InDi: Informative and Diverse Sampling for Dense Retrieval

Nachshon Cohen[1⋆], Hedda Cohen-Indelman[2⋆⋆], Yaron Fairstein[1], and Guy Kushilevitz[1]

[1] Amazon
[2] Technion

Abstract. Negative sample selection has been shown to have a crucial effect on the training procedure of dense retrieval systems. Nevertheless, most existing negative selection methods end by randomly choosing from some pool of samples. This calls for a better sampling solution. We define desired requirements for negative sample selection; the samples chosen should be *informative*, to advance the learning process, and *diverse*, to help the model generalize. We compose a sampling method designed to meet these requirements, and show that using our sampling method to enhance the training procedure of a recent significant dense retrieval solution (coCondenser) improves the obtained model's performance. Specifically, we see a $\sim 2\%$ improvement in MRR@10 on the MS MARCO dataset (from 38.2 to 38.8) and a $\sim 1.5\%$ improvement in $Recall@5$ on the Natural Questions dataset (from 71% to 72.1%), both statistically significant. Our solution, as opposed to other methods, does not require training or inferencing a large model, and adds only a small overhead ($\sim 1\%$ added time) to the training procedure. Finally, we report ablation studies showing that the objectives defined are indeed important when selecting negative samples for dense retrieval.

Keywords: Dense Retrieval · Contrastive Learning · Hard Negative Selection

1 Introduction

Duel Encoder (DE) solutions have recently become a common practice for information retrieval systems [18,38,5,8,40]. These solutions use text encoders to encode queries and passages[3] separately as dense vectors. The query-passage relevance is modeled as a simple dot product, resulting in low latency during inference even for huge corpora. This makes DE-based solutions a great fit for production systems.

The standard practice for training these systems is using the contrastive learning paradigm. In this paradigm, during training the model is shown a query along with a positive passage and some negative passages. The model is encouraged by the loss to push the query embedding and the positive passage embeddings closer together than the query embedding and the negative passage embeddings. During inference, the passages closest to the query in the embedding space are retrieved.

⋆ All authors contributed equally to this work
⋆⋆ Work done as an intern at Amazon
[3] In this work we focus on retrieving passages, but dense retrieval methods are used for retrieving other items as well such as products, documents, images etc.

While collecting a positive passage for a query can be done manually, labeling all passages as positive or negative is practically impossible due to the huge size of modern corpora. It is a common practice for datasets to provide a few positive examples per query, but no negative examples [29,21].

Since training with contrastive learning methods requires more negatives than are usually labeled in the dataset, the challenge of selecting which passages to use as negative samples during training arises (referred to as *negative sampling*). Previous work has shown that training with hard negative samples (i.e. irrelevant passages that are similar to a relevant passage) helps achieve better models [18,38]. Therefore, standard practice is to randomly sample passages from the top-retrieved passages for the query using some base retrieval method (e.g. DPR [18] uses BM25 retrieval, while ANCE [38] uses a slightly older version of the training model for retrieval). While this assures the passages chosen are fairly hard, it does not assure they are negative. Some filtering methods can be used to make sure the passages chosen are not actually positive, but this is not trivial since methods that filter out false negatives are prone to also filter out the *hardest* true negatives. RocketQA [34] deals with this by using a large Cross Encoder model for the filtering task, but this solution is costly in resources.

While there are many different techniques and heuristics to select the pool from which the hard negatives are sampled, ultimately most existing works sample the actual negatives used during training randomly from this pool. This common practice raises the following question: after forming a pool of passages considered as hard negatives for a train query, can we choose the passages to be used as the actual negatives during training in a way better than random sampling?

In this work, we try to answer this question. Specifically, we propose two important qualities we want the selected negatives to hold: *Informativeness* and *Diversity* (InDi). By requiring *diversity* we strive for a sampling method that can select multiple negatives for a single query that are not very similar, so that the model does not overfit. By requiring *informativeness*, we aim at choosing examples that the model is unsure whether they are negative or positive, as opposed to easier negatives that the model already knows are irrelevant for the given query.

We devise a sampling method aiming to optimize these qualities by utilizing the gradient embeddings of the model given the candidate samples. Our method selects the negative samples that will have the best effect on the training process and uses them during training. This enhancement to the training process allows us to achieve better performance, as tested on MS-MARCO [30] and Natural Questions [20] datasets, without adding any labeling costs and just a minor added compute. We publish our code and models for reproducibility and to encourage future research in this area[4].

