Descriptive and Discriminative Document Identifiers for Generative Retrieval

Jiehan Cheng, Zhicheng Dou*, Yutao Zhu, Xiaoxi Li

Gaoling School of Artificial Intelligence, Renmin University of China
jiehancheng@gmail.com, dou@ruc.edu.cn, {yutaozhu94, lixiaoxi45}@gmail.com

Abstract

Generative document retrieval is a novel retrieval framework, which represents documents as identifiers (DocID) and retrieves documents by generating DocIDs. It has the advantage of end-to-end optimization over traditional retrieval methods and has attracted much research interest. Nonetheless, the development of efficient and precise DocIDs for document representation remains a pertinent issue within the field. Existing methods for designing DocIDs tend to consider only the relevance of DocIDs to the corresponding documents, while neglecting the ability of the DocIDs to distinguish the corresponding documents from similar ones, which is crucial for the retrieval task. In this paper, we design learnable descriptive and discriminative document Identifiers (D2-DocID) for Generative Retrieval and propose the paired retrieval model D2Gen. The D2-DocID is semantically similar to the corresponding documents (descriptive) and is able to distinguish similar documents (discriminative) in the corpus, thus enhancing retrieval performance. We use a contrastive learning assisted generative retrieval task to enable the model to understand the document and then complete the generative retrieval. We then design a DocID selection method to select DocIDs based on the retrieval model's understanding of the documents. Our experimental results on the MS MARCO and NQ320k dataset illustrate the effectiveness of the approach.

Introduction

Information retrieval (IR) techniques play a crucial role in various domains, including search engines, social media and recommendation systems. Traditional IR paradigms consists of two methods: sparse retrieval and dense retrieval. The former, such as BM25 (Robertson and Zaragoza 2009), rely on bag-of-words representations, while the latter, like DPR (Karpukhin et al. 2020), utilize semantic embeddings. However, both of these approaches require separate steps for representation and retrieval, which can limit their efficiency and effectiveness.

In recent years, generative information retrieval has emerged as a promising end-to-end retrieval paradigm and has attracted significant research interest. In generative document retrieval, queries are directly mapped to the identifiers of relevant documents (DocID), making DocIDs the key bridge between the queries and the documents. This paradigm differs from traditional approaches in the training and inference phases. In the training phase, it takes generating the correct DocID as the optimization goal instead of queries and documents similarity metrics. And in the inference phase, it generates DocIDs for the queries end-to-end instead of searching for the most similar results by comparison. Therefore, the design of DocIDs is crucial for generative document retrieval (Zhang et al. 2023; Wang et al. 2023). Effective document identifier design can significantly impact the performance and efficiency of generative IR systems (Tay et al. 2022; Zhou et al. 2022; Wang et al. 2022b).

Existing DocIDs can be categorized in terms of data type, generation method, and data structure type (Li et al. 2024). The data type of DocIDs can be either numeric (Zhuang et al. 2022; Chen et al. 2023; Tay et al. 2022) or text (Wang et al. 2023; Zhang et al. 2023). The numeric DocIDs can be assigned a nice semantic structure. For example, DocIDs with the same prefix can represent semantically similar documents (Tay et al. 2022). The text DocIDs are readable and can utilize the internal knowledge of pre-trained models. Using titles and URLs as DocIDs (Zhou et al. 2022; De Cao et al. 2020) is a straightforward way to indicate the semantics of documents. SE-DSI (Tang et al. 2023) uses synthetic queries as DocIDs, getting rid of the reliance on structural information. Zhang et al. (2024a) propose using term sets rather than term sequences as DocIDs to address the false pruning problem during generation. The generation methods of the above DocIDs are static, meaning that the DocIDs are pre-defined before optimizing the generative retrieval model, whereas some other DocIDs are learnable (Sun et al. 2024; Yang et al. 2023; Zhang et al. 2024a; Wang et al. 2023; Liu et al. 2024).

However, existing methods did not take into account the discriminative nature of DocIDs in a corpus. The function of DocIDs is to help accurately retrieve the target document from a corpus. Therefore, the identifiers of the same document should differ depending on the corpus in which it is stored. Specifically, **a good DocID should not only be semantically associated with the corresponding document, but also be distinguished from the semantics of other similar documents in the corpus** in order to reduce the confusion of the generative retrieval engine. To address this challenge, we design learnable **descriptive and dis-**

*Corresponding author.

