ReSa2: A Two-Stage Retrieval-Sampling Algorithm for Negative Sampling in Dense Retrieval

Muyang Li[†] MYLI.ZZU@OUTLOOK.COM
Zihan Wang[†] ZHWANG.ZZU@OUTLOOK.COM
Sijia Chen CHENSIJIA@STU.ZZU.EDU.CN
Yijun Chen IE_YIJUNCHEN@OUTLOOK.COM
Jiayu Li ZZUCSLJY@STU.ZZU.EDU.CN
Yiming Qiao YIMINGJOE@OUTLOOK.COM
Xinyi Li XINYILI20041026@OUTLOOK.COM
Bo Ji IEBJI@ZZU.EDU.CN
School of Computer Science and Artificial Intelligence, Zhengzhou University, Zhengzhou, China

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Negative sampling algorithms are critical for training dense retrievers, which in turn impact retrieval performance in information systems. Among these, hard negative sampling is of great value, and the denoised negative sampling methods in particular. Strategically selecting relevant negative samples, these methods effectively enhance the effectiveness of model training. However, they are either restricted to single-stage retrieval, failing to fully explore potential effective negatives, or demand additional training for a filter, which compromises sampling efficiency. To address this issue, the paper introduces a **two**-stage **Re**trieval-**Sa**mpling Algorithm(ReSa2). It integrates document vector-based retrieval to refine candidate selection progressively while preserving semantic relevance. In Stage 1, ReSa2 uses query vectors for broad retrieval, generating a candidate subset from the corpus to narrow the search space. In Stage 2, it reuses the retriever to perform positive-centric retrieval within this subset, leveraging positive sample vectors to re-rank candidates and enrich hard negatives with semantic similarity to the query. During the whole process, the effect is further enhanced by conducting probability-weighted sampling on the candidate subset. Insight experiments on 40,000 query-sample pairs show ReSa2 suppresses false negatives by 69.1% compared to Top-K sampling. Specifically, on the Ms Pas dataset, it outperforms the state-of-the-art by 1.2% in MRR@10 and 0.5% in R@1000. Notably, an external validation on Natural Questions (unseen domain) demonstrates ReSa2 maintains robust performance when trained on MS MARCO, highlighting its generalization capability across diverse retrieval scenarios. Ablation experiments validate the complementary roles of the two stages. Our code and appendix are released in https://github.com/ad32q/ReSa2.

Keywords: Dense Retrieval; Negative Sampling; Probability-weighted.

1. Introduction

In the present information-saturated era, the demand for efficient and precise information retrieval technology has emerged as a critical necessity across a wide spectrum of fields.

[†]. Equal contribution

Dense retrieval methods, as exemplified by those in references Karpukhin et al. (2020); Qu et al. (2021); Xiong et al. (2021); Zhang et al. (2022); Ren et al. (2021), have enabled rapid and accurate retrieval within large-scale datasets. These methods encode queries and passages into dense vectors and leverage the vector similarity to gauge semantic relevance. Then, the Faiss library Johnson et al. (2021); Douze et al. (2024) can be utilized to build an approximate nearest neighbor (ANN) index, enabling effective and efficient retrieval.

The training process of a dense retriever can be divided into two stages: pre-training and fine-tuning. In the pre-training stage, it is usually based on pre-trained language models such as Devlin et al. (2019); Lu et al. (2022); Zhou et al. (2023), which are pre-trained on large-scale general text data to learn basic language knowledge and semantic representations.

In the dense retrieval task, the pre-trained model is fine-tuned using specific data. However, it faces a large candidate space, and it is infeasible to include all documents unrelated to the query in the training process. Hence, choosing several appropriate negative samples is crucial for enhancing model performance.

For sampling, Karpukhin et al. (2020); Xiong et al. (2021); Zhou et al. (2022a) indicate that negatives that are too simplistic contribute little to model learning. Hard negatives refer to texts that are semantically similar to the query but actually irrelevant Zhao et al. (2024). Effectively incorporating them into training can enhance the model's ability to distinguish between relevant and irrelevant texts, while Qu et al. (2021); Zhou et al. (2022a); Yang et al. (2024); Wang et al. (2024) discovered that excessively hard negatives carry risks of false negatives, which may adversely affect training outcomes Chuang et al. (2020); Zhou et al. (2022b).

Although some methods have achieved certain results in solving the problems mentioned above, these methods still have limitations. Zhou et al. (2022a); Xiong et al. (2021); Karpukhin et al. (2020); Yang et al. (2024) only take advantage of single-stage retrieval each time during sampling. This limits their ability to explore potentially sufficiently effective negative samples. Some methods Wang et al. (2024); Qu et al. (2021) also necessitate additional training for a filter. This requires more computing power and time.

