國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master's Thesis

# Multi-DSI: 可微搜尋索引的非確定性標識符和概念對齊

# Multi-DSI: Non-deterministic Identifier and Concept Alignment for Differentiable Search Index

柳宇澤

Yu-Ze Liu

指導教授: 鄭卜壬 博士

Advisor: Pu-Jen Cheng, Ph.D.

中華民國 113 年 7 月

July 2024

doi:10.6342/NTU202402960

# 誌謝

首先在生命上，我想要感謝父母讓我誕生於這個世界，給我無後顧之憂的自由，進而讓我能夠好好享受這個世界的隨機性，而「隨機」勢必有好有壞，但我想一路上，就算是暫時遇到的壞與失敗，若我不把它當成是終點，就長途來看也未必不會是下一個成功的原石。

第二我也非常想要感謝我的祖父母、外祖父母，為我的家庭奠定善良的根基，使整個家庭的價值觀有著非常正面且積極的態度，我想說我能考上以及完成碩士學位，有很大的一部分的動力來源，是想要看到他們的笑臉！我也想對他們說，若假設過往的時勢與家庭因素沒有阻礙他們，以他們認真的個性來就讀學士、碩士學位，一定是比我厲害 100 倍以上的大學霸！

在學術上，我絕對必須好好感謝這一年來指導我碩士論文的鄭卜壬教授以及姜俊宇教授 (學長)、PJJ 組合！這個碩士論文也是我首次自己尋找研究題目的歷程，一路上有需多的第一次，非常可怕也刺激，我認為在我培養研究的獨立性上有非常大的幫助，若沒有兩位的幫助也絕對不可能獲得 CIKM 的認可的，教授跟學長辛苦了！

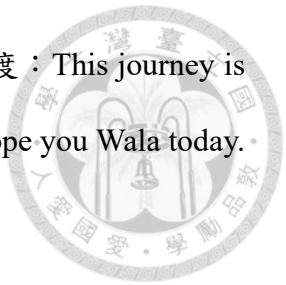
而在就讀碩士前，我在中研院也有幸受到蔡明峰教授以及王釧茹教授的指導，初窺了前輩們在做研究上的流程是如何進行的，也收穫了許多模型與實作上的知識，不僅如此，我也特別想感謝王教授的嚴格監督，讓我在知識性的思考與交流上，有更深一層的見解！我也感謝在中研院的學長姐與同學們的共同奮鬥。

最後我想說，我真的很幸運能加入 IR Lab，認識了大家，也認識了善良單純的女友。這個實驗室的氣氛真的很健康、友善，大家在水深火熱的作業與研究之餘，也可以互相臭、互相講幹話，真是太 Wala 啦。



doi:10.6342/NTU202402960

分享給正在看這篇論文的你，我與 Wala 兄弟們的研究態度：This journey is too Wala to Wala. If you are Wala, please don’ t hesitate to Wala. Hope you Wala today. Keep Wala! Walalalalalalalalala :).




doi:10.6342/NTU202402960

# Multi-DSI: 可微搜尋索引的非確定性標識符和概念對齊

研究生: 柳宇澤　　指導教授: 鄭卜壬 博士

國立臺灣大學　資訊工程學系

# 摘要

信息檢索（IR）已經被研究了很長一段時間。為解決 IR 問題，提出了許多方法，這些方法大致分為兩個方向：統計方法和深度學習方法。統計方法通常利用詞語的分佈來計算查詢與文檔的相似性，而深度學習模型則傾向於學習編碼器，並將查詢和文檔投射到向量空間中進行檢索。隨著生成性深度學習模型的出現，生成性信息檢索（Generative IR）引起了越來越多的關注。生成性信息檢索為解決信息檢索問題提供了新視角，並且透過生成模型直接生成文檔的標示符，減少了在推理過程中計算相似性所需的複雜度，該複雜度極大地受語料庫規模的影響。然而，現有方法面臨兩個問題：（1）當文檔僅用一個語義標識符（ID）表示時，檢索模型可能無法捕捉到文檔多方面且複雜的內容；（2）當生成的訓練數據存在語義模糊時，檢索模型可能難以區分相似文檔內容之間的差異。為了解決這些問題，我們提出了 Multi-DSI，旨在（1）提供多個非確定性的語義標識符 (Non-deterministic Semantic Identifier)；（2）對齊查詢和文檔的概念以避免模糊性。在兩個基準數據集上的大量實驗表明，所提出的模型比基線方法顯著提高了 7.4% 的性能。

**關鍵字：**

生成式資訊檢索、可微分搜尋索引、查詢生成應用、概念對齊、多重索引點檢索



doi:10.6342/NTU202402960

# Multi-DSI: Non-deterministic Identifier and Concept Alignment for Differentiable Search Index

Student: Yu-Ze Liu Advisor: Pu-Jen Cheng, Ph.D.

Department of Computer Science and Information Engineering

National Taiwan University

## ABSTRACT

There are many methods proposed to tackle IR problems. They are roughly divided into two directions, statistical methods and deep learning methods. While statistical methods usually utilize the distribution of words to calculate the similarities of the queries and documents, deep learning models tend to learn encoders and project queries and documents to a vector space for retrieval. With the advent of generative deep learning models, generative IR has gained increasing attention. However, existing methods face two issues: (1) when a document is represented by a single semantic ID, the retrieval model may fail to capture the multifaceted and complex content of the document; and (2) when the generated training data exhibits semantic ambiguity, the retrieval model may struggle to distinguish the differences in the content of similar documents. To address these issues, we propose Multi-DSI to (1) offer multiple non-deterministic semantic identifiers and (2) align the concepts of queries and documents to avoid ambiguity. Extensive experiments on two benchmark datasets demonstrate that the proposed model significantly outperforms baseline methods by 7.4%.

**Keywords:**

Generative Information Retrieval, Differentiable Search Index, Query Generation, Con-



doi:10.6342/NTU202402960

cept Alignment, Multiple Indexing Point Retrieval



doi:10.6342/NTU202402960



doi:10.6342/NTU202402960

# CONTENTS





doi:10.6342/NTU202402960

doi:10.6342/NTU202402960

# LIST OF FIGURES





doi:10.6342/NTU202402960



doi:10.6342/NTU202402960

# LIST OF TABLES





doi:10.6342/NTU202402960



doi:10.6342/NTU202402960

# Chapter 1

# Introduction

Generative information retrieval (GenIR)[1]–[15: Lee *et al.* 2022] is a novel approach that retrieves documents by directly generating their identifiers. Specifically, GenIR leverages the encoder-decoder structure to bind the mapping between queries and document identifiers (IDs). It fully parameterizes the entire indexing and retrieval process, which is typically managed by methods such as the dual-encoder architecture [16]–[21: Khattab and Zaharia 2020]. GenIR allows for end-to-end training and has attracted increasing attention in recent research.

Differentiable search index (DSI) [1: Tay *et al.* 2022] is a pioneer transformer-based solution in this domain. It indexes documents and then fine-tunes a generative model with labeled query-document pairs so that the model can retrieve relevant documents with any input query. However, the differences in length and semantic expression between queries and documents lead to inconsistencies in their data distributions, which can further affect retrieval accuracy.

To address this issue, some works [3], [5], [7: Wang *et al.* 2022] have shifted focus to generating pseudo queries for documents. These pseudo queries are treated as indexing points for their corresponding documents. They subsequently learn to map pseudo queries to document IDs. Conceptually, the methods of query generation involve transforming the document space into a corresponding pseudo query space and then mapping that pseudo query space to the document ID space. Such transformation aligns the characteristics of pseudo queries more closely with those of real queries. By generating multiple pseudo


doi:10.6342/NTU202402960

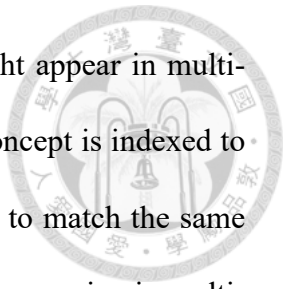
queries for each document, the training data can be augmented, potentially capturing the diversity within the documents.