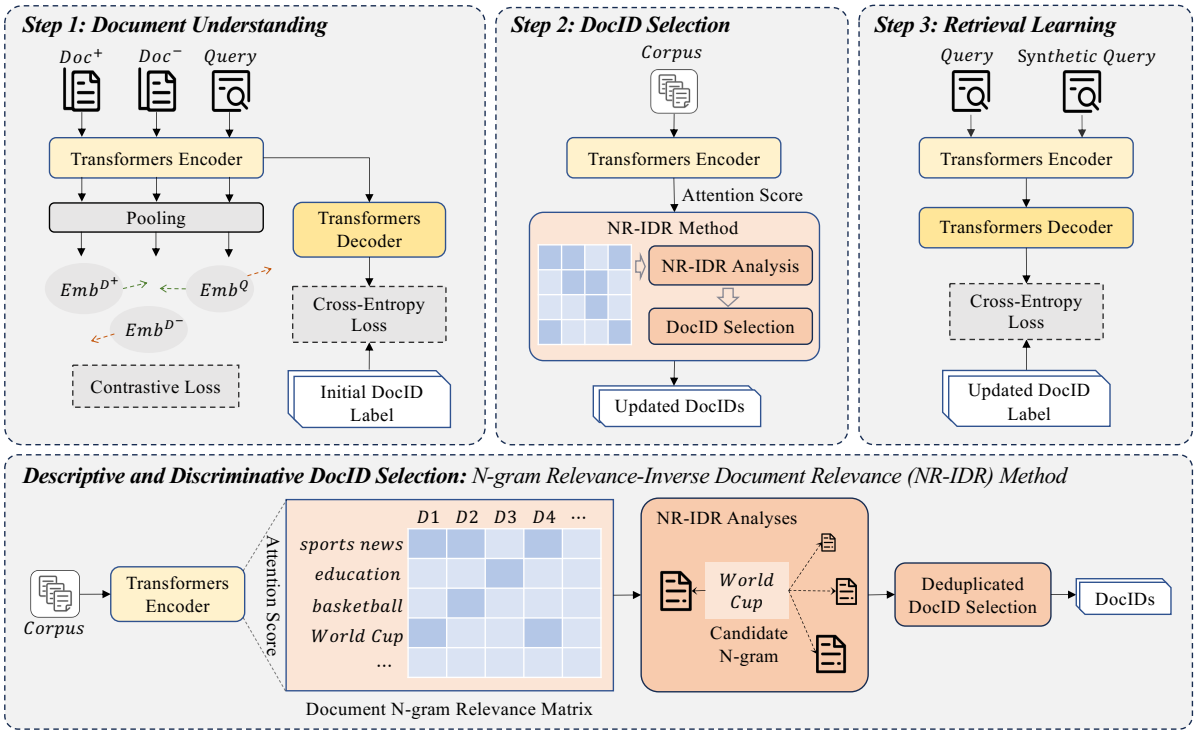

Figure 1: Overview of our D2Gen model optimization pipeline, which includes document understanding, descriptive and discriminative DocID selection and retrieval task learning.

criminative document identifiers (D2-DocID) based on **n-grams**. Specifically, as shown in the Document Understanding stage in Figure 1, we first use a retrieval learning task assisted by contrastive learning to equip the generative retrieval model with document comprehension, which means that the focused n-grams of a document will have higher attention scores when encoding it. Inspired by SE-DSI(Tang et al. 2023), we use synthetic queries to initialize the DocIDs. Then, in the DocID Selection stage, we extract all n-grams within the document and obtain a document-ngram semantic relevance matrix based on the attention scores, where each row of the matrix represents a document in the corpus, each column of the matrix represents an n-gram, and a value in the matrix represents the attention score of the n-gram in the document. This is a huge sparse matrix, since each document is only relevant to a very small proportion of n-grams. This matrix includes semantic information of all documents in the corpus. Inspired by TF-IDF (Ramos et al. 2003), we design the ngram Relevance-Inverse Document Relevance (NR-IDR) method to efficiently analyze and select DocIDs at the corpus level. Specifically, for each n-gram in a document, we separately compute the NR-IDR score based on the above matrix, which reflects the relevance of the n-grams to the document as well as their distinguishability from other documents. Then, we sequentially de-duplicate and filter the specified number of n-grams with the highest NR-IDR scores as DocID for this document. Our experiments demonstrate the effectiveness of such DocIDs with different generative retrieval models.

To further exploit the advantage of D2-DocID, in Retrieval Learning Stage, we use query and data augmentation to continually train the generative retrieval model based on the newly selected D2-DocID. Data augmentation, such as document fragments and synthetic queries, is widely used in generative retrieval model training(Wang et al. 2022b; Tang et al. 2023; Zhang et al. 2024a). However, the generated queries may suffer from poor quality and homogenization. We design a query filtering method based on query quality and diversity to obtain controlled, high-quality and diverse data augmentation, which enables the generative model to understand DocIDs from multiple perspectives and increase its robustness. We use the newly generated D2-DocID to update the DocID tags in the Document Understanding stage, thus iteratively learning and updating the DocIDs.

The main contributions of this paper are threefold:

(1) We design descriptive and discriminative DocIDs for generative retrieval, which not only describe the semantic information of documents, but also distinguish between similar documents, thus improving generative retrieval results.

(2) We design a generative retrieval model D2Gen, which can explicitly understand documents and iteratively learn and generate the DocIDs.

(3) The effectiveness of the method is validated on MS300k and NQ320k. For example, on MS300k, our method outperforms the best baseline by 4.5% on R@1.

Related Work

Text Retrieval

Text retrieval techniques can be broadly classified into two main approaches: sparse retrieval and dense retrieval. Sparse retrieval methods, such as BM25 (Robertson and Zaragoza 2009), SPLADE (Formal, Piwowarski, and Clinchant 2021) and UniCOIL (Lin and Ma 2021), rely on bag-of-words representations and have been widely used in traditional information retrieval systems. These methods represent documents and queries as sparse vectors based on term frequencies and inverse document frequencies. Dense retrieval methods, like DPR (Karpukhin et al. 2020), E5 (Wang et al. 2022a) and RepLLaMA (Ma et al. 2024), utilize semantic embeddings to represent documents and queries in a dense vector space. Dense retrieval methods aim to encode the semantic meaning of text and have shown promising results in various retrieval tasks. However, both sparse and dense retrieval approaches require separate steps for representation and retrieval, which can limit their efficiency and effectiveness in certain scenarios.

