Understanding Differential Search Index for Text Retrieval

Xiaoyang Chen[1,2], **Yanjiang Liu**[1,2], **Ben He**[1,2*], **Le Sun**[2*], **Yingfei Sun**[1*]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]Institute of Software, Chinese Academy of Sciences, Beijing, China
{chenxiaoyang19, liuyanjiang22}@mails.ucas.ac.cn,
{benhe, yfsun}@ucas.ac.cn
sunle@iscas.ac.cn

Abstract

The Differentiable Search Index (DSI) is a novel information retrieval (IR) framework that utilizes a differentiable function to generate a sorted list of document identifiers in response to a given query. However, due to the black-box nature of the end-to-end neural architecture, it remains to be understood to what extent DSI possesses the basic indexing and retrieval abilities. To mitigate this gap, in this study, we define and examine three important abilities that a functioning IR framework should possess, namely, exclusivity, completeness, and relevance ordering. Our analytical experimentation shows that while DSI demonstrates proficiency in memorizing the unidirectional mapping from pseudo queries to document identifiers, it falls short in distinguishing relevant documents from random ones, thereby negatively impacting its retrieval effectiveness. To address this issue, we propose a multi-task distillation approach to enhance the retrieval quality without altering the structure of the model and successfully endow it with improved indexing abilities. Through experiments conducted on various datasets, we demonstrate that our proposed method outperforms previous DSI baselines[1].

1 Introduction

Recent advancements in the field of information retrieval (IR) have sparked a growing interest in Differentiable Search Index (DSI) (Tay et al., 2022). Unlike traditional methods, which involve building an index before retrieval (Dai and Callan, 2019; Nogueira et al., 2019a; Lin et al., 2020; Xiong et al., 2021), DSI and related techniques such as DSI-QG (Zhuang et al., 2022) and NCI (Wang et al., 2022) do not rely on external indexes to store data. Instead, these methods map user queries directly to the identifiers (IDs) of the relevant documents, providing a simpler and more efficient retrieval process. This novel autoregressive approach, represented by DSI, has expanded the potential IR applications due to its ease of use, minimal index storage requirements, and end-to-end retrievability.

However, despite the novel retrieval mechanism of DSI, current DSI models still rely on relevance signals of query-passage pairs for training. These models, which map short texts to specific document IDs, do not have an explicit interaction between the query and the document during retrieval, unlike dense retrieval models (Khattab and Zaharia, 2020; Hofstätter et al., 2021; Qu et al., 2021; Lin et al., 2021c; Karpukhin et al., 2020; Gao and Callan, 2021) and cross-attention rerankers (Nogueira et al., 2019b; Nogueira and Cho, 2019; Zheng et al., 2020; Li et al., 2020; Wang et al., 2020; Chen et al., 2022b). This training approach and the inherent properties of the model may lead to two problems. First, due to the lack of explicit modeling of inter-document associations and an explicit query-document relevance measurement, the model may only learn a unidirectional mapping from short texts to specific IDs, without understanding how the document is relevant to the query, leading to somewhat random output in the ranking list. Second, to reduce computational complexity, DSI models often simply represent a document by a small number of tokens or pseudo-queries. However, this approach may result in a reduced capacity to differentiate between documents and capture crucial relevant information.

This study aims to deepen the understanding of DSI by evaluating its suitability as an end-to-end model for indexing and retrieval. To achieve this, DSI-QG (Zhuang et al., 2022), a recent enhancement of DSI using pseudo queries for the model training, is used as a representative model for analysis. Our argument is that a usable index of a non-boolean retrieval model should meet the following three conditions: 1) The document content in the index should have a one-to-one corre-

[1]The code and data for this work can be found at https://github.com/VerdureChen/Understang_DSI

Findings of the Association for Computational Linguistics: ACL 2023, pages 10701–10717
July 9-14, 2023

spondence with the ID to ensure the stability of retrieval results; 2) The key information of the documents should be stored in the index as completely as possible to avoid the loss of information related to the query and thus affecting the retrieval results; 3) The model should be able to output documents in decreasing order of their relevance to the query. These three abilities are summarized as **exclusivity**, **completeness**, and **relevance ordering**. Our analytical experiments indicate that the currently available DSI models do not fully meet the requirements of a general end-to-end indexing-retrieval model, which limits their conditions of use and significantly reduces their effectiveness, especially when compared to the state-of-the-art dense retrieval models, which, in contrast, is shown to better meet those requirements.

To this end, we investigate whether DSI models can be better trained to improve retrieval abilities while maintaining their simple structure and low storage cost. Specifically, we propose utilizing a dense retrieval approach to provide effective supervision signals for training DSI models. To enhance *exclusivity* and *completeness* of DSI, we propose to improve the document representation to capture information from different granularities and filter key information using the document representation encoded by the dense retrieval model. To improve the ability to discriminate the relevance degree of different documents of DSI, we propose a new distillation-based training approach. By explicitly modeling the connections between documents, the model is able to reduce the randomness of the output results and improve retrieval performance, especially on datasets with deep pool annotations.

Major contributions of this paper are tri-fold. 1) An empirical analysis of basic IR abilities indicates the potential weaknesses of existing DSI approaches. 2) Based on the analysis above, we propose a multi-task distillation approach to improve the effectiveness of DSI by learning from dense retrieval while keeping its advantages. 3) Further evaluation shows that our approach substantially improves retrieval effectiveness of DSI-QG.

