NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR

Zihan Wang
wangzihan0527@ruc.edu.cn
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China

Yujia Zhou
Yiteng Tu
zhouyujia@ruc.edu.cn
yitengtu16@gmail.com
Renmin University of China
Beijing, China

Zhicheng Dou
dou@ruc.edu.cn
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China

ABSTRACT

Model-based Information Retrieval (Model-based IR) has gained attention due to advancements in generative language models. Unlike traditional dense retrieval methods relying on dense vector representations of documents, model-based IR leverages language models to retrieve documents by generating their unique discrete identifiers (docids). This approach effectively reduces the requirements to store separate document representations in an index. Most existing model-based IR approaches utilize pre-defined static docids, i.e., these docids are fixed and are not learnable by training on the retrieval tasks. However, these docids are not specifically optimized for retrieval tasks, which makes it difficult to learn semantics and relationships between documents and achieve satisfactory retrieval performance. To address the above limitations, we propose **N**eural **O**ptimized **VO**cabularial (NOVO) docids. NOVO docids are unique n-gram sets identifying each document. They can be generated in any order to retrieve the corresponding document and can be optimized through training to better learn semantics and relationships between documents. We propose to optimize NOVO docids through query denoising modeling and retrieval tasks, allowing for optimizing both semantic and token representations for such docids. Experiments on two datasets under the normal and zero-shot settings show that NOVO exhibits strong performance in more effective and interpretable model-based IR.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Information Retrieval; Model-based IR; Large Language Models

ACM Reference Format:
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 International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614993

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614993

1 INTRODUCTION

Information Retrieval (IR) has advanced rapidly in recent years, benefiting a wide range of applications such as search engines. Traditional IR methods like TF-IDF [24] and BM25 [25] rely on term-matching but struggle with lexical mismatches [14] when dealing with differences in phrasing or wording. Semantic-based approaches [11, 18] have been introduced to mitigate the challenges posed by lexical variations. Recent developments in pre-trained language models [2, 7, 15, 22, 30] have led to a revolution in traditional IR methodologies. This innovation has extended to both the sparse retrieval [5, 10] and the dense retrieval [12, 29] methods, which excel at semantic matching between queries and documents. Nevertheless, a notable challenge persists in effectively capturing fine-grained relationships between queries and documents. Additionally, the substantial memory requirements for indexing document representations remain a concern for these approaches.

To address these challenges, model-based Information Retrieval (model-based IR) [1, 6, 27] has gained interest. Unlike traditional approaches, model-based IR uses language models to retrieve documents by directly generating their unique discrete identifiers (docids). Such a generation process facilitates fine-grained interaction between queries and docids, allowing for end-to-end optimization. Additionally, the adoption of discrete document representations results in a notable decrease in memory usage. However, designating appropriate docids for documents poses a noteworthy challenge in the current model-based IR systems. The model's capacity in retrieving documents will be hindered when docids fail to 1) effectively **convey document information,** or 2) accurately **represent document relationships** within the corpus.

Early text-based docids identify documents with a string of text, such as document titles [6] or URLs [32], as shown in Figure 1(a). By employing a prefix tree (i.e., trie) for the docids as a generation constraint, it can ensure that the language model generates an in-corpus docid. While this type of docids shows interpretability, it lacks clear connections between different docids, hampering the establishment of meaningful **semantic** relationships. On the other hand, semantic docids [27, 28, 32], as illustrated in Figure 1(b), have shown promising capability by 1) building semantic document representations, 2) clustering document representations, and 3) designating docids based on the identified clusters. By clustering document semantic representations, it empowers different docids with better semantic connections. However, semantic docids consist of unreadable category numbers, making it challenging to interpret the model's understanding of the corpus. Furthermore, both text-based and semantic docids have static **tokens** that remain fixed

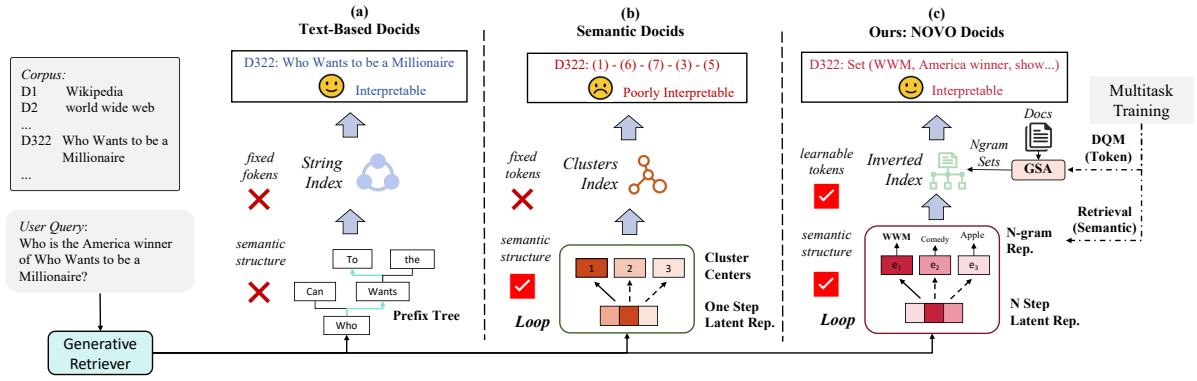

Figure 1: An overall illustration for the three types of docids. NOVO docids consist of n-grams of the documents obtained from global self-attention. A document can be retrieved by generating its NOVO in any order, constrained by an Inverted Index. To learn effective NOVO docids, we propose to train two tasks to improve the semantic and token representation of NOVO, achieving more effective optimization and interpretability.

after initialization, constraining their adaptability for retrieval tasks and potentially limiting system performance.

Our goal is to endow docids with learnability in both semantic and lexical spaces. By doing so, we aim to attain effective and interpretable model-based IR systems. Specifically, we propose **N**eural **O**ptimized **VO**cabularial (NOVO) docids, as shown in Figure 1(c). A NOVO docid is an n-gram set extracted from the corresponding document through an encoder. The encoder is trained to extract an n-gram set from documents by estimating the n-grams' confidence based on the document's global self-attention information. By doing so, high-confidence n-grams will be selected as docids. As the encoder is trained with all documents in the corpus (detailed in the next paragraph), these n-gram sets can be learned and periodically updated to better represent documents' semantic meanings and relationships. At the inference stage, the model can retrieve a document by generating the n-grams of its NOVO docid in any order. The inference process is constrained by a small inverted index to ensure that a valid (i.e., in-corpus) docid is generated. With learnability in tokenization, NOVO docids have much more expressive power than existing docids. In addition, since NOVO docids are composed of natural language tokens, it brings interpretability to the indexing and retrieval process.

