Bootstrapped Pre-training with Dynamic Identifier Prediction for Generative Retrieval

Yubao Tang[1,2] **Ruqing Zhang**[1,2] **Jiafeng Guo**[1,2*] **Maarten de Rijke**[3]
Yixing Fan[1,2] **Xueqi Cheng**[1,2]
[1]CAS Key Lab of Network Data Science and Technology, ICT, CAS
[2]University of Chinese Academy of Sciences
[3]University of Amsterdam
{tangyubao21b,zhangruqing,guojiafeng,fanyixing,cxq}@ict.ac.cn
m.derijke@uva.nl

Abstract

Generative retrieval uses differentiable search indexes to directly generate relevant document identifiers in response to a query. Recent studies have highlighted the potential of a strong generative retrieval model, trained with carefully crafted pre-training tasks, to enhance downstream retrieval tasks via fine-tuning. However, the full power of pre-training for generative retrieval remains underexploited due to its reliance on pre-defined static document identifiers, which may not align with evolving model parameters. In this work, we introduce BootRet, a bootstrapped pre-training method for generative retrieval that dynamically adjusts document identifiers during pre-training to accommodate the continuing memorization of the corpus. BootRet involves three key training phases: (i) initial identifier generation, (ii) pre-training via corpus indexing and relevance prediction tasks, and (iii) bootstrapping for identifier updates. To facilitate the pre-training phase, we further introduce noisy documents and pseudo-queries, generated by large language models, to resemble semantic connections in both indexing and retrieval tasks. Experimental results demonstrate that BootRet significantly outperforms existing pre-training generative retrieval baselines and performs well even in zero-shot settings.

1 Introduction

Document retrieval is an important task with widespread applications, such as question answering (Karpukhin et al., 2020a; Lee et al., 2019) and fact verification (Chakrabarty et al., 2018; Olivares et al., 2023), which aims to retrieve candidate documents from a huge document collection for a given query (Gao and Callan, 2022; Nie et al., 2020). Currently, the dominant implementation is dense retrieval (Xiong et al., 2017; Zhan et al., 2020a), which encodes the query and documents into dense embedding vectors to capture rich semantics.

Generative retrieval. An emerging alternative to dense retrieval in document retrieval is *generative retrieval* (GR) (Tang et al., 2023; Tay et al., 2022). It employs a sequence-to-sequence (Seq2Seq) architecture to generate relevant document identifiers (docids) for queries. In this manner, the knowledge of all documents in the corpus is encoded into the model parameters, similar to the human cognitive associative mechanism (Anderson and Bower, 2014; Kounios et al., 2001). To achieve this, GR involves two basic operations (Tay et al., 2022): (i) *indexing*, which memorizes the entire corpus by associating each document with its identifier, and (ii) *retrieval*, which uses the indexed corpus information to produce a ranked list of potentially relevant docids for a given query.

Using general language models, such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2020), as the base Seq2Seq model has become a popular choice in GR (Bevilacqua et al., 2022; De Cao et al., 2021; Zhuang et al., 2023). On top of this, some work has designed pre-training objectives for GR. For example, Zhou et al. (2022) proposed indexing- and retrieval-based pre-training tasks; document pieces or pseudo-queries are used as input, and docids (e.g., product quantization code) are predicted as output with maximum likelihood estimation (MLE). Similarly, Chen et al. (2022) proposed retrieval-based tasks, which aim to construct and learn pairs of pseudo-queries and docids (i.e., Wikipedia titles) from the corpus. These works demonstrate that applying specialized pre-trained models to GR yields superior results compared to using general language models.

Research challenges. While pre-training methods have shown their effectiveness, important limitations remain in the following: (i) *The construction process of pre-defined docids is independent from*

*Corresponding author.

the pre-training process. This results in a semantic gap between both processes, which could potentially hinder the retrieval performance. (ii) *Docids remain unchanged during pre-training.* If the initial docids are not suitable, they cannot be further adjusted after the training begins. Consequently, it may become challenging to learn semantics and relationships between documents, impeding the achievement of satisfactory retrieval performance. (iii) *Existing pre-training methods do not explicitly consider the interrelations between document-docid or query-docid pairs.* The widely-used MLE objective may result in difficulties in distinguishing among similar documents and docids. Therefore, we argue that the model should enhance its discriminative and generalization ability.

Approach. To address these challenges, we introduce a general bootstrapped pre-training method for GR, called BootRet. Our objective is to dynamically adjust docids in accordance with the evolving model parameters during pre-training. The key idea is inspired by that the human brain updates the organization of existing knowledge to better match updated goals or contents in learning (Mack et al., 2016). BootRet includes three key steps: (i) *Initial docid generation*. We leverage the encoder of the initial model to encode documents and then obtain the product quantization code (Ge et al., 2013; Zhan et al., 2021) as the initial docids. (ii) *Pre-training*. We design two pre-training tasks, i.e., corpus indexing task and relevance predication task. The corpus indexing task aims to memorize corpus information and distinguish among similar documents and docids. We construct pairs of original documents and corresponding identifiers to simulate the indexing operation. To enhance discrimination and generalization, we use a large language model (LLM) to generate noisy documents similar to the originals, creating pairs of noisy documents and identifiers. Besides, we design contrastive losses to help the model memorize and contrast these pairs. The relevance prediction task aims to learn relevance information from the corpus. We construct pairs of pseudo-queries and relevant docids to simulate the retrieval operation. We also use a LLM to generate high-quality pseudo–queries for original documents as input and design a contrastive loss for the model to predict and contrast docids. These two tasks are jointly learned, with the docids remaining fixed throughout this process. (iii) *Enhanced bootstrapping*. The encoder of the model pre-trained with the above two tasks is further used to encode documents, updating document representations, and then updating the PQ code, i.e., docids. These updated docids are further used to retrain the model based on the pre-training tasks. Steps (ii) and (iii) iteratively update the model parameters and docids.

We pre-train BootRet based on two kinds of large scale text corpus, i.e., MS MARCO (Nguyen et al., 2016) and Wikipedia (Wikipedia, 2022). We then fine-tune BootRet on two representative downstream datasets widely used in GR research. The empirical experimental results show that BootRet can achieve significant improvements over strong GR baselines.

Contributions. Our main contributions are: (i) We propose a bootstrapped pre-training framework for GR to iteratively update the model parameters and docids. (ii) BootRet demonstrates superior performance in downstream retrieval tasks. For instance, on the MS MARCO dataset, it outperforms the strong pre-training GR baseline, Ultron (Zhou et al., 2022), by 11.8% in terms of Hits@1. (iii) Additionally, BootRet exhibits better zero-shot performance than other general language models.