2 Preliminaries and Notation

In the passage retrieval task, given a query q and a corpus of M passages, the task is to retrieve a set of passages that are most relevant to q. In dense retrieval systems, this is done by encoding each passage and query into a dense, low-dimensional embedding. For

[4] https://github.com/nachshonc/indi-hard-negative-selection

every query, relevant passages are expected to be mapped close in the embedding space using some similarity measure. In most settings, an index of passages is built offline using a kNN tool such as FAISS [16] or ScaNN [10]. Then, when a query arrives, it is encoded online into a dense vector and the kNN index is queried for similar passages. While multiple similarity metrics are possible, most systems use a simple dot product similarity, which we denote by $s(a, b)$.

Encoding of passages and queries is done using a BERT-like [2] architecture (encoder-only transformer). The model is trained to encode query-passage relevance in the embedding space via a contrastive loss. The training dataset consists of tuples of a query (q), a positive passage (p^+), and a set of negative passages (N). The loss encourages the model to push queries and their relevant passages closer together in the embedding space, while pushing the query and irrelevant passages further apart. Given a query encoder E_q and a passage encoder E_p, the constrastive loss can be written as $\ell(q, p^+, N) =$

$$-\log \frac{e^{s(E_q(q),E_p(p^+)}}{e^{s(E_q(q),E_p(p^+))} + \sum_{p^-\in N} e^{s(E_q(q),E_p(p^-))}},$$

The selection of the negative passage set N has a crucial impact on the quality of the model achieved [40]. Following recent work, we assume an early version of the DE model, denoted $\mathcal{S}_1$, which is used to retrieve a pool of hard negative passages $\mathcal{T} = \{p_1, p_2, \ldots, p_t\}$. Out of these t passages, a subset of passages N of size k ($k \ll t$) are selected to serve as the negative passages during training (also referred to as *negative sampling*). Our work focuses on how these k samples are chosen out of the t available samples, and does not alter with the curation process of the pool $\mathcal{T}$. In this work, following our main baseline coCondenser [7], we use $t = 200$ and $k = 21$.

3 Methods

In this section, we present two methods used for selecting hard negative samples. The first is our main contribution; a sampling method focused on the Informativeness and Diversity (InDi) of the negative samples selected, replacing the currently widely-used random approach. The second is a complementary method addressing the redundancy of negative samples when using in-batch training.

3.1 Informative & Diverse Sampling

For each train query, existing systems sample k hard negatives randomly out of t top results retrieved for this query with some existing retrieval method. We argue that this approach does not realize the full potential of the hard negative examples. Therefore, we devise a new selection method aiming to select the samples that will bring the most benefit to the model. **Requirements** of the hard negative set chosen:

1. Be *diverse*; given we can choose k negative examples per query, we don't want them to be similar so that the model can generalize well.

2. Be *informative*[5]; contain passages that the model is *uncertain* about and cannot easily distinguish from a positive passage.

Inspiration. The Active Learning (AL) setting (see Section 6.3) shares our objectives as it also aims at selecting diverse and informative examples. A recent work in this field, BADGE [1], shows that clustering the gradient vectors of textual embeddings is beneficial for selecting diverse and informative examples. While BADGE is tailored for the classification use case and can not be directly applied for dense retrieval, it inspired our sampling method.

Method. We compute for each passage the gradient of its embedding with respect to the contrastive learning loss, assuming it is selected as a negative passage. We then take all gradient vectors and cluster them, using the K-means algorithm with Euclidean distance metric. The number of clusters k is selected as the number of negatives we use throughout training. From each cluster, we select the medoid (i.e., the example whose gradient embedding is closest to the cluster's center) as a representative negative example. An illustration of this method appears in Figure 1.

In the contrastive loss setting, it is not possible to compute the actual gradient vectors. This is due to the fact that the loss depends on multiple negative passages, which are unknown at the selection time. Therefore, we need to find a way to approximate the loss. We do this by computing a point-wise approximation of the loss, depending on just the query q, the positive example p^+, and a single negative p_j^- from the pool $\mathcal{T}$:

$$\tilde{g}(p_j^-) = \frac{\partial \ell(q, p^+, p_j^-)}{\partial E_p(p_j^-)}.$$

We note that most existing dense retrieval systems work in multiple phases. In the first step, an early version of the model, denoted $\mathcal{S}_1$, is trained. Then, the pool of negatives (from which the actual training negatives are later selected) is created by computing the embedding of all passages in the corpus, and running a kNN retrieval algorithm. Later, the final model is trained using the acquired negative samples. Since the passages embeddings are required regardless of our technique (to run the kNN retrieval algorithm), the gradient vector of each passage can be computed while incurring only a small overhead on top of the existing method.