In order to learn effective NOVO docids, we propose a denoising query modeling (DQM) task. The objective of the task is to learn to denoise documents and generate queries. Our intuition is that n-grams relevant to the queries will be highlighted after training, with a high-confidence global self-attention score. These n-grams can be seen as reasonable docids, and they are periodically assigned to documents for the retrieval task, helping to learn effective **tokens** (i.e. n-grams) of docids. On the other hand, the retrieval task can improve the **semantic** representation (i.e., embeddings) of docids, as the token embeddings are updated during the training process of the retrieval tasks. This dual optimization can be performed iteratively by multi-task training for the denoising query modeling and retrieval tasks. To avoid the distribution mismatch problems [33], we perform this by leveraging an encoder-decoder for the denoising query modeling task, and the decoder only for the retrieval task.

We conduct extensive experiments on two widely used datasets: MS MARCO and Natural Questions. Results show that NOVO can consistently outperform the baselines for both normal and zero-shot evaluation. In addition, we perform case studies to show the decent interpretability of NOVO docids for model-based IR: explaining how the model understands the documents by n-gram extraction in the indexing process, and how it understands the queries by n-gram generation in the retrieval process.

2 RELATED WORK

2.1 Traditional IR

Traditional IR methods can be categorized into two groups: sparse retrieval and dense retrieval. Early sparse retrieval methods like BM25 [25] and TF-IDF [24] focus on the exact lexical matching between document terms and query terms. These methods build an inverted index and calculate document relevance with term-based features like term frequencies. Recently, pre-trained language models (PLM) [7, 23] are used to improve the performance of sparse retrieval by estimating term weights [5], learning sparse representation [9, 10], and generating explicit document expansion [21].

On the other hand, dense retrieval approaches like DPR [12] and ANCE [29] project queries and documents into dense vectors and calculate relevance scores based on inner-product or cosine-similarity. These methods focus more on semantic similarity instead of term matching, eliminating the lexical mismatch problems [14]. However, a set of hard negative data must be sampled for contrastive learning, which leads to computational inefficiency. Besides, given the reliance on vector-based interactions, dense retrieval has challenges in maintaining an efficient memory footprint and accurately capturing fine-grained query-document connections.

2.2 Model-based IR

Firstly proposed by Metzler et al. [17], model-based IR approaches have emerged as promising alternatives to traditional retrieval methods. In model-based IR, each document is assigned a unique discrete identifier (docid), and retrieval is performed by directly

generating the docid with a language model. Recent studies have explored various aspects of model-based IR. Several works have demonstrated the effectiveness of model-based IR in fact verification [3], entity linking and disambiguation [6], and document retrieval [6, 27, 28, 31]. Some works discussed different learning objectives for language models. For example, De Cao et al. [6] present to generate docids directly when given a query. Tay et al. [27] propose to generate docids given document content and queries, respectively, to align the representation of them. Techniques for better optimization have also been investigated, including query generation [28, 33], pre-training [4, 32], and continual learning [16]. Furthermore, Bevilacqua et al. [1] propose to generate substrings and match them with the content of documents, presenting the potential for model-based IR even in the absence of unique docids.

Model-based IR utilizes various **docids** to represent the semantics and relationships of the documents. We review two important forms of them: text-based and semantic. Text-based docids compose of strings, such as title [3, 6], URL [32], or random numbers [27]. While they are mostly interpretable, there is no clear semantic relationship between different docids, which leads to ineffective document organization and a lack of generalization ability for *unseen* documents [27]. On the other hand, semantic docids are proposed, firstly in [27]. Semantic docids are obtained by first deriving documents' dense representations using a language model like BERT [7], and then getting clustering results from techniques such as Hierarchical Clustering [27, 28], Product Quantization [32] or Diverse Clustering [26]. Despite the benefits of refined semantic representations, such docids primarily comprise unreadable category numbers, posing challenges in interpreting the document retrieval process. Moreover, both text-based and semantic docids remain static after initialization, limiting their ability to acquire and adjust document semantics and relationships throughout the training process. In contrast to previous methods, we introduce NOVO docids that consist of unique n-gram sets. We propose to enhance their semantic and token representation through denoising query modeling and retrieval tasks, achieving effective and interpretable model-based IR systems.

3 METHODOLOGY

In this section, we first provide a definition of model-based IR. We then outline the desired property of a docid: dual learnability, that is, the capability of optimizing both semantic and lexical representations during training. Docids with dual learnability can be optimized to capture the semantics of documents and learn the relationships between them, thereby enhancing retrieval performance.

Next, we delve into the reasons behind our choice in using n-gram sets as docids, and demonstrate its dual learnability. We further detail the approach we employ to optimize NOVO docids with the denoising query modeling as well as the retrieval tasks, and introduce the training and inference processes.

3.1 Preliminaries

3.1.1 ***Task Formulation.*** In this paper, we formulate moel-based IR as retrieving a document d from a corpus C for a query q by generating its docid $o_d = (e_1, \ldots, e_{|o_d|})$ with a language model M. The modeling process is formulated as:

$$p_\theta(d|q) = M_\theta(o_d|q), \tag{1}$$

$$= \prod_i M_\theta(e_i|q, e\ddot{y}_i), \tag{2}$$

where e_i denotes the i^{th} element (token or n-gram) of the docid, $e\ddot{y}_i$ denotes generated elements of the docid, and θ denotes the learnable parameters of M. The docid o_d is obtained from:

$$o_d = \mathrm{AF}_\phi(d), \tag{3}$$

where AF is an assignment function that designates a docid for a document and ϕ denotes the learnable parameters of AF. In some cases, the corpus is also an input variable of AF, where $o_d = \mathrm{AF}_\phi(d, C)$, for better inter-document relationship learning. The challenge of designing appropriate docids lies in how to correctly represent d and reflect the relationships of documents in C.