2 Related Work

Generative retrieval. GR marks a new paradigm in document retrieval that generates identifier strings of documents as the retrieval target (Metzler et al., 2021; Tay et al., 2022). The current design of docids can be categorized into two types. (i) *Pre-defined static docids.* They remain unchanged during training, such as document titles (De Cao et al., 2021), URLs (Ren et al., 2023), product quantization code (Chen et al., 2023; Mehta et al., 2023). This design is simple and shows decent performance (Chen et al., 2022), but the pre-defined process is independent of training. (ii) *Learnable docids.* They are optimized jointly with the retrieval task (Sun et al., 2023; Wang et al., 2023). Though these docids are dynamic, their optimization primarily targets retrieval. Nevertheless, docids serve functions in both indexing and retrieval.

In addition to widely-used supervised learning approaches (Sun et al., 2023; Zhang et al., 2023; Zhou et al., 2023), recent studies (Chen et al., 2022; Zeng et al., 2023; Zhou et al., 2022) have explored pre-training for GR. However, each study adopts fixed docids, ignoring the potential mismatch between docids and the updated model. In contrast, our work dynamically updates both the docids and

the evolving model to enhance the effectiveness.

Bootstrapping. The idea of bootstrapping training methods have garnered significant interest in various natural language processing tasks (Deepika and Geetha, 2021; Song and Roth, 2014; Wu et al., 2009). The approach involves generating new training data or information based on the previous model to iteratively enhance its capabilities. While the techniques are sometimes used in conjunction with supervised learning (Deepika and Geetha, 2021; Song and Roth, 2014), our scenario involves an unlabeled corpus without ideal docids. Thus, we adopt a more unsupervised approach, iteratively refining the GR model and docids.

Dense retrieval. Dense retrieval (Gao and Callan, 2022; Karpukhin et al., 2020a) is currently the de facto solution for document retrieval. It focuses on representing documents and queries as dense vectors in continuous spaces, capturing semantic relationships. Efficient vector search is facilitated by approximate nearest neighbor (Xiong et al., 2020) algorithms. Further enhancements include using pre-trained models within a dual-encoder architecture (Nie et al., 2020; Zhan et al., 2020b) and hard negative mining techniques (Karpukhin et al., 2020b; Zhan et al., 2020a). Compared to dense retrieval, GR could achieve end-to-end global optimization. However, its current performance lags behind state-of-the-art methods in dense retrieval.

3 Method

This section introduces the details of the BootRet model proposed in this paper. As shown in Figure 1, (i) Given a corpus $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$, we first construct an initial docid id_i for each document d_i in $\mathcal{D}$. The initial docid set is denoted as $\mathcal{I}^0_{\mathcal{D}}$. We employ an encoder-decoder language model as the base model, where initial parameters are denoted as θ^0. (ii) Then, while keeping $\mathcal{I}^0_{\mathcal{D}}$ unchanged, we carefully design two pre-training tasks. During pre–training, the model parameters are updated from θ^0 to θ^1. (iii) Subsequently, fixing θ^1, we update the docids to $\mathcal{I}^1_{\mathcal{D}}$, thus completing one iteration. The updated docids can be used to further retrain the model for a next iteration. We define the t-th iteration as updating the model parameters from θ^{t-1} to θ^t with fixed $\mathcal{I}^{t-1}_{\mathcal{D}}$ in Step (ii), and then based on fixed θ^t, updating $\mathcal{I}^{t-1}_{\mathcal{D}}$ to $\mathcal{I}^t_{\mathcal{D}}$ in Step (iii).

3.1 Model Architecture

Like previous GR research (Chen et al., 2022; Tay et al., 2022; Wang et al., 2022), we leverage a transformer-based model comprising: (i) An encoder, a bidirectional encoder to encode documents or pseudo-queries. (ii) An identifier decoder, operating through a sequential generation process to produce document identifiers. We initialize the model with T5-base (Raffel et al., 2020), and the initial model parameter is denoted as θ^0.

3.2 Initial Docid Generation

Docids with semantic ties to the document content aid the model's learning (Tay et al., 2022). For effective bootstrapping, docids need efficient updates based on the model's progress. Considering these needs, we choose the widely used PQ code (Chen et al., 2023) as the docid.

Specifically, we first encode all the documents to obtain document vectors with the encoder of θ^0. Following (Zhou et al., 2022), vectors are evenly divided into g groups. For each group, we apply the K-means clustering algorithm to obtain k cluster centers. Then, the docid can be represented by cluster indices of length g corresponding to the clusters. And we denote the initial docid set as $\mathcal{I}^0_{\mathcal{D}}$. To facilitate the generation of docids, we include all cluster indices from all groups obtained in the docid generation process as new tokens added to the model vocabulary. $\mathcal{I}^0_{\mathcal{D}}$ will be used for subsequent iterative pre-training and updates.

3.3 Pre-training Tasks

The core idea is to construct pseudo document-docid pairs and query-docid pairs to simulate the indexing and retrieval operations, respectively. Our two pre-training tasks are: (i) *Corpus indexing task.* We first construct pairs of original documents and their corresponding identifiers. For original documents, we use a LLM to construct similar but noisy documents $\tilde{\mathcal{D}}$. $\tilde{d}^h_i \in \tilde{\mathcal{D}}$ is the h-th noisy version of d_i. And we design multiple losses to guide the model to learn the associations between original or noisy documents and their identifiers. (ii) *Relevance prediction task.* We use a LLM to construct pseudo-queries $\mathcal{Q}$, and pair them with relevant docids. For d_i, we construct a total of X queries, and $q^x_i \in \mathcal{Q}$ denotes the x-th pseudo-query.

3.3.1 Corpus Indexing Task

We introduce the construction of noisy documents and pre-training objectives in detail.

Noisy document construction. The noisy documents should maintain semantic consistency with the originals while remaining distinguishable. We