Generative Retrieval

Generative Retrieval revolutionizes information retrieval by directly mapping queries to DocIDs using generative models, eliminating separate indexing and retrieval stages. The design of docids is crucial for system performance, with two main types: numeric DocIDs (Zhuang et al. 2022; Chen et al. 2023; Tay et al. 2022) and text DocIDs (Wang et al. 2023; Zhang et al. 2023; Bevilacqua et al. 2022). The numeric DocIDs can be assigned a nice semantic structure. The specific construction process is firstly obtaining the semantic embedding of documents, then categorizing the articles by clustering (Tay et al. 2022; Wang et al. 2022b; Chen et al. 2023) or calculating PQ values (Zhou et al. 2022), and finally designing the DocIDs according to the categories. The semantic DocIDs of similar documents often have the same prefix, and this semantic structure can help to improve the effect of the retrieval engine (Tay et al. 2022). The text DocIDs are readable and can utilize the internal knowledge of pre-trained models. Ultron (Zhou et al. 2022), SE-DSI(Tang et al. 2023), etc. use static DocIDs. NOVO (Wang et al. 2023), TSGen (Zhang et al. 2024a) design learnable and set-based DocIDs that makes the decoding process more flexible. However, they do not fully utilize the corpus background to make DocID discriminative and distinguish similar documents. Therefore, we design the descriptive and discriminative DocIDs (D2-DocID)

Methodology

In generative retrieval, each document D in a corpus $\mathcal{C} = \{D_1, D_2, \cdots, D_{|\mathcal{C}|}\}$ is represented by its identifiers $\mathcal{I}(D)$, which is called DocID. When processing a query Q, a generative model generates the DocID of the relevant documents D for it. In this paper, we design learnable Descriptive and Discriminative Document Identifiers D2-DocID, and the paired retrieval model D2Gen. D2-DocID is a sequence of extracted n-grams of length 1-3 in the corresponding document. The model contains the following three modules.

(1) **Document Understanding**. The n-grams composing the DocIDs should be able to represent the semantics of the document, therefore, a good representation model is needed to represent the document. In order to extract the key information in a whole document, we design specialized optimization methods to help the generative retrieval model learn the end-to-end retrieval task while learning to extract key information from documents. The model's ability to extract information is critical for the next step of document identifier selection.

(2) **DocID Selection**. For each document, we extract all its n-grams of length 1-3 as DocID candidates. We use the pre-trained model from the first step to obtain the document-ngram semantic relevance matrix for the entire corpus. Each row of this matrix represents a document, and each column represents an n-gram. The values of the matrix stand for the semantic relevance of the document to the n-gram, which is the average of the attention score in the last layer of the encoder. This is a large sparse matrix because only n-grams that appear in a document are scored. Next, in order to obtain a DocID that can both represent the semantics of the corresponding document and distinguish it from other similar documents in the corpus, we design a corpus-aware ngram filtering function to compute the relative relevance of the ngrams to the document, and accordingly obtain k n-grams to compose the DocID.

(3) **Retrieval Learning**. To adapt the retrieval model to the DocIDs generated in the second step, we take them as new DocIDs to continue to train the model. In this step, we use filtered synthetic queries for data augmentation and mix them with the queries as training data.

The details of each step will be introduced in the remaining part of this section.

Document Understanding

To train the model to extract key semantics from documents, we design contrastive learning assisted retrieval tasks to enhance the model's document understanding ability.

We use the encoder-decoder architecture to extract document semantics. Give the training dataset $\mathcal{D} = \{(Q, D)\}$, the query Q is encoded by the encoder model, and the generation probability of the DocID is estimated as follows:

$$\mathbf{e}_i = \text{Decoder}(\text{Encoder}(Q), \text{id}_{<i}), \quad (1)$$

$$P(\text{id}_i | q, \text{id}_{<i}; \Theta_{e,d}) = \text{Softmax}(\text{Lmhead}(\mathbf{e}_i)), \quad (2)$$

where e_i is the output embedding of the decoder given the prefix DocID id_i. $\Theta_{e,d}$ represents the trainable parameters in the encoder and the decoder of the retrieval model. $Lmhead$ is the language model head, which is a linear layer used to map the output of the Transformer model to the size of the vocabulary table. thereby generating the probability distribution of the next word. The retrieval loss L_r can then be expressed as:

$$\mathcal{L}_r(\Theta_{e,d}) = \sum_{i}^{l} \text{logP}\left(id_i \middle| q, id_{<i}; \Theta_{e,d}\right), \quad (3)$$

where l represents the length of DocId.