3.1.2 ***Dual Learnability.*** In model-based IR, if the docids are expected to be trained for retrieval tasks, there are two main properties needed to be satisfied: semantic-space (SS) learnability and tokenization-space (TS) learnability. SS-learnable docids exist clear relationships in a semantic space. To be more specific, the *distance* of two docids can be defined. As shown in Figure (1), text-based docids are not SS-learnable because there is not a well-defined semantic distance between them. Meanwhile, semantic docids are SS-learnable because the semantic distance between two docids can be calculated by approaches such as the sum of L2-norm of the numerical tokens that formed them. TS-learnability means the tokens of a docid $o_d = \mathrm{AF}(d)$ are learnable (i.e., AF is a learnable function). However, current semantic docids are not TS-learnable, because the tokens of such docids only rely on initialization, being static afterward. Our goal is to construct docids that are both SS-learnable and TS-learnable. Such docids can be optimized for specific tasks, leading to more effective model-based IR systems.

3.2 Our Approach: NOVO docid

In this section, we will provide the definition of our docid, its dual learnability properties, how to learn effective docids through the denoising query modeling and retrieval task, and how to conduct training and inference. Figure 1 (c) illustrates the construction of the NOVO docid, and Figure 2 shows the model.

3.2.1 ***Definition.*** A NOVO docid is a set of n-grams from documents. The n-grams of a NOVO docid can be generated in an arbitrary order to retrieve the corresponding document. For example, a document with a NOVO docid $set(A, B, C)$ can be retrieved once the model generates BAC, ACB, or any other order of $set(A, B, C)$. By extracting n-grams from documents rather than the whole vocabulary list, our approach ensures that relevant document information is encapsulated within docids while avoiding *exhaustive computations* in the vocabulary list (V^N possibilities, where V denotes vocabulary size and N denotes n-gram length). Furthermore, we adopt a set-based approach for NOVO docids rather than fixed sequences, which enhances retrieval outcomes by enabling arrangement of generated n-grams' relevance in descending order, facilitating document filtering based on query relevance.

The retrieval process of NOVO docids is to iteratively shrink the candidate document list as the model generates n-grams one by one.

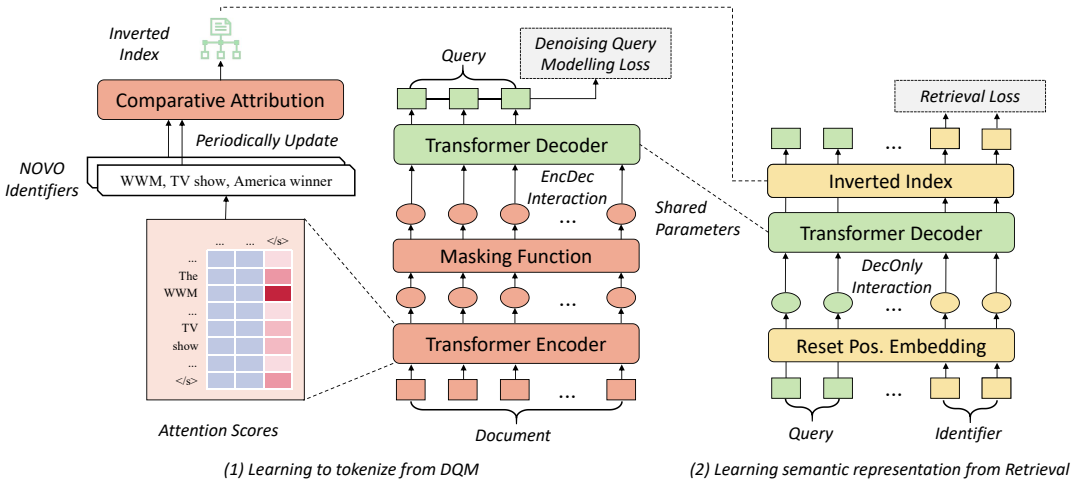

Figure 2: An overall illustration of our approach. The vectors are represented by ellipses and the tokens are represented by rectangles. The encoder learns through the denoising query modeling task to filter n-grams that are relevant to the query from global attention scores. These n-grams are periodically updated as NOVO docids. Meanwhile, we optimize the docids' semantic representation through retrieval tasks and jointly train our system on the two tasks.

Denote all the docids in the corpus as $O_C = (o_{d_1}, o_{d_2}, \ldots, o_{d_{|C|}})$ and all n-grams in O_C as $E(O_C) = (e_1, e_2, \ldots, e_m)$. We use $E(O_C)$ to build an inverted index $I_0 = I(E(O_C), O_C)$, and query it with a query element e_q to obtain a retrieved subset of corpus $C_1 = Q(I_0, e_q)$. We set $e_q \in E(O_C) \setminus e_{\text{gen}}$, where e_{gen} denotes generated elements (now an empty set). The overall process can be summarized as follows:

$$C_1 = Q(I(E_{O_C}, O_C), e_q). \tag{4}$$

Considering E_{O_C} and O_C only depend on C and AF, the above equation can be simplified as:

$$C_1 = Q(C, \text{AF}, e_q). \tag{5}$$

Let $e_{\text{gen}} := e_q \oplus e_{\text{gen}}$, the process can be iterated until C_i only has one document or a manually designed stop element e_s is generated. We can use Eq. (2) to derive the probability $p(d|q)$ of the retrieved document from a language model. For multi-document retrieval, since $\log p_\theta(d|q) = \sum_{e_i} \log M_\theta(e_i|q, e_{\text{gen}})$ is a continuous addition, a beam search constrained on the inverted index can be leveraged, where the beam score function is $\log M_\theta(e_{\text{beam}}|q)$.

3.2.2 ***Properties.*** NOVO docid is dual-learnable, and this enhances its capability to learn document representations and relationships. Specifically, we define the distances between two docids to achieve SS-learnability, and define how to obtain docids through a learnable assignment function to achieve TS-learnability.

To achieve SS-learnability, we can obtain the distance l between two docids o_1 and o_2 as follows:

$$\mathbf{h}_{ij} = \text{MP}(\text{Embedding}(e_{ij})), \tag{6}$$

$$\mathbf{q}_i = \text{MP}(\text{Att}(\mathbf{h}_{i1}, \mathbf{h}_{i2}, ..., \mathbf{h}_{i|o_i|})) \quad (i = 1, 2), \tag{7}$$

$$l(o_1, o_2) = \exp(-\mathbf{q}_1 \cdot \mathbf{q}_2), \tag{8}$$

where $\mathbf{h}_{ij}$ is the hidden state of e_{ij}, MP is the max-pooling operation at the dimension of tokens and Att is an attention layer. Due to the need to introduce additional parameters for Att, we do not leverage this distance directly in this work, but it demonstrates the *learnability* of document relationships for NOVO and provides inspiration for possible future learning methodologies based on docid semantic distance.