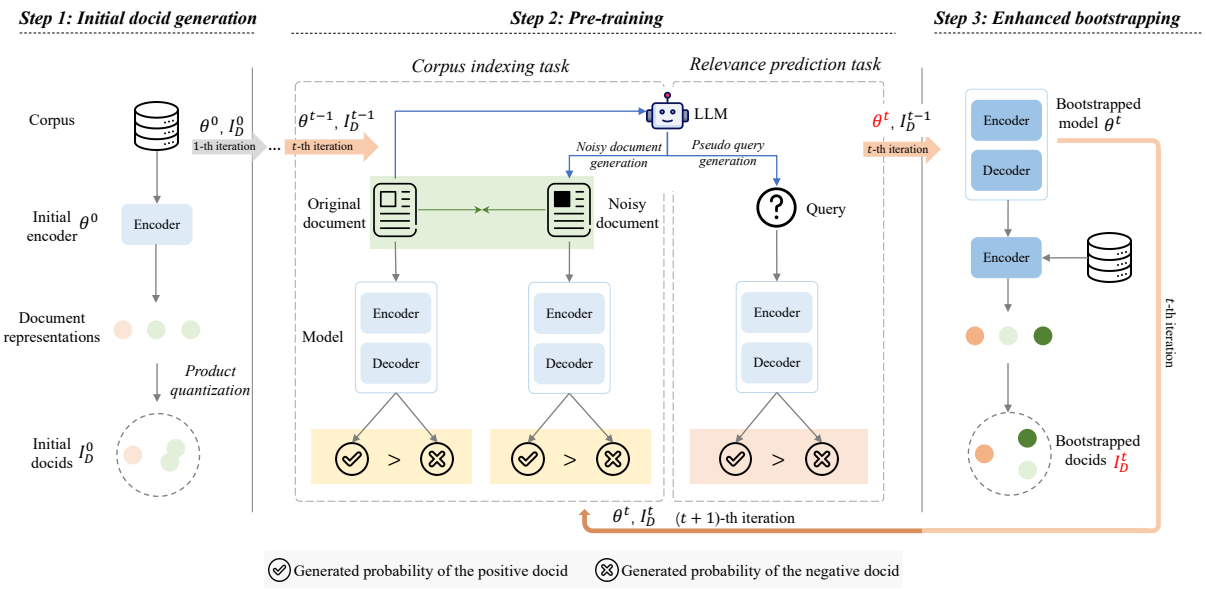

Figure 1: The bootstrapped pre-training pipeline of BootRet. (1) The initial docids $\mathcal{I}_{\mathcal{D}}^{0}$ are obtained with the initial model parameters θ^{0}. (2) To perform the t-th iteration, we design the corpus indexing task and relevance prediction task for pre-training. We construct noisy documents and pseudo-queries with a LLM, and design contrastive losses (the yellow and the orange rectangles) and a semantic consistency loss (the green rectangle) to learn the corpus and relevance information discriminatively. After pre-training, the model updates from θ^{t-1} to θ^{t}. (3) The bootstrapped θ^{t} is used to dynamically update the docids $\mathcal{I}_{\mathcal{D}}^{t-1}$ to $\mathcal{I}_{\mathcal{D}}^{t}$, i.e., bootstrapped docids, which are further used in the next iteration. (Figure should be viewed in color.)

propose leveraging a LLM to effectively achieve this. Inspired by (Raffel et al., 2020), we design the following four prompts to guide LLM generation:

- A synonym replacement prompt: "`Replace some words in the following document with their synonyms while maintaining the overall semantic meaning: {d}.`"
- A sentence removal prompt: "`Remove one or more sentences from the following document, while maintaining the overall semantic meaning: {d}.`"
- A sentence shuffling prompt: "`Rearrange the sentences in the following document to create a new flow, while maintaining the overall semantic meaning: {d}.`
- A word masking prompt: `Mask some words with [Masked] in the following document, while maintaining the overall semantic meaning: {d}.`"

Combining these prompts with an original document as the input, LLM generates four noisy documents, sharing the same docid with the original.

Pre-training objective. In the t-th iteration, the objective consists of three parts as the follows.

- **Semantic consistency loss**: It aims at maintaining overall semantic consistency between original and noisy documents. Specifically, in a mini-batch, there are a total of $4N$ document-docid pairs, where N pairs correspond to the original pairs, and each original document has four noisy pairs. This loss $L_{SC}(\mathcal{D}, \tilde{\mathcal{D}}; \theta^{t-1})$ is defined as:

$$\sum_{i=1}^{N}\sum_{h=1}^{4} 1 - \mathrm{sim}(\mathrm{Enc}(d_i), \mathrm{Enc}(\tilde{d}_i^{h})), \quad (1)$$

where θ^{t-1} denotes model parameters of the previous iteration and $\mathrm{sim}(\cdot)$ is the cosine function.

- **Contrastive losses for corpus indexing**: Conditioned on original document-docid pairs, we encourage the model to generate a docid that corresponds to the document rather than the docids of other documents. In the same mini-batch, we aim for the model to generate the docid corresponding to the document with a higher probability than generating others. Inspired by contrastive learning (Khosla et al., 2020), this loss $L_{C1}(\mathcal{D}, \mathcal{I}_{\mathcal{D}}; \theta^{t-1})$ is formalized as:

$$-\sum_{i=1}^{N} \log \frac{\exp(P(id_i \mid d_i)/\tau)}{\sum_{j=1}^{N} \exp(P(id_j \mid d_i)/\tau)}, \quad (2)$$

where τ is the temperature hyperparameter. $P(id_i \mid d_i)$ is the generated likelihood probability of id_i conditioned on d_i. Similarly, for

noisy pairs, the loss $L_{C2}(\tilde{\mathcal{D}}, \mathcal{I}_{\mathcal{D}}; \theta^{t-1})$ is:

$$-\sum_{i=1}^{N}\sum_{h=1}^{4}\log\frac{\exp(P(id_i \mid \tilde{d}_i^h)/\tau)}{\sum_{j=1}^{N}\exp(P(id_j \mid \tilde{d}_i^h)/\tau)}. \quad (3)$$

The pre-training objective of the corpus indexing task $L_{CI}(\mathcal{D}, \tilde{\mathcal{D}}, \mathcal{I}_{\mathcal{D}}; \theta^{t-1})$ is a weighted sum of the three aforementioned losses, denoted as:

$$L_{SC}(\cdot) + \alpha L_{C1}(\cdot) + \beta L_{C2}(\cdot), \quad (4)$$

where α and β are hyperparameters.

3.3.2 Relevance Prediction Task

We introduce the construction process of pseudo-queries, and the pre-training objective.

Pseudo-query construction. To generate high-quality pseudo-queries for the original documents, we employ a LLM using the prompt: "`Given the following document {d}, generate {X} insightful queries that a reader might have after reading the content. Ensure the queries cover key concepts.`" When the prompt is combined with a document d_i and the required number of pseudo-queries X as input, we obtain well-written pseudo-queries. They share the same docids as the input original document.

Pre-training objective. Similarly, we ensure that the model tends to generate relevant docids than irrelevant ones. In the same mini-batch, the loss $L_{RP}(\mathcal{Q}, \mathcal{I}_{\mathcal{Q}}; \theta^{t-1})$ in the t-th iteration is:

$$-\sum_{i=1}^{N}\sum_{x=1}^{X}\log\frac{\exp(P(id_i \mid q_i^x)/\tau)}{\sum_{j=1}^{N}\exp(P(id_j \mid q_i^x)/\tau)}. \quad (5)$$

3.3.3 Joint Learning

We jointly pre-train the model with two above objectives and two sequence generation objectives. In the t-th iteration, the overall loss $L_{Pre}(\mathcal{D}, \tilde{\mathcal{D}}, \mathcal{I}_{\mathcal{D}}, \mathcal{Q}, \mathcal{I}_{\mathcal{Q}}; \theta^{t-1})$ is :

$$\gamma L_{CI}(\cdot) + \rho L_{RP}(\cdot) + \lambda L_{ID}(\cdot) + \lambda L_{RE}(\cdot), \quad (6)$$

where γ, ρ and λ are hyperparameters; $L_{ID}(\mathcal{D}, \tilde{\mathcal{D}}, \mathcal{I}_{\mathcal{D}}; \theta^{t-1})$ is the widely used standard MLE loss based on document-docid pairs:

$$-\sum_{i=1}^{|\mathcal{D}|}\log P(id_i \mid d_i) - \sum_{i=1}^{|\mathcal{D}|}\sum_{h=1}^{4}\log P(id_i \mid \tilde{d}_i^h). \quad (7)$$