To address these limitations, we introduce a novel two-stage retrieval-sampling algorithm for hard Negative Sampling named ReSa2. First, in the retrieval-sampling of the first stage, we use query vectors for retrieval to obtain a subset from the entire text corpus, which effectively reduces the search space. Then, we conduct the first-stage sampling based on the probability distribution of the similarity score distances, screening out false negative samples as much as possible to improve the overall quality. In the retrieval-sampling of the second stage, we reuse the retriever with positive sample vectors to filter among the results of the first-stage sampling. While repositioning the rankings of each sample in the sample pool, we can more effectively enrich the samples with high information value. Finally, we select the area with the highest information value and sample using the uniform distribution to ultimately obtain a high-quality negative sample set.

To further demonstrate the effectiveness of our algorithm, we designed an insightful experiment. In this task, we leveraged large language models and manual review to screen 40,000 extracted query-sample pairs. For ease of processing, we adopted the following definitions:

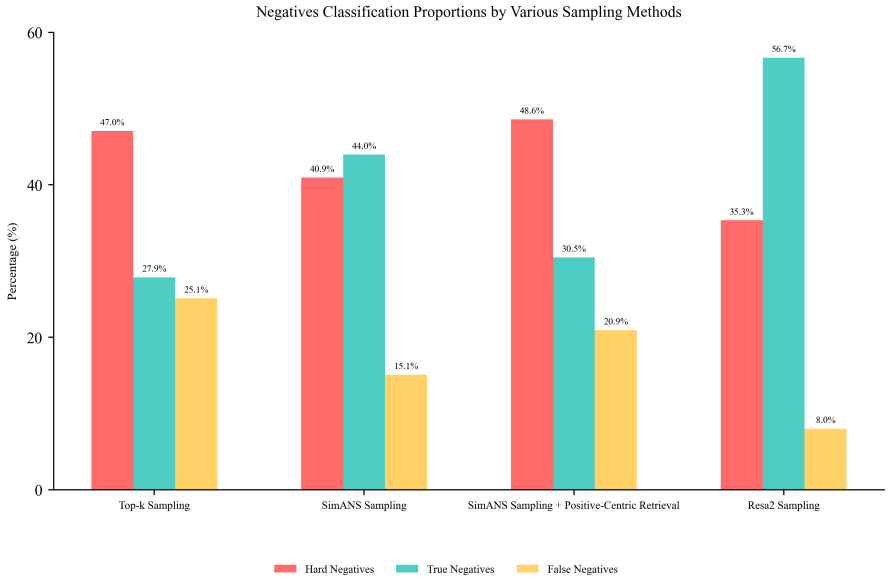

Figure 1: Insight experiments of Common Sampling Algorithm to illustrate the False Negative Suppression Performance

- **False Negatives** The paragraph under judgment is closer to the relevant passage than the query.
- **True Negatives** The paragraph under judgment is completely irrelevant to the query.
- **Hard Negatives** The paragraph under judgment is irrelevant to the query but relatively close to the correct answer.

The result is explicitly demonstrated in Figure 1, where our method shows remarkable effectiveness in suppressing false negatives. The ReSa2 method achieves the lowest proportion of false negative samples, demonstrating the most significant improvement in false negative suppression with a 69.1% reduction compared to Top-K sampling algorithms. While the SimANS method also reduces false negatives, ReSa2 exhibits a more substantial advantage. For more processing details, please refer to Appendix A in the GitHub repository.

Although positive samples have demonstrated effectiveness in selecting informative examples, relying solely on positive samples will lead to serious false negative problems. In addition, we have designed ablation experiments, including the reversed ReSa2. Specifically, in the retrieval-sampling of the first stage, we use vectorized positive samples, and in the retrieval-sampling of the second stage, we use query vectors. By using two different types of vectors for retrieval in two stages, our method can more comprehensively and accurately find valuable negative samples, effectively overcoming the shortcomings of previous single-retrieval negative sampling methods.

2. Related Work

Dense retrieval models have emerged as a powerful approach in text retrieval. Zhao et al. (2024) have explored dense retrieval on Pretrained Language Models (PLMs), and further investigations have been conducted on leveraging Large Language Models (LLMs) as the foundation for the next-generation of dense retrieval Luo et al. (2024). The negative log-likelihood loss is widely used in the training of dense retrieval models Karpukhin et al. (2020); Chen et al. (2022); Xiong et al. (2021); Qu et al. (2021); Wang et al. (2023); Shen et al. (2023). It aims to maximize the probability of relevant texts with respect to queries, enabling the model to learn more effective text representations. In actual calculations, due to the usually large size of the text collection, it is computationally expensive to exhaustively list all negative samples Zhao et al. (2024). Therefore, the negative sampling technique is often employed, where a small portion is selected from numerous negative samples for calculation.