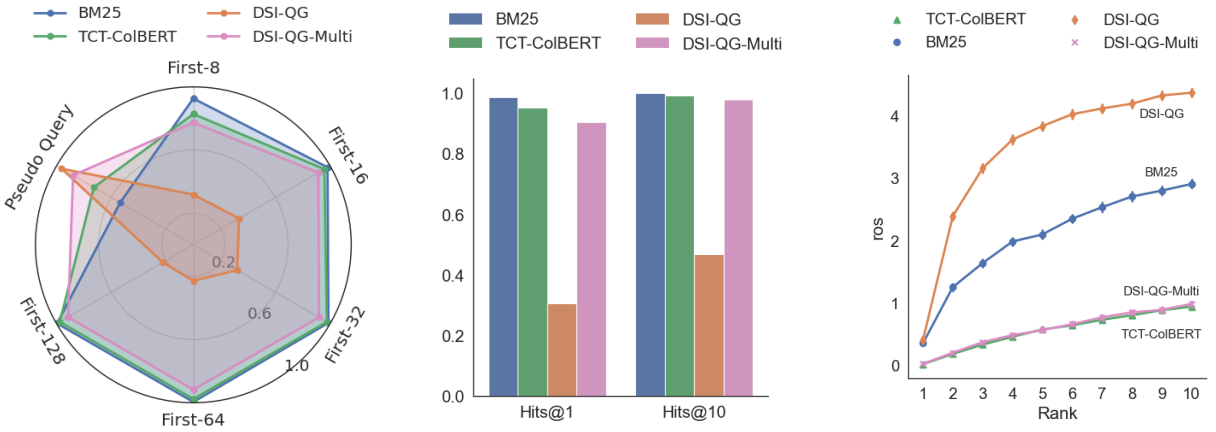
2 Empirical Analysis

In this section, we conduct an empirical analysis to examine to what extent the DSI framework satisfies the basic requirements of a functioning IR model. Specifically, we summarize *exclusivity*, *completeness*, and *relevance ordering*, as three essential abilities required for an IR framework, as defined below. While acknowledging that there are certain retrieval problems that do not require an ordered list, it is crucial to emphasize that our research specifically focuses on ranked retrieval, which inherently involves non-boolean ranking. The notions used throughout this paper are listed in Table 3 in Appendix A.1.

2.1 Definitions

Exclusivity refers to the uniqueness of documents in an IR system, i.e., the one-to-one correspondence between document content and its identifier, which determines the extent to which a retrieval framework can distinguish different documents in a collection. Although it is typical for document content and identifiers to have a one-to-one mapping, it is important to note that certain collections may contain duplicate documents, particularly in real-world scenarios. While exclusivity is primarily intended for stable experimental evaluations and reproducibility, it is not an absolute requirement for an index and is contingent upon the specific document collection. For the sake of simplicity and considering the majority of cases, we assume that the documents utilized in our experiments are not duplicated. In actual implementation, DSI is only trained for a one-way mapping (i.e., from document representation to document identifier) (Tay et al., 2022; Zhuang et al., 2022; Wang et al., 2022; Zhou et al., 2022). Therefore, in our analysis, we examine the injective relationship for various models by testing the case of a unidirectional mapping from a document to its identifier.

Completeness is the ability of an index to retrieve all eligible documents in a collection based on a specific query or search intent. This means that if a document has relevant content and meets the search criteria, then that document should appear in the search results. To obtain more comprehensive search results, the index structure needs to store as complete document information as possible (e.g., sparse search indexes) or to maximize the identification and memory of more valuable key document information. In this context, the completeness of an index is narrowly defined as its ability to store essential information contained within a document that can be used to respond to a variety of queries, taking into consideration the various differences in index structures and practical application requirements.



(a) Exclusivity (b) Completeness (c) Relevance ordering

Figure 1: Results of empirical analysis. (a) Hits@1 by querying different retrieval models with the first-k tokens or a random pseudo query generated from the passage. **Exclusivity** shows the ability to retrieve a document by using its own content as a query. (b) Hits@1 and Hits@10 by querying different retrieval models with the key fragments within individual documents. **Completeness** shows the ability to retrieve a document by using its key content as a query. (c) The relevance ordering scores (y-axis) at different positions of the retrieval results (x-axis) for different retrieval methods. A higher **relevance ordering** score indicates a lower ability of the model to distinguish relevant documents from random ones. As DSI-QG-Multi (proposed in Section 3.1.3) is trained using TCT-ColBERT's output as supervision signals, their curves (the bottom two) are almost indistinguishable.

Relevance Ordering is an essential ability of non-boolean IR models which outputs a sorted list of documents in decreasing order of relevance. For two documents d_i and d_j that both appear in the result list of a one-time retrieval operation $\mathrm{R}(q_t, \mathrm{M})$ on query q_t based on a certain retrieval paradigm M, relevance ordering can be described as:

$$d_j \preceq_{q_t} d_i, \text{ if } \mathrm{s}(q_t, d_j) \leq \mathrm{s}(q_t, d_i), \text{ for } \forall d_i, d_j \in R(q_t, \mathrm{M}) \quad (1)$$

where $d_j \preceq_{q_t} d_i$ means a binary relation that d_i is more relevant to q_t than d_j, and $\mathrm{s}(q_t, d_*)$ is the predicted relevance score of d_*, which is in the top-k recall list $(id^1, id^2, ..., id^k)$ given by $\mathrm{R}(q_t, \mathrm{M})$.

For d_i in the top-k recall list of $\mathrm{R}(q_t, \mathrm{M})$ and d_o in the index pool but not in the top-k list, relevance ordering can be given as:

$$d_o \preceq_{q_t} d_i, \text{ for } \forall d_i \in R(q_t, \mathrm{M}) \text{ and } \forall d_o \in \mathcal{D} \setminus R(q_t, \mathrm{M}) \quad (2)$$

where $\mathcal{D} \setminus R(q_t, \mathrm{M})$ is the absolute complement of the set $R(q_t, \mathrm{M})$ in the indexing set $\mathcal{D}$.

2.2 Analysis Methodology

To assess the **exclusivity** of various models, we query different models with a cut-off of the first fragment from the original document, or pseudo-queries derived from the original documents, and assess their ability to retrieve the corresponding documents as the highest-ranked retrieval outcomes. Following the DSI-QG setup (Zhuang et al., 2022), a series of experiments are conducted utilizing the training set of MS MARCO 100k (Nguyen et al., 2016; Zhuang et al., 2022; Zhou et al., 2022) to evaluate the exclusivity of various models. MS MARCO 100k is a subset of the MS MARCO passage dataset, which has been used in quite a few recent studies on DSI (Zhuang et al., 2022; Zhou et al., 2022). The length of first-k cut-off from the original document is varied within 16, 32, 64, 128. The pseudo queries are generated by docT5query (Nogueira et al., 2019a), based on the full text of the passage and followed the same procedures as the DSI-QG training data generation process (Zhuang et al., 2022).