Note, Eq. (2) and Eq. (3) ensure that the model's probability of generating the corresponding docid is greater than generating other docids. Eq. (7) does not explicitly contrast with other docids. $L_{RE}(\mathcal{Q}, \mathcal{I}_{\mathcal{D}}; \theta^{t-1})$ is based on query-docid pairs:

$$-\sum_{i=1}^{|Q|}\log P(id_i \mid q_i). \quad (8)$$

During training, we construct two types of batch data. One type has original and noisy documents-docid pairs, optimized using Eq. (4) and Eq. (7). The other type has pairs of the pseudo-query and relevant docid only, optimized using Eq. (5) and Eq. (8). After jointly training during the t-th iteration, θ^{t-1} updates to θ^t with docids fixed.

3.4 Enhanced Bootstrapping Strategy

Based on the updated θ^t, we introduce how to update docids $\mathcal{I}_{\mathcal{D}}^{t-1}$ and retrain the model.

Docid update. Fixing θ^t, we use the encoder of θ^t to encode documents as in Section 3.2, to update docids of the previous iteration $\mathcal{I}_{\mathcal{D}}^{t-1}$, to $\mathcal{I}_{\mathcal{D}}^{t}$. We refer to the version following the initial iteration's completion (i.e., $\mathcal{I}_{\mathcal{D}}^{1}$ and θ^1) as **BootRet-Bs**.

Retrain the model. To proceed to the next iteration, we retrain the model with $\mathcal{I}_{\mathcal{D}}^{t}$ as described in Section 3.3. After multiple iterations, we achieve continuous dynamic alignment and enhancement. We refer to this version as **BootRet-Mt**.

4 Experimental Settings

Pre-training corpus. For pre-training we use two large, publicly available corpora: (i) *English Wikipedia*, which contains tens of millions of well-written documents and we downloaded this dump (Wikipedia, 2022) for pre-training, and (ii) the *MS MARCO Document Collection* (Nguyen et al., 2016), which has about 3 million documents extracted from web documents using the Bing search engine. For each corpus, we sample 500K documents and generated four noisy documents and five pseudo-queries, i.e., X, for each document. This results in 2.5M documents and 2.5M pseudo-queries for pre-training. BootRet-BS^{Wiki} and BootRet-BS^{MS} denote the model pre-trained on Wikipedia and MS MARCO, respectively.

Downstream retrieval datasets. We leverage two representative retrieval datasets. (i) *MS MARCO Document Ranking dataset* (Nguyen et al., 2016). Following the setup of (Zhou et al., 2022), we sample a subset of 300K documents for experimentation, denoted as MS 300K, containing 360K training queries, 6980 evaluation queries. These

documents do not overlap with the ones used in pre–training. (ii) *Natural Question* (NQ) (Kwiatkowski et al., 2019) has real questions and Wikipedia documents, having about 228K documents with 307K training queries and 7.8K test queries.

Baselines. Following typical GR research (Tay et al., 2022; Wang et al., 2022), we examine three types of baseline: (i) *Sparse retrieval baselines*: BM25 (Robertson et al., 1995), and DocT5Query (Nogueira and Lin, 2019b). (ii) *Dense retrieval baselines*: RepBERT (Zhan et al., 2020b), DPR (Karpukhin et al., 2020a), and ANCE (Xiong et al., 2020). (iii) *Advanced GR baselines*: DSI (Tay et al., 2022), GENRE (De Cao et al., 2021), SEAL (Bevilacqua et al., 2022), DSI-QG (Zhuang et al., 2023), NCI (Wang et al., 2022), Ultron-PQ (Zhou et al., 2022), Corpusbrain (Chen et al., 2022), GenRet (Sun et al., 2023), and NOVO (Wang et al., 2023) . For more details on our baselines, please refer to Appendix A.1.

Evaluation metrics. Following GR work (Li et al., 2023; Tay et al., 2022), for NQ, we use hit ratio (Hits@K) with $K = \{1, 10\}$ as the metric. For MS 300K, we also use mean reciprocal rank (MRR@K) with $K = \{3, 20\}$ (Li et al., 2023).

Implementation details. For pre-training, we use the LLaMA-13b model (Touvron et al., 2023) to generate noisy documents and pseudo-queries. We initialize our model with T5-base (220M) (Raffel et al., 2021). For docids, we set the length g of PQ codes to 24, the number of clusters k to 256, and the dimension of vectors to 768. The hyperparameters for pre-training are set to $\alpha = \beta = \lambda = 1$, $\gamma = \rho = 2$ and $\tau = 0.2$. The batch size is 256. The Adam optimizer with a learning rate of 5e-5 is used, and the sequence length of documents is set to 512. The max training step is 500K, with the first iteration occurring at step 100K, followed by iterations every 40K steps thereafter.

For fine-tuning, we use the pre-trained model obtained from the last iteration to generate docids. Models are further fine-tuned with document-docid pairs and labeled query-docid pairs with MLE (Tay et al., 2022). Following (Wang et al., 2022; Zhou et al., 2022), we additionally generate 10 pseudo-queries for each document to enhance training. We set the learning rate as 1e-3, and the max training steps as 30K. Other settings remain consistent with the pre-training stage.

All models are trained on eight NVIDIA Tesla A100 80GB GPUs. For inference, we build a prefix trie (De Cao et al., 2021) for docids and use constrained beam search with 20 beams to decode docids. For more details, please see Appendix A.2.

Method	MRR		Hits	
	@3	@20	@1	@10
BM25	22.57	26.67	24.78	40.73
DocT5query	27.38	29.63	30.13	46.93
RepBERT	31.47	33.68	33.16	55.83
DPR	34.84	36.79	36.52	58.68
ANCE	30.76	34.25	33.63	53.62
DSI	23.21	28.93	28.14	49.72
GENRE	31.12	33.49	33.18	53.56
SEAL	31.35	33.57	33.34	53.74
DSI-QG	33.64	35.81	34.96	58.62
NCI	33.86	36.20	35.02	59.21
Corpusbrain	34.72	37.25	36.14	60.32
Ultron-PQ	35.25	38.41	39.53	62.85
GenRet	37.26	40.53	41.68	64.92
NOVO	38.36	41.29	43.14	64.55
BootRet-Bs^{Wiki}	36.28*	39.25*	40.73*	63.78*
BootRet-Bs^{MS}	37.13*	40.48*	41.56*	64.89*
BootRet-Mt^{Wiki}	38.83*	41.36*	43.97*	65.83*
BootRet-Mt^{MS}	**39.35***	**42.79***	**44.21***	**66.73***

Table 1: Retrieval performance on MS 300K. The best results are shown in **bold**. $*$ indicates statistically significant improvements over the best performing GR baseline NOVO ($p \leq 0.05$).