Intuition. We provide some intuition into why the described method should meet the aforementioned requirements. First, it is intuitive that taking a representative from each cluster will assure *diversity*. As for *informativeness*, since short vectors (i.e., with a low norm) fall in a dense area, when clustering using Euclidean distance many of them can be efficiently clustered in only a few clusters[6]. Therefore, only a few short representatives are likely to be chosen by our method, resulting in a set of negative samples containing mostly long vectors. Long gradient vectors represent negative samples the model is uncertain about, which are considered more informative to the model [23,15].

[5] Informativeness measures the ability of a sample to reduce the uncertainty of a model. Informativeness is commonly approximated by measuring the loss a sample causes the model [23,15]. High loss means high uncertainty, suggesting high informativeness of the sample.

[6] For example, many close-to-zero gradient vectors, pointing in different directions, will all be in the same cluster.

The informativeness and diversity requirements are revisited after the main experimental results are presented (Sections 5.7 and 5.5).

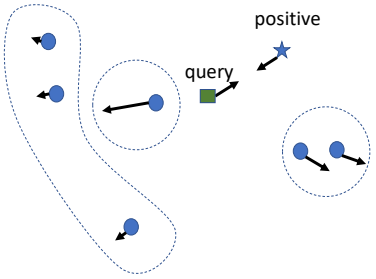

Fig. 1: Illustrating our negative selection process. The blue dots are the negative example. In arrows we mark the approximation of the gradient. We group negatives that, if chosen, will have a similar impact on the model. We then select one representative from each group.

3.2 In-batch Deduplication

Repeating training samples in general and especially in the same training batch was shown to negatively affect the performance of the trained model [22]. Luckily, in the standard training scheme used for dense retrieval, this is unlikely. This scheme, utilizes in-batch negatives, i.e., passages selected (as either positive or negative) for one query are also used as negative passages for other queries in the same batch. While this is expected to to increase the probability of duplication, in existing training methods, the negative passages of each query are selected at random from a large pool (e.g., of size 200) of top relevant passages. Thus, even if similar queries with intersecting pools belong to the same batch, it is still unlikely for the randomly selected passage to fall within this intersection.

When using our method this is no longer true. By focusing on highly informative negatives the method removes some of the randomness in the selection, making some passages more likely to be selected. As a result, the event of two similar queries belonging to the same batch may suffice to have the same passage selected more than once. Thus, addressing this potential hazard becomes essential.

We deduplicate each batch by iterating over the batch queries and removing passages already selected for the batch from the query's pool before selecting the hard negatives. The extent of this phenomenon and the effect of this method is demonstrated and discussed in Section 5.6.

Dataset	**Corpus size**	**Train/Test size**
MS-MARCO	8.8M	500K/6,980
NQ	21M	58K/3,610

Table 1: Corpus size is measured in passages, Train/Test size is measured in queries. Following DPR [18], We use the NQ dataset version published in the Tevatron toolkit.

4 Experimental Setup

We implemented our method in the Tevatron toolkit [8] for dense retrieval. For pre-training, we used the corpus-specific pre-trained checkpoint of coCondenser [7] for MS-MARCO and Natural Questions, which is trained in an unsupervised manner. Following coCondenser, training proceeds in two steps. First, we train the dense retrieval model using BM25 negatives. Based on this model, we gather top 200 results per query and select negatives from this pool either randomly (for the baseline), or via our sampling technique. The model is then trained from scratch using the hard negative examples.

Datasets. We follow many previous works and train and test our methods on the MS-MARCO dataset [30]. To demonstrate robustness and further validate our method, we also evaluate our sampling method on the Natural Questions dataset (NQ) [20]. Datasets information can be found in Table 1. Note, while there are passages published as negative in some versions of these datasets, there are no passages actually verified (i.e. human annotated) as being negative. In both datasets, the given annotation task is to mark/generate an answer for questions and not to annotate passages for relevance. Sometimes negatives are later generated with some heuristic (e.g. passages that do not contain the answer) but are not verified as actually being negative or to being hard.