While these methods have demonstrably improved retrieval performance, the document IDs they generate are all deterministic, where each document has only one unique ID within the retrieval model. Past studies have explored incorporating semantic meaning into document IDs. For example, some uses hierarchical k-means methods to ensure documents with similar meaning share prefixes in their identifiers [1: Tay *et al.* 2022]. Others construct substrings as document IDs by leveraging partial sentences within the documents [4]–[6], [22], [23: Zhou *et al.* 2022]. Although these methods offer benefits like improved human readability and a potentially reduced search space for semantically structured IDs, representing a complex document with a single ID inevitably leads to a loss of crucial semantic information. Essentially, various concepts mentioned in the document become conflated with the unique ID, hindering the ability to distinguish or retrieve detailed concepts.

In addition, the methods of query generation might potentially produce the same pseudo query for different documents. Such query ambiguity issue can easily reduce the discriminability of documents in retrieval, leading to the inability of the trained model to effectively distinguish the differences in local semantics between different documents.

To address the two issues mentioned above, we propose Multi-DSI, a model that generates multiple non-deterministic semantic identifiers and supports concept alignment for both queries and documents. To generate the non-deterministic semantic identifiers, we apply a query generation model to produce queries from a document, treating these generated queries as concepts mentioned in the document. These concepts are then iteratively


doi:10.6342/NTU202402960



clustered to create identifiers. However, since a single concept might appear in multiple documents, it can lead to the problem of ambiguity if the same concept is indexed to multiple documents. To address this, we perform concept alignment to match the same concept with document-level information. This ensures that concepts appearing in multiple documents are aligned with the unique main document-level information in each case. We conduct experiments on two datasets, MSMARCO and Natural Questions. The results demonstrate that Multi-DSI achieves performance comparable to other baselines. Additionally, we perform ablation studies for each component and visualiz the effectiveness of our model.


doi:10.6342/NTU202402960

# Chapter 2

# Related Work

Information retrieval (IR) has been well-explored for a long time. There are mainly three classes of traditional IR models, Boolean, Vector space, and Probabilistic.

Boolean Model is based on the set theory and boolean algebra. Vector space model is aimed to create vectors of text documents and calculate the similarity between the search document and the representative user query. Probabilistic model estimates how likely it is that a document is relevant for a query by the statistical distribution of the terms in both the relevant and non-relevant documents.

From the initial implementation of TF-IDF, the Vector Space Model incorporated semantic information through Latent Semantic Indexing (LSI). LSI employs Singular Value Decomposition (SVD) to reduce the dimensionality of the original high-dimensional term vectors into a latent semantic space. With the increasing computational power and data availability, neural networks have been progressively applied to the Vector Space Model. Word2Vec [24: Rong 2014], a word vector representation method, maps words into a low-dimensional continuous vector space. Doc2Vec [25: Lau and Baldwin 2016] extends Word2Vec to generate document-level vector representations. With the advent of Transformers [26: Vaswani *et al.* 2017], deep neural networks and attention mechanisms have been utilized to generate context-sensitive word and document representations.

Compared to the traditional TF-IDF sparse vector representation used in Sparse Retrieval, methods leveraging neural networks to generate word or document vectors can be classified as Dense Retrieval. Within Dense Retrieval, the introduction of the encoder


doi:10.6342/NTU202402960

concept further categorizes it into Cross-encoder and Dual-encoder architectures:

## 2.1 Dense Retrieval

### 2.1.1 Cross-encoder architecture

The Cross-encoder architecture takes the query and document as a joint input, encoding them through the same encoder—typically a Transformer model [27], [28: Liu *et al.* 2019] such as BERT—and directly outputs a relevance score. This method captures fine-grained interaction information between the query and document, often resulting in higher retrieval accuracy.

BERT [27: Devlin *et al.* 2018] concatenates the query and document into a single input sequence, such as ”[CLS] query [SEP] document [SEP]”. This sequence is encoded by the BERT model, and the relevance score is obtained from the output vector of the ”[CLS]” token. RoBERTa [28: Liu *et al.* 2019] is similar to BERT but utilizes a larger scale of training data and a longer training duration.

### 2.1.2 Dual-encoder architecture

The Dual-encoder architecture employs two separate encoders to independently encode the query and the document, resulting in two distinct vector representations. The relevance is then assessed by calculating the similarity between these vectors, using methods such as dot product or cosine similarity. Because the encoding of the query and document is independent, this approach offers significant efficiency advantages compared to Corss-encoder methods.

Sentence-BERT [29: Reimers and Gurevych 2019] employs two BERT models with shared parameters to encode the query and document separately. The relevance is evalu-


doi:10.6342/NTU202402960

ated by computing the dot product of the query vector and the document vector.

Dense Passage Retrieval (DPR) [16: Karpukhin *et al.* 2020], on the other hand, uses two independent BERT encoders to encode the query and document separately. During training, it is optimized through contrastive learning to make the vectors of related queries and documents closer together.

ColBERT (Contextualized Late Interaction over BERT) [21: Khattab and Zaharia 2020] combines the advantages of Dual-encoder and Cross-encoder architectures. It first uses BERT to independently encode the query and document, then performs local interactions during the similarity computation phase. This approach maintains the efficiency of retrieval while capturing some interaction information between the query and document.

## 2.2 Generative Retrieval

The tradition language model approach can be regarded as an implementation of a Probabilistic model, which assumes that a query is generated from the language model of a document. The typical formula is:

$$P(Q|D) = \prod_{i=1}^{n} P(q_i|D) \tag{2.1}$$

where $P(Q|D)$ is the probability of the query $Q$ given the document $D$, and $P(q_i|D)$ is the probability of the $i^{th}$ term of the query given the document.

Traditional Language Models evaluate relevance by estimating the probability of generating the query from the document. In contrast, Generative Language Models, a modern deep learning model, learn language patterns from large amounts of text data and can generate new text consistent with the input context. The training objective of these models is


doi:10.6342/NTU202402960

to predict the next word in a sequence:

$$P(w_t|w_{1:t-1}) = P(w_t|w_1, w_2, \ldots, w_{t-1}) \tag{2.2}$$

According to [30: Tang *et al.* 2023], the definition of generative retrieval is formulating IR task as a sequence-to-sequence (Seq2Seq) generation problem. GENRE (Generative ENtity REtrieval) [22: De Cao *et al.* 2020] is one of the first attempts to apply generative models to the Information Retrieval (IR) field. It is based on generative pre-trained models such as BART[31: Lewis *et al.* 2019] or T5[32: Raffel *et al.* 2020] and generates the target entity's title in an autoregressive manner. Specifically, the model receives a query (or contextual text) and then generates the entity name most relevant to the query through autoregressive generation. This approach leverages the generative capabilities of pre-trained models to produce accurate and contextually appropriate entity names, enhancing the retrieval process by focusing on generating relevant entity titles.

Subsequent developers have advanced the application of Generative Language Models, primarily by enhancing their capability to generate unique document identifiers (IDs), including both numeric and text identifiers. Unlike GENRE, which focuses on generating textual identifiers, DSI (Differentiable Search Index) [1: Tay *et al.* 2022] pioneered the generation of numeric identifiers in an autoregressive fashion. This approach requires the model to learn to map textual content to numeric outputs. Furthermore, DSI explores various methods for indexing and defining IDs to generate accurate numeric identifiers.

For models generating text identifiers, although GENRE does not explicitly label them as identifiers, titles can indeed serve as a form of text identifier. SEAL [4: Bevilacqua *et al.* 2022] generates substrings as document identifiers, using the FM-index to map these


doi:10.6342/NTU202402960

substrings to their corresponding documents. This method combines the advantages of generative modeling with efficient indexing, thereby enhancing document retrieval.

As Generative Retrieval is a nascent and exploratory field, recent advancements have introduced various techniques to mitigate existing issues in Generative Retrieval Models. For instance, DSI-QG [3: Zhuang *et al.* 2022] identified a data mismatch between training and testing data in the original training method of DSI, which affected the model's effectiveness in memorizing corpus information. To address this, DSI-QG incorporated the Query Generation technique, replacing document text in the training data with pseudo-queries, resulting in significant improvements.

Beyond Query Generation, numerous challenges faced by Generative Retrieval Models, such as handling dynamic corpora or training strategy optimization, are being actively explored and addressed using cross-disciplinary techniques [3], [5]–[8], [11]–[15], [33: Tang *et al.* 2024]. These ongoing efforts aim to enhance the robustness and applicability of Generative Retrieval Models in diverse and dynamic information retrieval scenarios.