These works Xiong et al. (2021); Karpukhin et al. (2020); Zhou et al. (2022a); Qu et al. (2021); Yang et al. (2024); Wang et al. (2024) indicate that hard negative sampling plays a crucial role in the performance of dense retrieval models. Regarding the research directly utilizing hard negative samples, there are Karpukhin et al. (2020); Xiong et al. (2021); Wang et al. (2024); Zhan et al. (2021). Among them, DPR Karpukhin et al. (2020) investigated using hard negative samples obtained by the BM-25 algorithm for training, and ANCE Xiong et al. (2021) achieved performance improvement by using hard negative samples retrieved by the model itself. The STAR and ADORE algorithms proposed by Zhan et al. (2021) and the contrastive confidence regularizer proposed by Wang et al. (2024) can be directly applied to the training of dense retrieval models. Zhan et al. (2021) also made a distinction between dynamic negative samples and static negative samples

In the denoised(also known as debiased) hard negative sampling algorithms, RocketQA Qu et al. (2021) and Passage-sieve Wang et al. (2024), which adopt a filter mechanism, use a cross-encoder and a dual-encoder optimized by contrastive confidence regularizer respectively to remove false negatives. Among the heuristic denoised hard negative algorithms, SimANS Zhou et al. (2022a) tends to select ambiguous negative samples through probability distribution, and TriSampler Yang et al. (2024) further improves the sampling quality by introducing the quasi-triangular principle on this basis. In the method we proposed, we reuse the retriever as a filter by utilizing document vectors. Without the need for additional model training, we have achieved state-of-the-art results in removing false negatives and improving sample quality. However, while the filters in previous methods are used to eliminate false negative samples, we use a probability distribution to remove false negative samples like Zhou et al. (2022a) and employ the filter to enrich the samples with the most informative value.

Apart from hard negative sampling methods, random sampling and in-batch sampling are also commonly used. Random sampling is the simplest and easiest method to apply. The trick of in-batch negatives has been applied not only in dense retrievers Karpukhin et al. (2020); Zhan et al. (2021) but also in the full-batch setting Yih et al. (2011) and mini-batch training Gillick et al. (2019); Henderson et al. (2017).

3. The Negative Sampling Algorithm for Dense Retrieval Model

3.1. Task Formulation

Let's assume we have a set of documents $D = \{d_1, d_2, \cdots, d_N\}$, where d_i represents the $i-$th document. And we have a query q.

3.1.1. Text Representation

We use a function f to transform each document d_i and the query q into vector representations.

$$\mathbf{v}_{d_i} = f(d_i) \tag{1}$$

$$\mathbf{v}_q = f(q) \tag{2}$$

Here, f could be based on techniques such as word embeddings and neural network-based encoders.

3.1.2. Index Construction

We build an index structure I from the vector representations of the documents $\{\mathbf{v}_{d_1}, \mathbf{v}_{d_2}, \cdots, \mathbf{v}_{d_N}\}$. Mathematically, we can denote this process as:

$$I = \text{BuildIndex}(\mathbf{v}_{d_1}, \mathbf{v}_{d_2}, \cdots, \mathbf{v}_{d_N}) \tag{3}$$

where BuildIndex is a function that constructs an appropriate index, such as a KD-tree or a hash-based index for vector similarity search.

3.1.3. Similarity Computation

We define a similarity metric s to measure the similarity between the query vector $\mathbf{v}_q$ and the document vectors $\mathbf{v}_{d_i}$. Common similarity metrics include: Cosine similarity:

$$s(\mathbf{v}_q, \mathbf{v}_{d_i}) = \frac{\mathbf{v}_q \cdot \mathbf{v}_{d_i}}{|\mathbf{v}_q||\mathbf{v}_{d_i}|} \tag{4}$$

Dot product similarity:

$$s(\mathbf{v}_q, \mathbf{v}_{d_i}) = \mathbf{v}_q \cdot \mathbf{v}_{d_i} \tag{5}$$

3.1.4. Ranked Retrieval Process

Given a query vector $\mathbf{v}_q$, the ranked retrieval process is formalized as:

$$R = \text{Retrieve}(I, \mathbf{v}_q, s, k) \tag{6}$$

where Retrieve function retrieves from index I the top-k documents with highest similarity to $\mathbf{v}_q$ The returned result $R = (d_1, d_2, ..., d_k)$ is an ordered tuple of document identifiers satisfying:

$$\forall i < j,\ s(\mathbf{v}_q, \mathbf{v}_{d_i}) \geq s(\mathbf{v}_q, \mathbf{v}_{d_j}) \tag{7}$$

3.2. Design of Two-Stage Retrieval-Sampling Algorithm ReSa2

Our sampling algorithm employs a two-stage retrieval and sampling process. In each stage, retrieval is performed first, followed by sampling a smaller subset using a specific distribution—either for the subsequent stage or the final output. In the first stage, the original query is used to conduct an initial search within the text collection, aiming to obtain a subset of texts most similar to the query. A first probability distribution is then employed to sample a subset from this initial result. In the second stage, a secondary retrieval is carried out within this sampled subset based on the positive samples, with the objective of precisely identifying relevant content while reshaping the distribution of samples within the subset. Finally, a second probability distribution is applied to perform the final sampling, selecting high-quality negative samples from the candidate pool. The algorithm flow is shown in Algorithm. 1. This approach provides valuable data for model training and contributes to enhancing the model's performance. The two-stage negative sampling process of ReSa2 is illustrated in Figure. 2.