Measure. Following previous work and official metrics, for MS-MARCO we report $MRR@10$ as the main metric and $Recall@50/@1000$ as complementary metrics and for NQ we report $Recall@5/@20/@1000$. The top results in the same column is marked **bold**, and results that are also better than all other results in the same column in a statistically significant manner, using a relative t-test with $p < 0.01$, are marked with an asterisk (*).

Baselines. Our proposed method is a train-procedure improvement method. Therefore, we mainly want to show using it during training of some base-model leads to a better model compared to training the same base-model without our method. We choose coCondenser as our base model for the following reasons: 1. It is a highly popular dense retrieval solution achieving competitive results. 2. It is not a distillation model which is very expensive to train. 3. The authors publish their code and framework, making the training procedure for this model relatively easy to reproduce. For reference, we also report results on some additional retrieval solutions. These include the classic BM25 lexical matching results, doc2query [32], DeepCT [28], docT5query [31] and SPLADEv2 [3], which are solutions that combine lexical matching and a transformer-based model. We also include dense retrieval methods. These include ANCE [38] that collects hard negatives in an online manner by training a model and building a kNN index in parallel, ME-BERT [27], which uses multiple vectors for representing passages, and ColBERT [19], a multistage model that represents both queries and passages using multiple queries. Last, we include RocketQA [34], which filters out hard negatives suspected as positives based on a Cross Encoder. The latter four models do not start from a corpus-aware pretraining, and therefore show lower-quality results compared to coCondenser.

Method	MRR@10	R@50	R@1000
BM25	18.7	59.2	85.7
doc2query	21.5	64.4	89.1
DeepCT	24.3	69.0	91.0
docT5query	27.7	75.6	94.7
SPLADEv2	36.8	-	97.9
ANCE	33.0	-	95.9
ME-BERT	33.8	-	-
ColBERT	36.0	82.9	96.8
RocketQA	37.0	85.5	97.9
coCondenser	38.2	86.5	98.4
InDi	**38.8***	**86.6**	**98.5**

Table 2: Main results on MS-MARCO.

5 Results

5.1 Main Results

Main results are presented in Table 2. Our Informative and Diverse (InDi) negative sample selection method improves the training procedure of coCondenser reaching statistically significant better results compared to all baselines on the main official metric ($MRR@10$). Results on other measures ($Recall@50/@1000$) are comparable, this is further discussed and analyzed in Section 5.2.

5.2 Fine-grained Topical Understanding

In the Dual Encoder (DE) training procedure, it is a common practice to use in-batch negatives. When computing the loss for a query q, apart from the negatives chosen specifically for q (i.e., hard negatives) passages chosen for other batch queries are also considered as negatives for q. Since batch queries are usually selected randomly, the passages selected for other queries are likely easy negatives (i.e. very different from positive passages) for q. It is a common belief that easy negatives teach the model a coarse-grained topical understanding while the hard negatives chosen specifically for q teach the model a fine-grained understanding of relevance. For example, easy negatives might teach the model that passages discussing cats or trees are irrelevant for a query about dogs, while hard negatives might teach it that passages discussing giant poodles are not relevant to queries asking about toy poodles.

InDi only affects the hard negative selection. Thus, it is expected for InDi to mainly affect the top-of-the-list results. Indeed, this is demonstrated in the main results in Table 2. InDi provides a statistically significant improvement to $MRR@10$ results, while a numerically higher, but not statistically different result in $Recall@50/1000$. To further corroborate this, we show additional $Recall@k$ values in Table 3, focusing on the top of the list. We find that the improvement of InDi is focused on the top-of-the-list

retrieved passages, suggesting it mainly improves the fine-grained understanding of the model, and not the general, more coarse-grained topical understanding.

R@k	coCondenser	InDi	Improvement
1	25.5	**26.2***	0.7
2	37.5	**38.7***	1.2
3	45.8	**46.5***	0.7
4	51.4	**52.3***	0.9
5	58.9	**59.2**	0.3
10	67.3	67.3	0.0
20	77.0	77.0	0.0
50	86.5	**86.6**	0.1

Table 3: MS-MARCO Recall@k with (InDi) and without (coCondenser) our selection method. Statistical significance in this table is calculated with respect to the rows.

