Multiview Identifiers Enhanced Generative Retrieval

Yongqi Li[1], **Nan Yang**[2], **Liang Wang**[2], **Furu Wei**[2], **Wenjie Li**[1]
[1]The Hong Kong Polytechnic University [2]Microsoft
liyongqi0@gmail.com
{nanya,wangliang,fuwei}@microsoft.com cswjli@comp.polyu.edu.hk

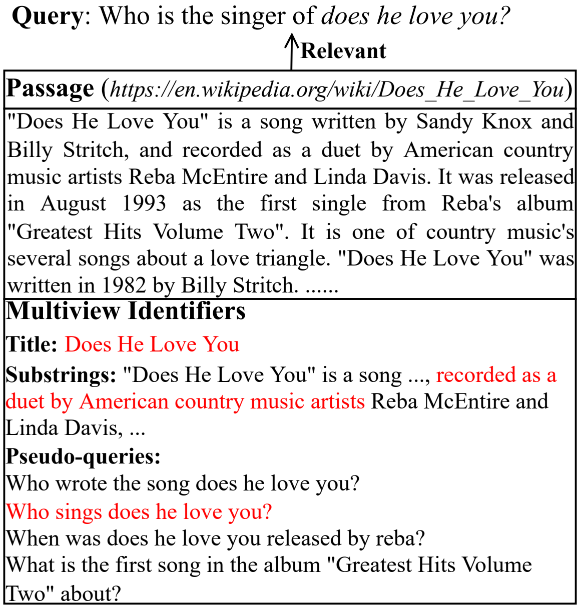
Abstract

Instead of simply *matching* a query to pre-existing passages, generative retrieval *generates* identifier strings of passages as the retrieval target. At a cost, the identifier must be distinctive enough to represent a passage. Current approaches use either a numeric ID or a text piece (such as a title or substrings) as the identifier. However, these identifiers cannot cover a passage's content well. As such, we are motivated to propose a new type of identifier, synthetic identifiers, that are generated based on the content of a passage and could integrate contextualized information that text pieces lack. Furthermore, we simultaneously consider multiview identifiers, including synthetic identifiers, titles, and substrings. These views of identifiers complement each other and facilitate the holistic ranking of passages from multiple perspectives. We conduct a series of experiments on three public datasets, and the results indicate that our proposed approach performs the best in generative retrieval, demonstrating its effectiveness and robustness. The code is released at https://github.com/liyongqi67/MINDER.

1 Introduction

Text retrieval is a fundamental task in information retrieval and plays a vital role in various language systems, including search ranking (Nogueira and Cho, 2019) and open-domain question answering (Chen et al., 2017). In recent years, the dual-encoder approach (Lee et al., 2019; Karpukhin et al., 2020), which encodes queries/passages into vectors and matches them via the dot-product operation, has been the de-facto implementation. However, this approach is limited by the embedding space bottleneck (Lee et al., 2022a) and missing fine-grained interaction (Wang et al., 2022b).

An emerging alternative to the dual-encoder approach is generative retrieval (De Cao et al., 2020; Tay et al., 2022; Bevilacqua et al., 2022). Generative retrieval utilizes autoregressive language models to generate identifier strings of passages, such as titles of Wikipedia pages, as an intermediate target for retrieval. The predicted identifiers are then mapped as ranked passages in a one-to-one correspondence. Employing identifiers, rather than generating passages directly, could reduce useless information in a passage and makes it easier for the model to memorize and learn. At a cost, the identifier must be distinctive enough to represent a passage. Therefore, high-quality identifiers have been the secret to effective generative retrieval.



Query: Who is the singer of *does he love you?*

Relevant

Passage (*https://en.wikipedia.org/wiki/Does_He_Love_You*)

"Does He Love You" is a song written by Sandy Knox and Billy Stritch, and recorded as a duet by American country music artists Reba McEntire and Linda Davis. It was released in August 1993 as the first single from Reba's album "Greatest Hits Volume Two". It is one of country music's several songs about a love triangle. "Does He Love You" was written in 1982 by Billy Stritch. ......

Multiview Identifiers

Title: Does He Love You

Substrings: "Does He Love You" is a song ..., recorded as a duet by American country music artists Reba McEntire and Linda Davis, ...

Pseudo-queries:
Who wrote the song does he love you?
Who sings does he love you?
When was does he love you released by reba?
What is the first song in the album "Greatest Hits Volume Two" about?

Figure 1: An example of multiview identifiers for a passage. Corresponding to the query "Who is the singer of does he love you?", the semantic-related identifiers are highlighted in red.

Previous studies have explored several types of identifiers, such as titles of documents (De Cao et al., 2020), numeric IDs (Tay et al., 2022), and distinctive substrings (Bevilacqua et al., 2022). However, these identifiers are still limited: numeric IDs require extra memory steps and are ineffective in the large-scale corpus, while titles and substrings are only pieces of passages and thus lack contextualized information. More importantly, a

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6636–6648
July 9-14, 2023 ©2023 Association for Computational Linguistics

passage should answer potential queries from different views, but one type of identifier only represents a passage from one perspective.