To achieve TS learnability, we find an assignment function AF with learnable parameters ϕ. Specifically, we set:

$$\text{AF}_\phi(d) = C(\arg_k \text{GSAScore}(d_k; d, \text{Encode}_\phi) | |d| > \tau) \tag{9}$$

where d_k is the k^{th} token of d, $|d|$ is the document length, Encode_ϕ is the document encoder, GSAScore is a global self-attention score that calculates the attention of the document global token to another token, τ is a confidence threshold hyperparameter. $\arg_k$ gets all the tokens that satisfy the condition $\text{GSAScore} * |d| > \tau$, and C is a function to merge these consecutive tokens to set up an n-gram set. Specifically, let d_g be the global token, we have:

$$\text{GSAScore}(d_k; d, \text{Encode}_\phi) = \frac{\exp(\mathbf{Q}_{d_g}^{L\phi} \cdot \mathbf{K}_{d_k}^{L\phi} / \sqrt{dim})}{Z}, \tag{10}$$

where L is the number of layers of the encoder, $\mathbf{Q}_{d_g}^{L\phi}$ is the last-layer query vector of d_g, $\mathbf{K}_{d_k}^{L\phi}$ is the last-layer key vector of d_k, dim is dimension of vectors, and Z is a normalization value that

$$Z = \sum_{i=1}^{|d|} \exp(\mathbf{Q}_{d_g}^{L\phi} \cdot \mathbf{K}_{d_i}^{L\phi} / \sqrt{dim}). \tag{11}$$

The score can be maximized or averaged across heads for multi-head attention. To avoid assigning identical docids to different documents, we use a Comparative Assignment approach for docid generation. Specifically, we initialize the docid with only the n-gram of the highest GSAScore (score is averaged for n>1) of each document. While multiple documents share the same docid, we

select their remaining n-gram of the highest GSAScore, adding to their respective docids. This comparative assignment approach can ensure that multiple documents don't share the same docid while maintaining assignment efficiency.

3.2.3 ***Denoising Query Modeling and Retrieval Tasks.*** NOVO docids are learned on the denoising query modeling (DQM) task and the retrieval task. In this section, we demonstrate how to optimize NOVO docids in both semantic and lexical spaces with the two tasks (as shown in Figure 2).

To learn in the tokenization space, we designed the denoising query modeling task (left side of Figure 2). By learning to generate queries with noised documents, the model can implicitly learn to filter out document n-grams that are more likely to be relevant to the queries. Queries and documents are accessible from IR datasets, and the obtained n-grams can serve as effective docids. The denoising process facilitates filtering out unimportant parts of the document. Specifically, the denoising query modeling task is represented as:

$$q \rightarrow \text{Decode}_\theta \quad T \quad \text{Encode}_\phi(d) \quad , (s) \quad , \tag{12}$$

where d is a document (i.e., encoder input), q is the query to fit (i.e., target), Encode_ϕ is the encoder, Decode_θ is the decoder, s is a fixed starting token of the decoder (i.e., decoder input), and T represents the mask function for denoising modeling:

$$T\ (\mathsf{h}_i) = \begin{cases} \mathsf{h}_i, & \text{if } i \text{ is global token,} \\ \mathsf{h}_i \sqrt{w_1}, & \text{otherwise.} \end{cases} \tag{13}$$

The decoding process is basically the same as Transformer. To achieve denoising, the cross attention layer is represented as:

$$\text{Attention}(\mathsf{Q}, \mathsf{K}, \mathsf{V}) = \text{softmax} \quad \frac{\mathsf{Q} \cdot \mathsf{K}}{\sqrt{dim}} \quad \mathsf{V} \cdot w_2, \tag{14}$$

where Q comes from the decoder, and K and V come from the encoder. The two mask hyperparameters $w_1, w_2 \in [0, 1]$ raise/drop respectively during training to gradually raise the difficulty of denoising query modeling, forcing cross attention to focus on global tokens and decreasing the effect of the whole cross attention layer. The final loss function for the task is represented as:

$$L_{DQM} = \tilde{\sum_i} -\log p_{q_i} \quad \text{Decode}_\theta^1 \quad T \quad \text{Encode}_\phi(d) \quad , (s, q_{\ddot{Y}i}) \quad , \tag{15}$$

where Decode_θ^1 denotes one-step decoding, q_i is the i^{th} token of q, $q_{\ddot{Y}i}$ is the prefix of q up to the i^{th} token, and p_{q_i} is the probability to generate q_i derived from the language model.

To learn in the semantic space, we propose to leverage supervision from retrieval tasks (right side of Figure 2) to update docid semantics. Specifically, the retrieval task is represented as:

$$o \rightarrow \text{Decode}_\theta^{\text{RPE}}\ (None, (s, q)), \tag{16}$$

and each step is represented as:

$$e_i \rightarrow \text{Decode}_\theta^{\text{RPE}}\ (None, (s, q, e_{\ddot{Y}i})), \tag{17}$$

where $\text{Decode}_\theta^{\text{RPE}}$ means decoding with a **R**eset of **P**osition **E**ncoding: the first token of each n-gram is reset to zero. For example, for $|q| = 2, |e_1| = 1, |e_2| = 3$, The positional encoding with $\text{Decode}^{\text{RPE}}$ for the sequence qe_1e_2 is $[0, 1, 0(\text{reset}), 0(\text{reset}), 1, 2]$. RPE ensures that shuffling the order of generated docid elements does not affect the generation of the current element, i.e., $M_\theta(e_i|q, e_{\ddot{Y}i}) = M_\theta(e_i|q, \text{shuffle}(e_{\ddot{Y}i}))$. The retrieval loss is represented as:

$$L_R = \tilde{\sum_i} -\log p_{e_i} \quad \text{Decode}_\theta^{\text{RPE}}(None, (s, q, e_{\ddot{Y}i})) \quad , \tag{18}$$

where p_{e_i} is the i^{th} element of o and p_{e_i} is the probability to generate e_i derived from the language model.