5.3 Additional Dataset

To further validate our sampling method, we apply it on the Natural Question (NQ) dataset, and report the results in Table 4. The main theme observed on MS-MARCO (and discussed in Sections 5.1 and 5.2) is maintained in NQ as well; our sampling method improves results of top-of-the-list retrieved passages, and preserves results when looking at a larger portion of top retrieved passages.

Method	R@5	R@20	R@100
coCondenser	71.0	82.6	88.7
InDi	**72.1***	**83.2***	88.7

Table 4: Results with (InDi) and without (coCondenser) our sampling method on the Natural Questions dataset.

5.4 Diversity and Informativeness Importance

In this section, we aim to verify that both informativeness and diversity are indeed important when selecting hard negatives for Dense Retrieval. We do this by devising two more selection methods, one focusing solely on diversity and the other solely on informativeness. In a diversity-focused paradigm, the different samples are chosen to be as diverse as possible. We achieve this with **D-sampling** by first clustering passages using their embeddings (as generated by $\mathcal{S}_1$) and then choosing a representative from each cluster. This approach is similar to InDi, but it uses the contextual embeddings directly as opposed to the gradient embeddings. While the gradient vectors incorporate the informativeness of a sample (i.e. the model's uncertainty) in them, this is not the case when using the model embedding directly. In an informativeness-focused paradigm the samples that the model is least certain about are chosen. To achieve this we devise

I-sampling, an informativeness-based method that selects the negative samples that produce the highest loss, i.e., those that are most similar to the query. Therefore, for this method, we take the top k passages retrieved for the query as the hard negative samples.

Results are shown in Table 5. It can be seen that InDi, combining both paradigms, performs best. I-sampling achieved subpar results. We hypothesize this is actually due to a data issue: the MS-MARCO dataset, like many other retrieval datasets, contains many unlabeled positives. I-sampling is prone to choosing these samples as false negatives since they are likely to be ranked high. Other methods may also select unlabeled positives, but are unlikely to select many of them since in their selection they take diversity into consideration, and positives are likely to be somewhat similar to each other.

To verify this hypothesis, we combine I-sampling with a precursory filtering step, where we attempt to eliminate false-negative samples. We follow the line of RocketQA [34], which uses a strong, capable Cross Encoder (CE) model to filter out false negatives[7]. Unlike the DE model, where the query-passage interaction is limited to a single dot product computation, CEs concatenate the query and passage and pass them through transformer layers. This allows the query and passage to interact via the self-attention mechanism, resulting in a more capable model that can be used for negative selection, but naturally comes with a vast computational cost (see Section 5.8). CE filtering vastly improves I-sampling (Table 5), suggesting that false negatives are indeed a problem. Still, results are inferior to InDi, showing both diversity and informativeness are important.

Selection Method	MRR@10
I-sampling	32.22
D-sampling	34.67
InDi	**38.84***
I-sampling + CE	37.81

Table 5: MRR@10 on MS-MARCO of different methods for negative selection.

Selection Method	**Diversity**
I-sampling	0.187
Random	0.203
InDi	0.203
D-sampling	0.223

Table 6: Diversity of different selection methods.

5.5 Diversity and Informativeness Measures

We measure informativeness and diversity and compare methods discussed in this paper.

Informativeness. Informativeness is considered as the ability of a sample to reduce the uncertainty of a model [15]. It is commonly approximated by measuring the loss a sample causes the model [23]. High loss means that currently the model is highly uncertain about this sample, suggesting it contains information that the model was not exposed to. Thus, teaching the model using this sample is highly informative.

Building on this intuition, we measure informativeness by calculating for each query the contrastive loss of its selected negatives using the embeddings generated by $\mathcal{S}_1$. Then, for each query we assess which selection method produced the highest loss. Finally, we used the fraction of queries for which the loss of one method is higher than the other as a proxy-metric for informativeness. Unsurprisingly, the I-sampling selection

[7] We find that the best result is achieved with a threshold of 0.8 on the CE score.

method always presents the highest loss. It is expected as this method selects the samples which $\mathcal{S}_1$ is most uncertain about. Setting I-sampling aside, our method showed higher informativeness compared to the random (67% of the queries) and D-sampling (57% of the queries) selection methods. This suggests that it tends to select negatives that are highly informative to the model.