In this work, we argue that generative retrieval could be improved in the following ways:

(1) Synthetic identifiers. To address the limitations of titles and substrings in providing contextual information, we propose to create synthetic identifiers that are generated based on a passage's content. In practice, we find the pseudo-queries, that are generated upon multiple segments of a passage, could serve as effective synthetic identifiers. For example, as shown in Figure 1, the pseudo-query "What is the first song in the album Greatest Hits Volume Two about?" spans multiple sentences in the passage. Once a query could be rephrased into a potentially-asked pseudo-query, the target passage could be effectively retrieved.

(2) Multiview identifiers. We believe that a single type of identifier is not sufficient to effectively represent a passage. Using multiple types of identifiers, such as titles, substrings, and synthetic identifiers, can provide complementary information from different views. (i) One type of identifier, like the title, may be unavailable in some scenarios. In this case, synthetic identifiers could alternatively work. (ii) Different views of identifiers are better suited for different types of queries. Titles could respond to general queries, while substrings are more effective for detailed ones. And the synthetic identifiers could cover some complex and difficult queries that require multiple segments. (iii) For one specific query, passages could be scored and ranked holistically from different views.

Based on the above insights, we propose the Multiview Identifiers eNhanceD gEnerative Retrieval approach, MINDER, as illustrated in Figure 2. To represent a passage, we assign three views of identifiers: the title, substring, and synthetic identifiers (pseudo-queries). MINDER takes a query text and an identifier prefix indicating the type of identifier to be generated as input, and produces the corresponding identifier text as output. Passages are ranked based on their coverage with the predicted three views of identifiers. We evaluate MINDER on three public datasets, and the experimental results show MINDER achieves the best performance among the current generative retrieval methods.

The key contributions are summarized:

- We are the first to propose synthetic identifiers (generated based on the passage's content) to integrate contextualized information. In practice, we find pseudo-queries could serve as effective synthetic identifiers.
- This is the first work that considers multiple views of identifiers simultaneously. Passages could be ranked holistically from different perspectives.
- Our approach achieves state-of-the-art performance in generative retrieval on three widely-used datasets.

2 Related Work

2.1 Generative Retrieval

Recently, we have witnessed an explosive development in autoregressive language models, such as the GPT-3/3.5 series (Brown et al., 2020; Ouyang et al., 2022). This motivates the generative approach to retrieve passages. In some retrieval scenarios, like entity retrieval and sentence retrieval, the entire items could be regarded as identifiers. De Cao et al. (2020) proposed GENRE (Generative ENtity REtrieval), which retrieves an entity by generating the entity text itself. GENRE also could be applied in page-level retrieval, where each document contains a unique title as the identifier. Lee et al. (2022b) introduced generative retrieval to the multi-hop setting, and the retrieved items are short sentences. In 2022, Tay et al. (2022) proposed the DSI (Differentiable Search Index) method, which takes numeric IDs as identifiers for documents. Wang et al. (2022b) later improved the DSI by generating more queries as extra training data. However, the numeric Ids-based methods usually were evaluated on the small NQ320K datasets, partially because they suffer from the large scaling problem. Bevilacqua et al. (2022) proposed SEAL, which takes substrings as identifiers. The retrieval process is effectively completed upon the FM-Index structure. In this work, we mainly improve the SEAL method via synthetic identifiers and multiview identifiers. This is the first work that takes pseudo-queries as identifiers and considers multiple kinds of identifiers.

2.2 Query Generation in Text Retrieval

Query generation is originally introduced to the IR community to improve the traditional term-based methods. Nogueira et al. (2019) showed that appending the T5-generated queries to the document

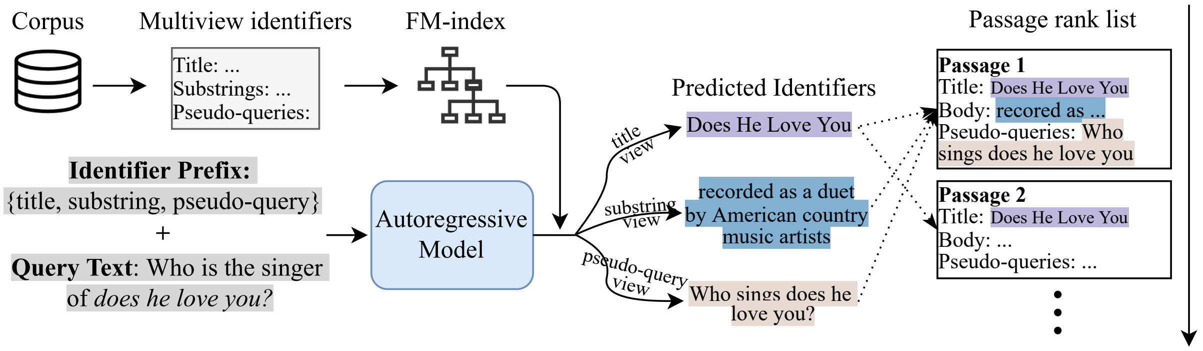


Figure 2: Illustration of our proposed MINDER method. MINDER adopts multiview identifiers: the title, substrings, and pseudo-queries. For a query with different identifier prefixes, MINDER generates corresponding identifiers in different views. Passages are ranked holistically according to the coverage with these generated identifiers.