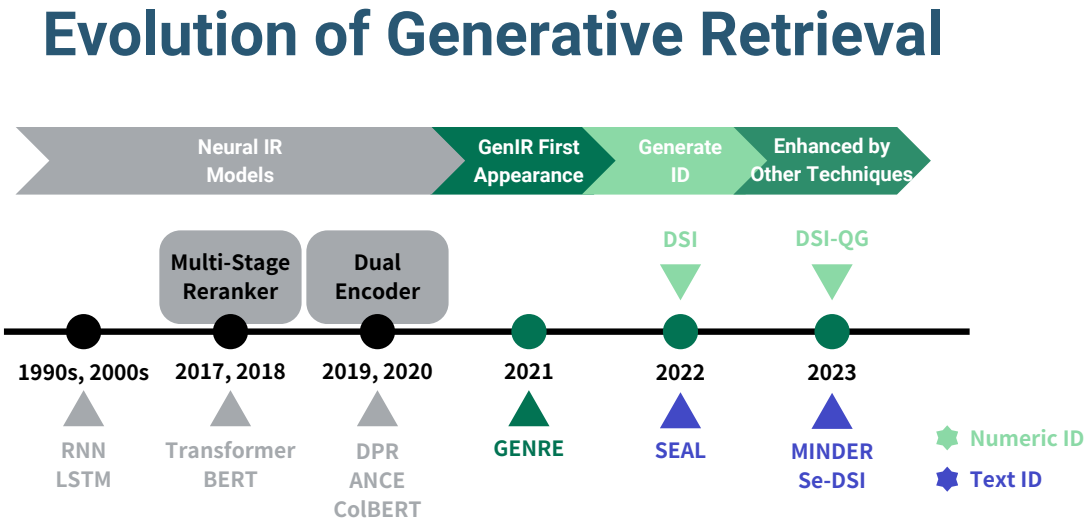



**Figure 2.1: The Evolution of Generative Retrieval.**


doi:10.6342/NTU202402960

# Chapter 3

# Methodology

## 3.1 Problem Statement.

Given the corpus $D$ and a query $q$, we aim to train a Transformer-based sequence-to-sequence model to retrieve documents by directly generating top-$k$ document identifiers from $D$ so that relevant documents can be ranked as high as possible.

## 3.2 Model Overview.

As shown in Figure 3.1, our model first generates $n$ pseudo queries for each document as the document concepts. Simultaneously, the model extracts $m$ n-gram keyphrases (word-level n-gram as keyphrases) from each document as explicit information for alignment. A Transformer-based language model is then trained to map each pseudo query and the $m$ extracted keyphrases to a concept-aware semantic identifier ($CSID$). This $CSID$ encompasses concept-level semantics and supports multiple non-deterministic indexing points for the document. Specifically, each document can have up to $n$ $CSID$s for retrieval in a hierarchical structure based on the embeddings of pseudo queries and documents.

## 3.3 Concept Alignment with Document Information.

Since we treat each generated pseudo query as a concept and bind it to a corresponding $CSID$, the model might struggle to distinguish between $CSID$s constructed by similar concepts in different documents. To address this, the generated concepts must be


doi:10.6342/NTU202402960

aligned with the belonging documents. In particular, a pseudo query from a document is subtly tuned to reflect the key of the overall document content.

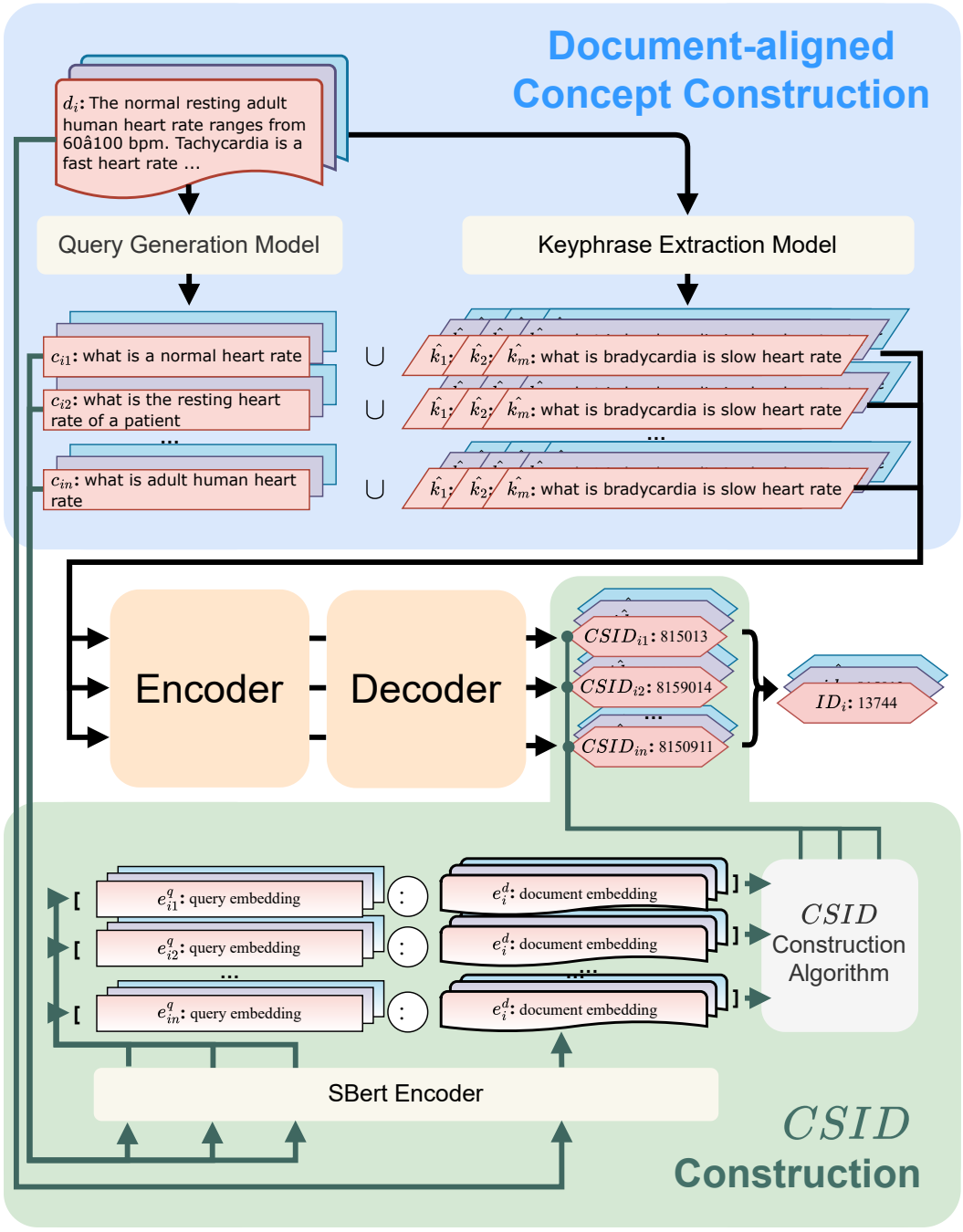



**Figure 3.1: The proposed Multi-DSI model. We construct document-aligned concepts as the input of Transformer. Meanwhile, we construct non-deterministic concept-aware semantic identifiers as the constrained output space.**


doi:10.6342/NTU202402960

## 3.4 Components in Multi-DSI

### 3.4.1 Document-aligned Concept Construction

We first construct concepts as the input for the seq-to-seq model for DSI. DSI-QG [3: Zhuang *et al.* 2022] highlights the importance of bridging the gap between input data for effective indexing and retrieval. Specifically, the input formats for indexing and retrieval should be similar. We follow this approach by generating pseudo queries to form the concepts in each document $d_i$. Unlike DSI-QG, which binds all $n$ pseudo queries to a single identifier, our method uses these $n$ pseudo queries to provide up to $n$ diverse concepts and retrieval paths for $d_i$. We employ a query generation model QG to generate question set $\hat{Q}_i$, which represent the concepts $C_i^q$ mentioned in $d_i$:

$$C_i^q = \hat{Q}_i = \{\ c_{ij} \in \text{QG}(d_i)\ \}, \tag{3.1}$$

where each $c_{ij} \in C_i^q$ represents $j^{th}$ concept of $d_i$ in the question format to mimic the query in testing phase.

Different documents $d_a$ and $d_b$ ($a \neq b$) could contain the same concept $c_{as} \in C_a^q$ and $c_{bs} \in C_b^q$, so concepts $c_{ij} \in C_i^q$ are better to be aligned with corresponding documents $d_i$ to tackle potential ambiguity during retrieval. Although a concept could occur in multiple documents, each document can have some unique keyphrases. For example, two medical documents could mention the same concept (e.g., normal heartbeat rate), but their main themes could be completely different, such as one listing symptoms and the other teaching how to self-check.