Diversity. We calculate the diversity w.r.t. each query separately and compare the average value across queries. Following Zhdanov [42], given a query q we define the diversity of a subset N selected out of $\mathcal{T}$[8], as follows.

$$D(N) = \left(\frac{1}{|\mathcal{T}|} \sum_{x_i \in \mathcal{T}} \min_{x_j \in N} d(x_i, x_j) \right)^{-1},$$

where $d(x_i, x_j)$ is the Euclidean distance[9] between the embeddings of x_i and x_j (as calculated by $\mathcal{S}_1$). Results appear in Table 6. D-sampling, as expected, produces the most diverse negatives. I-sampling produces the most homogeneous results. Our method is able to achieve a diversity score equal to that of the random selection. We note that random selection, especially when samples are relatively uniformly distributed, leads to diverse selections. For dense retrieval models using contrastive loss, this is usually the case as shown by Wang et al. [37]. InDi preserves the diversity compared to the already diverse baseline while improving informativeness.

5.6 Ablation

In Section 3 we presented the methods on which InDi is based. In this section, we evaluate the contribution of each of the methods separately. The results appear in Table 7. As can be seen, the lion's share of the improvement produced by InDi originates from our selection method.

Nevertheless, we see that our in-batch deduplication method is helpful. When the negative samples are selected randomly $\sim 0.05\%$ of batches include at least one duplicated sample. Even though this is only a small fraction of the batches, deduplicating these samples led to a non-negligible performance improvement. In the case of negative samples selected by InDi this phenomenon is more common as $\sim 1.6\%$ of batches include at least one duplicated sample. Indeed, deduplication with InDi leads to an increased improvement in performance.

	w/o deduplication	deduplication
Random	38.2	38.33
InDi	**38.66***	**38.84***

Table 7: Ablation study comparing MRR@10 on MS-MARCO of each of our methods.

[8] $\mathcal{T}$ represents the top t retrieved samples for q.

[9] We opt to use Euclidean distance as it is desired that the distance to a sample is minimized by the sample itself.

5.7 Cross Encoder Filtering with InDi

In Section 5.4 we saw that the I-sampling method, which prioritizes informativeness, vastly benefits from CE-filtering where false-negative samples are filtered. As InDi, like I-sampling, also selects highly informative samples, it raises the question whether this filtering step can further improve our results.

We enhance the filtering step by defining two thresholds, ce_{min} and ce_{max}, bounding the allowed relevance score of the hard negative passages. In some cases insufficient number of passages pass the filter[10], which makes our clustering step degenerate, leading to subpar performance. To alleviate this issue, we expand the acceptable relevance range by first increasing ce_{max} up to 0.95 until there are enough passages to choose from. If this does not suffice, we also decrease ce_{min} downward (down to 0) until there is a sufficient number of candidates.

In Table 8 we show results using CE-filtering. Results show that performance can be improved by using CE filtering, but only slightly. Further, results suggest that while using $ce_{min} > 0$ consistently improves results, showing again the importance of selecting hard negatives, for ce_{max} improvement is less clear. This is likely due to the following trade-off: as increasing ce_{max} should reduce the number of false-negative passages, it also decreases the hardness and informativeness of the passages surviving the filter.

CE-filtering		**MS-MARCO**
Min	Max	MRR@10
-	-	38.84
0.0	0.5	38.25
0.0	0.6	38.42
0.0	0.7	38.66
0.1	0.5	38.99
0.1	0.6	39.03
0.1	0.7	39.03
0.2	0.5	39.07
0.2	0.6	39.11
0.2	0.7	39.02

Table 8: Parameter study for InDi with and without CE filtering. Min and Max are ce_{min} and ce_{max}.

5.8 Computational Surplus

Our sampling solution, while demonstrating strong results, is also very efficient. It does not require training a new model or running inference (as opposed to using a CE[11]) and adds only a small overhead to the training procedure of the DE. While training a

[10] To make sure the number of passages is sufficient, we define a minimal ratio between the number of passages that pass the CE-filtering and the number of negatives selected.

[11] RocketQA trains and inferences an ERNIE-large model.

DE on the MS-MARCO train set takes roughly 48 hours[12], sampling negatives using our method adds only ~ 0.5 hours ($\sim 1\%$ degradation)[13]. Negative sampling using a CE (assuming a trained model is already available) adds more than 50 hours ($\sim 100\%$ degradation).