before building the inverted index can bring substantial improvements over BM25. More recently, Mallia et al. (2021) used generated queries as term expansion to learn better sparse representations for documents. In the context of dense retrieval, the generated pseudo-queries were used as extra data to improve the training process of dense retrieval. For example, Ma et al. (2020) aimed to generate synthetic queries on the target domain for model training. Dai et al. (2022) achieved excellent performance in few-shot retrieval with prompt enhanced query generation. In generative retrieval, Wang et al. (2022b) also explored the use of pseudo-queries as extra data to train DSI. In this paper, we are the first to use pseudo-queries as one view of identifiers for generative retrieval.

2.3 Dense Retrieval

In recent years, text retrieval has witnessed a paradigm shift from traditional BM25-based inverted index retrieval to neural dense retrieval (Lee et al., 2019; Karpukhin et al., 2020; Li et al., 2022). Dense retrieval is further developed via hard negative sample mining (Xiong et al., 2020; Qu et al., 2021) and better pre-training design (Chang et al., 2019; Wang et al., 2022a), and has achieved excellent performance. Zhang et al. (2022) argued that a single vector representation of a document is hard to match with multi-view queries and proposed the multi-view document representation vectors. This is similar to our work, but we focus on using multi-view identifiers to improve generative retrieval.

Compared to dense retrieval that relies on the dual-encoder architecture, generative retrieval is promising to overcome the missing fine-grained interaction problem via the encoder-decoder paradigm. However, as a recently proposed technique route, generative retrieval still lags behind the state-of-the-art dense retrieval method and leaves much scope to investigate.

3 Method

Given a query text q, the retrieval system is required to retrieve a list of passages $p_1, p_2, \dots, p_n$, from a corpus $\mathcal{C}$. Both queries and passages are a sequence of text tokens. Besides, there are k relevant query-passage pairs $\{q_i, p_i\}^k$ for training, where $p_i \in \mathcal{C}$.

3.1 Multiview Identifiers

For all passages in the corpus $\mathcal{C}$, we assign them multiview identifiers, including the titles, substrings, and pseudo-queries. These different types of identifiers could represent a passage from different perspectives.

Title. A title is usually a very short string that indicates the subject of a passage. Titles have been verified as effective identifiers in page-level retrieval. We denote a title as t for a passage p and select it as one view of identifiers in our work.

Substrings. For a query, some substrings in the relevant passage are also semantically related. For example, for the query “Who is the singer of does he love you?” in Figure 1, the substring “recorded as a duet by” is corresponding to the “Who is the singer of” in the query. For implementation, we directly store the whole content of the passage, denoted as $\mathcal{S}$, and sample substrings from $\mathcal{S}$ for model training.

Pseudo-queries. In this work, we generate pseudo-queries for a passage as synthetic identifiers to augment the title and substrings. Since pseudo-queries are generated based on the content of the passages, these synthetic identifiers could integrate multiple segments and contextualized information. For example, as shown in Figure 1, the pseudo-query "What is the first song in the album

Greatest Hits Volume Two about?" covers multiple sentences in the passage.

We first use the labeled query-passage pairs $\{q_i, p_i\}^k$ to train a query generation model **QG**. And then we generate a set of queries with top-k sampling strategy to encourage the query generation diversity. For each passage p in corpus $\mathcal{C}$, we generate pseudo-queries $\mathcal{Q}$ as follows,

$$\mathcal{Q} = \mathbf{QG}(p). \tag{1}$$

As such, for each passage in $\mathcal{C}$, we have obtained three views of identifiers $\{t, \mathcal{S}, \mathcal{Q}\}$. These identifiers could well represent a passage's content from different views.

3.2 Model Training

We train an autoregressive language model (denoted as **AM**) like BART (Lewis et al., 2020) or T5 (Raffel et al., 2020) to generate corresponding identifiers using the standard sequence-to-sequence loss. The input is the query text along with an identifier prefix, and the target is the corresponding identifier of the relevant passage, formulated as:

$$identifier = \mathbf{AM}(prefix; q). \tag{2}$$

The $prefix$ text is "title", "substring", and "pseudo-query", for the three different views, respectively. For the title view, the target text is the title t of the relevant passage. For the substring view, we randomly select a substring s from $\mathcal{S}$ as the target text. And to guarantee the semantic relevance between the input and the target, we only keep those substrings with a high character overlap with the query. As for the query view, we randomly select a pseudo-query pq from $\mathcal{Q}$ as the target. Since both the user query q and the pseudo-query pq are conditioned on the same passage, they are usually about the same subject and even are different forms of the same question. The three different training samples are randomly shuffled to train the autoregressive model.