In this paper, we apply a keyphrase extraction model KE to derive $m$ n-gram keyphrases


doi:10.6342/NTU202402960

$\hat{k} \in \hat{K}_i$ for $d_i$. The extracted keyphrases are then transformed into the question format by a template:

$$\hat{K}_i = \text{KE}(d_i);\ C_i^d = \{\ \text{template}(\hat{k}) \mid \hat{k} \in \hat{K}_i\ \}, \tag{3.2}$$

where $|C_i^d| = m$. Compared with $C_i^q$, $C_i^d$ represents more general document information.

Finally, all document-aligned concept as a question set $C_i$ for the document $d_i$ can be derived by combining each $c_{ij} \in C_i^q$ with $C_i^d$:

$$C_i = \{\ A_{ij} = \text{assemble}(c_{ij}, C_i^d) \mid c_{ij} \in C_i^q\ \}, \tag{3.3}$$

where assemble$(x, y)$ denotes the question set constructed by adding pseudo query $x$ into question set $y$. The action treats $c_{ij}$ as a concept for $d_i$ and incorporates document information $C_i^d$ to distinguish concepts that appear in multiple documents. We then map all $c$ in each assembled question set $A_{ij}$ to corresponding $CSID_{ij}$:

$$\text{LM}_\theta(c) = CSID_{ij},\ \forall c \in \text{A}_{ij} \tag{3.4}$$

Note that the models QG and KE can be arbitrary query generation and n-gram extraction models. We refer more implementation details in Section 4.

### 3.4.2 Non-deterministic Concept-aware Semantic Identifier (CSID)

To construct non-deterministic concept-aware semantic identifiers $\text{CSID}_{ij}$ for $A_{ij} \in C_i$ of $d_i$, concept alignment is also inevitable. Every single concept $c_{ij}$ should be coupled with document information from $d_i$, thereby ensuring the balance in the mapping information between the inputs of the encoder ($A_{ij}$) and the outputs of the decoder ($\text{CSID}_{ij}$).


doi:10.6342/NTU202402960

We first encode $c_{ij}$ to $\mathbf{e}_{ij}^q \in \mathbb{R}^l$ and take it as the information about a single concept. As for the information of the document $d_i$, we encode the whole content of $d_i$ (if the content exceeds the max sequence length, we will truncate it) and acquire $\mathbf{e}_i^d \in \mathbb{R}^l$. Then, we concatenate them and take the result $\mathbf{e}_{ij} \in \mathbb{R}^{2l}$ as the start of generating $\text{CSID}_{ij}$:

$$\mathbf{e}_{ij} = [\mathbf{e}_{ij}^q : \mathbf{e}_i^d] \tag{3.5}$$

Finally, we generate concept-aware semantic identifiers $CSID$ by applying a clustering-based method similar to DSI[1: Tay *et al.* 2022] with $\mathbf{e}_{ij}$ across the entire corpus $D$. It's worth noting that a document will ultimately acquire multiple $CSID$, making this construction process non-deterministic. The vectors will be clustered into $k$ clusters using K-means. This process will be iterated in each derived cluster until the number of vectors in the derived cluster is less than $evoke\_size$. This algorithm finally constructs a tree-like structure. To determine the $CSID$ of a concept, we can traverse to the leaf cluster containing the concept, and the path will compose the $CSID$. More specifically, the level of the cluster determines the digit, and the order of the cluster at the level is the value of the digit. For the details, please refer to Algorithm 1 and Figure for the details of CSID construction.

With document-aligned concepts $C_i$ for every $d_i \in D$, Multi-DSI learns a Transformer-

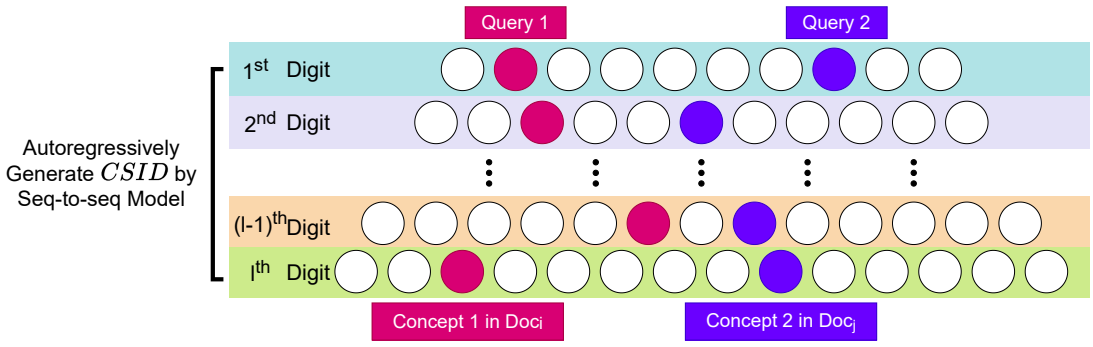



**Figure 3.2: How Multi-DSI retrieves concepts by $CSID$.**


doi:10.6342/NTU202402960

based model to map each question set $A_{ij} \in C_i$ to $CSID_{ij}$ as Equation 3.4. Following DSI [1: Tay *et al.* 2022], we leverage T5[32: Raffel *et al.* 2020][1] as our backbone model. Accordingly, $d_i$ has $n$ pairs of $A_{ij}$ & $\text{CSID}_{ij}$ as non-deterministic indexing points.

## 3.5 Model Training.

We use normal sequence-to-sequence binary cross-entropy loss to learn the mapping

[1]Due to the limitation to computation resource, we use T5-base in our experiment: https://hugging-face.co/google-t5/t5-base