5 Experimental Results

This section presents the experimental findings.

5.1 Main Results

The comparison between our BootRet and baselines on MS 300K and NQ are shown in Table 1 and Table 2, respectively. We observe: (i) Dense retrieval baselines generally outperform sparse retrieval baselines, indicating that dense vectors capturing rich semantics are more beneficial for retrieval. (ii) Dense retrieval baselines outperform naive GR methods, such as DSI and SEAL, demonstrating the challenge of learning with only labeled data for GR. (iii) DSI-QG and NCI with data augmentation perform better than dense retrieval baselines, suggesting that GR requires more labeled data. (iv) Pre-trained baselines, i.e., Ultron and Corpusbrain, outperform supervised learning GR baselines, highlighting the necessity of pre-training for GR. (v) Our BootRet-Mt^{Wiki} and BootRet-Mt^{MS} outperform base versions and Ultron, demonstrating the effec-

Method	Hits@1	Hits@10
BM25*	29.27	60.16
DocT5query*	39.13	69.72
RepBERT	50.20	78.12
DPR*	52.63	79.31
ANCE	45.42	72.75
DSI*	27.40	56.60
GENRE*	26.30	71.20
SEAL*	26.30	74.50
DSI-QG*	63.49	82.36
NCI	64.24	83.11
Corpusbrain	65.12	84.09
Ultron-PQ	64.61	84.45
GenRet	65.42	85.67
NOVO	66.13	86.24
BootRet-Bs^{Wiki}	66.71*	85.53*
BootRet-Bs^{MS}	65.88	85.04
BootRet-Mt^{Wiki}	**67.32***	**87.59***
BootRet-Mt^{MS}	66.15*	86.31*

Table 2: Retrieval performance on NQ. Methods marked with $\star$ indicate results are obtained from (Bevilacqua et al., 2022; Tay et al., 2022; Zhuang et al., 2023). The best results are shown in **bold**. $*$ indicates statistically significant improvements over the best performing GR baseline NOVO ($p \leq 0.05$).

tiveness of bootstrapped pre-training with dynamic identifiers. (vi) In MS300K, our BootRet-Bs does indeed perform slightly worse compared to strong GR baselines such as GenRet, and NOVO. However, the performance of BootRet-Mt is better than them, which also demonstrates the effectiveness of our approach. Similar conclusions are observed in NQ. (vii) BootRet-Bs^{MS} performs better on MS 300K than BootRet-Bs^{Wiki}, while the opposite is observed on NQ, indicating that the performance of pre-trained models improves when downstream data and pre-training corpora are more similar. Additionally, Table A.3 in the appendix, shows that the performance of GR methods lags behind cross-encoder methods, suggesting ample room for exploration in GR.

5.2 Ablation Study

To analyze the impact of each part of BootRet, we conduct ablation study on the Wikipedia pre-training corpus. From Table 3, we observe the following: (i) When not using dynamic identifiers (i.e., the 2nd row), wherein the model solely undergoes repeated pre-training using fixed docids, the performance significantly deteriorates compared to BootRet-Mt^{Wiki}, affirming the effectiveness of

Method	MS 300K	NQ
	Hits@10	Hits@10
BootRet-Mt^{Wiki}	65.83	87.59
w/o dynamic identifiers	63.14	83.81
BootRet-Bs^{Wiki}	63.78	85.53
w/o pre-training	59.95	83.26
w/o retrieval prediction	63.01	83.82
w/o corpus indexing	63.28	83.91
w/o noisy documents	63.47	84.17
w/o contrastive losses	63.31	83.94

Table 3: Ablation study of the pre-training components.

dynamic identifiers. (ii) When pre-training is not performed, and docids are directly obtained using the initial T5-base model (i.e., the 4th row), the model's performance is lower than that of Ultron. This underscores the necessity of pre-training for GR. (iii) When pre-training does not involve the retrieval prediction task or corpus indexing task (i.e., the 5th-6th rows), the performance is lower than BootRet-Bs^{Wiki} (i.e., the 3rd row). This confirms that pre-training should consider both relevance and corpus information. (iv) Not learning the corpus indexing task (i.e., the 6th row) leads to better performance compared to Ultron, indicating that the contrastive loss in the retrieval prediction task enhances the discriminative ability. (v) When the corpus indexing task does not use noisy documents (i.e., the 7th row), the performance is even lower, demonstrating that both noisy documents and contrastive losses contribute to discriminating similar documents and docids. (vi) When not using contrastive losses, i.e., the 8th row, where pre-training solely uses MLE losses (Eq. (7) and (8)) and Eq. (1), there is a significant decrease in performance compared to BootRet-Bs^{Wiki}, indicating the effectiveness of contrastive losses. The ablation results based on the MS MARCO pre-training corpus show similar trends, as shown in Table 4.

5.3 Zero- and Low-resource Settings

To show whether BootRet can perform well with limited data, we randomly sample 2K, 4K, 6K, and 8K queries from the training set of both datasets. From Figure 2, we observe the following: (i) Under the zero-shot setting, where the model learns solely from the corpus without annotated queries, BootRet-Bs^{Wiki} initially performs worse than Ultron on MS 300K. However, as fine-tuning with annotated queries progresses, BootRet-Bs^{Wiki}

Methods	MS 300K	NQ
	Hits@10	Hits@10
BootRet-Mt^{MS}	66.73	86.31
w/o dynamic identifiers	63.55	84.62
BootRet-Bs^{MS}	64.89	85.04
w/o pre-training	59.57	83.71
w/o retrieval prediction	63.02	84.51
w/o corpus indexing	63.46	84.76
w/o noisy documents	63.95	84.96
w/o contrastive losses	63.24	84.62

Table 4: Ablation study of the pre-training components based on the MS MARCO pre-training corpus.

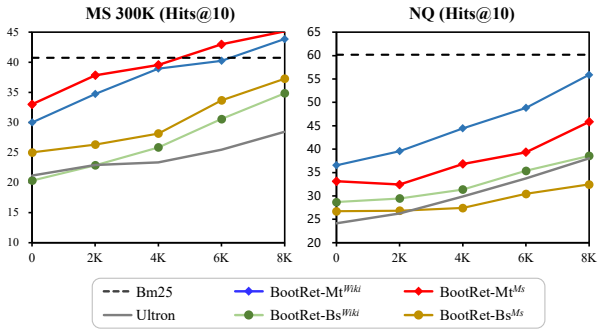

Figure 2: Results under zero- and low-resource setting. The x-axis indicates the number of labeled queries.

quickly surpasses Ultron. This is possibly due to Ultron directly pre-training on the downstream dataset's corpus, while BootRet-Bs^{Wiki}'s pre-training corpus differs significantly from MS 300K. It also indicates that BootRet-Bs^{Wiki} requires less annotated data to achieve rapid performance improvement. (ii) Under the low-resource setting, both base versions of BootRet exhibit performance gaps compared to BM25, highlighting the importance of annotated data for GR. (iii) Both versions of BootRet-Mt demonstrate better performance over base versions. Additionally, they achieve performance comparable to BM25 at approximately 1.3%, i.e., 5K, queries fine-tuning on MS 300K. Similar trends are observed for all methods on NQ, but all GR models perform worse than BM25.