For any $(Q, D) \in \mathcal{D}$, we use the encoder of the retrieval model to compute their embedding and design the comparison loss as follows:

$$\mathbf{h}_q = \text{MeanPooling}(\text{Encoder}(Q)), \tag{4}$$

$$\mathbf{h}_d = \text{MeanPooling}(\text{Encoder}(D)), \tag{5}$$

$$\mathcal{L}_c = -\log \frac{e^{s(\mathbf{h}_q, \mathbf{h}_d)/\tau}}{e^{s(\mathbf{h}_q, \mathbf{h}_d)/\tau} + \sum_{(q,d^-)\notin\mathcal{C}} e^{s(\mathbf{h}_q, \mathbf{h}_d)/\tau}}, \tag{6}$$

where h_Q and h_D are the embedding representations of the query Q and the document D, respectively. (Q, D^-) is the negative instance. $s(\mathbf{h}_q, \mathbf{h}_d)$ stands for cosine similarity. We want the encoder of the retrieval model to produce similar embeddings of (Q,D) pairs, and thus more adapted to the retrieval task. We merge the objective functions to train the model:

$$\mathcal{L}_p = \mathcal{L}_r + \lambda * \mathcal{L}_c. \tag{7}$$

DocID Selection

For every document D in the corpus $\mathcal{C}$, we utilize the encoder of the retrieval model D2Gen to compute its correlation with each n-grams in the document separately. Specifically, the computation process is as follows. For each document D, we use the nltk toolkit to get the n-gram set $\mathcal{G}_D$ of documents and tokenize the documents as $D = (d_1, d_2, \cdots, d_n)$. Every n-gram $g \in \mathcal{G}_D$ corresponds to several tokens of the document, denoted as $g = (d_i, \cdots, d_j)$. We feed D into Encoder and the correlation between g and D can be computed as:

$$\text{Rel}(D, g) = \text{MP}(\text{Att}(\text{Encoder}(D))[d_i : d_j]), \tag{8}$$

where Att stands for Attention Score of the last layer of the model and MP stands for Mean Pooling. Then, we traverse all the documents $\mathcal{C}$ and record the relevance score in the matrix $\mathcal{M}$, i.e.,

$$\mathcal{M}[D, g] = \text{Rel}(D, g), \text{ for every } D \in \mathcal{C}, g \in \mathcal{G}_D. \tag{9}$$

In this way, we obtain the doc-ngram correlation matrix $\mathcal{M}$, which is a sparse matrix because a single document contains very few n-grams out of the total n-grams and we set the position in $\mathcal{M}$ that is not assigned a value to None.

This matrix records the relevance between all the documents in the corpus and their n-grams. Next, we design a matrix analysis method called ngram relevance-inverse document relevance (NR-IDR) to select DocIDs. Through the NR-IDR method, we can select descriptive and discriminative DocIDs. **Descriptive** means that the selected DocIDs are semantically similar to the documents and can represent the document content, while **discriminative** means that similar document identifiers in the corpus can be distinguished from each other, thus leading to more accurate retrieval results. The NR-IDR method is designed as follows:

We denote the set of n-grams formed by all documents of copurs C as $\mathcal{G} = (g_1, g_2, \cdots, g_{|\mathcal{G}|})$. For any n-gram $g_j \in \mathcal{G}$, we compute its inverse document relevance IDR$[j]$ by:

$$\begin{aligned} \text{IDR}[j] &= \log\left(\frac{1}{\text{Mean}_{k\in I_j}(\mathcal{M}[k, j])}\right) \\ &= \log\left(\frac{|I_j|}{\sum_{k\in I_j} \mathcal{M}[k, j]}\right), \end{aligned} \tag{10}$$

where I_j represents the index of the row of $\mathcal{M}$ whose element in column j does not have the value None, which can be formulated as:

$$I_j = \left\{k \in [0, |\mathcal{C}|) \middle| \mathcal{M}[k, j] \neq \text{None}\right\}. \tag{11}$$

According to Equation (10), IDR$[j]$ is determined by the average relevance of g_j with all relevant documents. Similar to TF-IDF, the lower the IDR value of g_j, the higher the average relevance with all relevant documents, which also indicates the lower discriminative power of g_j. The opposite is also true. Then, we compute ngram relevance-inverse document relevance (NR-IDR) score for any document D_i and any of its n-grams g_i as follows:

$$\text{NR-IDR}[i, j] = \mathcal{M}[i, j] * \sqrt{\text{TF}[i, j]} * \text{IDR}[i]. \tag{12}$$

NR-IDR score evaluates n-grams comprehensively from the perspectives of semantic relevance to documents, counting frequency, and inverse document relevance. The higher the NG-IDR score of a (D, g) pair is, the higher the relevance of the n-gram g to the document D is, and the higher the differentiation of g from other documents is.

We then select DocIDs based on the NR-IDR score. Specifically, for any document D_i, we sort the set of ngrams $\mathcal{G}_{D_i}$ based on NR-IDR$[i, :]$ and sequentially de-duplicate and select top n_g ngrams as the DocID of D_i. Specifically, our de-duplication approach is that, for an n-gram, if each of its word prototypes duplicates a word prototype of an already selected DocID, then skip this n-gram and continue to determine the next one.

Retrieval Learning

In this section, we present the construction of training data and the training and inference process.

Training Data Construction Work such as NCI (Wang et al. 2022b), Ultron(Zhou et al. 2022) and DSI-QG(Zhuang et al. 2022) tends to use shorter pseudo-queries as training data. Multi also uses passages to generate pseudo queries and filter based on dense retrieval capabilities. However, our experiments finds that automatically generated pseudo queries often have homogeneity problems. We use document fragments to generate pseudo queries, and innovatively propose a filtering method based on query diversity and quality to build diverse data sets.