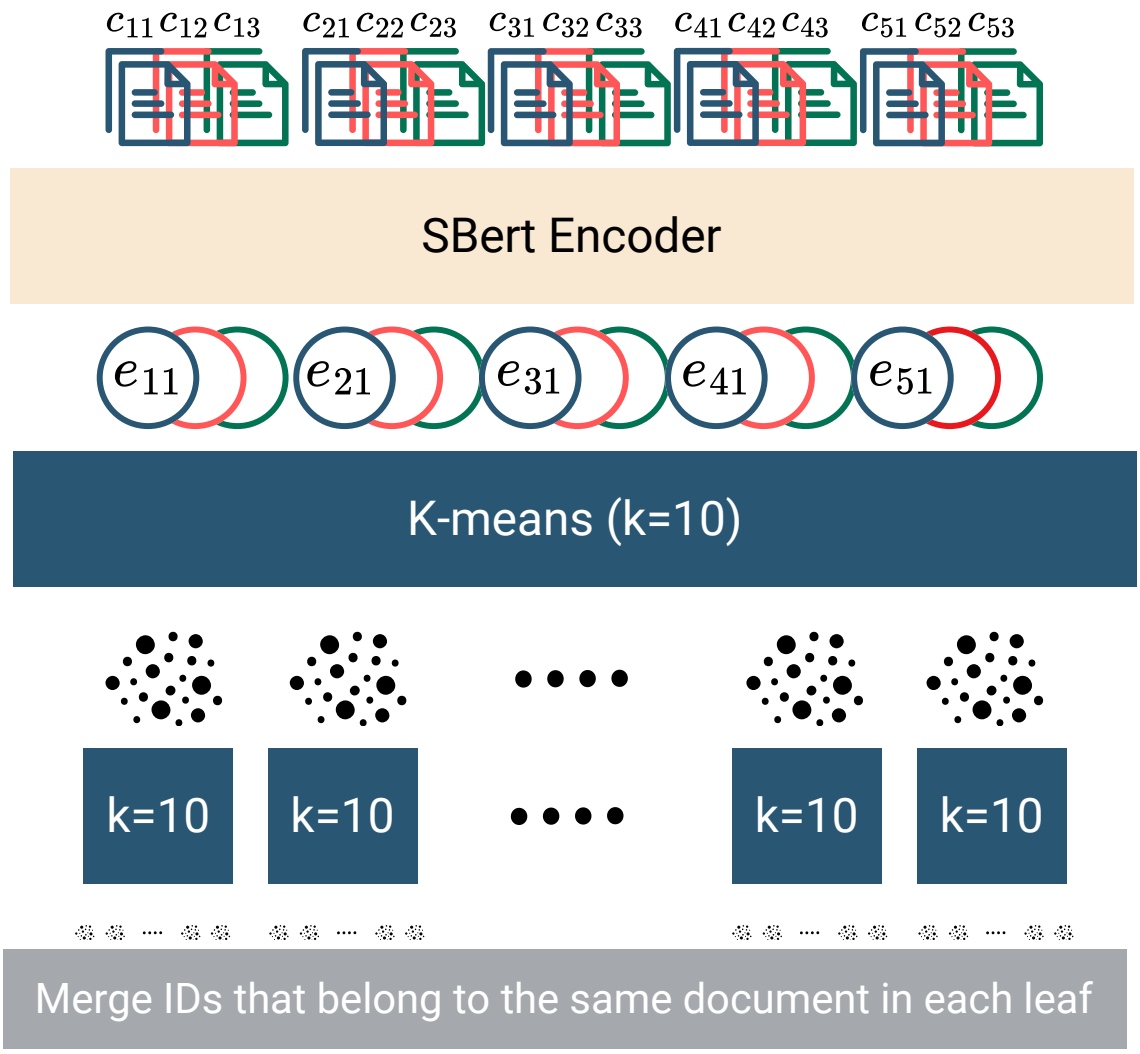



**Figure 3.3: How $CSID$s are constructed. Take 5 documents for example, the query generation model will generate $n$ concepts for each document. Then the encoder will produce concept embeddings. These embeddings will then be iteratively clustered until the number of concepts in each cluster is smaller than the threshold $evoke\_size$.**


doi:10.6342/NTU202402960

**Algorithm 1:** Clustering algorithm for generating $CSID$

**Input:** All $e_{ij}$ in $D$
**Output:** $CSID\_dict$
**Result:** Corresponding $CSID$ for each $e_{ij}$
**Data:** $k$ = # of cluster K-Means generated ;
$evoke\_size$ = the size evokes K-means for child clusters ;
$CSID\_dict$ = dictionary whose key is $e$ and value is $CSID$ ;

```
Function generateCSIDs( es = [ e1, e2, ... ] ) is
    foreach cluster in K-Means( es, k ) do
        es_c = [ those e ∈ es in cluster ] ;
        foreach e in es_c do
            Add cluster_number after CSID_dict[ e ] ;
        end
        if | es_c | > evoke_size then
            generateCSIDs( es_c ) ;
        else
            // merge CSIDs of the same doc in this leaf ;
            mergeAndUpdateCSIDs( es_c ) ;
        end
    end
end
```

relations between the pseudo-query texts and CSIDs. That is,

$$\mathcal{L}(\theta) = -\sum_{d_i \in D} \sum_{c_{ij} \in C_i^q} \sum_{c \in A_{ij}, C_i^d} \log \mathcal{P}(\text{CSID}_{ij} \mid \text{LM}_\theta(c)) \tag{3.6}$$

where LM means the Transformer-based Language Model.

## 3.6 Retrieval with a User Query.

Taking a user query as the input, the model autoregressively generates top-k CSIDs with beam search, which can be mapped to the respective documents in the corpus. The retrieval process is illustrated in Figure 3.2.


doi:10.6342/NTU202402960

| | MSMARCO | | Natural Questions | |
|---|---|---|---|---|
| | HIT@1 | HIT@10 | HIT@1 | HIT@10 |
| BM25 | 0.4670 | 0.7322 | 0.2927 | 0.6015 |
| DPR | 0.4716 | 0.7999 | 0.3820 | 0.6792 |
| $\text{DSI}_{naiveID}$ | 0.0099 | 0.0469 | 0.0670 | 0.2100 |
| $\text{DSI}_{semanticID}$ | 0.0943 | 0.2980 | 0.2740 | 0.5660 |
| SEAL | 0.3054 | 0.6851 | 0.3483 | 0.7266 |
| DSI-QG | 0.5893 | 0.8179 | **0.4725** | 0.6916 |
| Multi-DSI | **0.6424**$^{\dagger\ddagger}$ | **0.8915**$^{\dagger\ddagger}$ | 0.4713$^{\ddagger}$ | **0.7307**$^{\dagger}$ |

**Table 4.1: Retrieval performance on MSMARCO and Natural Questions datasets. $^{\dagger}$ and $^{\ddagger}$ indicate significant improvements against DSI-QG and SEAL in a paired t-test at 99% level.**

# Chapter 4

# Experiments

## 4.1 Experimental Settings

### 4.1.1 Datasets.

We conduct experiments on two benchmark datasets, MSMARCO-100K (MSMARCO) [34: Nguyen *et al.* 2016] and Natural Questions 320K (NQ) [35: Kwiatkowski *et al.* 2019]. We randomly sample 93,020 and 6,980 training and evaluation query-document pairs from MSMARCO. Similarly, NQ has 307,373 and 7,830 training and evaluation pairs. There are 100,000 and 109,715 documents for MSMARCO and NQ, respectively. Note that the maximum document length in MSMARCO is 216, which is much shorter than 100,569 in NQ.

Additionally, in the MSMARCO dataset, only 2% of the documents have multiple labeled queries, whereas in the Natural Questions dataset, 70% of the documents have multiple labeled queries. This means that for MSMARCO, most of the documents' labeled queries appear either in the training set or in the testing set. This characteristic directly


doi:10.6342/NTU202402960

impacts DSI's performance on MSMARCO in Table 4.1, highlighting its vulnerability to zero-shot scenarios where the document's queries have not been seen during training. This lack of resilience in handling unseen queries during testing underscores a significant challenge for DSI in datasets like MSMARCO.

### 4.1.2 Baseline Methods.

We compare Multi-DSI against several statistical and deep learning methods. For lexical-based retrieval method BM25, we set $k1 = 1.5$, $b = 0.75$, and $\epsilon = 0.25$. DPR [16: Karpukhin *et al.* 2020] serves as a dense retrieval baseline. We also compare with generative IR baselines, including DSI [1: Tay *et al.* 2022] with naive string and semantic string identifiers, SEAL [4: Bevilacqua *et al.* 2022], and DSI-QG [3: Zhuang *et al.* 2022]. Note that MINDER [6: Li *et al.* 2023] is not compared in this paper because it additionally leverages the provided document structure rather than solely using document content as what this paper does. Except for BM25 and DSI, we conduct baseline experiments with source codes provided by the paper authors. It is also worthwhile to mention that we are unable to use the provided implementation to reproduce the reported scores in DSI-QG.

### 4.1.3 Evaluation Metrics

Our retrieval task is Question-Answer Document Retrieval. Given a query, we calculate the Hit@1, and Hit@10 for the retrieval document ID list. We demonstrate an example in Figure 4.1.

In information retrieval or recommendation systems, hit@k is used to measure the hit rate of the system in the top k returned results. Common metrics include hit@1 and hit@10. The formula is as follows:


doi:10.6342/NTU202402960

$$\text{Hit@}k = \frac{\text{Number of Hits at rank } k}{\text{Total Number of Queries}} \tag{4.1}$$

### 4.1.4 Implementation Details.

We utilize the doc2query T5 model pre-trained on MSMARCO as the QG model. Each document in MSMARCO-100K has 10 generated queries as 10 concepts, which could be enough for relatively short documents. For longer content, we generate 50 queries for each document in NQ. We leverage KeyBERT [36: Issa *et al.* 2023] as the KE model to select n-gram candidates by comparing the similarity between BERT embeddings of keywords and documents. In particular, we extract the top-5 keywords from 1-gram to 5-gram keywords for each document and then apply the *"What is"* template. To generate $e_{ij}$, we use a pre-trained 12-layer 768-dimension BERT model as the encoder. $evoke_size$ for generating $CSID$ is set to 100. Finally, Multi-DSI trains T5 with learning rate $5e$-4, warmup steps $100k$, and batch size 128 on 2 NVIDIA RTX A6000.