5.4 Impact of the Number of Iteration

The iteration of updating docids and model parameters is important in our proposed bootstrapping pre-training method. We analyze the retrieval performance of the number of iterations on the downstream task, MS 300K, pre-training on the MS MARCO corpus. In Figure 3, we find that performance generally improves as the number of

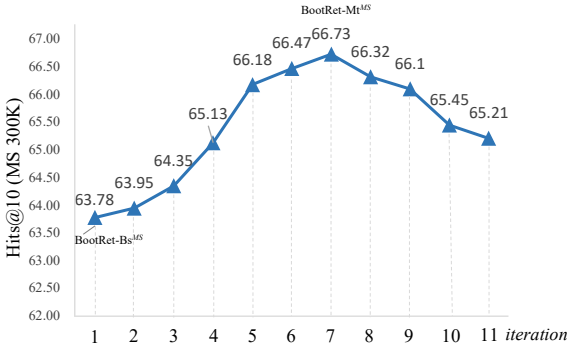

Figure 3: Retrieval performance of different number of iterations on MS 300K.

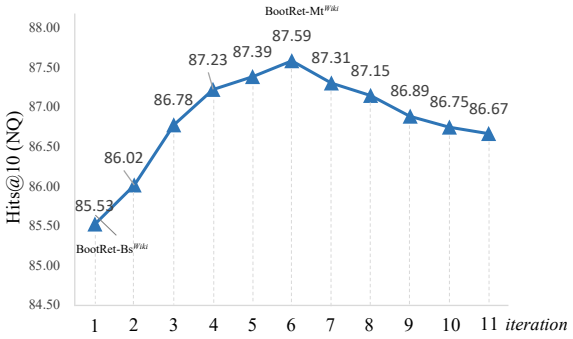

Figure 4: Retrieval performance of different number of iterations on NQ.

iterations increases from 1 to 7, indicating the effectiveness of the bootstrapping pre-training method. However, performance begins to decline gradually after exceeding 7 iterations, possibly due to the model overfitting to the pre-training data.

As shown in Figure 4, it show the retrieval performance of different number of iterations on NQ, which aligns with the trend on MS 300K. The phenomenon that the performance degrades substantially after a certain iteration, is reasonable. Because different datasets have different characteristics and properties. Therefore, the optimal number of iterations may vary. However, the optimal iteration range is similar across datasets. For example, we found that the best performance is achieved around the 7th iteration in MS 300K, while it is around the 6th iteration in NQ (Figure 4). Therefore, for computational efficiency, when generalizing to other datasets, one can initially choose the number of iterations within a similar range.

5.5 Impact of Noisy Documents

To analyze the impact of different prompts for generating noisy documents, we remove noisy documents generated using a certain type of prompt during pre-training to train BootRet-Bs^{MS} and eval-

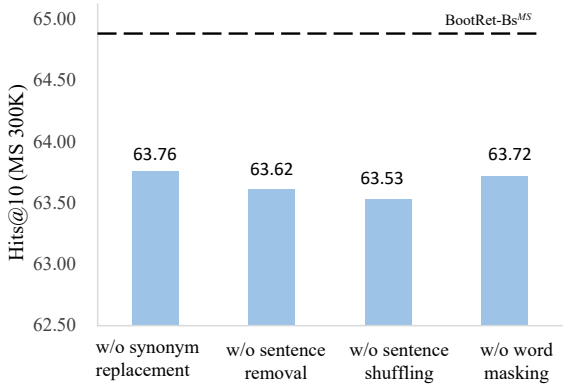

Figure 5: Retrieval performance of different prompts for generating noisy documents during pre-training.

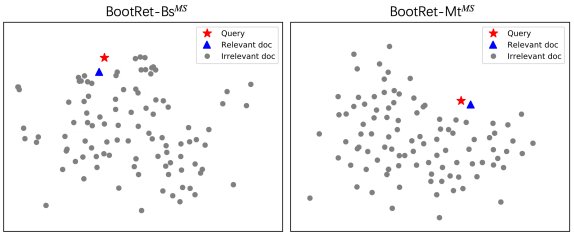

Figure 6: t-SNE plot of representations of a query (QID:1039861) from MS 300K validation set and documents corresponding to the generated top-100 docid list by BootRet-Bs^{MS} and BootRet-Mt^{MS}.

uate its retrieval performance on the downstream MS 300K dataset. Based on Figure 5, we observe the following. (i) When the noisy documents generated by the shuffling prompt are removed, the performance dropped the most, likely due to the significant semantic differences introduced by altering sentence order, reinforced by semantic consistency loss, improving discrimination ability. The sentence removal prompt shows a similar result. (ii) Next, the word masking prompt yields moderate results, possibly due to the omission of masked token prediction (as the initial T5 is already pre–trained for this task), thereby weakening the masking effect. (iii) Lastly, the synonym replacement prompt performs the most modestly, possibly because it introduces minimal semantic changes, thus having the same effect as original documents.

5.6 Visual Analysis

To further analyze the bootstrapped pre-training, we conduct visual analysis on BootRet-Bs^{MS} and BootRet-Mt^{MS} on MS 300K. We sample a query, " germany gasoline cost" (QID: 194592), from the validation set and visualize the documents corresponding to the decoded docid lists (top 100) generated by BootRet-Bs^{MS} and BootRet-Mt^{MS}. Specifically, we visualize the query and document representations encoded by the encoders of both models.

From Figure 6, we observe that compared to BootRet-Bs^{MS} (left), BootRet-Mt^{MS} (right) exhibits the relevant docid (the blue triangle) closer to the query (the red star), while irrelevant documents (the grey circles) are farther away. Additionally, we observe that irrelevant documents near the query are more clustered in BootRet-Bs^{MS} compared to BootRet-Mt^{MS}, indicating that dynamic identifiers and pre-training tasks could effectively distinguish between documents.

6 Conclusion

In this work, we proposed BootRet, a bootstrapped pre-training method for GR, addressing the mismatch between pre-defined fixed docids and evolving model parameters in existing pre-training approaches. It dynamically adjusts docids based on the model pre-trained with two tasks. Extensive experiments validate that BootRet achieves superior performance compared to strong GR baselines on downstream tasks, even in the zero-shot setting.