3.3 Model Inference

In this section, we detail how to retrieve passages using the trained autoregressive model, **AM**.

FM-index. MINDER requires a data structure that can support generating valid identifiers. Following the work (Bevilacqua et al., 2022), we use the FM-index (Ferragina and Manzini, 2000) to store all types of identifiers. For easy understanding, FM-index could be regarded as a special prefix tree that supports search from any position. Specifically, we flatten multiview identifiers into a sequence of tokens with special split tokens. For example, the identifiers of the passage in Figure 1 are flattened into "<TS> Does He Love You <TE> Does He Love You is a song written by Sandy Knox and Billy Stritch, and recorded as ..., <QS> Who wrote the song does he love you? <QE> <QS> Who sings does he love you? ...", where "<TS>, <TE>, <QS>, <QE>" are special tokens indicating the start and end of different types of identifiers. Given a start token or a string, FM-index could provide the list of possible token successors in $O(Vlog(V))$, where V is the vocabulary size. Therefore, we could force the **AM** model to generate valid identifiers.

Constrained generation. Upon the FM-index, MINDER could generate valid identifiers via constrained generation. For the title view, we input the prefix text "title" and query text into the **AM** model, and force it to generate from the token "<TS>". As such, MINDER could generate a set of valid titles via beam search, denoted as $\mathcal{T}_g$. For the substring view, the **AM** model receives the prefix "substring" and query as input, and generates substrings $\mathcal{S}_g$ via constrained beam search. Similarly, the **AM** model could generate valid pseudo-queries $\mathcal{Q}_g$ with the start token "<QS>" and end token "<QE>". We also save the language model scores for each generated text and utilize them in the following passage ranking stage. Notably, the language model score for a string is influenced by its length, which makes long strings, like pseudo-queries, have lower scores. Therefore, we add a biased score for the pseudo-query view to offset the influence.

Passage ranking. Previous generative retrieval methods (Tay et al., 2022; De Cao et al., 2020) could rank items directly using the constrained beam search, since their identifiers could map to passages one-to-one. Differently, MINDER considers multiview identifiers to rank passages comprehensively. To address this issue, we propose a novel scoring formulation that aggregates the contributions of multiview identifiers. Each passage's score is holistically computed according to its coverage with the predicted identifiers, $\mathcal{T}_g$, $\mathcal{S}_g$, and $\mathcal{Q}_g$.

We follow the work (Bevilacqua et al., 2022) to rank passages with the generated identifiers. For a passage p, we select a subset $\mathcal{I}_p$ from the predicted identifiers. One identifier $i_p \in \{\mathcal{T}_g, \mathcal{S}_g, \text{and } \mathcal{Q}_g\}$ is selected if i_p occurs at least once in the identifiers of passage p. To avoid repeated scoring of substrings, we only consider once for substrings that overlapped with others. Finally, the rank score of the passage p corresponding to the query q is formulated as the sum of the scores of its covered identifiers,

$$s(q,p) = \sum_{i_p \in \mathcal{I}_p} s_{i_p}, \tag{3}$$

where s_{i_p} is the language model score of the identifier i_p.

According to the rank score $s(q,p)$, we could obtain a rank list of passages from the corpus $\mathcal{C}$. In practice, we could use the FM-index to conveniently find those passages that contain at least one predicted identifier rather than score all of the passages in the corpus.

4 Experiments

4.1 Datasets

We conducted experiments on widely-used NQ (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) datasets with the DPR (Karpukhin et al., 2020) setting. NQ and TriviaQA are open-domain QA datasets, where the queries are natural language questions and the passages are from Wikipedia. Each page in Wikipedia is chunked into several passages with no more than 100 words. Therefore, several passages may share the same Wikipedia title. Besides, we also evaluated generative retrieval methods on the MSMARCO dataset (Nguyen et al., 2016). MSMARCO is sourced from the Web search scenario, where queries are web search queries and passages are from Web pages.

4.2 Baselines

We compared MINDER with the generative retrieval methods, DSI (Tay et al., 2022) and SEAL (Bevilacqua et al., 2022). GENRE (De Cao et al., 2020) was excluded because it relies on unique titles of documents and thus cannot perform passage-level retrieval. Besides, we also included the term-based method, BM25, DPR (Karpukhin et al., 2020), and GAR (Mao et al., 2021) for comparison. Most of the results of baselines are from their paper, and the rest are reproduced by using publicly released code.