To measure the **completeness** of information stored in an index, we randomly select 10k documents from the training set of MS MARCO 100k (Nguyen et al., 2016; Zhuang et al., 2022; Zhou et al., 2022) and their corresponding queries. Assuming that the BERT reranker with cross-attention (Nogueira and Cho, 2019) behaves in a manner that closely approximates the ranking preferences of human annotators for documents, we employ a strategy to identify the most pertinent segments within each document. To achieve this, we divide the documents into equally-sized chunks

with overlaps. Each chunk is then scored by the BERT reranker for its relevance to the query. It is important to acknowledge that it can be challenging to fully determine all of the key components of a document due to the lack of human annotations. Therefore, it is assumed that the best-scored chunks to the user queries constitute a subset of the document's essential content. This subset is then used as queries in various retrieval models, and their ability to accurately recall the corresponding document is evaluated as a measure of completeness.

To assess the **relevance ordering** ability, in the MS MARCO passage dataset (Nguyen et al., 2016), we employ the same BERT reranker as above. We randomly select 10k queries from the training set of MS MARCO 100k and use each retrieval model to predict the top 10 documents for each query. The binary group (q, d_m) comprising of q and the document d_m in the result list is incorporated into a set referred to as $\mathcal{S}$. In the context of a retrieval model M, the ranking of document d_m in the result list returned for a query q is represented by $r_{\mathrm{M}}(q, d_m)$, where $d_m \in \mathcal{S}$. Additionally, a random selection of 10 documents from the dataset is made for q. For each randomly selected document d_r, the binary group (q, d_r), is incorporated into a set referred to as $\mathcal{T}$. Subsequently, the BERT reranker is utilized to evaluate the relevance of the documents in the sets $\mathcal{S}$ and $\mathcal{T}$ to their corresponding queries, resulting in the ranking of document d_i in the combined result list as $r_{BERT}(q, d_i)$.

To evaluate the performance of a given model M, we define a *relevance ordering score* of M at the pth position in the result list returned for q as:

$$ros_q(\mathrm{M}, p) = ct(r_{BERT}(q, d_r) < r_{BERT}(q, d_m)), \text{ for } d_r \in \mathcal{T} \text{ and } r_{\mathrm{M}}(q, d_m) = p \quad (3)$$

where $ct(\cdot)$ is a counting function that determines the number of randomly selected documents that possess a higher relevance score given by BERT than d_m. Upon conducting an average computation across all queries, the relevance ordering score for the retrieval method M at position p is obtained:

$$ros(\mathrm{M}, p) = \underset{q \in Q}{Mean}(ros_q(\mathrm{M}, p)) \quad (4)$$

It is important to note that, as we assume that none of the random documents are relevant to the query if a significant proportion of these documents exhibit higher scores than those returned by the model, this serves as an indication that the model is returning less relevant documents. Models from different IR paradigms, including DSI-QG (Zhuang et al., 2022) which improves upon DSI, the BM25 sparse retrieval model (Robertson and Zaragoza, 2009), and the state-of-the-art dense retrieval model TCT-ColBERT (V2) (Lin et al., 2021c), are involved in our analysis.

2.3 Analysis Results

For **exclusivity**, Figure 1a reports Hits@1 to evaluate the ability of different models to recall the document ID when using either a cut-off of the initial text from the document or pseudo-queries as input. The results indicate that both BM25 and TCT-ColBERT exhibit exceptional performance in accurately recalling the corresponding document ID when being queried with the original text cut-off, and the accuracy increases with the length of the cut-off. However, DSI-QG can recall the correct document ID for pseudo-queries, but struggles to do so when queried with text cut-off from the document itself. This indicates that the specificity of DSI-QG is limited mostly to mapping from pseudo-queries to the associated document IDs, and lacks the ability to determine the document to which a segment of the primary text pertains, which could negatively impact its retrieval effectiveness.

For **completeness**, Figure1b shows the effectiveness of different retrieval methods in identifying important information in a document. Both BM25 and TCT-ColBERT have a high probability of accurately returning the correct document, with a probability above 95% when considering the top result, and over 99% when looking at the top 10 results. This suggests that both sparse and dense indexing methods effectively retain crucial content of the original document in relation to the user's query. In contrast, DSI-QG lacks proficiency in identifying the correct document, indicating that under the current training methodology, the differentiable search index may fail to capture important information, resulting in suboptimal performance.

Figure 1c plots the **Relevance Ordering** score (y-axis) at each ranking position (x-axis). As the ranking progresses further down, the difference between the returned documents of and randomly selected documents becomes increasingly insignificant, which aligns with expectation. However, the second-ranked document of DSI-QG shows a significant decline in ros compared to TCT-ColBERT and BM25. As the ranking lowers, the differentia-

tion between the returned documents of DSI-QG and randomly selected documents becomes increasingly indistinguishable. By the tenth document, approximately half of the random documents score higher, indicating a near-random distribution of the results of DSI-QG at that position. Based on the observations above, we propose to improve DSI-QG by a multi-task distillation approach.

3 Model Improvement

3.1 Method

In this section, we propose to improve the DSI-QG framework in order to gain enhanced abilities in terms of exclusivity, completeness, and relevance ordering, with the help of supervision signals from dense retrieval models. Specifically, the dense models are utilized to search through text fragments comprising all indexed documents and pseudo-query texts generated from these fragments. The text fragments that effectively recall the original documents are then selected and added to the training data to provide a more unique and comprehensive collection of information. Furthermore, a new distillation-based model training task, utilizing document IDs recalled by the dense model as a supervision signal, is also proposed to address the semantic gap between short input text and single document ID label. Finally, the training tasks of the DSI model are reclassified in accordance with their respective task characteristics. By utilizing this classification, the model is trained with a multi-task setting, allowing for the improvement of various abilities through the aid of different tasks.