Algorithm 1 ReSa2 Negative Sampling Algorithm

Input:
Queries and their positive documents $\{(q, D_q^+)\}$
Document corpus D
Pre-learned dense retrieval model M
Let $D_q^- = D \setminus D_q^+$ denote non-positive documents for query q

Output: Negative documents $\hat{D}_q^- \subseteq D_q^-$

Build an ANN index on D using M
Let Retrieve be the retrieval function
Stage 1: Initial Retrieval and Sampling
$R_1 \leftarrow \text{Retrieve}(I, q, D_q^-, k_1)$ ▷ Retrieve top-k_1 from non-positives D_q^-
Define $p_s(d^-) = \frac{\exp\left(-\frac{1}{4}(s(\mathbf{v}_q,\mathbf{v}_{d^-}) - s(\mathbf{v}_q,\mathbf{v}_{d^+}))^2\right)}{\sum_{d\in R_1}\exp\left(-\frac{1}{4}(s(\mathbf{v}_q,\mathbf{v}_d) - s(\mathbf{v}_q,\mathbf{v}_{d^+}))^2\right)}$ ▷ equation(10)
$S_1 \leftarrow \text{Sample}(R_1, p_s, k_1')$ ▷ Sample using first-stage distribution
Stage 2: Refined Retrieval and Sampling
$R_2 \leftarrow \text{Retrieve}(I, d^+, S_1, k_2)$ ▷ Retrieve top-k_2 for positive doc d^+
Define $p_\mu(d^-) = \frac{1}{|R_2 \setminus \{d^+\}|}$ ▷ equation(13)
$S_2 \leftarrow \text{Sample}(R_2 \setminus \{d^+\}, p_\mu, n)$ ▷ Sample after removing d^+
Return $\hat{D}_q^- := S_2$

3.2.1. Key Improvements

- **Mitigates False Negatives** Stage 1's distribution (p_s) explicitly reduces the chance of sampling false negatives by down-weighting candidates too dissimilar to the positive ($s^- \ll s^+$).
- **Enriches Hard Negatives** Stage 2's retrieval based on the positive (d^+) specifically targets hard negatives within the safer pool S_1.

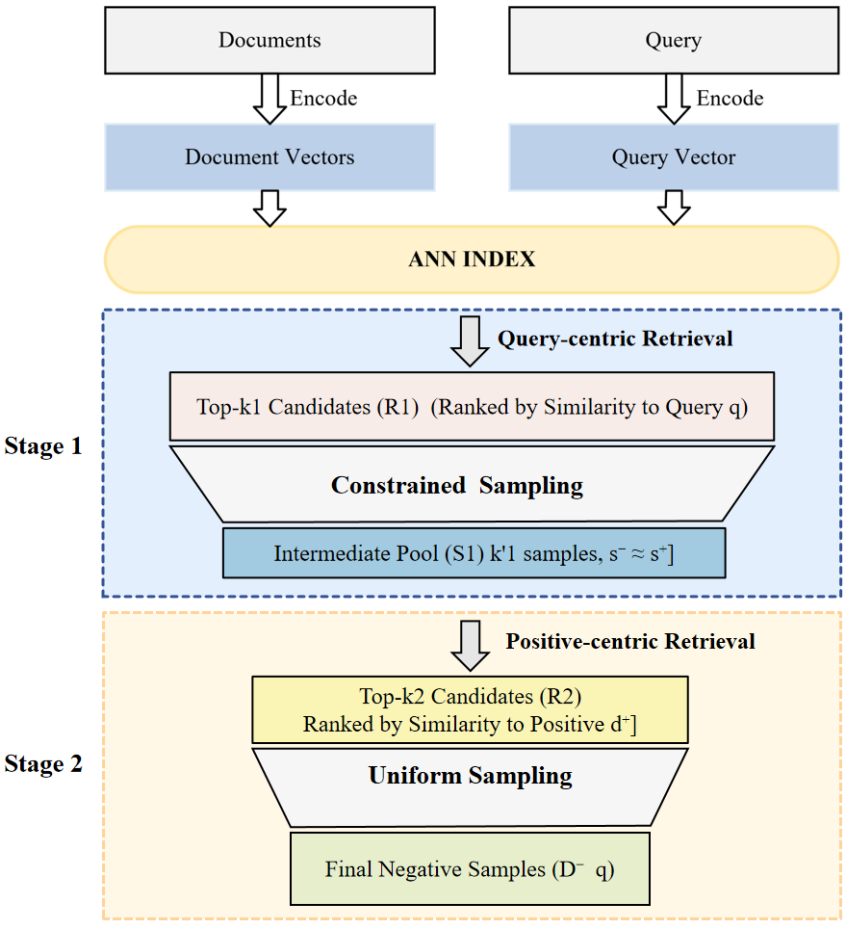

Figure 2: Architecture of the ReSa2 Two-Stage Negative Sampling Algorithm

- **Computational Efficiency** Stage 1 drastically reduces the search space ($D \to R_1 \to S_1$) before Stage 2 performs a more refined (and potentially expensive) similarity search.
- **Progressive Refinement** Combines broad query context (Stage 1) with precise positive context (Stage 2) for better negative selection.
- **Reduced Bias** Simple uniform sampling in the final stage (p_μ) avoids introducing complex biases after careful candidate selection.