Specifically, for each document $D \in \mathcal{C}$, we first divide it into joint passages $P = \{p_1, p_2, \cdots\}$, where each passage consists of $s = 3$ sentences, and the overlap is $o = 1$ sentence. Subsequently, we generate pseudo queries based on the original text D and the passages P. We generate $n_d = 10$ pseudo queries from the original text D , denoted as $Q_d = \{q_1, q_2, \cdots\}$, and $n_p = 3$ pseudo queries from each passage, denoted as $Q_p = \{q_{ij}\}$, where $1 \leq i \leq |P|, 1 \leq j \leq n_d$. Then we take $Q = Q_d \cup Q_p$ as the original pseudo-query set. Subsequently, in order to improve training efficiency, we use query-filter to filter pseudo queries and select diverse pseudo queries with retrieval capabilities as training input.

In detail, we denote the selected pseudo query set as Q_s and initialize Q_s as an empty set. We first utilize a dense retrieval model $\mathcal{M}_\beta$ to retrieve each query $q \in Q$ in the

Dataset	#Docs	#Train Queries	#Test Queries
MS300k	324,311	367,008	5,193
NQ320k	109,712	307,373	7,830

Table 1: Statistics of the document retrieval datasets.

document corpus $\mathcal{C}$, and calculate the MRR@10 value of each query as a measure of the retrieval capability of the query, denoted as mrr_q, and sort based on MRR@10. Then we traverse each query $q \in Q$ and calculate its semantic similarity score sim_q with all queries in Q_s respectively:

$$\text{sim}_q = \max_{q' \in Q_s} \left[\mathcal{M}_\beta(q) \cdot \mathcal{M}_\beta(q') \right]. \tag{13}$$

If semantic similarity sim_q of query q is less than the similarity threshold λ_1 and the MRR@10 value mrr_q is greater than the threshold λ_2, then we set $Q_s = Q_s \cup \{q\}$, and continue processing the next query. Finally, we use Q_s as a representative pseudo query set for document d as input of the training data and the docid of d as output, together with the original training dataset, to compose the training dataset.

Training and Inference We continue to use the same encoder-decoder structure as in Section and inherit its parameters. We then optimize the retrieval performance of the model using the newly generated DocID in Section . We mix real queries and selected synthetic queries to train the model based on the following generative retrieval loss $\mathcal{L}_{gr}$:

$$\mathcal{L}_{gr}(\Theta_{e,d}) = \sum_{i}^{l} \log P\left(id_i | q, id_{<i}; \Theta_{e,d}\right). \tag{14}$$

Experiment Setup

Datasets and Evaluation Metrics

Datasets We experiment on two widely recognized datasets: MS MARCO (Bajaj et al. 2016) and Natural Questions (NQ) (Kwiatkowski et al. 2019). MS MARCO contains 300k query-document pairs, in which the queries are extracted from Bing's query logs and the documents are extracted from web documents retrieved by Bing. Natural Questions contains 320k query-document pairs extracted from Wikipedia, in which the queries are real and the documents are Wikipedia pages. Following NOVO (Wang et al. 2023), we eliminate duplicate documents in NQ based on document titles and use the training set and the validation set divided in NQ as our training set and testing set. For MSMARCO, however, the documents come from web pages and do not have structural information as regular as NQ, so we eliminate duplicate documents in MSMARCO based on the URLs as Ultron (Zhou et al. 2022) does, and use the training set and the dev set divided in MS MARCO as our training set and testing set. We named the processed datasets NQ320k and MS320k respectively, and Table 1 summarizes their statistical information.

Evaluation Metrics Following existing studies (Zhang et al. 2024a; Tay et al. 2022), we adopt the widely used MRR@K (M@K) and Recall@K (R@K) to measure the retrieval performance.

Baselines

For traditional retrieval methods, we compare sparse retrieval, such as BM25, UniCOIL, SPLADEv2 (Formal et al. 2021), and dense retrieval, such as DPR (Karpukhin et al. 2020), ANCE (Xiong et al. 2020), GTR-BASE (Ni et al. 2021). We also compare with generative retrieval methods. In order to validate the effects of D3, we choose generative retrieval models with different kinds of DocIDs. Specifically, DSI (Tay et al. 2022), NCI (Wang et al. 2022b) use Kmeans-based semantic numeric DocID, GENRE (De Cao et al. 2020) and Ultron (Zhou et al. 2022) use tiltes or urls. SEAL(Bevilacqua et al. 2022) and MINDER use multi DocIDs for one document, the former using all n-grams of a document, and the latter using titles, synthetic queries and n-grams. GENRET(Sun et al. 2024) uses a learned numeric DocID, NOVO (Wang et al. 2023) and TSGen (Zhang et al. 2024a) use collection DocIDs instead of sequences.

Implementations

We use T5-base as the base model with the structure of transformer encoder-decoder. We use nltk to split words and select N-grams in the range 1-3. We choose the number of n-grams $n_g = 3$ to compose the DocIDs on both datasets and analyze it in the ablation experiments with different n_g. On MS300k, we choose similarity threshold $\lambda_1 = 0.99$, MRR threshold $\lambda_2 = 0.1$ to improve the diversity of the synthetic queries so as to reflect the document from multiple perspectives, while on NQ320k, we set similarity threshold $\lambda_1 = 0.99$, MRR threshold $\lambda_2 = 0.6$ to improve the retrieval performance of the query.