3.1.1 Training Data Construction

Current DSI methods mostly largely rely on supervised learning from training data to guarantee effectiveness. Due to the inevitable bottleneck of model memory against the sheer size of document corpus, DSI-QG, along with other DSI models like NCI (Wang et al., 2022) and Ultron (Zhou et al., 2022), resort to memorizing the much shorter pseudo queries, hence the gap between the query and the document representations. To this end, in order to improve the exclusivity of DSI-QG, we employ a two-phase procedure for constructing training data. The first step involves dividing each document d_i in the set $\mathcal{D}$ of n documents into equal-length segments with overlaps $\mathcal{O}_i = \{d_i^1, d_i^2, \cdots, d_i^m\}$. The resulting segments of all documents after this process form the set $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_n\}$. Subsequently, for each text segment d_i^j, in $\mathcal{O}$, we utilize docT5query (Nogueira et al., 2019a) to generate a pseudo query pq_i^j. Similar to the construction of $\mathcal{O}$, the pseudo query set $\mathcal{P}_i = \{pq_i^1, pq_i^2, \cdots, pq_i^m\}$ is obtained for each document, and $\mathcal{P}_i$ constitute the pseudo query set $\mathcal{P}$ of the entire dataset. A combination of these two sets, $\mathcal{U} = \mathcal{O} \cup \mathcal{P}$, is expected to ensure a satisfactory level of information retention and effectively align the document text with the query.

Utilizing the set $\mathcal{U}$ for training may present challenges, such as difficulty in memorization due to a large number of text fragments and the inclusion of excessive irrelevant information. To mitigate these issues, filtering of $\mathcal{U}$ is implemented to optimize retention of crucial information. As previous analysis in section 2 demonstrates that the dense retrieval model effectively preserves textual information in its representation, we employ the dense method M for further processing of $\mathcal{U}$ by selectively filtering key content from it. The process involves inputting individual text fragment t, originating from the document of id_t in $\mathcal{U}$ as queries into the model, and obtaining a list, denoted as $\mathrm{R_k}(t, \mathrm{M}) = (id^1, id^2, \cdots, id^k)$, of the top k document IDs returned by the model. If id_t of the original text fragment is present within $\mathrm{R_k}(t, \mathrm{M})$, it is determined that the fragment t possessed key information relevant to the original document, and is included in the corpus $\mathcal{T}'$ for the model training.

In our implementation, the value of k is set to 1 for the original text segments and to 5 for the pseudo queries. $\mathcal{T}'$ was included in the initial training data for DSI-QG along with the original queries to ensure a valid comparison of results, resulting in the final training data $\mathcal{T}$.

3.1.2 Explicit Modeling of Relevance

Our previous analysis has shown that in order to improve the accuracy of retrieval results, the differentiable search index model should prioritize the enhancement of its relevance weighting capabilities in addition to its current unidirectional mapping to document IDs. The current training approach for the DSI-QG model, which maps brief text to a single document identifier, is constrained by the limited amount of information present in the brief query text, which leads to the gap between the pseudo queries and document contents. Moreover, the filtering of training data may cause the loss of information for certain documents, as the model would not be able to learn the relevant information

of omitted documents if their text is not included.

In light of the aforementioned considerations, a distillation-based training protocol for DSI is proposed in order to explicitly model the relevance of various documents to a given query. For text fragment t from the training set $\mathcal{T}$, the dense retrieval model M is utilized to query the corpus with t. The result list of the top f document IDs returned by the model, denoted as $\mathrm{R_f}(t, \mathrm{M}) = (id^1, id^2, \cdots, id^f)$, in conjunction with the identifier id_t of the document to which t belongs, serves as the supervision signals for the entire distillation task. Formally, the supervision signal for text segment t in the distillation task is defined as:

$$\begin{aligned} supsig_t^{dis} &= (id_t, \mathrm{R_f}(t, \mathrm{M})) \\ &= (id_t, id^1, id^2, \cdots, id^f) \end{aligned} \tag{5}$$

During the implementation of training procedures, we set f to 10 and utilize commas as a means of combining all identifiers into a cohesive string format, and the training objective of this task is:

$$L_{dis}(\theta) = \sum_{t \in \mathcal{T}} \log p(supsig_t^{dis} | \mathrm{M}(t), \theta) \tag{6}$$

where $p(supsig_t^{dis} | \mathrm{M}(t), \theta)$ denotes the probability of generating the supervised string given t as the input of the model.

The distillation process is expected to enable the model to acquire knowledge from the precise ID correspondence with the text, and associate the (pseudo) query to the list of document identifiers that the dense model deems most pertinent. It is our contention that learning the relevance relationship between the input text and different documents could improve the model's ability of relevance ordering, resulting in more relevant documents appearing at the top of the list of results. However, prior to this endeavor, the DSI approaches have been evaluated on datasets that come with only shallow relevance annotations, where a query usually has a single labeled relevant document. To validate the model in the context of deeply annotated data, we have constructed an MS MARCO 300k dataset, comprised of TREC DL19 (Craswell et al., 2020) and 20 (Craswell et al., 2021) data based on the MS MARCO passage set with 80 relevant documents per query on average.

3.1.3 End-to-end Multi-task Training

Remind that in DSI (Tay et al., 2022), the training task is divided into two sub-tasks, indexing and retrieval, depending on whether the input text is an original text fragment or a query. Our proposed model utilizes a multi-task setup, with however redefined tasks. Empirically, both sub-tasks of DSI in fact lead the model to learn the features of a single document, thus we uniformly attribute them to the indexing task. Therefore, we formally define the indexing task as:

$$t \rightarrow id_t, \text{ for } t \in \mathcal{T} \tag{7}$$

where $\rightarrow$ indicates an injective function that maps a text segment to its corresponding identifier. During training, the loss function of the indexing task is:

$$L_{index}(\theta) = \sum_{t \in \mathcal{T}} \log p(id_t | \mathrm{M}(t), \theta) \tag{8}$$

which maximizes the probability of generating id_t with t as the input of the model, and t can be a natural text fragment in any form.