To summarize for the learning process, the assignment function (i.e., docid tokens) is updated through the denoising query modeling task, and the embeddings (i.e., docid semantics) are updated through the retrieval task. Two tasks work jointly towards a dual optimization of the NOVO docids. Furthermore, we leverage mask hyperparameters to train a denoised encoder, and leverage RPE to reduce the complexity of document retrieval learning by generating unordered docids in the auto-regressive decoder models.

3.2.4 ***Training and Inference.*** In the training stage, we optimize NOVO docids' semantic and tokenization representation alternatively. Specifically, we share the embedding parameters between Encoder and Decoder and follow these steps:

1) Obtain a docid o_d for each document d in the corpus through the current Encoder to form O_C;

2) Derive an inverted index from O_C. Then, conduct multi-task training of the denoising query modeling task and the retrieval task by mixing their samples, in order to learn docid semantic representation with a fixed token representation, such that

$$L = \lambda L_{DQM} + L_R; \tag{19}$$

3) After the training loss converges, update O_C with the updated $Encoder_\phi$, leading to a better token representation for the docids;

4) Repeat 2) and 3) until O_C stopped changing, where it is believed to reach the optimized token representation.

In the inference stage, we cache the first-step inverted index I_0 because it is invariant for any query, and then retrieves the document following the process shown in Section 3.2.1

4 EXPERIMENT SETUP

4.1 Datasets and Evaluation Metrics

4.1.1 ***Datasets.*** We performed training and testing using two widely recognized datasets for model-based IR: Natural Questions [13] and MS MARCO [19]. Natural Questions, curated by Google, contains 320k query-document pairs from Wikipedia, featuring naturally posed questions. MS MARCO from Microsoft Bing comprises 300k query-document pairs. The queries are real Bing questions and the documents are web pages. We selected the Document Ranking subtask of this dataset for training and testing. For both datasets, we follow the identical data processing approach as outlined in NCI [28]. We eliminate duplicate documents by comparing their titles and proceed to randomly choose a validation set, comprising 500 samples from the training set. Table 1 summarizes the statistics of the above datasets.

4.1.2 ***Evaluation Metrics.*** Building upon prior research [26, 28], we employ a set of five widely recognized metrics, Recall@1, Recall@10, Recall@100, and Mean Reciprocal Rank (MRR), to conduct a comprehensive assessment of retrieval performance. Recall@n

Dataset	# Docs	# Test Queries	# Train Queries
Natural Questions	109,739	7,830	307,373
MS MARCO	323,569	5,187	366,235

Table 1: Statistics of the used document retrieval datasets.

quantifies the frequency with which the desired document is successfully retrieved among the top-n candidates. Meanwhile, MRR determines the ordinal position at which the first relevant document is retrieved. For our evaluation on the Natural Questions dataset, we adopt Recall@{1, 10, 100} and MRR@100 as evaluation standards, in alignment with the methodology proposed by Wang et al. [28]. On the MS MARCO dataset, we adhere to the evaluation criteria outlined by Zhou et al. [32], employing Recall@{1, 10, 100} and MRR@10 as our metrics. Following Zhou et al. [32], We conduct paired t-tests with p<0.05.

4.2 Baselines

We compare our approach against two categories of baseline methods: traditional IR methods and model-based IR methods.

4.2.1 ***Traditional IR Methods.*** Within the category of traditional IR methods, we consider several prominent models commonly employed for Document Retrieval, encompassing both sparse retrieval and dense retrieval techniques.

- BM25 [25] stands as a highly effective retrieval model, representative of the classical probabilistic retrieval approach.
- SPLADE [9, 10] is a sparse retrieval model capable of encoding a text sequence into a sparse lexical representation, leveraging a BERT-based encoder.
- ANCE [29] presents a BERT-based dense retriever trained using dynamic global hard negatives.
- DPR [12], a BERT-based dual-encoder model, is trained with in-batch negatives and hard negatives selected via BM25.
- Sentence-T5 [20], a T5-based dual-encoder model, is trained using in-batch negatives.

4.2.2 ***Model-Based Retrieval Methods.*** Furthermore, we consider several advanced model-based retrieval baselines.

- GENRE [6] retrieves documents by generating their titles. It leverages a BART trained on BLINK and KILT datasets.
- DSI [27] leverages three types of docids: hierarchical cluster centers, random numbers, and unique single-token docids, which are referred to as semantic, naive, and atomic docids.
- SEAL [1] generates a series of n-grams given a query and retrieves documents by calculating the matching score between the generated n-grams and the n-grams of documents in the corpus based on FM-index.
- Ultron [32] adopts a three-stage training pipeline, utilizing document URLs and PQ cluster centers as docids to improve performance.
- NCI [28] employs a prefix-aware weight-adaptive (PAWA) decoder alongside diverse query generation strategies.
- Genret [26] leverages diverse constrained cluster centers as docids and trains the docids' representations through document tokenization, retrieval, and reconstruction tasks.

4.3 Implementation Details

Model Architecture. We implement NOVO based on the transformer encoder-decoder T5-base [23], where the hidden size is 768, the feed-forward layer size is 3072, the number of transformer layers is 6, and the number of self-attention heads is 12, for both the encoder and decoder.

Docid Settings. We set the confidence threshold in GSAScore τ to be 3, maximize GSAScore across heads for multi-head attention, and set the maximum size of docid set N to be 20.

Training Hyperparameters. Experiments are conducted using four Nvidia-Titan V GPUs. Specifically, we employ a total batch size of 64, and an Adam optimizer whose learning rate is set to 2e-5. The proportion between denoising query modeling and the retrieval task is set at 1:1. The loss weight λ is established as 0.1. During training, a linear transition is applied to the mask parameters: w_1 transitioned from 1 to 0.5, and w_2 transitioned from 0 to 0.5. The reference sequence of docids for the retrieval task is randomized. Query and document lengths are truncated to 64 and 512, respectively. An upper limit of $5x10^5$ optimization steps is set, and corpus reindexing are conducted every $5x10^4$ steps.

Inference Hyperparameters. We use constrained beam search with a beam size of 100 and implemented the beam-scorer mentioned in Section 3.2.1 for document ranking.