## 4.2 Experimental Results & Discussion

### 4.2.1 Comparison of Retrieval Performance

As shown in Table 4.1, Multi-DSI outperforms DSI-QG by 7.36% and 3.91% on MSMARCO and NQ in HIT@10. In NQ, SEAL performs better because it uses *BART*-

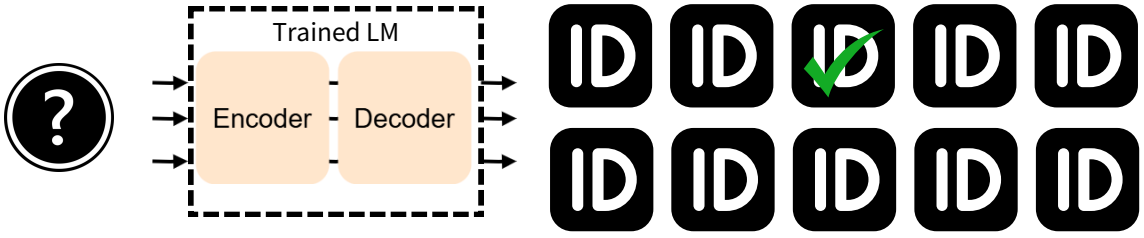
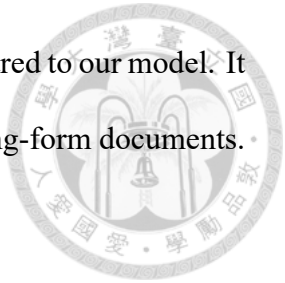



**Figure 4.1: In this example, LM generated $3^{th}$ ID is relevant to the query. Hence, Hit@$1 = 0$ and Hit@$10 = 1$**


doi:10.6342/NTU202402960



*Large*[31: Lewis *et al.* 2019] with 1.85 times model parameters compared to our model. It can also deal with twice longer documents but we need to truncate long-form documents. However, Multi-DSI can still obtain better performance.

When it comes to generative IR methods, we also observe that both DSI-QG and Multi-DSI utilizing query generation techniques have made significant progress from DSI. Multi-DSI performs even better because of multiple concept-level information with training texts in question format and identifiers.

It is also interesting that DSI performs worse in MSMARCO. This is because only a few documents in MSMARCO own more than 1 labeled query and thus a document won't exist in testing set if it is already in training set.

### 4.2.2 Ablation Study.

Table 4.2 shows the performance after removing each component in Multi-DSI on MSMARCO. The general document-level concept $C_i^d$ plays the most important role since it is a part of the training source. Removing it would result in an 8.56% drop in HIT@10. Similarly, removing the document embedding $e_i^d$ would also drop the HIT@10 performance by 4.19%. It could be because removing document information would fail in concept alignment, one of the key features in our model. Concepts are too ambiguous to be distinguished without being aligned to documents, especially when a concept exists in multiple documents. Besides, CSID also has positive impacts to Multi-DSI. Without CSID, documents are not equipped with multiple non-deterministic indexing points and can have a hard time being retrieved by different concepts.

### 4.2.3 Effectiveness of Multiple Document IDs.

To verify the effectiveness of providing multiple document IDs, we randomly select


doi:10.6342/NTU202402960

| | MSMARCO | | | |
|---|---|---|---|---|
| | HIT@1 | | HIT@10 | |
| Multi-DSI | 0.6424 | - | 0.8915 | - |
| $-C_i^d$ | 0.5577 | -8.47% | 0.8059 | -8.56% |
| $-e_i^d$ | 0.5831 | -5.93% | 0.8496 | -4.19% |
| $-CSID$ | 0.6279 | -1.45% | 0.8665 | -2.50% |

**Table 4.2: Performance changes after removing each component in Multi-DSI on MSMARCO.**

98 correctly retrieved documents that are relevant to multiple queries on MSMARCO and check how many unique $CSID$ are used for each document as shown in Figure 4.2. 81.6% of the documents have multiple generated CSIDs with the potential of providing multiple $CSID$ when 43.8% of these documents would utilize multiple CSIDs during retrieval. It demonstrates that multiple $CSID$ indeed plays an important role in Multi-DSI.

Additionally, we have observed that Multi-DSI shows more significant improvements on documents with multiple labeled queries compared to overall queries in Figure 4.3. SEAL, using substrings as identifiers, also exhibits multiple identifier characteristics, resulting in smaller improvements on documents with multiple labeled queries. However, progress is still observed, and the breakdown of documents to the concept level enables Multi-DSI to outperform SEAL overall.

In the case of DSI, due to the constraints imposed by the zero-shot nature of the testing

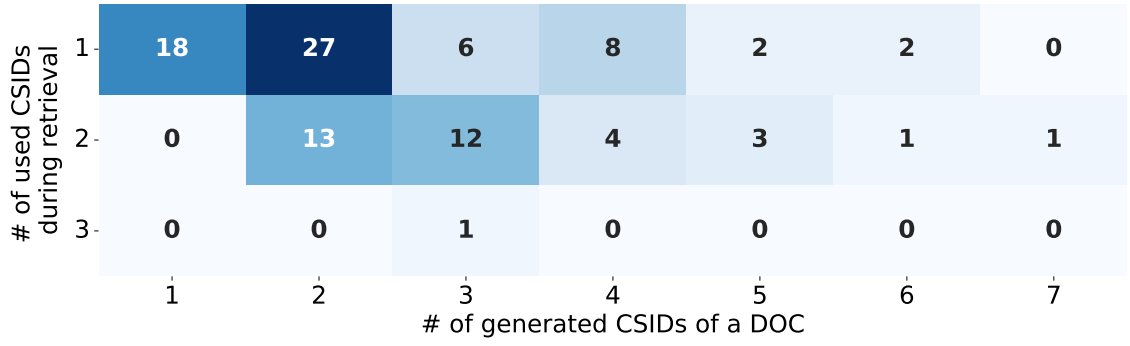



**Figure 4.2: The number of documents over different numbers of used CSIDs during retrieval and different numbers of generated CSIDs out of 98 correctly retrieved documents relevant to multiple queries on MSMARCO.**


doi:10.6342/NTU202402960

set, the overall performance was poor, leading to disproportionately large improvements for Multi-DSI. This indicates that while Multi-DSI excels in scenarios with multiple labeled queries, challenges remain in zero-shot contexts that need further exploration.

### 4.2.4 Need of Concept Alignment.

Figure 4.4 illustrates a concrete example of a generated query that requires concept alignment from MSMARCO. Although two documents generate the same query "what is normal heart rate" in query generation, they focus on distinct subjects about diseases and health checkups. Obviously, the generated query cannot distinguish the two documents.

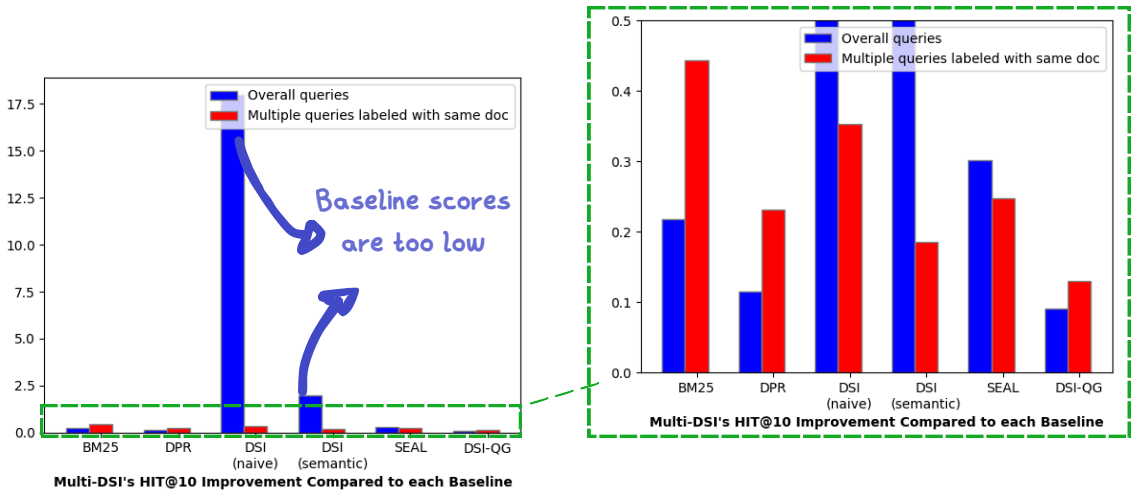



**Figure 4.3: The improvement of Multi-DSI compared with different baselines on overall queries and multiple queries labeled with the same document. Note that the unnormal high improvements compared to DSI are due to the zero-shot testing set problem mention in Section. 4.**