3.2.2. Gradient Analysis of Negative Sampling

We use $\nabla_\theta Loss$ to represent the gradient of the loss function with respect to the model parameter θ, where $Loss$ is determined by the specific loss function of the model.

For the contrastive loss function, the gradients of positive and negative samples are defined as:

$$\nabla_\theta Loss_c = \begin{cases} -\frac{\sum_{i=1}^{n} \exp(s(\mathbf{v}_q, \mathbf{v}_{d_i}^-))}{\exp(s(\mathbf{v}_q, \mathbf{v}_d^+)) + \sum_{i=1}^{n} \exp(s(\mathbf{v}_q, \mathbf{v}_{d_i}^-))}, & \text{for positive samples} \\ \frac{\exp(s(\mathbf{v}_q, \mathbf{v}_{d_j}^-))}{\exp(s(\mathbf{v}_q, d^+)) + \sum_{i=1}^{n} \exp(s(\mathbf{v}_q, \mathbf{v}_{d_i}^-))}, & \text{for negative samples} \end{cases} \tag{8}$$

From the gradient formula, we know that the gradient of negative samples $\frac{\partial \mathcal{L}}{\partial s(\mathbf{v}_q, \mathbf{v}_{d_j}^-)}$ is proportional to the negative similarity score $\exp(s(\mathbf{v}_q, \mathbf{v}_{d_j}^-))$. The similarity scores of randomly sampled negative samples are extremely low, and their gradients can be almost ignored. On the other hand, negative samples extracted from the top K nearest irrelevant documents have higher similarity scores, which can accelerate the convergence of the retrieval model.

When the negative similarity score $s(\mathbf{v}_q, \mathbf{v}_d^-)$ significantly exceeds the positive similarity score $s(\mathbf{v}_q, \mathbf{v}_d^+)$, the gradient of the positive samples $\frac{\partial \mathcal{L}}{\partial s(\mathbf{v}_q, \mathbf{v}_d^+)}$ will be restricted to a fixed value, which will affect the training effect of the model.

3.2.3. ReSa2 Algorithm Workflow and Probability Sampling Mechanism

Let the original query be denoted as q, and the text collection be $\mathcal{D}$. The core innovation of our work lies in the design of a two-stage retrieval and sampling method, aiming to establish a more effective foundation for sampling.

Stage 1: Query-Centric Retrieval & Constrained Sampling In the first stage of the retrieval process, we utilize query q to conduct a search within the text collection $\mathcal{D}$. The goal is to identify the k_1 texts in $\mathcal{D}$ that are most similar to q. Then, the retrieval result of the first stage can be formalized as:

$$\mathcal{R}_1 = \text{Retrieve}(I,\ q, \mathcal{D}, k_1) \tag{9}$$

where I is the index structure constructed from the vector representations of the entire document collection. $\mathcal{R}_1$ is a subset of the text collection $\mathcal{D}$, i.e., $\mathcal{R}_1 \subseteq \mathcal{D}$.

After retrieval, we perform sampling on $\mathcal{R}_1$ based on the first-stage distribution. The primary goal of the one-stage negative sampling method is to design an effective probability distribution to filter out false negative samples as much as possible while sampling high-quality negative samples from negative candidates.

We formulate the first distribution with the range constraint $s(\mathbf{v}_q, \mathbf{v}_{d^+}) \approx s(\mathbf{v}_q, \mathbf{v}_{d^-})$. The distribution is:

$$p_s(d^-) \propto \exp\left(-\frac{1}{4} \cdot (s^- - s^+)^2\right) \tag{10}$$

where s^- and s^+ represent $s(\mathbf{v}_q, \mathbf{v}_{d^-})$ and $s(\mathbf{v}_q, \mathbf{v}_{d^+})$ respectively. This distribution is essentially a Gaussian kernel function and is also employed in Zhou et al. (2022a) and Yang et al. (2024).

This distribution aims to reduce overly hard negatives (probably false negatives) and strengthen the constraint $s^+ \approx s^-$. Then, the probability distribution $p_s(d^-)$ is applied to

sample from the first-stage retrieval result pool $\mathcal{R}_1$, obtaining the transitional intermediate sample pool $\mathcal{S}_1$ containing k_1' samples, which is formalized as follows:

$$\mathcal{S}_1 = \text{Sample}(\mathcal{R}_1, p_s, k_1') \tag{11}$$

Stage 2: Positive-Centric Retrieval & Uniform Sampling In the second stage, we shift our focus back to the positive sample d^+. We perform a retrieval operation within the subset $\mathcal{S}_1$ obtained from the first stage. The result of this second-stage retrieval serves as the basis for subsequent sampling. Mathematically, the retrieval result can be represented as:

$$\mathcal{R}_2 = \text{Retrieve}(I, d^+, \mathcal{S}_1, k_2) \tag{12}$$

where k_2 is the number of texts to be returned in the second - stage retrieval, which can be flexibly adjusted according to specific sampling requirements.$\mathcal{R}_2$ forms the final basis for our second-stage sampling operations.