Acknowledgements

This work was funded by the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, the National Key Research and Development Program of China under Grants No. 2023YFA1011602 and 2021QY1701, the National Natural Science Foundation of China (NSFC) under Grants No. 62372431, the Youth Innovation Promotion Association CAS under Grants No. 2021100, the Lenovo-CAS Joint Lab Youth Scientist Project, and the project under Grants No. JCKY2022130C039. This work was also (partially) funded by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl, project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO), and the FINDHR (Fairness and Intersectional NonDiscrimination in Human Recommendation) project that received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101070212. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

7 Limitations

While BootRet has shown certain results in GR, it still has several limitations. (i) In the relevance prediction task, although we incorporate negative samples, the computational cost limits our ability to conduct comprehensive comparisons beyond batch-level contrasts. Future work could explore integrating dynamic hard negative mining techniques from traditional retrieval methods into GR. (ii) We design prompts with minimal hyperparameters to generate noisy documents. Future research could explore corresponding hyperparameter designs, such as determining the extent of sentence shuffling/removal strategies. (iii) Compared to other GR pre-training methods, our pre-training incurs slightly higher computational costs due to the need to update docids at each iteration. In future work, we can further explore how to trade off iteration costs and performance. (iv) For handling incremental documents, we ignore this issue in this work. For future work, inspired by (Chen et al., 2023), we could adaptively adjust cluster centers in the docid generation process based on the similarity between new and old documents. (v) Scalability is a significant challenge in current GR, requiring targeted solutions. Currently, a few works (Pradeep et al., 2023; Zeng et al., 2023) are exploring this issue. Differently, our work focuses on pre-training for GR which can provide suitable base model for GR. Therefore, the size of our experimental datasets follows that of most current GR works (Li et al., 2023; Sun et al., 2023; Tay et al., 2022; Wang et al., 2022, 2023; Zhou et al., 2022). We leave the scalability issue in the future. (vi) Jin et al. (2023) is concurrent work with ours, proposing to conduct language model indexer pretraining and Docid learning jointly. We do not consider this in the present study.

References

John R. Anderson and Gordon H. Bower. 2014. *Human Associative Memory*. Psychology press.

Anserini. 2020. Anserini. https://github.com/castorini/anserini.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. In *Advances in Neural Information Processing Systems*, pages 31668–31683.

Tuhin Chakrabarty, Tariq Alhindi, and Smaranda Muresan. 2018. Robust document retrieval and individual evidence modeling for fact extraction and verification. In *Proceedings of the First Workshop on Fact Extraction and VERification*, pages 127–131.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Continual learning for generative retrieval over dynamic corpora. In *Proceedings of the 32nd ACM Conference on Information and Knowledge Management*, pages 306–315.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2022. Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 191–200.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

S.S. Deepika and T.V. Geetha. 2021. Pattern-based bootstrapping framework for biomedical relation extraction. *Engineering Applications of Artificial Intelligence*, 99:104130.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 2843–2853.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):744–755.

Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, Suhang Wang, Jiawei Han, and Xianfeng Tang. 2023. Language models as semantic indexers. *ArXiv*, abs/2310.07815.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, and Wu. 2020a. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020b. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.

John Kounios, Roderick W Smith, Wei Yang, Peter Bachman, and Mark D'Esposito. 2001. Cognitive association formation in human memory revealed by spatiotemporal brain imaging. *Neuron*, 29(1):297–306.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, and Ankur Parikh. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview identifiers enhanced generative retrieval. In *61st Annual Meeting of the Association for Computational Linguistics*, pages 6636–6648.

Michael L Mack, Bradley C Love, and Alison R Preston. 2016. Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proceedings of the National Academy of Sciences*, 113(46):13203–13208.

Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2023. DSI++: Updating transformer memory with new documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8198–8213.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: Making domain experts out of dilettantes. *SIGIR Forum*, 55(1):1–27.

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 *CoCo@NIPS2016*.

Ping Nie, Yuyu Zhang, Xiubo Geng, Arun Ramamurthy, Le Song, and Daxin Jiang. 2020. DC-BERT: Decoupling question and document for efficient contextual encoding. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1829–1832. ACM.

Rodrigo Nogueira and Jimmy Lin. 2019a. Doct5query. https://github.com/castorini/docTTTTTquery.

Rodrigo Nogueira and Jimmy Lin. 2019b. From doc2query to doctttttquery. An MS MARCO Passage Retrieval Task Publication. University of Waterloo.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424*.

Daniel Guzman Olivares, Lara Quijano, and Federico Liberatore. 2023. Enhancing information retrieval in fact extraction and verification. In *Proceedings of the Sixth Fact Extraction and VERification Workshop*, pages 38–48.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Ronak Pradeep, Kai Hui, Jai Gupta, Adam D Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q Tran. 2023. How does generative retrieval scale to millions of passages? In *Gen-IR@SIGIR 2023: The First Workshop on Generative Information Retrieval*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2021. T5 base. https://huggingface.co/t5-base.

Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. TOME: A two-stage approach for model-based retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 6102–6114.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at TREC-3. In *TREC*, pages 109–126.

Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, pages 1579–1585.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. 2023. Learning to tokenize for generative retrieval. In *Advances in Neural Information Processing Systems*, volume 36.

Yubao Tang, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. 2023. Recent advances in generative information retrieval. In *SIGIR-AP 2023: 1st International ACM SIGIR Conference on Information Retrieval in the Asia Pacific*, pages 294–297. ACM.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*, volume 35, pages 21831–21843.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. https://huggingface.co/huggyllama/llama-13b. Meta AI.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2022. A neural corpus indexer for document retrieval. In *Advances in Neural Information Processing Systems*, volume 35, pages 25600–25614.

Zihan Wang, Yujia Zhou, Yiteng Tu, and Zhicheng Dou. 2023. NOVO: Learnable and interpretable document identifiers for model-based IR. In *Proceedings of the 32nd ACM Conference on Information and Knowledge Management*.

Wikipedia. 2022. Data dumps. https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2.

Dan Wu, Wee Sun Lee, Nan Ye, and Hai Leong Chieu. 2009. Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 3*, pages 1523–1532. Association for Computing Machinery.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dcn+: Mixed objective and deep residual coattention for question answering. In *International Conference on Learning Representations*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Caleb Zearing. 2023. Article title generator. https://huggingface.co/czearing/article-title-generator.

Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2023. Scalable and effective generative information retrieval. *arXiv preprint arXiv:2311.09134*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2020a. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Jointly optimizing query encoder and product quantization to improve retrieval performance. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2487–2496.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020b. RepBERT: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.

Peitian Zhang, Zheng Liu, Yujia Zhou, Zhicheng Dou, and Zhao Cao. 2023. Term-sets can be strong document identifiers for auto-regressive search engines. *arXiv preprint arXiv:2305.13859*.

Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2023. Enhancing generative retrieval with reinforcement learning from relevance feedback. In *EMNLP 2023: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022. Ultron: An ultimate retriever on corpus with a model-based indexer. *arXiv preprint arXiv:2208.09257*.

Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2023. Bridging the gap between indexing and retrieval for differentiable search index with query generation. In *Gen-IR@SIGIR 2023: The First Workshop on Generative Information Retrieval*.