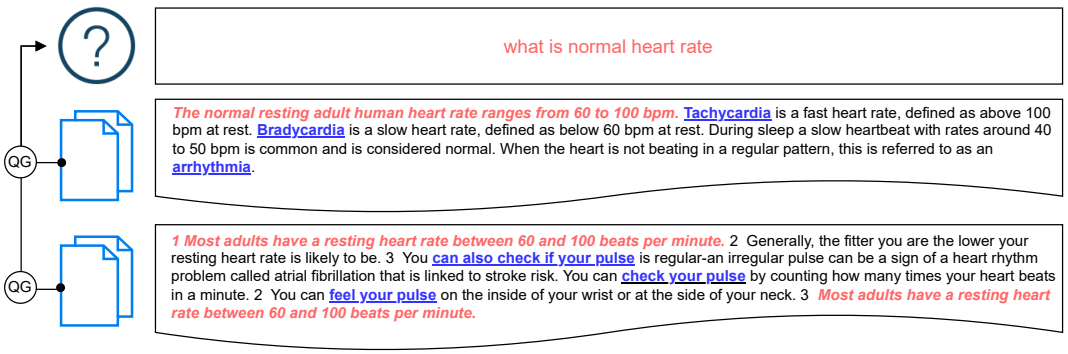



**Figure 4.4: A real example from MSMARCO about a generated query could happen in multiple documents. Red sentences are the concept of the generated query, while the blue sentences are the other concepts.**


doi:10.6342/NTU202402960

However, with certain keywords extracted from the document, concepts can be aligned to the documents, thereby easing the retrieval process.

### 4.2.5 Study on k of K-means used for CSID construction

In the construction of CSID referring to Algorithm 1, $k$ not only represents the number of clusters in iterative K-means clustering but also the numerical range of each digit in the CSID. For example, $k = 10$ indicates that the digit range is from 0 to 9, and $k = 2$ indicates a range of 0 to 1, and so on. Ideally, a larger $k$ corresponds to a higher semantic precision for each clustering iteration. With the same number of concepts or documents, a larger $k$ means fewer K-means iterations are required, resulting in shorter CSID lengths.

With this foundational understanding, we can observe that as $k$ increases, the CSID length shortens, and the performance of Multi-DSI improves in Figure 4.5. This improvement is likely due to the reduced number of decodings required by the model when CSID length is shorter, along with the higher semantic precision embedded in each decoding step.

### 4.2.6 Study on $clustering\_evoke\_size$ used for CSID construction

In CSID construction referring to Algorithm 1, $clustering\_evoke\_size$ affects the upper limit on the number of concepts or documents contained within a leaf cluster. When the number of concepts or documents is less than the $clustering\_evoke\_size$, the concepts or documents within that cluster are assigned sequentially with non-semantic numerical digits. As $clustering\_evoke\_size$ increases, the proportion of non-semantic content within a CSID also increases, leading to a deterioration in performance. This is because a higher $clustering\_evoke\_size$ results in more assignments of non-semantic digits, thereby reducing the overall semantic richness of the CSID in Figure 4.6.