After the second retrieval, a fine-grained filtering is performed within $\mathcal{S}_1$ by calculating the similarity scores between all samples and the positive samples. This enriches the hard negative samples in the top-k_2 of the sample pool $\mathcal{S}_1$, providing a more targeted and effective basis for sampling.

For the probability distribution of the second-stage sampling, we adopt a uniform distribution, which is a simple distribution, to further reduce the impact of false negative samples. The probability distribution is given by:

$$p_\mu(d^-) \propto \frac{1}{|\mathcal{R}_2|} \tag{13}$$

In this formula, $\mathcal{R}_2 = \{d_1^-, d_2^-, \ldots, d_{k_2}^-\}$ represents the negative candidate pool. The uniform distribution ensures equal sampling probability for each candidate, proportional to $\frac{1}{|\mathcal{R}_2|}$.

Finally, using this distribution, we sample from the retrieval results $\mathcal{R}_2$ to obtain the final subset of negative samples $\hat{\mathcal{D}}_q^-$ containing n negative samples .

$$\hat{\mathcal{D}}_q^- = \text{Sample}(\mathcal{R}_2, p_\mu, n) \tag{14}$$

In essence, the biretrieval sampling method first uses the original query to reduce the large text collection to a more manageable subset. Then, it computes the similarity with the positive sample to perform a refined retrieval within this subset, providing a more targeted and effective basis for sampling.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets

We mostly use a publicly available dataset Ms marco Passage Nguyen et al. (2016) in our experiments. It contains a large number of real-world queries and passages/documents collected from Bing search logs. In addition, we also conducted experiments on the TREC-2019 Craswell et al. (2020), and TREC-2020 Craswell et al. (2021) to verify the effectiveness

Table 1: Statistics of the text retrieval datasets.

Dataset	Train	Dev	Test	#Passage
MS MARCO Passage Ranking (MS-Pas)	502,939	6,980	-	8,841,823
TREC 2019 Deep Learning Track (TREC-2019)	-	-	200	8,841,823
TREC 2020 Deep Learning Track (TREC-2020)	-	-	200	8,841,823
Natural Questions (NQ)	58,880	8,757	3,610	21,015,324

of our method on different datasets. TREC (Text Retrieval Conference) datasets (TREC-2019 and TREC-2020) are Datasets from NIST-organized evaluation conferences, containing text data and query tasks for evaluating information retrieval systems to help compare methods and advance the field. The dataset information is shown in Table. 1.

4.2. Results and Analysis

Due to computational constraints, all experiments were conducted under the standard DPR framework with BERT-base-uncased as the backbone PLM. This ensures a fair and reproducible setting, further implementation details are provided in the Appendix of our GitHub repository.

From the results of Teble 2, we can see that TriSampler performs the best among the previous negative sample sampling methods. In terms of the effectiveness of various negative sampling methods on DPR alone, we have achieved the best results. The experimental results show that when the negative samples obtained by our sampling method are used for training, compared with the model trained by the BM-25 sampling method, MRR@10 is increased by 8.9%. And compared with the TriSampler method, it is increased by 1.2%. It also achieved excellent performance in the nDCG@10 metric on the TREC-2019 and TREC-2020 datasets.

The TopK sampling method has shown excellent results in the experimental environment we set up. It might even outperform the denoising method in some metrics. The reason could be that sampling is only carried out once during the DPR training process, while in the model training that requires multiple samplings, it will be more restricted compared to the denoising method. SimANS is of great significance because it can not only be combined with negative sample training algorithms but also be applied to many sampling methods, such as TriSampler and the sampling method proposed by us.

5. Discussion

5.1. Ablation Study

In the ablation study, for the first experiment, the first-stage probability distribution in negative sampling was removed, relying solely on the original sampling method for training and evaluation. In the second one, the first-stage operation remained, but the second-stage retrieval was removed, with only the first-stage retrieval being done and no follow-up screening based on its results. The third experiment skipped the first-stage retrieval while keeping the second-stage one, using the original, unscreened document set for the second-stage retrieval. In the fourth experiment, the retrieval order was reversed. First, the retrieval using positive documents was carried out, and probability distribution sampling

Table 2: Results on three web search datasets. The column "Negative Selection" indicates the types of hard negative samples used. The best results are bolded, and the second-best results are underlined.