6 Related Work

6.1 Retrieval

Retrieving relevant passages from a textual corpus given a textual query is a highly studied problem with many practical use-cases. Early solutions, such as IDF [17], TF-IDF [36] and BM-25 [35] are based on lexical matching. These solutions suffer from the vocabulary mismatch problem [14], where a query does not share any token with a relevant passage. Modern retrieval systems use SOTA embedding models to embed both queries and passages and search for passages similar to the query in the embedding space [11,18,38,12,34,6,7]. Training such models is a highly active area of research [41] due to the many benefits in terms of quality and efficiency. Today, SoTA results are achieved using distillation from a CE model [26,24]. However, these solutions suffer from the need to train (possibly multiple) CE models and continue training them during distillation, which is very costly. Hence, the lightweight training scenario (where we cannot run CE training or inference) remains an important challenge.

6.2 Negative Selection for Dense Retrieval

Since dense retrieval solutions have been introduced, much effort has been put into negative sample selection. It was shown that the number of negatives, as well as their difficulty, has a large impact on the model's quality [18,38,34]. DPR [18] uses in-batch negatives to increase the number of negatives per query, as well as BM25-retrieved hard negatives. In TAS [13], authors try to make in-batch negatives harder by clustering similar queries together and sampling queries from the same cluster to build batches. RocketQA [34] introduced cross-batch negatives; When using multiple GPUs for training it's possible to share information and use passages from other GPUs as negatives also. This increases the number of negative examples available compared to the in-batch technique, but does not assure the selected negatives are hard. While multiple works suggest different hard-negative sampling heuristics [18,38,25] as far as we know, they all end with some random sampling.

6.3 Active Learning

As stated in Section 3.1, the Active Learning (AL) [33,4] setting shares the uncertainty and diversity objectives discussed in our work. In this line-of-work, samples are selected for labeling out of a pool of unlabeled samples. The selected samples are then labeled and used for training a model. The goal is to maximize the performance of the model

[12] Times are measured using an NVIDIA T4 GPU.

[13] Our sampling method requires only a CPU. Time was measured on a 4-core machine.

while minimizing the number of needed annotations. Recent AL works in the NLP domain mainly focus on optimizing uncertainty and diversity [1,9,39]. BADGE [1], the algorithm inspiring our selection method, combines the two by using the gradient vectors each sample generates for the model. These embeddings are clustered and a representative from each cluster is sent to annotators.

7 Conclusions

In this work we define two desired qualities for negatives selected for training dense retrieval models: *Informativeness* and *Diversity*. We present InDi, a negative sampling technique aiming at optimizing these qualities. We show the aforementioned qualities are indeed important (Section 5.4) and that InDi is able to balance between them (Section 5.5). This results in improved fine-grained understanding of the model, directly leading to a statistically significant increase in the $MRR@10$ result on MS MARCO as well as $Recall@5$ and $Recall@20$ on NQ. InDi is also superior to filtering negatives using CE scores, while being more than $100x$ faster. Both of these techniques can be combined for further benefit. Overall, we believe that InDi should be considered as an alternative to the existing random method for negative selection.

References

1. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) 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). pp. 4171–4186. Association for Computational Linguistics (2019)
3. Formal, T., Lassance, C., Piwowarski, B., Clinchant, S.: Splade v2: Sparse lexical and expansion model for information retrieval. arXiv preprint arXiv:2109.10086 (2021)
4. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. Knowledge and information systems **35**(2), 249–283 (2013)
5. Gao, L., Callan, J.: Condenser: a pre-training architecture for dense retrieval. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. pp. 981–993. Association for Computational Linguistics (2021)
6. Gao, L., Callan, J.: Condenser: a pre-training architecture for dense retrieval. arXiv preprint arXiv:2104.08253 (2021)
7. Gao, L., Callan, J.: Unsupervised corpus aware language model pre-training for dense passage retrieval. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2843–2853. Association for Computational Linguistics, Dublin, Ireland (May 2022)