A Appendix

A.1 Baseline Details

The baseline methods are described as follows:

Sparse retrieval baselines: (i) BM25 (Robertson et al., 1995) is a widely used strong term-based method. We implement it based on the Anserini toolkit (Anserini); (ii) DocT5Query (Nogueira and Lin, 2019b) expands a document with pseudo–queries predicted by a fine-tuned T5 (Raffel et al., 2020) conditioned on the original document. And then we perform the BM25 retrieval.

Dense retrieval baselines : (i) DPR (Karpukhin et al., 2020a) is a BERT-based dual-encoder model using dense embeddings for texts; (ii) ANCE (Xiong et al., 2020) leverages ANN algorithm and hard negative techniques for training a dual-encoder model; and (iii) RepBERT (Zhan et al.,

2020b) is also a dual-encoder model with brute force searching.

Advanced GR baselines: (i) DSI (Tay et al., 2022) employs semantic structured numbers as docids via a hierarchical k-means clustering algorithm. (ii) GENRE (De Cao et al., 2021) uses document titles as docids. It learns the document-docid pairs. For NQ, it has unique document titles as docids. For MS 300K which might lack of titles, we use a document title generator (Zearing, 2023) to generate high-quality titles for documents. (iii) SEAL (Bevilacqua et al., 2022) uses n-grams as docids, and generates docids based on FM-index. It uses BART-large as the backbone. (iv) DSI-QG (Zhuang et al., 2023) generates pseudo-queries conditioned on the document using docT5query (Nogueira and Lin, 2019b) and pairs them with docids for training. It uses unique integer strings as identifiers. (v) NCI (Wang et al., 2022) employs semantic structured numbers as identifiers. It trains the model using pairs of pseudo-queries and docids, and designs a prefix-aware decoder. (vi) Ultron (Zhou et al., 2022) employs the product quantization code as docids. It starts with pre-training using document piece-docid pairs, followed by supervised fine-tuning with annotated queries and generated pseudo-queries on downstream tasks. (vii) Corpusbrain (Chen et al., 2022) employs unique document titles as docids for Wikipedia during pre-training. For MS MARCO, it might lack of titles; hence, we use the document title generator (Zearing, 2023) to generate titles for documents. It undergoes pre-training using pseudo-queries constructed from documents. For downstream MS 300K, we also generate document titles as docids, and then undergoes fine-tuning on downstream tasks using annotated queries. (viii) GenRet (Sun et al., 2023) introduces an autoencoder to generate identifiers for documents. This autoencoder learns to compress documents into docids and to reconstruct docids back into documents. It learns jointly with the retrieval task. (ix) NOVO (Wang et al., 2023) selects important words from the document as docids. The model is trained through supervised learning with annotated information. All GR baselines are optimized with an encoder-decoder architecture using MLE.

A.2 Additional Implementation Details

For T5-base, the hidden size is 768, the feed-forward layer size is 12, the number of self-attention heads is 12, and the number of transformer layers is 12. Decoder-only structures like the GPT (Ouyang et al., 2022) series models are left for future exploration.

BootRet and the reproduced baselines are implemented with PyTorch 1.9.0 and HuggingFace transformers 4.16.2; we re-implement DSI, and utilize open-sourced code for other baselines.

For data augmentation during fine-tuning, we leverage the pre-trained model, DocT5Query (Nogueira and Lin, 2019b) to generete pseudo-queries for documents. For MS 300K, we directly use the off-the-shelf pseudo-queries (Nogueira and Lin, 2019a). For NQ, we use the labeled queries to fine-tune DocT5Query. For each document, we generate 10 queries with the first 512 tokens of the document as input and constrain the maximum length of the generated query as 64. During training, we pair these pseudo-queries with docids corresponding to the document, and learn these pairs with standard MLE.

Method	MRR@20	Hits@10
Ultron-PQ	38.41	62.85
BootRet-Mt^{MS}	42.79	66.73
monoBERT	**46.83***	**71.88***

Table 5: Comparison between GR methods and the full-ranking baseline on MS 300K. Best results are shown in **Bold**. $*$ indicates statistically significant improvements over BootRet ($p \leq 0.05$).

A.3 Additional Comparisons

As depicted in Table 5, for evaluating current GR methods against full-ranking methods, we adopt a cross-encoder baseline, namely monoBERT (Nogueira et al., 2019). Firstly, BM25 retrieves the top 1000 candidate documents, and monoBERT subsequently ranks them. monoBERT concatenates the query and document as input, and utilizes [CLS] for relevance calculation. It is optimized with cross-entropy.

A.4 Case Study

To better explain the changes in identifiers over bootstrapping iterations, we conducted a case study. Specifically, we sampled two documents. Below are their PQs at the initial stage, after training one round (BootRt-Bs), and after training multiple rounds (BootRt-Mt).

As shown in Table 7, we found that as identifiers evolve, the PQs for semantically similar documents

Method	Memory	Latency
DocT5query	3.76 MB	5.61 ms
DPR	940.00 MB	18.35 ms
BootRet-Bs^{Wiki}	65.40 MB	8.87 ms

Table 6: Results about inference efficiency on MS 300K.

gradually become more discriminative, while still maintaining appropriate similarity. This makes the semantic hierarchy of the docid prefix tree clearer.

A.5 Inference Efficiency

Since inference efficiency is critical for practical use, we further evaluate memory costs and inference speed on MS 300K. Table 6 in the Appendix highlights BootRet's significant reduction in memory and latency compared to DPR. BootRet only needs a prefix tree for inference, resulting in 93% less memory usage than DPR's index based on dense vectors. Additionally, BootRet outperforms DPR, with latency dropping from 18.35ms to 8.87ms for a 300K corpus. While ANN methods speed up, dual encoder latency may increase with larger corpora. However, the inference speed of BootRet only depends on the prefix tree's structure.

Original index	Initial PQ	PQ obtained with BootRt-Bs	PQ obtained with BootRt-Mt
D2169186 (topic: Germany Gasoline Prices)	12-45-67-11-4-56-2-21-53-67-1-8-5-42-13-53-64-78-120-63-4-113-2-4	12-45-67-11-4-56-2-21-53-67-1-8-5-42-13-53-61-72-115-67-8-121-8-9	12-46-70-12-4-56-2-24-53-67-1-8-5-42-13-53-61-72-115-67-8-121-8-9
D3126635 (topic: Heating oil average prices in Germany)	12-45-70-11-4-56-2-21-53-22-1-8-5-42-13-53-73-78-127-56-4-113-2-4	12-47-70-11-4-56-2-21-53-22-1-8-5-45-13-53-73-78-127-56-4-110-1-2	12-52-79-9-4-56-2-21-53-22-1-8-5-45-13-53-73-78-127-56-4-110-1-2

Table 7: Two sampled documents and their corresponding initial state PQ, obtained with BootRet-Bs and BootRet-Mt respectively.