Method	Negative selection	MS-MARCO			TREC-19	TREC-20
		MRR@10	R@50	R@1k	nDCG@10	nDCG@10
BM25 Yang et al. (2017)	—	18.5	58.5	85.7	51.2	47.7
DeepCT Dai and Callan (2019)	—	24.3	69.0	91.0	57.2	-
docT5query Nogueira et al. (2019)	—	27.7	75.6	94.7	64.2	-
ANCE Xiong et al. (2021)	Dynamic Hard Negatives	33.0	-	95.9	64.5	64.6
STAR Zhan et al. (2021)	Static Hard Negatives	34.0	-	-	64.2	-
ADORE Zhan et al. (2021)	Dynamic Hard Negatives	34.7	-	-	<u>68.3</u>	-
SEED Lu et al. (2021)	Dynamic Hard Negatives	33.9	-	96.1	-	-
DPR+BM25 Neg	Static Hard Negatives	32.4	79.0	95.4	59.4	62.7
DPR+TopK Neg Xiong et al. (2021)	Static Hard Negatives	34.2	<u>81.2</u>	<u>96.2</u>	67.3	**66.9**
DPR+SimANS Zhou et al. (2022a)	Denoised Hard Negatives	34.8	80.2	96.1	67.6	**66.9**
DPR+TriSampler Yang et al. (2024)	Denoised Hard Negatives	<u>34.9</u>	80.9	95.9	65.4	62.8
DPR+ReSa2	Denoised Hard Negatives	**35.3**	**81.8**	**96.4**	**68.6**	<u>66.3</u>

was calculated based on the similarity distances to the positive documents. Then the query retrieval was performed. All ablation experiments were conducted on the MS MARCO Passage Ranking dataset.

Table 3: Ablation Experiments

Experimental Conditions	MRR@10	Recall@1000
Remove the first-stage probability distribution	33.69	95.30
Remove the second-stage retrieval	35.14	96.48
Remove the first-stage retrieval	33.34	94.62
Swap the first and second-stage retrieval	33.60	95.06

From the results, it can be seen that the retrieval sampling in the first stage contributes the most significant improvement. Considering the ablation experiment of "Remove the second-stage retrieval", we adopted a distribution similar to that of SimANS. However, the reason why the results are significantly better than those of SimANS in Table 3 is that in our method, we do not directly use the distribution to extract the negative samples for final training. Instead, we use this probability distribution to filter out some samples that are more likely to be false negatives. The comparison between the results of this ablation experiment and the full experiment shows that the second-stage retrieval sampling indeed plays a role in enriching more informative samples. And as can be seen from the results of the other three ablation experiments, without pre-filtering some false negative samples in advance, directly utilizing the similarity ranking of positive samples is more likely to lead to a poor result.

5.2. External Validation

To evaluate ReSa2's generalization to unseen domains, we tested it on the Natural Questions (NQ) dataset, distinct from the training corpus (MS MARCO). Trained exclusively on MS MARCO without NQ exposure, DPR+ReSa2 outperformed baseline DPR across all metrics (Table 4).

Table 4: Generalization Ability Comparison among DPR, DPR+SimANS and DPR+ReSa2

Metric	Method	@1	@3	@5	@10	@100
NDCG	DPR	0.28563	0.37971	0.41444	0.44838	0.49494
	DPR+SimANS	0.27723	0.36200	0.39368	0.42388	0.47367
	DPR+ReSa2	**0.30041**	**0.38955**	**0.42206**	**0.45463**	**0.50337**
MAP	DPR	0.25381	0.34557	0.36601	0.38111	0.39113
	DPR+SimANS	0.24889	0.33074	0.34931	0.36272	0.37359
	DPR+ReSa2	**0.26895**	**0.35647**	**0.37543**	**0.39032**	**0.40100**
Recall	DPR	0.25381	0.44899	0.52909	**0.62865**	0.83959
	DPR+SimANS	0.24889	0.42454	0.4978	0.58596	0.80927
	DPR+ReSa2	**0.26895**	**0.45553**	**0.53107**	0.62522	**0.84323**
Precision	DPR	0.28563	0.17285	0.12370	**0.07433**	0.01011
	DPR+SimANS	0.27723	0.16329	0.11593	0.06909	0.00975
	DPR+ReSa2	**0.30041**	**0.17545**	**0.12381**	0.07390	**0.01017**
MRR	DPR	0.28563	0.37553	0.39401	0.40669	0.41459
	DPR+SimANS	0.27723	0.35849	0.37551	0.38712	0.3961
	DPR+ReSa2	**0.30070**	**0.38644**	**0.40366**	**0.41582**	**0.42460**

Gains were most significant in precision-critical metrics, while recall improvements confirmed cross-domain semantic retrieval. ReSa2 preserves relevance via query/positive-centric retrieval and mitigates false negatives through probabilistic filtering and uniform sampling, validating its domain-agnostic effectiveness for dense retrieval.

5.3. Time Complexity of ReSa2

The time complexity of the ReSa2 algorithm is determined by three core operations: ANN index construction, two-stage retrieval, and two-stage sampling. Herein, we assume the employed ANN indexing algorithm is Hierarchical Navigable Small Worlds (HNSW) Malkov and Yashunin (2018), and conduct a quantitative analysis by leveraging the structural characteristics of HNSW. Index construction costs $O(N \log N)$ for N vectors. Each retrieval adds $O(\log N)$, Stage-1 samples k_1 candidates in $O(k_1 + k_1')$ via Gaussian weights, stage-2 samples n negatives in $O(n)$. With $k_1, k_1, m \ll N$, total complexity is $O(N \log N)$, which is consistent with that of the SimANS and TriSampler algorithms.