Experimental Results

Main Results

Table 2 shows the results of our experiments on MS MARCO and NQ320k. We have the following observations.

Firstly, our model **D2Gen significantly outperforms previous generative retrieval models on the MS300k.** For example, on MS300k, it outperforms NOVO by +3.9% on M@10; **We also get outstanding results on the NQ320k.** For example, it performs best on the R@1. Differences in advancement on the two datasets are reasonable. The more significant improvement of D2Gen on MS300k is due to the fact that MS300k's corpus is multi-sourced, while NQ's corpus is only from Wikipedia. In comparison, MS300k lacks uniform structural information, and we design D2-Docid to understand documents and extract key semantics, and design accurate document identifiers, which significantly improves retrieval performance. The corpus of NQ300K is highly structured, and the headline can often summarize the documents well, which limits the improvement of D2-DocID. The results demonstrate the high retrieval ability of D2Gen, especially on a corpus consisting of multi-source, unstructured documents.

Secondly, **our DocIDs achieve significantly superior results compared to other DocID types**. Existing work has tried many kinds of document identifiers. For example, NCI uses clustering-based semantic identifiers, Ultron uses title+URL, and SEAL uses substrings as document identifiers.

Category	Method	MS300K					NQ320K				
		M@10	M@100	R@1	R@10	R@100	M@10	M@100	R@1	R@10	R@100
Sparse	BM25†	0.248	0.255	0.186	0.391	0.573	0.480	0.487	0.376	0.704	0.881
	UniCOIL	0.425	0.435	0.284	0.766	0.951	0.710	0.713	0.619	0.862	0.926
	SPLADEv2	0.443	0.452	0.328	0.779	**0.956**	0.726	0.731	0.624	0.873	0.954
Dense	DPR	0.424	0.433	0.271	0.764	0.948	–	0.599	0.502	0.777	0.909
	ANCE	0.451	0.455	0.299	0.785	0.953	–	0.602	0.502	0.785	0.914
	GTR-Base†	0.576	0.581	0.471	0.785	0.912	0.658	0.663	0.567	0.836	0.936
Generative	DSI†	0.318	0.327	0.239	0.507	0.643	0.588	0.592	0.542	0.706	0.804
	NCI	0.408	0.417	0.301	0.643	0.851	–	0.731	0.659	0.852	0.924
	GENRE	0.361	0.368	0.266	0.579	0.751	0.653	0.656	0.591	0.756	0.814
	Ultron	0.432	0.437	0.304	0.676	0.794	0.726	0.729	0.654	0.854	0.911
	SEAL	0.393	0.402	0.259	0.686	0.899	–	0.677	0.599	0.812	0.909
	MINDER	0.431	0.435	0.289	0.728	0.916	0.709	0.713	0.627	0.869	0.933
	TSGen	0.502	0.505	0.384	0.781	0.931	0.771	**0.774**	0.708	0.889	0.948
	GenRet	0.581	–	0.479	0.798	0.916	–	0.759	0.681	0.888	0.952
	NOVO	0.592	-	0.491	0.808	0.925	–	0.767	0.693	**0.897**	**0.959**
	D2Gen	**0.615***	**0.620***	**0.513***	**0.813***	0.915	**0.772**	**0.774**	**0.710**	0.876	0.936

Table 2: Evaluation of the retrieval performance on NQ320K and MS300K. The methods marked with † are from our implementation, and the others are from their official implementation and (Zhang et al. 2024b). * indicates significant improvements over MINDER and TSGen on MS300K and NQ320K respectively with p-value ≤ 0.05.

These approaches attempt to represent a document with a short sequence, and our D2Gen can significantly outperform them on both datasets by deeply analyzing document semantics and generating descriptive and discriminative identifiers. It is worth noting that model effectiveness is also affected by other factors such as optimization methods and training data. To illustrate the effect of our Docid more convincingly, we control the variables and validate it further in the following section.

Thirdly, Compared with traditional retrieval methods, **D2Gen has a significant advantage in the metric of small cutoffs.** For example, on the MS300k, the model outperformed the GTR by 26% on M@10. Compared with traditional retrieval methods, our approach can balance excellent retrieval performance and semantic representation, meaning that the model directly generates readable DocIDs representing the semantics of the corresponding documents. This is heuristic for further end-to-end application to RAGs.

Analysis on Effectiveness of D2-DocID

In the previous section, we briefly discussed the superiority of D2Gen. In order to further prove that the improvement of the DocID indeed brings about an improvement in retrieval ability, and not an illusion brought about by the improvement of other factors such as the model's optimization method, we designed the following experiment. We selected two generative retrieval models that are innovative in their optimization methods, and training data, respectively, and verified D2-DocID by replacing the document identifiers they use with D2-DocID. In addition, we also replaced the DocID used in D2Gen for comparison.

As shown in Table 3, **D2-DocID outperforms other DocIDs on both their primitive model and ours.** This demonstrates the generalizable help of D2-DocID on the retrieval power of generative retrieval.