doi:10.6342/NTU202402960

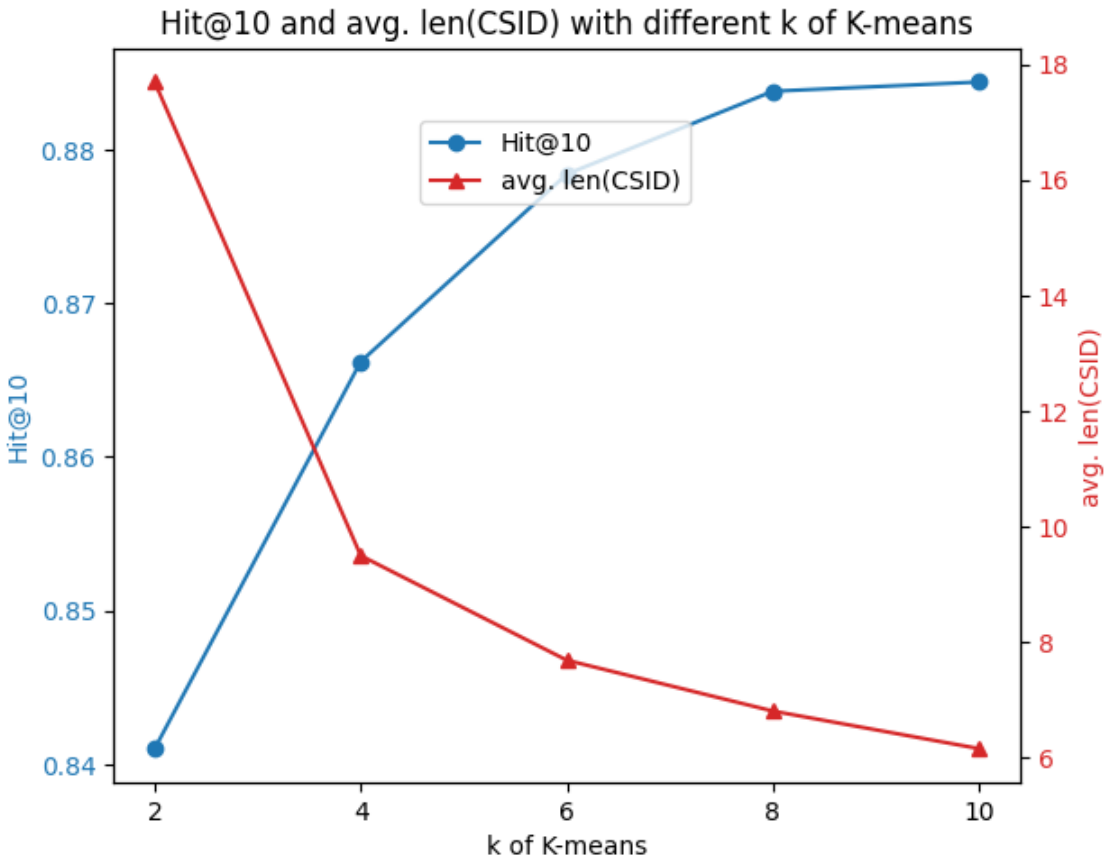



**Figure 4.5: Hit@10 and average of $CSID$ length with different $k$ of K-means.**



doi:10.6342/NTU202402960

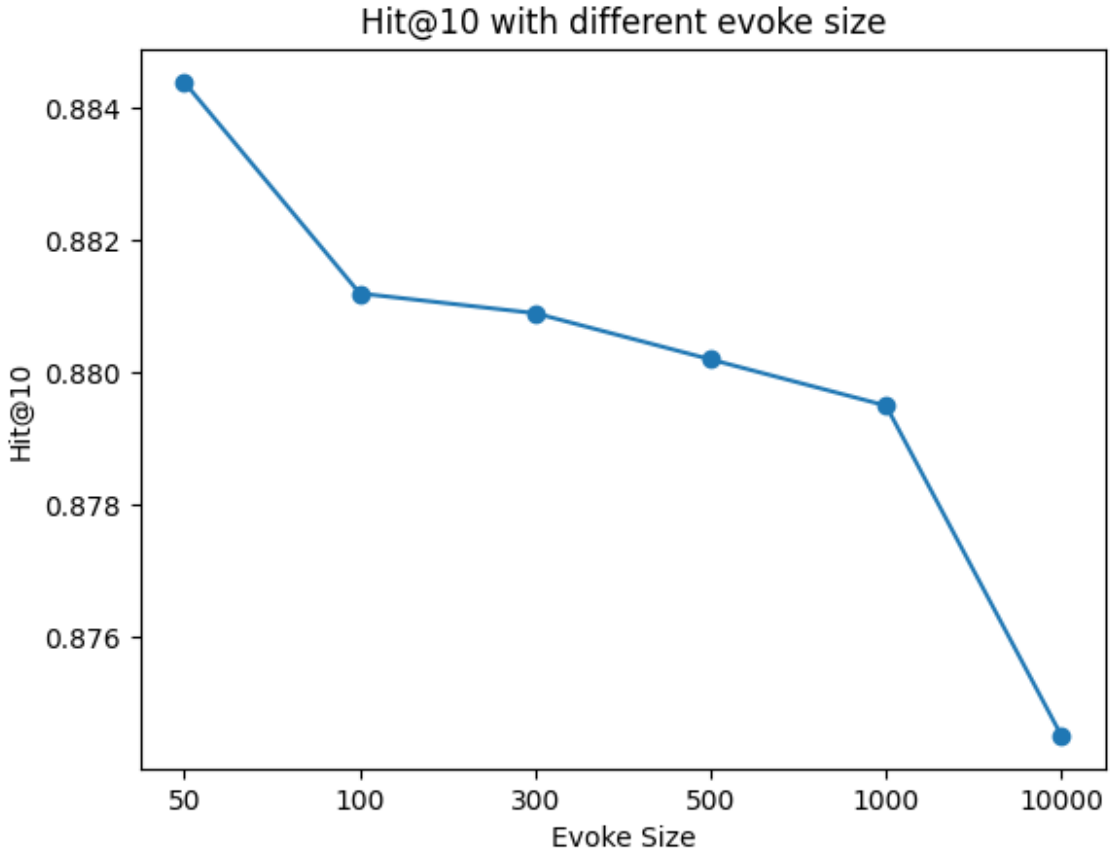



**Figure 4.6: Hit@10 with different** $clustering\_evoke\_size$**.**


doi:10.6342/NTU202402960

# Chapter 5

# Conclusion

In this paper, we propose the Multi-DSI model to enhance DSI by constructing non-deterministic semantic identifiers as multiple indexing points for each document. Moreover, we introduce the idea of concept alignment to deal with the issue of generated ambiguous queries. Experimental results on two benchmark datasets demonstrate that Multi-DSI can significantly improve the retrieval performance against several baseline methods. In-depth analysis also indicates the effectiveness of each component in Multi-DSI. Our conclusion can be summarized in three-fold: (1) A single identifier might be insufficient as a representation for a document with diverse topics; (2) Aligning concepts or generated pseudo queries to a document is critical to distinguish similar documents; (3) For generative IR, it is important to learn models with the inputs in the same format as queries. We leave (1) Improve non-deterministic ID by extending its diversity. (e.g. more than 10 numbers in each digit or incorporating with other information.) (2) Make CSID construction algorithm adapt to datasets rather than using predefined parameters (3) The scalability for large datasets or real-time applications as our future work.



doi:10.6342/NTU202402960


doi:10.6342/NTU202402960

# References


[1: Tay, Tran, Dehghani, Ni, Bahri, Mehta, Qin, Hui, Zhao, Gupta, *et al.* 2022]

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, *et al.*, "Transformer memory as a differentiable search index," *Advances in Neural Information Processing Systems*, vol. 35, pp. 21 831–21 843, 2022.

[2: Metzler, Tay, Bahri, and Najork 2021]

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork, "Rethinking search: Making domain experts out of dilettantes," in *Acm sigir forum*, ACM New York, NY, USA, vol. 55, 2021, pp. 1–27.

[3: Zhuang, Ren, Shou, Pei, Gong, Zuccon, and Jiang 2022]

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

[4: Bevilacqua, Ottaviano, Lewis, Yih, Riedel, and Petroni 2022]

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni, "Autoregressive search engines: Generating substrings as document identifiers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 668–31 683, 2022.

[5: Tang, Zhang, Guo, Chen, Zhu, Wang, Yin, and Cheng 2023]

Yubao Tang, Ruqing Zhang, Jiafeng Guo, Jiangui Chen, Zuowei Zhu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng, "Semantic-enhanced differentiable search index inspired by learning strategies," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4904–4913.

[6: Li, Yang, Wang, Wei, and Li 2023]

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li, "Multiview identifiers enhanced generative retrieval," in *Proceedings of the 61st Annual Meeting of the*




doi:10.6342/NTU202402960

*Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Jul. 2023, pp. 6636–6648.

[7: Wang, Hou, Wang, Miao, Wu, Chen, Xia, Chi, Zhao, Liu, *et al.* 2022]

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, *et al.*, "A neural corpus indexer for document retrieval," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 600–25 614, 2022.

[8: Li, Yang, Wang, Wei, and Li 2024]

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li, "Learning to rank in generative retrieval," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 8716–8723.

[9: Nguyen and Yates 2023]

Thong Nguyen and Andrew Yates, "Generative retrieval as dense retrieval," *arXiv preprint arXiv:2306.11397*, 2023.

[10: Sun, Yan, Chen, Wang, Zhu, Ren, Chen, Yin, Rijke, and Ren 2024]

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren, "Learning to tokenize for generative retrieval," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[11: Mehta, Gupta, Tay, Dehghani, Tran, Rao, Najork, Strubell, and Metzler 2022]

Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler, "Dsi++: Updating transformer memory with new documents," *arXiv preprint arXiv:2212.09744*, 2022.

[12: Wang, Zhou, Tu, and Dou 2023]

Zihan Wang, Yujia Zhou, Yiteng Tu, and Zhicheng Dou, "Novo: Learnable and interpretable document identifiers for model-based ir," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 2656–2665.



doi:10.6342/NTU202402960

[13: Zhang, Liu, Zhou, Dou, and Cao 2023]

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

[14: Ren, Zhao, Liu, Wu, Wen, and Wang 2023]

Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang, "Tome: A two-stage approach for model-based retrieval," *arXiv preprint arXiv:2305.11161*, 2023.

[15: Lee, Kim, Chang, Oh, Yang, Karpukhin, Lu, and Seo 2022]

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

[16: Karpukhin, Oğuz, Min, Lewis, Wu, Edunov, Chen, and Yih 2020]

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[17: Lee, Sung, Kang, and Chen 2020]

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen, "Learning dense representations of phrases at scale," *arXiv preprint arXiv:2012.12624*, 2020.

[18: Xiong, Xiong, Li, Tang, Liu, Bennett, Ahmed, and Overwijk 2020]

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk, "Approximate nearest neighbor negative contrastive learning for dense text retrieval," *arXiv preprint arXiv:2007.00808*, 2020.

[19: Wang, Yang, Huang, Jiao, Yang, Jiang, Majumder, and Wei 2022]

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei, "Text embeddings by weakly-supervised contrastive pre-training," *arXiv preprint arXiv:2212.03533*, 2022.

[20: Wang, Yang, Huang, Jiao, Yang, Jiang, Majumder, and Wei 2022]

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang,



doi:10.6342/NTU202402960

Rangan Majumder, and Furu Wei, "Simlm: Pre-training with representation bottleneck for dense passage retrieval," *arXiv preprint arXiv:2207.02578*, 2022.

[21: Khattab and Zaharia 2020]

Omar Khattab and Matei Zaharia, "Colbert: Efficient and effective passage search via contextualized late interaction over bert," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 39–48.

[22: De Cao, Izacard, Riedel, and Petroni 2020]

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni, "Autoregressive entity retrieval," *arXiv preprint arXiv:2010.00904*, 2020.

[23: Zhou, Yao, Dou, Wu, Zhang, and Wen 2022]

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

[24: Rong 2014]

Xin Rong, "Word2vec parameter learning explained," *arXiv preprint arXiv:1411.2738*, 2014.

[25: Lau and Baldwin 2016]

Jey Han Lau and Timothy Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," *arXiv preprint arXiv:1607.05368*, 2016.

[26: Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin 2017]

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27: Devlin, Chang, Lee, and Toutanova 2018]

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-



doi:10.6342/NTU202402960

training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[28: Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer, and Stoyanov 2019]

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[29: Reimers and Gurevych 2019]

Nils Reimers and Iryna Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[30: Tang, Zhang, Guo, and Rijke 2023]

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

[31: Lewis, Liu, Goyal, Ghazvininejad, Mohamed, Levy, Stoyanov, and Zettlemoyer 2019]

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[32: Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li, and Liu 2020]

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[33: Tang, Zhang, Guo, Rijke, Chen, and Cheng 2024]

Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng, "Listwise generative retrieval models via a sequential learning process," *ACM Transactions on Information Systems*, vol. 42, no. 5, pp. 1–31, 2024.



doi:10.6342/NTU202402960

[34: Nguyen, Rosenberg, Song, Gao, Tiwary, Majumder, and Deng 2016]

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

[35: Kwiatkowski, Palomaki, Redfield, Collins, Parikh, Alberti, Epstein, Polosukhin, Devlin, Lee, *et al.* 2

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, *et al.*, "Natural questions: A benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.

[36: Issa, Jasser, Chua, and Hamzah 2023]

Bayan Issa, Muhammed Basheer Jasser, Hui Na Chua, and Muzaffar Hamzah, "A comparative study on embedding models for keyword extraction using keybert method," in *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)*, IEEE, 2023, pp. 40–45.



doi:10.6342/NTU202402960