5 EXPERIMENTAL RESULTS AND ANALYSIS

We conduct experiments targeting the following research questions:

- **RQ1:** In both normal and zero-shot settings, how does the performance of NOVO compare with that of strong retrieval baselines?
- **RQ2:** To what extent does the performance of NOVO get influenced by the characteristics of dual learnability?
- **RQ3:** How does the nature of the *set* docid impact the effectiveness of NOVO?
- **RQ4:** What impact do our training and evaluation settings have on the effectiveness of NOVO?
- **RQ5:** How does the efficiency in terms of memory and time of NOVO compare with traditional baselines?
- **RQ6:** Can the interpretability of NOVO be better understood through the analysis of specific case studies?

5.1 Main Result

To answer **RQ1**, we conduct a comprehensive comparison between NOVO and several baselines using the Natural Questions and MS MARCO datasets. For normal evaluation, we adopt the same setting as Wang et al. [28]. We assess NOVO's zero-shot performance on Natural Questions, simulating real applications like intelligent search engines. We train the model using in-domain documents, including updated news, and later augment the corpus with test set documents for evaluation. Performance results for normal setting are summarized in Table2, while Table 3 details the zero-shot performance.

For normal evaluation, NOVO consistently perform better than other baselines on two datasets, underscoring its remarkable efficacy in document retrieval tasks. Especially for MRR@{10, 100}, We observed relative performance gains of 1.9% and 1.1% in the MS MARCO and Natural Questions datasets, respectively, demonstrating the powerful ability of NOVO. The superior effectiveness

Table 2: Results on MS MARCO (MS300K) and Natural Questions (NQ320K). Results from methods denoted with †reflect our independent reimplementation, while others are from their official implementations and Sun et al. [26]. Instances highlighted with * signify noteworthy enhancements over the leading baselines with a p-value of < 0.05. The most exceptional outcomes for each metric are highlighted boldly.

			MS MARCO				Natural Questions			
Type	Model	docid	R@1	R@10	R@100	MRR@10	R@1	R@10	R@100	MRR@100
Sparse Retrieval	BM25	Term Weights	39.1	69.1	86.2	48.6	29.7	60.3	82.1	40.2
	SPLADE†	Term Weights	45.3	74.7	90.5	54.6	50.5	77.8	92.3	61.4
Dense Retrieval	ANCE	Dense Vector	45.6	75.7	89.6	55.6	50.2	78.5	91.4	60.2
	DPR	Dense Vector	-	-	-	-	50.2	77.7	90.9	59.9
	Sentence-T5	Dense Vector	41.8	75.4	91.2	52.8	53.6	83.0	93.8	64.1
Model-based Retrieval	GENRE	Text-Based	35.6	57.6	79.1	42.3	55.2	67.3	75.4	59.9
	$DSI_{semantic}$†	Category Nums	25.7	43.6	53.8	33.9	38.2	55.3	65.3	44.2
	DSI_{naive}	Text-Based	-	-	-	-	6.7	21.0	-	-
	DSI_{atomic}†	Unique docid	32.4	63.0	69.9	44.3	49.4	65.2	76.1	55.2
	SEAL	All n-grams	25.9	68.6	87.9	40.2	59.9	81.2	90.9	67.7
	NCI	Category Nums	30.1	64.3	85.1	41.7	65.9	85.2	92.4	73.1
	$Ultron_{PQ}$†	Category Nums	31.6	64.0	73.1	45.4	52.0	70.1	80.6	58.7
	$Ultron_{URL}$†	Text-Based	29.6	56.4	67.8	40.0	61.2	77.4	85.9	67,5
	Genret	Category Nums	47.9	79.8	91.6	58.1	68.1	88.8	95.2	75.9
	NOVO (Ours)	n-gram Set	**49.1**	**80.8**	**92.5***	**59.2**	**69.3**	**89.7***	**95.9***	**76.7**

of NOVO can be attributed to two aspects: (i) NOVO can optimize the semantic representation of docids through the retrieval task; (2) NOVO also supports optimizing the tokenization of docids through the denoising query modeling task, learning effective tokens for docids to convey document representations and document relationships.

For the zero-shot evaluation, NOVO consistently outperforms all other model-based IR methods in all four metrics. This proves that NOVO docids can be generalized to documents that the model has not seen at training with excellent performance. Specifically, NOVO significantly outperforms DSI_{atomic}, DSI_{naive} methods, which require inputting documents into the retrieval model to memorize them and cannot obtain the generalization capability to new documents. NOVO is also superior to methods that require pseudo-query generation, such as NCI. These methods only use queries generated through documents to enhance retrieval performance. However, they do not directly involve the documents themselves in the modeling process, leading to a loss of information. We will analyze how the model assigns docids to unseen documents in Section 5.4.

5.2 Ablation Studies

To answer **RQ2**, **RQ3**, and **RQ4**, We conduct ablation studies in terms of dual-learnability, properties of the n-gram set, and our training and evaluation settings, respectively.

For **RQ2**, we target on evaluating model effectiveness when its dual-learnability properties are disabled. To disable SS-learnability, we freeze the embedding table of docid tokens, which will fix the semantic space. To disable TS-learnability, we freeze the tokens of all docids after initialization. For **RQ3**, we target on evaluating our model when its properties obtained from its set structure are disabled. We first use a fixed, random order for retrieval task training and inference to fully disable its set properties. Then, we conducted experiments that disable the reset-of-position-encoding technique to partially disable the set properties. For **RQ4**, we replace the denoising query modeling task with a normal query generation task (i.e., set $w_1 = 1$ and $w_2 = 0$), and disable comparative assignment and search constraint with an inverted index, respectively, and analyze how these techniques affect model performance.

Table 4 shows the result for all our ablation studies. We notice a consistent performance ranking across both datasets, indicating that altering any configuration of the current model would result in a decline in performance. These findings validate the necessity of our approach to achieve strong model-based IR systems.

5.3 Efficiency Analysis

To address **RQ5**, we conduct a comparison between NOVO and typical traditional model-based IR approaches on Natural Questions, considering factors such as model parameters, index memory footprint, and inference time. The baselines utilized different indexing methods, such as inverted index, dense vector-base, prefix tree, and FM-index [8]. To evaluate inference time for model-based IR methods, we randomly sample 256 queries from the Natural Questions dataset and inference with a beam size of 100 and record the time cost of different methods.