4.3 Implementation Details

For a fair comparison with previous work (Bevilacqua et al., 2022), we utilized the BART-large as the backbone. We finetuned the model using training samples, title, substrings, and pseudo-queries, with the portion of 3:10:5. Inspired by SEAL that exposes the model to more possible pieces of evidence, we also add some "unsupervised" examples to the training set. In each of these examples, the model takes as input a random pseudo-query and generates the corresponding passage's identifiers. We discuss its influence in Section 4.7. Lewis et al. have generated pseudo-queries for half of the passages on Wikipedia. Therefore, we generate queries for another half of the passages on Wikipedia. And for the MSMARCO corpus, we take the pseudo-queries from the work (Nogueira et al., 2019).

We trained MINDER with the fairseq[1] framework. We adopted the Adam optimizer with a learning rate of 3e-5, warming up for 500 updates, and training for 800k total updates. Detailed training hyperparameters are illustrated in Appendix A for better reproduction. The experiments are conducted on 8×32GB NVIDIA V100 GPUs.

4.4 Retrieval Results on QA

The retrieval performance on NQ and TriviaQA is summarized in Table 1. By jointly analyzing the results, we gained the following findings.

(1) Among the generative retrieval methods, MINDER achieves the best performance. We found that SEAL which takes natural identifiers surpasses DSI based on numeric identifiers. This is because numeric identifiers lack semantic information and DSI requires the model to memorize the mapping from passages to their numeric IDs. As such, it becomes more challenging for DSI on the NQ and TriviaQA datasets with more than 20 million passages. Despite the superiority of SEAL, MINDER still outperforms it. Specifically, the improvements in terms of hits@5 are 4.5% and 1.6% on NQ and TriviaQA, respectively. This verifies the effectiveness of our proposed multiview identifiers, which could rank passages from different perspectives.

(2) On NQ, MINDER achieves the best performance in terms of hits@100 and the second-best results in terms of hits@5, 20. However, generative retrieval methods, including MINDER, perform worse than dual-encoder approaches on TriviaQA. Generative retrieval methods rely on the identifiers to represent passages, and cannot "see" the content of the passage. Although the QG module in

[1] https://github.com/facebookresearch/fairseq.

Methods	Natural Questions			TriviaQA		
	@5	@20	@100	@5	@20	@100
BM25	43.6	62.9	78.1	67.7	77.3	83.9
DPR(Karpukhin et al., 2020)	**68.3**	**80.1**	86.1	72.7	80.2	84.8
GAR(Mao et al., 2021)	59.3	73.9	85.0	**73.1**	**80.4**	**85.7**
DSI-BART(Tay et al., 2022)	28.3	47.3	65.5	-	-	-
SEAL-LM(Bevilacqua et al., 2022)	40.5	60.2	73.1	39.6	57.5	80.1
SEAL-LM+FM(Bevilacqua et al., 2022)	43.9	65.8	81.1	38.4	56.6	80.1
SEAL(Bevilacqua et al., 2022)	61.3	76.2	86.3	66.8	77.6	84.6
MINDER	65.8†	78.3†	**86.7**†	68.4†	78.1†	84.8†

Table 1: Retrieval performance on NQ and TriviaQA. We use hits@5, @20, and @100, to evaluate the retrieval performance. Inapplicable results are marked by "-". The best results in each group are marked in Bold, while the second-best ones are underlined. † **denotes the best result in generative retrieval**.

Methods	MSMARCO			
	R@5	R@20	R@100	M@10
BM25	28.6	47.5	66.2	18.4
SEAL	19.8	35.3	57.2	12.7
MINDER	**29.5**	**53.5**	**78.7**	**18.6**
only pseudo-query	24.9	48.9	72.5	15.5
only substring	18.7	38.7	64.9	11.5
only title	9.8	19.3	30.1	5.5

Table 2: Retrieval performance on the MSMARCO dataset. R and M denote Recall and MRR, respectively. SEAL and MINDER are trained only with labeled query-passage pairs.

Methods	Natural Questions		
	@5	@20	@100
only query	59.0	72.5	80.9
only substring	60.2	74.3	84.5
only title	60.4	74.9	84.1
w/o pseudo-query	63.4	77.2	86.1
w/o substring	63.1	77.0	85.0
w/o title	63.9	76.6	85.3
MINDER	65.8	78.3	86.7

Table 3: Ablation study on different views of identifiers. We use "w/o query", "w/o substrings", and "w/o title" to respectively denote new models without considering the query flow, substrings, and title as identifiers. We also evaluate MINDER with only one view of the identifier.

our work generates pseudo-queries based on a passage's content, the autoregressive language model AM still cannot directly "see" the original content of the passage. Besides, autoregressive generation has the error accumulation problem. These are the disadvantages of generative retrieval and why it may not perform as well as dense retrievers in some scenarios.