In contrast, our newly proposed training task utilizes a list of document IDs as a supervision signal, which enables the model to explicitly learn the relationship of relevance between documents and queries, as well as the relationship of relevance between documents themselves. Therefore, we define this task as the retrieval task:

$$t \rightarrow (id^1, id^2, \cdots, id^l), \text{ for } t \in \mathcal{T} \tag{9}$$

The objective of this task is to assign a text fragment t to a list of identifiers, which can be derived from any method. Here, we use $supsig_t^{dis}$ as the list, thus the loss function for the retrieval task is:

$$L_{retvl}(\theta) = L_{dis}(\theta) \tag{10}$$

During training, the model is randomly presented with either a single ID or a list of IDs as the supervision signal with equal probability, and a special symbol is appended to the beginning of the query to indicate the task type. The loss function of the multi-task training can be written as:

$$L_{multi}(\theta) = L_{index}(\theta) + L_{retvl}(\theta) \tag{11}$$

In the following, for the DSI-QG model that only uses the newly constructed training data, we denote it as **DSI-QG-Merge**. The model that only uses the distillation task for training is labeled as **DSI-QG-Distill**, and the model that uses the newly constructed training data for multi-task training, first training the index task and then training the distillation task, is labeled as **DSI-QG-M+D**. The model trained using all of the above improvements is denoted as **DSI-QG-Multi**.

3.2 Evaluation Setup

Datasets and Metrics. We experiment on the MS MARCO (Nguyen et al., 2016) and Natural Question (NQ) (Kwiatkowski et al., 2019) datasets. Akin to prior art (Tay et al., 2022; Zhuang et al., 2022), we employ a 100k subset of MS MARCO and the Dev queries with shallow annotations. Our constructed MS MARCO 300k is also used, which includes queries and documents from both the Dev set and TREC DL19 (Craswell et al., 2020) and 20 (Craswell et al., 2021). For the NQ dataset, following the DSI experimental procedures (Tay et al., 2022), we construct the NQ 320k dataset. Akin to (Tay et al., 2022), we report Hits@1 and Hits@10 on Dev set of the data. To accurately assess the retrieval effectiveness on datasets with deeper annotations such as TREC DL 19 and 20, NDCG@10 and P@10 are also reported. Statistical significance for paired two-tailed t-test is reported.

Baselines. We evaluate against the following baselines: the Pyserini implementation of **BM25** (Lin et al., 2021a), and the original **DSI** (Tay et al., 2022), **DSI-QG** (Zhuang et al., 2022), and the recently proposed **NCI** (Wang et al., 2022). The state-of-the-art dense retrieval model **TCT-ColBERT (V2)** (Lin et al., 2021c) and **SEAL** based on generative retrieval (Bevilacqua et al., 2022a) are also included. Further information about the data and baselines can be found in Appendix A.

Implementation Details. Following the DSI and DSI-QG settings (Zhuang et al., 2022), the models are initialized using standard pre-trained T5 models (Raffel et al., 2019). The T5-Base and T5-Large models are trained with a batch size of 128, the learning rate is 5e-4, and the maximum number of training steps is determined based on the scale of training data, with options among {1M,2M,3M}. All pseudo-queries are generated by docT5query (Nogueira et al., 2019a) based on T5-Large. For our proposed method, we adhered to the DSI approach (Tay et al., 2022) to generate semantic IDs based on the dense representation of TCT-ColBERT. Following DSI-QG (Zhuang et al., 2022), all DSI models, with the exception of NCI, are trained using Naive String Docids if there are no extra specifications. We plan to make our code and data available to public.

3.3 Evaluation Results

The enhanced DSI-QG variants outperform the DSI baselines. Results on datasets with shallow annotations are presented in Table1. The proposed method of multi-task training, DSI-QG-Multi, demonstrates a notable enhancement in comparison to existing DSI models, with statiscally significant improvement reported on three datasets. Additionally, our experiments demonstrate that scaling up the model size has limited impact on the original DSI-QG, whereas the our adjustment to the training data and training tasks allow model to obtain better results on larger models as the data size increases, as is evident on the MS MARCO 300k data. Furthermore, our utilization of multi-task training enables the selection of different subtask settings for prediction, with the indexing task yielding the best results for the MS MARCO dataset, and the retrieval setting producing better results for the NQ dataset, possibly due to the need to rely on different types of information for relevant documents for different datasets.

The proposed training approach facilitates the three aforementioned abilities of DSI-QG. The effectiveness of the proposed method in enhancing the basic abilities is evaluated through the DSI-QG-Multi model, for the three experiments described in Section 2. The results of these experiments are included in Figure 1. The present study demonstrates that our improvement of DSI-QG results in significant enhancement in all three model abilities. Figure 1a illustrates that the model is able to accurately return the corresponding IDs of the text of different lengths when the initial parts of documents are input, and its ability to identify documents corresponding to pseudo-queries is still maintained. Figure 1b further supports the validity of the model's stored information following optimized training, as it is able to accurately locate documents containing key content related to a given query. This demonstrates that our approach effectively picks up the unique and key information of the documents, and our model effectively encodes these contents in training, thus makes improvements on exclusivity and completeness.

Our proposed improvements has led to a significant reduction in the performance gap to TCT-ColBERT. This is particularly encouraging as it suggests that our proposed DSI improvements are able to achieve comparable performance to the dense model on datasets with more complete annotations, as seen in Table 2. Importantly, this is achieved while still maintaining the advantages of DSI, such as minimal storage cost and end-to-