Table 5 demonstrates the evaluation results of model efficiency. We observe that most model-based methods significantly reduce the index size while maintaining a similar inference time compared to sparse and dense retrieval methods, except SEAL [1], which recorded the entire document n-grams into an FM index, leading to much more index memory and inference cost. The results demonstrate the potential of model-based IR methods to achieve efficient IR systems in terms of indexing and inference efficiency.

Table 3: Zero-shot Performance Evaluation on Natural Questions (NQ320K). Results from methods denoted with †reflect our independent reimplementation, while others are from their official implementations and Sun et al. [26]. Instances highlighted with * signify noteworthy enhancements over the leading baselines with a p-value of < 0.05. The most exceptional outcomes for each metric are highlighted boldly.

Category	Model	docid	R@1	R@10	R@100	MRR@100
Unsupervised Retrieval	BM25	Term Weights	21.8	57.4	78.3	34.3
Model-based Retrieval	GENRE	Text-Based	6.0	10.4	23.4	7.8
	$\text{DSI}_{semantic}$	Category Nums	1.3	7.2	31.5	3.5
	DSI_{naive}	Text-Based	0.0	0.0	0.2	0.1
	DSI_{atomic}†	Unique Identifier	0.0	0.0	0.1	0.0
	NCI	Category Nums	15.5	-	-	-
	Ultron_{PQ}†	Category Nums	2.8	12.7	42.1	6.2
	Ultron_{URL}†	Text-Based	32.5	50.4	64.3	39.1
	Genret	Category Nums	34.1	-	-	-
	NOVO (Ours)	n-gram Set	**37.0***	**65.7***	**80.6***	**47.6***

Table 4: Performance comparisons on several variants of our approach. We evaluate the impact of dual-learnability, set properties, and training/inference settings on the model performance.

	MS MARCO				Natural Questions			
Model	R@1	R@10	R@100	MRR@10	R@1	R@10	R@100	MRR@100
NOVO (Ours)	**49.1**	**80.8**	**92.5**	**59.2**	**69.3**	**89.8**	**95.9**	**76.7**
w/o SS-learnability	46.7	77.0	88.1	56.4	66.0	85.4	91.4	73.1
w/o TS-learnability	46.1	76.2	87.2	55.8	65.3	84.5	90.4	72.3
w/o arbitrary order	47.3	77.6	88.8	56.9	66.5	86.1	92.1	73.7
w/o reset positional encoding	48.2	79.6	91.1	58.3	68.2	88.3	94.4	75.5
w/o denoising query modeling	46.5	77.6	88.9	56.9	66.6	86.2	92.1	73.7
w/o comparative assignment	48.7	80.2	91.8	58.7	68.8	89.0	95.1	76.1
w/o inverted index	48.5	79.6	91.1	58.3	68.3	88.4	94.5	75.6

Table 5: Efficiency analysis for different methods. We run BM25 on CPU and other methods on GPU. For Indexing methods, II denotes inverted index, DV denotes dense vector base, PT denotes prefix tree and FM denotes FM index.

Model	Index	Model Params	Index Size	Infer. Time
BM25	II	0M	448MB	1.0s
DPR	DV	220M	330MB	6.0s
GENRE	PT	406M	27MB	6.5s
DSI	PT	250M	12MB	5.7s
SEAL	FM	406M	210MB	58.1s
Ours	II	250M	25MB	7.2s

5.4 Case Studies

To answer **RQ6**, we mainly focus on two aspects: 1) How to interpret the model's understanding to extract n-grams from the document; 2) How to interpret the process by that NOVO docids are retrieved.

5.4.1 ***Understanding Document N-grams.*** To interpret how the model understands document n-grams, we pick two documents from the training-phase corpus and unseen corpus, respectively. We compare the high-confidence n-grams filtered before training, after corpus re-indexing once, and after corpus re-indexing 5 times. The results are shown in Figure 3, represented by red, blue, and green colors, respectively. The first document "Cholera" is from MS MARCO training set and the second document "Gun Interest Profile .." is an unseen document from MS MARCO dev set.

We discover that as training goes on, the model learns to filter out the n-grams relevant to the retrieval task for both seen documents and unseen documents. For the first document, the initial n-grams are "poverty", "investine", "ium", which make no sense. The selected n-grams catch better semantics after re-indexing for once, but there still exist mistakes. For example, although "Cholera (food)" is lexically relevant to the document's topic "Cholera", it's semantically irrelevant to the main idea of the document. However, after re-indexing 5 times, the selected n-grams "Cholera", "watery diarrhea", "clean water" can convey the semantics of the document, and are very relevant to the queries that asked about the causes of the Cholera disease. For the second document, although our model keeps failing to capture the important word "National Rifle Association", which is the main topic of the document, it still chooses "NRA", the abbreviation of the association as an important n-gram, which helps document retrieval since the queries all used the abbreviation rather than the full name.

Title	Document	Example Queries
Cholera	... (*content omitted*) This article is about the bacterial disease. For the dish, see Cholera (food). Cholera A person with severe dehydration due to cholera causing sunken eyes and wrinkled hands and skin. Specialty Infectious disease Symptoms Large amounts of watery diarrhea, vomiting, muscle cramps [1] [2]Complications Dehydration, electrolyte imbalance [1]Usual onset 2 hours to 5 days after exposure [2]Duration Few days [1]Causes Vibrio cholerae spread by fecal-oral route [3] [1]Risk factors Poor sanitation, not enough clean drinking water, poverty [1] Diagnostic method Stool test [1]Prevention Improved sanitation, clean water, cholera vaccines [4] [1]Treatment Oral rehydration therapy, zinc supplementation, intravenous fluids, antibiotics [1] [5]Frequency 3–5 million people a year [1]Deaths 28,800 (2015) [6]Cholera is an infection of the small intestine by some strains of the bacterium Vib... (*content omitted*)	1. what is the bacteria that causes cholera 2. rice-water stools are associated with disease caused by which organism?
Gun Interest Profile: NRA Contributions to 113th Congress and Senate Judiciary Committee	... (*content omitted*) NEWSGun Interest Profile: NRA Contributions to 113th Congress and Senate Judiciary Committee Pamela Behrsin | February 20, 2013Next week federal gun legislation is expected to take center stage in the Senate Judiciary Committee. The Hearing on the Assault Weapons Ban of 2013 is scheduled for February 27, 2013. Data: Map Light conducted an analysis of campaign contributions from the political action committee (PAC) of the pro-gun interest group National Rifle Association (NRA) ... (*content omitted*) About Map Light: Map Light is a 501 (c) (3) nonprofit, nonpartisan research organization that reveals money's influence on politics ... (*content omitted*)	1. how much money does the nra contribute to the dems/reps? 2. how much money did the nra give to ted cruz of texas senate?

