A. Ellis (Eds.). National Institute of Standards and Technology (NIST).

[4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. (2020). arXiv:2003.07820

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio (Eds.). ACL, 4171–4186.

[6] William Falcon and The PyTorch Lightning team. 2019. *PyTorch Lightning*.

[7] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, M.-F. Moens, X. Huang, L. Specia, and S. Wen-tau Yih (Eds.). ACL, 981–993.

[8] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complement Lexical Retrieval Model with Semantic Residual Embeddings. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12656)*, D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, and F. Sebastiani (Eds.). Springer, 146–160.

[9] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* 129, 6 (2021), 1789–1819.

[10] Mai Lan Ha and Volker Blanz. 2021. Deep Ranking with Adaptive Margin Triplet Loss. (2021). arXiv:2107.06187

[11] Sangchul Hahn and Heeyoul Choi. 2019. Self-Knowledge Distillation in Natural Language Processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2019, Varna, Bulgaria, September 2-4, 2019*, R. Mitkov and G. Angelova (Eds.). INCOMA Ltd., 423–430.

[12] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. (2015). arXiv:1503.02531

[13] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. (2020). arXiv:2010.02666

[14] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, and T. Sakai (Eds.). ACM, 113–122.

[15] Chengming Hu, Xuan Li, Dan Liu, Haolun Wu, Xi Chen, Ju Wang, and Xue Liu. 2023. Teacher-Student Architecture for Knowledge Distillation: A Survey. (2023). arXiv:2308.04268

[16] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data* 7, 3 (2021), 535–547.

[17] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, B. Webber, T. Cohn, Y. He, and Y. Liu (Eds.). ACL, 6769–6781.

[18] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. 2020. Boosting Few-Shot Learning With Adaptive Margin Loss. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 12573–12581.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692

[20] Yang Liu, Sheng Shen, and Mirella Lapata. 2021. Noisy Self-Knowledge Distillation for Text Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou (Eds.). ACL, 692–703.

[21] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*, T. R. Besold, A. Bordes, A. S. d'Avila Garcez, and G. Wayne (Eds.). CEUR-WS.org.

[22] Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. (2019). arXiv:1910.14424

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett (Eds.). 8024–8035.

[24] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. (2019). arXiv:1910.01108

[25] Harrisen Scells, Shengyao Zhuang, and Guido Zuccon. 2022. Reduce, Reuse, Recycle: Green Information Retrieval Research. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai (Eds.). ACM, 2825–2837.

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 815–823.

[27] Rohan Sharma, Kaiyi Ji, zhiqiang xu, and Changyou Chen. 2024. AUC-CL: A Batchsize-Robust Framework for Self-Supervised Contrastive Representation Learning. In *The Twelfth International Conference on Learning Representations*.

[28] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin (Eds.).

[29] Shuai Wang and Guido Zuccon. 2023. Balanced Topic Aware Sampling for Effective Dense Retriever: A Reproducibility Study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, H.-H. Chen, W.-J. (Edward) Duh, H.-H. Huang, M. P. Kato, J. Mothe, and B. Poblete (Eds.). ACM, 2542–2551.

[30] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, Q. Liu and D. Schlangen (Eds.). ACL, 38–45.

[31] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. 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.

[32] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, and T. Sakai (Eds.). ACM, 1503–1512.

[33] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense Text Retrieval based on Pretrained Language Models: A Survey. (2022). arXiv:2211.14876

[34] Yang Zhou, Zhanhao He, Keyu Lu, Guanhong Wang, and Gaoang Wang. 2022. Preserve Pre-trained Knowledge: Transfer Learning With Self-Distillation For Action Recognition. (2022). arXiv:2205.00506