Methods	Model Size/ Task	MS Marco 100k Dev		MS Marco 300k Dev		NQ 320k	
		Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
BM25	-/-	0.5398	0.8295	0.4404	0.7417	0.0834	0.3336
TCT-ColBERT	Base/ -	0.7074	0.9506	0.6032	0.9107	0.2411	0.7197
DSI	Base/ Index	0.0292	0.0682	0.0218	0.0543	0.0008	0.0083
DSI	Large/ Index	0.0874	0.1948	0.0214	0.0652	0.0057	0.0493
SEAL	Base/ -	0.2802	0.6219	0.2322	0.5818	0.1377	0.5679
DSI-QG	Base/ Index	0.6085	0.8026	0.5123	0.7703	0.2193	0.5180
DSI-QG	Large/ Index	0.6182	0.8024	0.5188	0.7589	0.2223	0.5189
NCI *(sem)*	Base/ Index	0.6133	0.8670	0.5289	0.8244	0.2120	0.6999
DSI-QG-Multi	Base/ Index	0.6711†	**0.9208**†	0.5390	0.8726 †	0.2190	0.6138
DSI-QG-Multi	Base/ Retrv	0.6711†	0.9143†	0.5219	0.8480†	0.2162	0.7003
DSI-QG-Multi*(sem)*	Base/ Index	0.6626†	0.9115†	0.5615†	0.8801†	0.2390†	**0.7202**†
DSI-QG-Multi*(sem)*	Base/ Retrv	**0.6739**†	0.9192†	0.5589†	0.8659†	**0.2392** †	0.7135†
DSI-QG-Multi	Large/ Index	0.6625†	0.9130†	**0.5746**†	**0.8868**†	0.2206†	0.6304
DSI-QG-Multi	Large/ Retrv	0.6593†	0.9138†	0.5741†	0.8754†	0.2285†	0.7110†

Table 1: Evaluation results with shallow annotations. Method names suffixed with *(sem)* indicates that the semantic IDs are used in the training process. Statistical significance at 0.05 relative to NCI is marked by †.

Methods	DL19		DL20	
	NDCG@10	P@10	NDCG@10	P@10
BM25	0.6843	0.8837	0.6873	0.4852
TCT-ColBERT	0.7977	0.9279	0.8012	0.6315
DSI	0.2156	0.2209	0.1907	0.1167
DSI-QG	0.6922	0.8256	0.7348	0.5630
NCI	0.6725	0.8419	0.7127	0.5407
DSI-QG-Merge	0.7215†	0.8767†‡	0.7501†	0.5815†
DSI-QG-Distill	0.7836†‡	0.9140†‡	0.7801†‡	0.6056†‡
DSI-QG-D+M	0.7838†‡	0.9209†‡	0.7851†‡	0.6074†‡
DSI-QG-Multi	**0.7920**†‡	**0.9279**†‡	**0.7983**†‡	**0.6278**†‡

Table 2: Results on TREC DL datasets with deeper annotations. DSI-QG-D+M and DSI-QG-Multi are evaluated on the retrieval task while others are evaluated on the indexing task. Statistical significance at 0.05 relative to NCI or DSI-QG is marked by † or ‡.

end retrievability. As depicted in Figure 1c, the distribution of the top 10 documents returned by DSI-QG following optimized training exhibits a high degree of similarity to that of TCT-ColBERT. This demonstrates that the model is capable of effectively modeling the correlation order among documents through distillation-based training. To further validate this conclusion, Table 2 presents the effects of various models that were trained on MS MARCO 300k data on the TREC DL19 and DL20 query sets. The data demonstrate that our method consistently outperforms the original training method by at least 8.6% on deeply annotated data, thus achieving a level of performance that is comparable to that of dense retrieval models. While our DSI-QG-Multi model still slightly underperforms TCT-ColBERT on datasets with shallow annotations, it is important to note that our objective is to enhance the retrieval effectiveness of DSI models. Previous studies, such as NCI (Wang et al., 2022) and Ultron (Zhou et al., 2022), have indicated that DSI models can outperform dense retrieval models when trained on similar smaller datasets. In our experiments, TCT-ColBERT was trained on the entire MSMARCO dataset, which likely contributes to its superior performance. By distilling the ranking capabilities of TCT-ColBERT, we have achieved significant improvements.

Effects of Document Identifiers. To investigate the impact of various identifiers, we assign semantic IDs to documents utilizing the hierarchical clustering process as in DSI (Tay et al., 2022) in our training process. The results presented in table 1 indicate that the semantic IDs can further improve the retrieval on larger datasets (i.e., MS MARCO 300k and NQ). For comparison of our model with other models in different tasks under semantic IDs, as well as the ablation analysis, please refer to Appendix A.4 & A.5.

4 Related Works

The field of information retrieval has recently seen a surge in interest in generative retrieval models. Examples of this approach include the docT5query (Nogueira et al., 2019a), which trains T5 models to generate document-related queries and adds them to the original documents for index construction, and SEAL (Bevilacqua et al., 2022b), which generates text fragments that the target document may contain and uses FM-Index (Ferragina and Manzini, 2000) for retrieval. GENRE (Cao et al., 2021) utilizes BART (Lewis et al., 2020) for entity name generation in entity retrieval tasks.

Furthermore, Tay et al. (2022) proposed the Differentiable Search Index (DSI), which explores various strategies for mapping string queries to

document IDs based on T5 (Raffel et al., 2019). This idea has been further developed in models such as DSI-QG (Zhuang et al., 2022), NCI (Wang et al., 2022), and Ultron (Zhou et al., 2022), which have all effectively improved the retrieval performance of generative retrieval models through the use of pseudo queries. More specifically, DSI-QG (Zhuang et al., 2022) adapted the model for multilingual retrieval tasks, Ultron (Zhou et al., 2022) attempted to incorporate hyperlink information into document IDs, and NCI (Wang et al., 2022) employed richer data, more fine-grained semantic ID mapping, and a novel decoder structure to obtain the best retrieval results. Differentiable indexing models have also been applied to a wide range of tasks such as knowledge-intensive language tasks (Chen et al., 2022a), long sequence retireval (Lee et al., 2022b) and QA tasks (Yu et al., 2022). Additionally, efforts have been made to improve the memory capability (Mehta et al., 2022) and decoding capability (Lee et al., 2022a) of these models. This paper aims to gain a deeper understanding of differentiable indexing models as a retrieval method and proposes a multi-task distillation approach to improve their performance.

5 Conclusions

We summarized three essential abilities of a functional non-Boolean IR framework. Through an empirical analysis of these abilities, we identified potential weaknesses in existing DSI approaches. To address these weaknesses, we propose a multi-task distillation approach to enhance the effectiveness of DSI by learning from dense retrieval while preserving the advantages of DSI, such as minimal storage cost and end-to-end retrievability. Our evaluation results indicate that our proposed approach improves the three abilities of DSI and, as a result, its retrieval effectiveness, particularly on data with more comprehensive human annotations. There are several directions to explore. Firstly, the three capabilities identified in our empirical analysis, although currently applied to DSI, can be further extended to analyze statistical retrieval methods as well. Additionally, investigating the trade-off between these three capabilities would be an interesting avenue for future research. Performing more comprehensive experiments to ascertain the effectiveness of retrieval models that rely on expanded queries for downstream tasks would yield valuable insights.