8. Gao, L., Ma, X., Lin, J., Callan, J.: Tevatron: An efficient and flexible toolkit for dense retrieval. CoRR **abs/2203.05765** (2022)
9. Gissin, D., Shalev-Shwartz, S.: Discriminative active learning. CoRR **abs/1907.06347** (2019)
10. Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., Kumar, S.: Accelerating large-scale inference with anisotropic vector quantization. In: International Conference on Machine Learning (2020)
11. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: International Conference on Machine Learning. pp. 3929–3938. PMLR (2020)
12. Hofstätter, S., Lin, S.C., Yang, J.H., Lin, J., Hanbury, A.: Efficiently teaching an effective dense retriever with balanced topic aware sampling. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 113–122 (2021)
13. Hofstätter, S., Lin, S., Yang, J., Lin, J., Hanbury, A.: Efficiently teaching an effective dense retriever with balanced topic aware sampling. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021. pp. 113–122. ACM (2021)
14. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338 (2013)
15. Huang, S.J., Jin, R., Zhou, Z.H.: Active learning by querying informative and representative examples. Advances in neural information processing systems **23** (2010)
16. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Transactions on Big Data **7**(3), 535–547 (2019)
17. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of documentation (1972)
18. Karpukhin, V., Oguz, B., Min, S., Lewis, P.S.H., Wu, L., Edunov, S., Chen, D., Yih, W.: Dense passage retrieval for open-domain question answering. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. pp. 6769–6781. Association for Computational Linguistics (2020)
19. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over bert. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 39–48 (2020)
20. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.W., Dai, A.M., Uszkoreit, J., Le, Q., Petrov, S.: Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics **7**, 452–466 (2019)
21. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A.P., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M., Dai, A.M., Uszkoreit, J., Le, Q., Petrov, S.: Natural questions a benchmark for question answering research. Trans. Assoc. Comput. Linguistics **7**, 452–466 (2019)
22. Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., Carlini, N.: Deduplicating training data makes language models better. arXiv preprint arXiv:2107.06499 (2021)
23. Lewis, D.D.: A sequential algorithm for training text classifiers: Corrigendum and additional data. In: Acm Sigir Forum. vol. 29, pp. 13–19. ACM New York, NY, USA (1995)
24. Lin, Z., Gong, Y., Liu, X., Zhang, H., Lin, C., Dong, A., Jiao, J., Lu, J., Jiang, D., Majumder, R., et al.: Prod: Progressive distillation for dense retrieval. arXiv preprint arXiv:2209.13335 (2022)

25. Lu, J., Ábrego, G.H., Ma, J., Ni, J., Yang, Y.: Multi-stage training with improved negative contrast for neural passage retrieval. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. pp. 6091–6103. Association for Computational Linguistics (2021)
26. Lu, Y., Liu, Y., Liu, J., Shi, Y., Huang, Z., Sun, S.F.Y., Tian, H., Wu, H., Wang, S., Yin, D., et al.: Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. arXiv preprint arXiv:2205.09153 (2022)
27. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Transactions of the Association for Computational Linguistics **9**, 329–345 (2021)
28. Mackenzie, J., Dai, Z., Gallagher, L., Callan, J.: Efficiency implications of term weighting for passage retrieval. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1821–1824 (2020)
29. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. In: Besold, T.R., Bordes, A., d'Avila Garcez, A.S., Wayne, G. (eds.) 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. CEUR-WS.org (2016)
30. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. In: Besold, T.R., Bordes, A., d'Avila Garcez, A.S., Wayne, G. (eds.) 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. CEUR-WS.org (2016)
31. Nogueira, R., Lin, J., Epistemic, A.: From doc2query to doctttttquery. Online preprint **6** (2019)
32. Nogueira, R., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. arXiv preprint arXiv:1904.08375 (2019)
33. Prince, M.: Does active learning work? a review of the research. Journal of engineering education **93**(3), 223–231 (2004)
34. Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W.X., Dong, D., Wu, H., Wang, H.: Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (eds.) 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. pp. 5835–5847. Association for Computational Linguistics (2021)
35. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M., et al.: Okapi at trec-3. Nist Special Publication Sp **109**, 109 (1995)
36. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management **24**(5), 513–523 (1988)
37. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 9929–9939. PMLR (2020)
38. Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021)

39. Yuan, M., Lin, H., Boyd-Graber, J.L.: Cold-start active learning through self-supervised language modeling. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. pp. 7935–7948. Association for Computational Linguistics (2020)
40. Zhao, W.X., Liu, J., Ren, R., Wen, J.R.: Dense text retrieval based on pretrained language models: A survey. arXiv preprint arXiv:2211.14876 (2022)
41. Zhao, W.X., Liu, J., Ren, R., Wen, J.R.: Dense text retrieval based on pretrained language models: A survey. arXiv preprint arXiv:2211.14876 (2022)
42. Zhdanov, F.: Diverse mini-batch active learning. arXiv preprint arXiv:1901.05954 (2019)