4.5 Retrieval Results on Web Search

Previous generative retrieval works (Tay et al., 2022; Bevilacqua et al., 2022) only verified the effectiveness on open-domain QA datasets, like NQ320k and NQ, but did not evaluate under the Web search scenario. To deeply analyze generative retrieval, we conducted experiments on the MSMARCO dataset and reported the results in Table 2. Notably, we tried to implement DSI on MSMARCO but achieved poor performance. This may be due to the large-scaling problem of DSI, which requires a huge amount of GPU resources to work on a large-scale corpus.

By analyzing the results in Table 2, we found: 1) Different from the results on the QA datasets, SEAL performs worse than BM25 under the Web search scenario. Queries in Web search may only contain several keywords, which makes it hard for SEAL to learn the semantic correlation between queries and the substrings of passages. 2) MINDER surpasses SEAL and achieves a bigger performance improvement compared with the results on the QA datasets. This benefits from the multiview identifiers, which improve MINDER's robustness under various scenarios. 3) MINDER outperforms BM25, particularly in terms of Recall@100. MINDER could recall passages from three different views, and thus achieves a better performance in Recall@100 than Recall@5.

4.6 Ablation Study

MINDER considers multiple types of identifiers: titles, substrings, and pseudo-queries. 1) Do the three views of identifiers all contribute to MINDER? 2) how much help does MINDER gain from

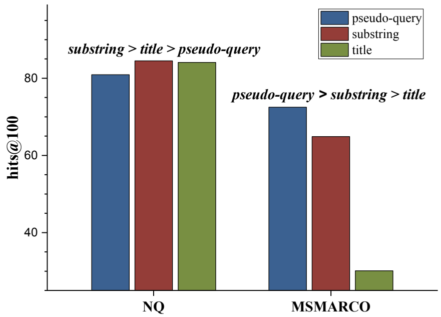


Figure 3: Illustrating the roles of various identifier views in different search scenarios.

Methods	Unsupervised data	Natural Questions		
		@5	@20	@100
SEAL	✗	58.9	74.8	85.4
SEAL	span as queries	61.3	76.2	86.3
SEAL	pseudo-queries	61.2	76.8	85.7
MINDER	✗	64.6	76.8	86.4
MINDER	span as queries	65.9	78.3	86.7
MINDER	pseudo-queries	65.8	78.3	86.7

Table 4: Retrieval performance with different unsupervised data. "span as queries" and "pseudo-queries" means taking a span from the passage or a pseudo-query as the input, respectively.

Methods	Natural Questions		
	@5	@20	@100
MINDER+ID view	64.6	77.1	86.1
MINDER	64.6	76.8	86.4

Table 5: Evaluation of numeric identifiers as one view identifier in MINDER. Both two variants are trained only with labeled query-passage pairs.

	BS	@5	@20	@100
TriviaQA	5	66.9	77.1	83.8
	10	67.8	77.9	84.6
	15	68.4	78.1	84.8
	20	68.4	78.4	84.8
MS MARCO	5	29.4	52.9	78.4
	10	29.4	53.9	79.3
	15	29.1	53.7	79.6
	20	27.8	52.8	79.8

Table 6: Retrieval performance of MINDER with beam size values in {5, 10, 15, 20}.

the three different identifiers? 3) Is there any difference among different datasets? To answer these questions, we conducted experiments by eliminating one type of identifier each time. The results are illustrated in Table 2 and Table 3. To better demonstrate the functions of different views on different datasets, we kept only one view identifier and reported results in Figure 3.

From the results, we gained the following insights. (1) No matter which view of identifiers is removed from MINDER, the performance significantly declines. In terms of hits@5, the decline is 2.4%, 2.7%, and 1.9%, while eliminating the pseudo-query view, substring view, and title view, respectively. This clearly reveals that all three views of identifiers contribute to the system's performance, and verifies the necessity to adopt multiview identifiers simultaneously. (2) Besides, comparing the three types of identifiers, we found that eliminating the substring view degrades the most on NQ. This may be due to the fact that the substrings could cover the most content of a passage. Although the "only title" and "only pseudo-query" variants perform worse than the substring view, they could complement each other and significantly improve the overall performance. 3) Comparing the results on NQ and MSMARCO, we found different views played different roles in different search scenarios. As illustrated in Figure 3, the substring view is vital on NQ while the pseudo-view contributes the most on MSMARCO. This is determined by the different natures between the QA and Web search scenarios. And it verifies the necessity to adopt multiview identifiers again.

4.7 In-depth Analysis

Unsupervised Data. Besides the labeled query-passage pairs, we also trained MINDER using pseudo-queries. SEAL conducted unsupervised data by randomly selecting a span from a passage as the input. (1) Are the unsupervised data useful for the training? (2) Which kinds of unsupervised data contribute most? We conducted experiments by using different kinds of unsupervised data, and the results are illustrated in Table 4. We found that both kinds of unsupervised data improve upon purely supervised training. Specifically, the performance gets improved by 2.3 and 1.2 points in terms of hits@5 for SEAL and MINDER respectively. There is no significant gap between the two kinds of unsupervised data. We think the unsupervised training mainly exposes passages to the model, and both two ways could meet this goal.