Limitations

In this study, we primarily focus on the examination and experimentation of the DSI-QG model, with plans to expand our research to include more recent models that utilize differentiable search indexing, such as the NCI model. While our approach has demonstrated effective improvements in DSI retrieval outcomes, and both TCT-ColBERT and our proposed DSI-QG-Multi performed well in our empirical analysis concerning relevance ordering, we cannot dismiss the possibility that these favorable results may be attributed to the extraction of insights from a specific BERT reranker model that shares similarities or correlations with the one used to define the desired ranking.

Despite showing improvement over DSI-QG, our model remains slightly less effective than state-of-the-art dense retrieval methods such as TCT-ColBERT V2. Our approach offers advantages over dense retrieval models such as reduced storage and maintenance overhead, as DSI models do not require additional index structures for online use. Though index structures are utilized during the training phase of DSI-QG-Multi, the generated index structures are temporary in nature.

Furthermore, due to the limitation of model memory, current research on DSI only experiments on a subset of the entire MS MARCO dataset or small dataset such as NQ. Therefore, an important future direction is to develop more efficient architectures to deal with the issue of memory bottleneck, for example, by using the current popular Large Language Models (LLM) or constructing aggregation structures for storing all information in hierarchical pieces.

Acknowledgements. This work is supported in part by the National Key Research and Development Program of China (No. 2020AAA0106400) and the National Natural Science Foundation of China (No. 62272439).

References

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

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

Andrzej Bialecki, Robert Muir, and Grant Ingersoll. 2012. Apache lucene 4. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval, OSIR@SIGIR 2012, Portland, Oregon, USA, 16th August 2012*, pages 17–24. University of Otago, Dunedin, New Zealand.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2022a. Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 191–200. ACM.

Xiaoyang Chen, Kai Hui, Ben He, Xianpei Han, Le Sun, and Zheng Ye. 2022b. Incorporating ranking context for end-to-end BERT re-ranking. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part I*, volume 13185 of *Lecture Notes in Computer Science*, pages 111–127. Springer.

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

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

Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *CoRR*, abs/1910.10687.

Paolo Ferragina and Giovanni Manzini. 2000. Opportunistic data structures with applications. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 390–398. IEEE Computer Society.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 981–993. Association for Computational Linguistics.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research*

and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, pages 113–122. ACM.

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

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. *CoRR*, abs/2004.12832.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. 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. *CoRR*, abs/2210.02068.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022b. Generative retrieval for long sequences. *CoRR*, abs/2204.13596.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: passage representation aggregation for document reranking. *CoRR*, abs/2008.09093.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021b. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2356–2362. ACM.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *CoRR*, abs/2010.11386.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021c. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173, Online. Association for Computational Linguistics.

Sanket Vaibhav Mehta, Jai Prakash Gupta, Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2022. DSI++: updating transformer memory with new documents. *CoRR*, abs/2212.09744.

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 *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

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

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019b. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424.

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 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. *CoRR*, abs/2202.06991.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *CoRR*, abs/2002.10957.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Allen Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2022. A neural corpus indexer for document retrieval. *CoRR*, abs/2206.02743.

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

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *CoRR*, abs/2209.10063.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: contextualized query expansion for document re-ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4718–4728. Association for Computational Linguistics.

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

Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *CoRR*, abs/2206.10128.

Table 3: Summary of notation.

Symbol	Definition
$\mathcal{D} = \{d_i\}$	Set of documents stored in the index.
$\mathcal{I} = \{id_i\}$	Set of docids of documents in $\mathcal{D}$.
$\mathcal{Q} = \{q_i\}$	Set of user queries.
$\mathrm{M}(d_i)$	An retrieval function take d_i as input.
$\mathrm{R}(q_i, \mathrm{M})$	A retrieval operation for q_i based on M.
$\mathrm{s}(q_i, d_i)$	Relevance score of q_i and d_i.
$d_j \preceq_q d_i$	A binary relation that d_i is more relevant to q than d_j

A Appendix

A.1 Notation Table

The notations are listed in table 3 .

A.2 Datasets Details

For MS MARCO 100k dataset, We adhere to the experimental design outlined in (Zhuang et al., 2022) by randomly selecting 93k passages from the dataset and incorporating all text from the validation set. For MS MARCO 300k dataset, we randomly select 293k passages from the dataset and incorporate all text from the validation set, as well as from TREC DL19 and 20. For NQ 320k dataset, we follow the DSI experimental procedures (Tay et al., 2022), constructing the data consisting of approximately 200k passages and 8k validation set queries after pre-processing.

A.3 Baseline Details

The details of our baselines are as follows:

- **BM25** (Robertson and Zaragoza, 2009) is a classical sparse retrieval model that utilizes lexical weights. In this study, we employ a pyserini-based implementation (Lin et al., 2021b), which leverages the Lucene (Bialecki et al., 2012) as the underlying infrastructure.

- **TCT-ColBERT (V2)** (Lin et al., 2021c) is a state-of-the-art single-vector dense retrieval model that utilizes knowledge distillation and hard negative example sampling techniques. The model combines the performance of ColBERT (Khattab and Zaharia, 2020) with the computational efficiency of a bi-encoder. The implementation of TCT-ColBERT is based on the Faiss vector index and is implemented through the pyserini library. This model is utilized to demonstrate the effectiveness of the current state-of-the-art model for retrieval and serves as a guide for training a differentiable search index. Note that the TCT-ColBERT model is exclusively trained utilizing the MS MARCO Passage dataset. Our analysis of the retrieval performance of various dense retrieval models on NQ data revealed that TCT-ColBERT displayed remarkable results despite not having been specifically trained on that dataset.