Model	DocID	M@10	R@1	R@10
Ultron	Title+URL	0.400	0.296	0.678
	PQ	0.454	0.316	0.731
	Atomic	0.469	0.328	0.741
	D2-DocID	**0.538**	**0.431**	**0.757**
SE-DSI	Pseudo-query	0.469	0.375	0.665
	D2-DocID	**0.536**	**0.434**	**0.742**
D2Gen	Title+URL	0.579	0.489	0.758
	Pseudo-query	0.585	0.486	0.782
	D2-DocID	**0.607**	**0.504**	**0.810**

Table 3: Evaluation of the D2-DocID on different models with their docids on MS300k.

Ablation Studies

Our ablation experiments on MS300k for the factors influencing the D2Gen are shown in Table 4.

- **DocID selection.** In order to validate the effectiveness of the document-ngram relevance matrix and the NR-IDR method, we compared the selection of DocIDs using the TF-IDF, relevance ranking, and NR-IDR methods. TR-IDR outperforms the TF-IDF method, which illustrates that the document-ngram relevance matrix characterizes the semantics of the documents better than the document-word frequency matrix, and verifies the effectiveness of its semantic representation. NR-IDR outperforms the Relevance Ranking method, which illustrates that the discriminative nature of DocID is critical for retrieval enhancement.

Factor	Setting	M@10	R@1	R@10
N-gram Selection	TF-IDF	0.586	0.480	0.800
	NR-IDR*	**0.607**	**0.504**	**0.810**
Iterations	1*	0.603	0.497	0.809
	2	**0.610**	**0.505**	**0.817**
Pipeline	w.o. $\mathcal{L}_r$	0.592	0.485	0.801
	D2Gen*	**0.607**	**0.504**	**0.810**
N-gram Number	2	0.594	0.491	0.796
	3*	**0.607**	**0.504**	**0.810**
	6	0.603	0.497	0.809
	9	0.580	0.473	0.793
Similarity Threshold	0.99*	**0.616**	**0.513**	**0.817**
	0.9	0.607	0.504	0.810

Table 4: Ablation studies on MS300k. The default settings of the ablation studies are marked with *.

- **Iterations.** We compare the effect of the number of iterations of model learning and iterative DocID updates on the results. The results show that iterative learning leads to a improvement, while iterating once has achieved results beyond the other baselines. This suggests that after one iteration, the model is already able to learn the document and extract the semantics adequately. The model retrieval ability was further improved after the model was iterated for multiple rounds. The reason is that multiple rounds of iteration of the model helps it to understand the document in more detail.
- **Pipeline.** We design contrast learning-assisted retrieval tasks to help retrieval models gain document comprehension. To verify the effect of contrast loss, we compare the model to a pipeline without contrast loss. It can be observed that the contrast loss does play a positive role on the retrieval performance.
- **N-gram number.** We compare the effect on the retrieval ability of using different numbers of n-grams to compose the DocID. The results show that using the number of n-grams of 3 works best on MS300k at one iteration, and more or less n-grams reduce the retrieval ability. This is because when the number of n-grams is too small, the DocID cannot adequately represent the document semantics, and the problem of DocID duplication is more serious. Whereas when the number of n-grams is too large, the model decoding difficulty increases and the model is prone to illusions. So choosing the right length of DocID is important for the model retrieval ability.
- **Similarity Threshold.** We explored the effect of different similarity thresholds in data augmentation on retrieval performance. The experiments show that too much differentiation instead leads to a decrease in the results, which may be due to the fact that the generative model is confused by the widely differing datasets.

Case Study

In this section, we will show the descriptive as well as the discriminative nature of DocID through concrete examples.

Title: T-Mobile To Go Refill - PIN **URL:** http://www.callingmart.com/products/wireless/ Product-Detail?ID=35 **Body:** ... To Go Prepaid (PAY AS YOU GO) has great rates and popular phones like the Sidekick II. The Sidekick pricing plan is a great deal, especially for teens. For only a $1 per day ... **DocID:** T Mobile, Refill, Sidekick
Title: Exchange a device under warranty **URL:** https://support.t-mobile.com/docs/DOC-1656 **Body:** ... Be warned: If you don't return the defective device within seven (7) days, T-Mobile charges a non-return fee. Be sure to return it as soon as possible ... **DocID:** T Mobile, Mobile charges, defective device
Title: Can i keep my tmobile phone number if i switch to sprint? **URL:** https://answers.yahoo.com/question/index?qid=20090215171615AA4IIyV **Body:** ... Best Answer: Yes, you absolutely can keep your number when switching wireless carriers. DO NOT cancel your service with T-Mobile. Simply go into a Sprint store and give them your T-Mobile phone number and account number ... **DocID:** phone number, T Mobile, to sprint

Table 5: Case study of D2-DocID on NQ320k. The documents presented talk about different perspectives on the same object. D2Gen demonstrates its descriptive and discriminative capabilities.

As seen in table 5, all three documents narrate topics related to T-mobile, and D2-DocID both reflects this key semantics and distinguishes where the documents differ. The first focuses on refills, so the DocID includes "refill" as well as an example of a discount, "Sidekick". The second focuses on after-sales, so the DocID includes the topics such as "charges" when over seven days and "defective devices". The third focuses on replacing T-mobile with sprint, so the DocID includes the document's focus on keeping the "phone number" and changing "to sprint".