5.4. Performance in the Low-Resource Setting

With a 10 k-query MS MARCO subset, we trained a deliberately weak retriever to simulate a low-resource setting. Under this setting, ReSa2 achieves an MRR@10 of 22.9, outperforming TopK (22.0) and TriSampler (22.7). This demonstrates its resilience and effectiveness even when retrieval quality is suboptimal. For additional details on these time complexity analyses and robustness experiments, please refer to the Appendix on GitHub.

6. Conclusion

We investigated the effects of various hard negative sample samplings, with a primary focus on debiased hard negative sample sampling. First, we employed large language models to conduct insightful experiments for simple judgment of retrieved samples, and analyzed negative sampling from the perspective of the impact of false negative samples and overly simple negative samples on gradients. Subsequently, to address the limitations of existing negative sampling algorithms, we proposed the two-stage ReSa2 algorithm for negative sampling in dense retrieval. Under the verification method we uniformly designed, experimental results on MS MARCO Passage, TREC-2019, and TREC-2020 datasets showed that our method achieves optimal performance. Ablation experiments indicated that the first stage makes the greatest contribution to denoising, while the second stage enriches samples containing useful information and improves effectiveness, proving that our design aligns with practical scenarios. Experiments on the untrained NQ dataset demonstrated that our model has strong generalization ability.

References

Xuanang Chen, Jian Luo, Ben He, Le Sun 0001, and Yingfei Sun. Towards robust dense retrieval via local ranking alignment. In *IJCAI*, pages 1980–1986, 2022.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820, 2020. URL https://arxiv.org/abs/2003.07820.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2020 deep learning track. *CoRR*, abs/2102.07662, 2021. URL https://arxiv.org/abs/2102.07662.

Zhuyun Dai and Jamie Callan. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of SIGIR 2019*, pages 985–988, 2019. URL https://doi.org/10.1145/3331184.3331303.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In Mohit Bansal and Aline Villavicencio, editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1049. URL https://aclanthology.org/K19-1049/.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *ArXiv e-prints*, 2017.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transaction's on Big Data*, 7(3):535–547, 2021. URL https://doi.org/10.1109/TBDATA.2019.2921572.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP 2020*, pages 6769–6781, 2020. URL https://aclanthology.org/2020.emnlp-main.550.

Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. Less is more: Pretrain a strong Siamese encoder for dense text retrieval using a weak decoder. In *Proceedings of EMNLP 2021*, pages 2780–2791, 2021. URL https://aclanthology.org/2021.emnlp-main.220.

Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, and Haifeng Wang. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *CoRR*, abs/2205.09153, 2022. URL https://doi.org/10.48550/arXiv.2205.09153.

Kun Luo, Minghao Qin, Zheng Liu, Shitao Xiao, Jun Zhao, and Kang Liu. Large language models as foundations for next-gen dense retrieval: A comprehensive empirical assessment. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1354–1365, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.80. URL https://aclanthology.org/2024.emnlp-main.80/.

Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset.

In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016*, volume 1773, 2016. URL http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. From doc2query to doctttttquery. *Online preprint*, 6(2), 2019.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.466. URL https://aclanthology.org/2021.naacl-main.466/.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of EMNLP 2021*, pages 2825–2835, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.224.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. LexMAE: Lexicon-bottlenecked pretraining for large-scale retrieval. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PfpEtB3-csK.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.125. URL https://aclanthology.org/2023.acl-long.125/.

Shiqi Wang, Yeqin Zhang, and Cam-Tu Nguyen. Mitigating the impact of false negative in dense retrieval with contrastive confidence regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19171–19179, 2024.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1253–1256, 2017. URL https://doi.org/10.1145/3077136.3080721.

Zhen Yang, Zhou Shao, Yuxiao Dong, and Jie Tang. Trisampler: a better negative sampling principle for dense retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9269–9277, 2024.

Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. Learning discriminative projections for text similarity measures. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 247–256, 2011.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1503–1512, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462880. URL https://doi.org/10.1145/3404835.3462880.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. Adversarial retriever-ranker for dense text retrieval. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=MR7XubKUFB.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.*, 42(4), February 2024. ISSN 1046-8188. doi: 10.1145/3637870. URL https://doi.org/10.1145/3637870.

Kun Zhou, Yeyun Gong, Xiao Liu, Wayne Xin Zhao, Yelong Shen, Anlei Dong, Jingwen Lu, Rangan Majumder, Ji-Rong Wen, Nan Duan, et al. Simans: Simple ambiguous negatives sampling for dense text retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022a.

Kun Zhou, Beichen Zhang, Xin Zhao, and Ji-Rong Wen. Debiased contrastive learning of unsupervised sentence representations. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6120–6130, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.423. URL https://aclanthology.org/2022.acl-long.423/.

Kun Zhou, Xiao Liu, Yeyun Gong, Wayne Xin Zhao, Daxin Jiang, Nan Duan, and Ji-Rong Wen. Master: Multi-task pre-trained bottlenecked masked autoencoders are better dense retrievers. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 630–647. Springer, 2023.