- **DSI** (Tay et al., 2022) is a T5-based approach for learning text-to-identifier mappings. Specifically, DSI defines the process of mapping original text to identifiers as an indexing task and the mapping of query text to identifiers as a retrieval task. This study reproduces the DSI model using the Naive String Docid and Semantic String Docid techniques, based on T5-base and T5-large architectures, utilizing open-source implementations[2]. As access to the original training data of DSI is not available, we followed the settings of the open-source implementation to set the indexing and retrieval ratio to 1:1 for MS MARCO and 3:1 for NQ.

- **DSI-QG** (Zhuang et al., 2022) improves upon DSI by incorporating pseudo-queries that are generated utilizing DocT5Query during the training process. Our research has involved reproducing the model on the MS MARCO 100k dataset, utilizing the available open-source code, and subsequently applying it to the MS MARCO 300k and NQ 320k datasets. This model serves as the baseline for our work but also serves as the primary focus for our investigations into potential improvements.

- **NCI** (Wang et al., 2022), is a recently proposed state-of-the-art differentiable search indexing model that utilizes a variety of techniques to enhance its performance. These include the generation of semantic identifiers, the implementation of query generation strategies, and the utilization of a prefix-aware weight-adaptive decoder. Through the use of open-source implementation, the effective-

[2] https://github.com/ArvinZhuang/DSI-transformers

ness of the model is validated using the three distinct datasets.

- **SEAL** (Bevilacqua et al., 2022a) is a novel methodology that incorporates an autoregressive language model with a compressed full-text substring index. The implementation of this model utilizes BART and an external index known as the FM-index. We evaluate it on three datasets based on its official implementation.

A.4 Impact of Document Identifier

To investigate the impact of various identifiers, we assign semantic IDs to documents utilizing the hierarchical clustering process as in DSI (Tay et al., 2022) and have subsequently trained various models based on these new IDs. From Table 4, it is observed that for DSI and DSI-QG, the utilization of semantic clustering-based document IDs results in improved effectiveness when compared to the use of Naive String IDs. For our proposed DSI-QG-Multi, the use of semantic IDs is effective on larger datasets, such as MS MARCO 300k and NQ320k. Across all three models, it is evident that semantic IDs are highly beneficial in enhancing Hits@10 on NQ.

A.5 Impact of Different Factors

As shown in table 5,the results of *DSI-QG-Distill* reveal that the distillation task effectively improves the performance of the DSI-QG model on Hits@10, while having comparable performance on Hits@1, except for a drop on the MS MARCO 300k data. This suggests that the model is able to recall more relevant documents for a given query in the 2nd to 10th positions of the returned list, which is in line with our objective of enhancing the relevance ordering capabilities of the model through distillation. In comparison to the original DSI-QG, *DSI-QG-Merge* exhibits a substantial enhancement in both Hits@1 and Hits@10. This illustrates the non-negligible impact that training data has on the overall performance of the model, thereby highlighting the importance of utilizing a more judicious and directed approach in the selection of training data. When the tasks are trained independently, the result of *DSI-QG-M+D* demonstrates an improvement in Hits@10 across all three datasets in comparison to prior single-task training. However, this improvement is less substantial than the gain achieved by *DSI-QG-Multi*.

Models	ID Type	MSMarco 100k		MSMarco 300k						NQ 320k	
		Dev		Dev		DL19		DL20		Dev	
		Hits@1	Hits@10	Hits@1	Hits@10	Ndcg@10	P@10	Ndcg@10	P@10	Hits@1	Hits@10
NCI	Semantic	0.6133	0.8670	0.5289	0.8244	0.6725	0.8419	0.7127	0.5407	0.2120	0.6999
DSI	Naive	0.0292	0.0682	0.0218	0.0543	0.2156	0.2209	0.1907	0.1167	0.0008	0.0083
	Semantic	0.0957	0.2597	0.0602	0.2054	0.3607	0.4279	0.2749	0.2000	0.0212	0.2370
DSI-QG	Naive	0.6085	0.8026	0.5123	0.7703	0.6922	0.8256	0.7348	0.5630	0.2193	0.5180
	Semantic	0.6103	0.8109	0.5252	0.7971	0.7022	0.8512	0.7317	0.5759	0.2287	0.6252
DSI-QG-Multi	Naive	**0.6711**†	**0.9208**†	0.5390	0.8726†	0.7920†	0.9279†	**0.7983**†	**0.6278**†	0.2190	0.6138
	Semantic	0.6626†	0.9115†	**0.5615**†	**0.8801**†	**0.7961**†	**0.9372**†	0.7844†	0.6019†	**0.2390**†	**0.7202**†

Table 4: Semantic ID Experiment Results. Results on the dev set of the three datasets are based on index task, and results on TREC DL are based on retrieval task. Statistical significance at 0.05 relative to NCI is marked by †.

Methods	Model Size/ Task	MS Marco 100k Dev		MS Marco 300k Dev		NQ 320k	
		Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
DSI-QG	Base/ Index	0.6085	0.8026	0.5123	0.7703	0.2193	0.5180
NCI	Base/ Index	0.6133	0.8670	0.5289	0.8244	0.2120	0.6999
DSI-QG-Distill	Base/ Retrv	0.6095	0.8851†	0.4755	0.8143	0.2188	0.6535
DSI-QG-Merge	Base/ Index	0.6457†	0.8692	0.5493†	0.8302	**0.2199**†	0.5957
DSI-QG-D+M	Base/ Retrv	0.6460†	0.9097†	0.5007	0.8471†	0.2163	0.6992
DSI-QG-Multi	Base/ Index	**0.6711**†	**0.9208**†	**0.5390**	**0.8726** †	0.2190	0.6138
DSI-QG-Multi	Base/ Retrv	**0.6711**†	0.9143†	0.5219	0.8480†	0.2162	**0.7003**

Table 5: Ablation study results with shallow annotations. Statistical significance at 0.05 relative to NCI is marked by †.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

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

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

2,3

☑ B1. Did you cite the creators of artifacts you used?
1,2,3,4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☑ 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)?
1,2,3

☐ 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?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ 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.
3

C ☑ Did you run computational experiments?

2, 3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
2, 3

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?
3

☑ 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?
3

☑ 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.)?
2, 3

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.?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.