Figure 3: Two cases demonstrating NOVO's understanding of documents. The red, blue, and green tokens are selected high-confidence n-grams before training, after re-indexing the corpus once and 5 times, respectively.

Title	Document	Example Queries and Retrieval Order
What is the Social Security Retirement Age?	... (*content omitted*) What is the **Social Security Retirement** Age? **Social Security**\'s full-benefit retirement age is increasing gradually because of **legislation** passed by Congress in 1983. Traditionally, the full **benefit** age was 65, and early retirement benefits were first available at age 62, with a permanent reduction to 80 percent of the full benefit amount. Currently, the full benefit **age** is 66 years and 2 months for people born in **1955**, and it will gradually rise to 67 for those born in 1960 or later. Early **retirement** benefits will continue to be available at age 62, but they will be reduced more. When the full-benefit **age** reaches 67, benefits taken at **age** 62 will be reduced to 70 percent of the full benefit and benefits first taken at age 65 will be reduced to 86.7 percent of the full benefit. There is a financial bonus for delayed retirement. ... (*content omitted*)	Q1: age one can retire draw social sec R: social security, age, full benefit, 1955, retirement, legislation Q2: normal retirement age fo someone born in 1954 R: retirement, age, 1955, social security, full benefit, legislation Q3: retire age for full benefit R: retirement, full benefit, 1955, age, social security, legislation Q4: what is the age to draw social security R: social security, age, retirement, full benefit, 1955, legislation

Figure 4: Cases demonstrating NOVO's understanding of queries. The green tokens selected from the document are high-confidence n-grams taken as docid. For different queries denoted as Q, the generated n-gram's relevance to the query is arranged in descending order, reflecting the model's understanding of how to find the relevant document n-gram given a query.

5.4.2 ***Interpreting Retrieval Process.*** To interpret how the model understands the retrieval process, we provide a document and four queries and examine the order in which the model generates n-grams. The results are shown in Figure 4. The n-grams of docid are represented by the green tokens. For each query denoted as Q, NOVO generates n-grams step by step to retrieve the document, and the order T in which the n-grams are generated reflects NOVO's understanding of how to find the most relevant document n-gram for a given query. We observe the order of the generated n-grams is arranged in descending order of relevance to different queries. Specifically, the docid consists of "social security", "retirement", "age", "full benefit", "1955" and "legislation". The first question is about the age that one can draw social security, so the model generates "social security" first, followed by "age". The second and third questions asked about the retirement age, so the model generates "retirement" first. The four queries are all irrelevant to "legislation", so this n-gram is consistently the last generated.

6 CONCLUSION AND LIMITATIONS

In this study, we introduce NOVO, a neural-optimized vocabularial document identifier. NOVO docid leverages denoising query modeling and retrieval tasks for enhanced document retrieval performance through improved tokenization and semantic representation. Experimental results on MS MARCO and Natural Questions datasets demonstrate its superiority over existing methods in IR, while also offering interpretability into the indexing and retrieval process. However, there are still limitations in this work: 1) The current dataset size is relatively small (< 1M documents), leaving the question of how dataset size might affect the model's performance; 2) the efficacy of the encoder might diminish because of denoising query modeling due to the presence of information within documents that extends beyond their relevance to retrieval queries, leaving the possibility of identifying alternative tasks that can better extract document features; 3) The semantic comprehension of documents based solely on N-grams could be constrained, especially when handling intricate or nuanced information. We plan to address these limitations in subsequent research.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by the National Natural Science Foundation of China No. 62272467, the fund for building world-class universities (disciplines) of Renmin University of China, Beijing Outstanding Young Scientist Program No. BJJWZYJH012019100020098, Public Computing Cloud, Renmin University of China, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods, and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

REFERENCES

[1] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems* 35 (2022), 31668–31683.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[3] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. GERE: Generative evidence retrieval for fact verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2184–2189.

[4] 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*. 191–200.

[5] Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687* (2019).

[6] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904* (2020).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] P. Ferragina and G. Manzini. 2000. Opportunistic data structures with applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 390–398. https://doi.org/10.1109/SFCS.2000.892127

[9] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. arXiv:2109.10086 [cs.IR]

[10] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2288–2292. https://doi.org/10.1145/3404835.3463098

[11] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 55–64.

[12] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[13] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics* (2019).

[14] Jimmy Lin, Rodrigo Frassetto Nogueira, and Andrew Yates. 2020. Pretrained Transformers for Text Ranking: BERT and Beyond. *CoRR* abs/2010.06467 (2020). arXiv:2010.06467 https://arxiv.org/abs/2010.06467

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

[16] Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2022. DSI++: Updating Transformer Memory with New Documents. *arXiv preprint arXiv:2212.09744* (2022).

[17] Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: making domain experts out of dilettantes. In *ACM SIGIR Forum*, Vol. 55. ACM New York, NY, USA, 1–27.

[18] Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137* (2016).

[19] 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. (November 2016). https://www.microsoft.com/en-us/research/publication/ms-marco-human-generated-machine-reading-comprehension-dataset/

[20] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pretrained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 1864–1874. https://doi.org/10.18653/v1/2022.findings-acl.146

[21] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* 6 (2019).

[22] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[23] 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 (2020), 5485–5551.

[24] S. E. Robertson and S. Walker. 1997. On Relevance Weights with Little Relevance Information. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Philadelphia, Pennsylvania, USA) *(SIGIR '97)*. Association for Computing Machinery, New York, NY, USA, 16–24. https://doi.org/10.1145/258525.258529

[25] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[26] 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. arXiv:2304.04171 [cs.IR]

[27] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems* 35 (2022), 21831–21843.

[28] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems* 35 (2022), 25600–25614.

[29] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=zeFrfgyZln

[30] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

[31] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2022. DynamicRetriever: A Pre-training Model-based IR System with Neither Sparse nor Dense Index. *CoRR* abs/2203.00537 (2022).

[32] 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* (2022).

[33] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128* (2022).