Numeric Identifiers. MINDER adopts multiview identifiers, including titles, substrings, and pseudo-queries, which are all semantic text. We excluded numeric identifiers in MINDER, because

Query	Predicted Identifiers	Relevant Passages
Question on NQ: Who got the first nobel prize in physics?	**Title view** 1. Alfred Nobel 2. Ernest Rutherford 3. Alfred Marshall	**Title:** Nobel Prize in Physics **Body:** Nobel Prize in Physics is a yearly award given by the Royal Swedish Academy of Sciences for those who have made the most outstanding contributions for mankind in the field of physics. It is one of the five Nobel Prizes established by the will of Alfred Nobel in 1895 and awarded since 1901; the others being the Nobel Prize in Chemistry, Nobel Prize in Literature, Nobel Peace Prize, and Nobel Prize in Physiology or Medicine. The **first Nobel Prize in Phys**ics was awarded to physicist Wilhelm Conrad Rntgen in recognition of the extraordinary services he **Pseudo-queries:** || who founded the nobel peace prize || who founded the nobel peace prize in 1901 || how many nobel prizes are there || **who won the first nobel prize for physics** || in which year was the nobel prize for physics established || in which year was the first nobel prize for physics awarded || what is the name of the nobel prize for physics || **who won the first nobel prize in physics** || who founded the nobel prize for physics || when was the nobel prize for physics established || **when was the first nobel prize for physics awarded** || in which year was the nobel prize for physics ......
	Substring view 1. first Nobel Prize in Phys 2. first Nobel Prize in Physiology 3. first Nobel Prize in Physiology or	
	Pseudo-query view 1. who won the first nobel prize for physics 2. who won the first nobel prize in physics 3.when was the first nobel prize for physics awarded	
Query on MSMARCO: Androgen receptor define	**Title view** 1. Androgen receptor 2. Definitions &Translations 3. difference between a gene and an allele?	**Title:** **Androgen receptor** **Body:** The **androgen receptor (AR**), also known as NR3C4 (nuclear receptor subfamily 3, group C, member 4), is a type of nuclear receptor that is activated by binding either of the androgenic hormones, testosterone, or dihydrotestosterone in the cytoplasm and then translocating into the nucleus.n some cell types, testosterone interacts directly with **androgen receptors,** whereas, in others, testosterone is converted by 5-alpha-reductase to dihydrotestosterone, an even more potent agonist for androgen receptor activation. **Pseudo-queries:** || what kind of androgen does a receptor || **androgen receptors definition** || what is ar receptor || what is androgen receptor || where is nr3c4 receptor || is testosterone a nuclear receptor || what types of receptors do a nr3c4 receptor have || what is ar receptor || **what is the function of androgen receptors** || what kind of receptor for testosterone|| what is androgen receptor || what type of androgen receptors activate testosterone || what is the name of the androgen receptor ......
	Substring view 1. androgen receptor 2. androgen receptors 3. androgen receptor (AR	
	Pseudo-query view 1. androgen receptor definition 2. what is the function of androgen receptors 3. what is the function of androgen receptor	

Figure 4: Case study. Two cases from NQ and MSMARCO. For the predicted identifiers from MINDER, we show three top-scored predictions for the title view, body view, and pseudo-query view, respectively. The predicted identifiers that occur in relevant passages are colored in red.

IDs are numbers and lack semantic information. As such, numeric identifiers require extra steps to memorize the mapping from passages to IDs. For exploration, we also added the ID view in MINDER and reported the results in Table 5. It is observed that there is no big difference in performance after including numeric identifiers. On the one hand, numeric identifiers are weak at large-scale corpus. Therefore, the ID view cannot contribute to MINDER on the NQ dataset. On the other hand, numeric identifiers fail to provide extra information to complement the three views identifiers in MINDER.

Beam Size. MINDER relies on beam search to predict a set of identifiers, and then these predicted identifiers are mapped as ranked passages. To evaluate the influence of beam size, we conducted experiments and reported results in Table 6. The results suggest that a bigger beam size, like 15 or 20, could achieve a better performance in terms of hits@100 on both two datasets. As for the top-ranked evaluation, TriviaQA prefers a bigger beam size, but MSMARCO requires a smaller one. One possible reason is that there are too many similar passages on MSMARCO and a bigger beam size introduces more noise.

Inference speed. On our equipment, MINDER takes about 135 minutes to complete the inference process on the NQ test set, while SEAL takes about 115 minutes. Both of them apply the same beam size of 15. MINDER requires 1.2 times more inference time than SEAL on our equipment, due to the increased identifier views.