Conclusion and Future Work

In this paper, we design descriptive and discriminative document identifiers for generative retrieval, which can both represent document semantics and distinguish between similar documents to improve retrieval performance. We also propose a paired generative retrieval model: D2Gen. we design document comprehension-assisted retrieval tasks to help the retrieval model understand documents, and extract DocIDs based on the corpus characteristics using the NR-IDR method. we also preserve document information more comprehensively by means of data augmentation through diversity filtering. Through experiments, we demonstrate the high retrieval performance of D2Gen on two widely used datasets and the generalization of D2-DocID across different models. In the future, we would like to continue to explore how comparative learning and document understanding can help in generative retrieval. Moreover, we will try to apply D2Gen to retrieval-augmented generation (RAG).

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. 62272467 and 62402497), Beijing Natural Science Foundation L233008, and Beijing Municipal Science and Technology Project No. Z231100010323009. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

References

Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Bevilacqua, M.; Ottaviano, G.; Lewis, P.; Yih, S.; Riedel, S.; and Petroni, F. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35: 31668–31683.

Chen, X.; Liu, Y.; He, B.; Sun, L.; and Sun, Y. 2023. Understanding differential search index for text retrieval. *arXiv preprint arXiv:2305.02073*.

De Cao, N.; Izacard, G.; Riedel, S.; and Petroni, F. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Formal, T.; Lassance, C.; Piwowarski, B.; and Clinchant, S. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Formal, T.; Piwowarski, B.; and Clinchant, S. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, 2288–2292. ACM.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP*, 6769–6781.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.-W.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7: 452–466.

Li, X.; Jin, J.; Zhou, Y.; Zhang, Y.; Zhang, P.; Zhu, Y.; and Dou, Z. 2024. From matching to generation: A survey on generative information retrieval. *arXiv preprint arXiv:2404.14851*.

Lin, J.; and Ma, X. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *CoRR*, abs/2106.14807.

Liu, Y.; Yang, T.; Zhang, Z.; Song, M.; Huang, H.; Deng, W.; Sun, F.; and Zhang, Q. 2024. ASI++: Towards Distributionally Balanced End-to-End Generative Retrieval. *arXiv preprint arXiv:2405.14280*.

Ma, X.; Wang, L.; Yang, N.; Wei, F.; and Lin, J. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2421–2425.

Ni, J.; Qu, C.; Lu, J.; Dai, Z.; Ábrego, G. H.; Ma, J.; Zhao, V. Y.; Luan, Y.; Hall, K. B.; Chang, M.-W.; et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Ramos, J.; et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, 29–48. Citeseer.

Robertson, S. E.; and Zaragoza, H. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4): 333–389.

Sun, W.; Yan, L.; Chen, Z.; Wang, S.; Zhu, H.; Ren, P.; Chen, Z.; Yin, D.; Rijke, M.; and Ren, Z. 2024. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems*, 36.

Tang, Y.; Zhang, R.; Guo, J.; Chen, J.; Zhu, Z.; Wang, S.; Yin, D.; and Cheng, X. 2023. Semantic-enhanced differentiable search index inspired by learning strategies. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4904–4913.

Tay, Y.; Tran, V.; Dehghani, M.; Ni, J.; Bahri, D.; Mehta, H.; Qin, Z.; Hui, K.; Zhao, Z.; Gupta, J.; et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35: 21831–21843.

Wang, L.; Yang, N.; Huang, X.; Jiao, B.; Yang, L.; Jiang, D.; Majumder, R.; and Wei, F. 2022a. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *CoRR*, abs/2212.03533.

Wang, Y.; Hou, Y.; Wang, H.; Miao, Z.; Wu, S.; Chen, Q.; Xia, Y.; Chi, C.; Zhao, G.; Liu, Z.; et al. 2022b. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35: 25600–25614.

Wang, Z.; Zhou, Y.; Tu, Y.; and Dou, Z. 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, 2656–2665. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701245.

Xiong, L.; Xiong, C.; Li, Y.; Tang, K.-F.; Liu, J.; Bennett, P. N.; Ahmed, J.; and Overwijk, A. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *ICLR*.

Yang, T.; Song, M.; Zhang, Z.; Huang, H.; Deng, W.; Sun, F.; and Zhang, Q. 2023. Auto Search Indexer for End-to-End Document Retrieval. In *Conference on Empirical Methods in Natural Language Processing*.

Zhang, P.; Liu, Z.; Zhou, Y.; Dou, Z.; and Cao, Z. 2023. Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines. *arXiv preprint arXiv:2305.13859*.

Zhang, P.; Liu, Z.; Zhou, Y.; Dou, Z.; Liu, F.; and Cao, Z. 2024a. Generative Retrieval via Term Set Generation.

In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, 458–468. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704314.

Zhang, P.; Liu, Z.; Zhou, Y.; Dou, Z.; Liu, F.; and Cao, Z. 2024b. Generative Retrieval via Term Set Generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 458–468.

Zhou, Y.; Yao, J.; Dou, Z.; Wu, L.; Zhang, P.; and Wen, J.-R. 2022. Ultron: An ultimate retriever on corpus with a model-based indexer. *arXiv preprint arXiv:2208.09257*.

Zhuang, S.; Ren, H.; Shou, L.; Pei, J.; Gong, M.; Zuccon, G.; and Jiang, D. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128*.