4.8 Case Study

To qualitatively illustrate why MINDER works, we analyzed the prediction results on NQ and MSMARCO in Figure 4. (1) It is observed that pseudo-queries are sufficient and could cover almost potential queries. In the first example, given the question "Who got the first nobel prize in physics?", MINDER generates either the same meaning question "who won the first nobel prize for physics" or another question about the same subject "when was the first novel prize for physics award". These predicted queries accurately locate the relevant passage. (2) As for the substring view, MINDER tends to generate almost the same ones. These substrings are not much distinctive and could be found in several passages of the corpus. This may be the reason why the substring view cannot work well on MSMARCO.

5 Conclusion and Future Work

In this work, we present MINDER, a novel retrieval system that combines an autoregressive language model with multiview identifiers. We find pseudo-queries are admirable identifiers that could work on different search scenarios. More importantly, MINDER simultaneously utilizes multiple types of identifiers, including titles, substrings, and pseudo-queries. These different views of identifiers could complement each other, which makes MINDER effective and robust in different search scenarios. The experiments on three widely-used datasets illustrate MINDER achieves the best performance in generative retrieval.

In the future, we aim to improve MINDER from the following aspects.MINDER adopts a heuristic function to aggregate predicted identifiers and rank passages. The heuristic rank function relies on manual hyper-parameters to balance different views of identifiers, which may not be suitable for all samples. As such, we are motivated to integrate the rank process into an auto-learned neural network. Besides, we plan to apply MINDER on more search domains, like the few-shot retrieval setting.

Acknowledgments

The work described in this paper was supported by Research Grants Council of Hong Kong(PolyU/5210919, PolyU/15207821, and PolyU/15207122), National Natural Science Foundation of China (62076212) and PolyU internal grants (ZVQ0).

Limitations

MINDER achieves the best performance among the current generative retrieval methods, but it is still not as good as the well-designed dual-encoder approaches and lags behind the current state-of-the-art on leaderboards. The reason for this is that the model's autoregressive generation way (generating from left to right) prevents it from "seeing" the entire content of a passage. Generative retrieval methods have advantages over dual-encoder approaches but also leave many research problems to be investigated. Another limitation of MINDER is the memory consumption of identifiers. Since MINDER considers multiview identifiers, it also consumes more memory to store these identifiers. Fortunately, we use the FM-index structure to process the identifiers, and the space requirements are linear in the size of the identifiers.

Ethics Statement

The datasets used in our experiment are publicly released and labeled through interaction with humans in English. In this process, user privacy is protected, and no personal information is contained in the dataset. The scientific artifacts that we used are available for research with permissive licenses. And the use of these artifacts in this paper is consistent with their intended use. Therefore, we believe that our research work meets the ethics of ACL.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *arXiv preprint arXiv:2204.10628*.

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*, volume 33, pages 1877–1901.

Wei-Cheng Chang, X Yu Felix, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2019. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1870–1879.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

P. Ferragina and G. Manzini. 2000. Opportunistic data structures with applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 390–398.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

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 International Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781. ACL.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Hyunji Lee, Jaeyoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vlad Karpukhin, Yi Lu, and Minjoon Seo. 2022a. Contextualized generative retrieval. *arXiv preprint arXiv:2210.02068*.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022b. Generative multi-hop retrieval. *arXiv preprint arXiv:2204.13596*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096. ACL.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Yongqi Li, Wenjie Li, and Liqiang Nie. 2022. Dynamic graph reasoning for conversational open-domain question answering. *ACM Transactions on Information Systems*, 40(4):1–24.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. *arXiv preprint arXiv:2004.14503*.

Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the International Conference on Research and Development in Information Retrieval*, pages 1723–1727.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docttttquery. *Online preprint*, 6.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 5835–5847.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Yi Tay, Vinh Q 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. *arXiv preprint arXiv:2202.06991*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022a. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *arXiv preprint arXiv:2207.02578*.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, et al. 2022b. A neural corpus indexer for document retrieval. *arXiv preprint arXiv:2206.02743*.

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

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000.

A Training Hyperparameters

Name	Value
arch	bart_large
task	translation
criterion	label_smoothed_cross_entropy
weight-decay	0.01
optimizer	adam
lr-scheduler	polynomial_decay
lr	3e-05
total-num-update	800000
patience	5

Table 7: Hyperparameters to train MINDER using the fairseq.

For better reproduction, we detail the training hyperparameters in Table 7. We train our model for serval runs with the fairseq, and the results of the different runs are reported in Table 8.

# Run	Natural Questions		
	@5	@20	@100
1	66.2	78.6	86.9
2	66.2	78.6	86.9
3	65.8	78.3	86.7
4	64.8	78.6	86.7

Table 8: Results of MINDER on NQ for different runs.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☑ A4. Have you used AI writing assistants when working on this paper?
Assistance purely with the language of the paper by using Grammarly

B ☑ Did you use or create scientific artifacts?

Section 4.3

☑ B1. Did you cite the creators of artifacts you used?
Section 4.3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4.3 and Section Ethics Statement

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section Ethics Statement

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section Ethics Statement

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.3 and Section A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.3 and Section A

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.3 and Section A

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4.3 and Section Ethics Statement

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.