Bottleneck-Minimal Indexing for Generative Document Retrieval

Xin Du [*1] **Lixin Xiu** [*2] **Kumiko Tanaka-Ishii** [3]

Abstract

We apply an information-theoretic perspective to reconsider generative document retrieval (GDR), in which a document $x \in \mathcal{X}$ is indexed by $t \in \mathcal{T}$, and a neural autoregressive model is trained to map queries $\mathcal{Q}$ to $\mathcal{T}$. GDR can be considered to involve information transmission from documents $\mathcal{X}$ to queries $\mathcal{Q}$, with the requirement to transmit more bits via the indexes $\mathcal{T}$. By applying Shannon's rate-distortion theory, the optimality of indexing can be analyzed in terms of the mutual information, and the design of the indexes $\mathcal{T}$ can then be regarded as a *bottleneck* in GDR. After reformulating GDR from this perspective, we empirically quantify the bottleneck underlying GDR. Finally, using the NQ320K and MARCO datasets, we evaluate our proposed bottleneck-minimal indexing method in comparison with various previous indexing methods, and we show that it outperforms those methods.

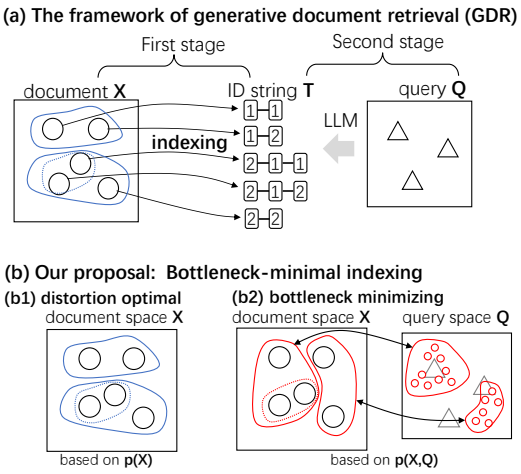


Figure 1: (a) Generative document retrieval (GDR) framework. (b) Our contribution using bottleneck-minimal indexing: (b-1) distortion-optimal indexing for documents $\mathcal{X}$; (b-2) optimal indexing for both documents and queries $\mathcal{Q}$.

1. Introduction

The importance of accurate document retrieval is increasing, especially with the limitation of recent large language models. Generative document retrieval (GDR) is a new, promising framework for information retrieval. First, every document $x \in \mathcal{X}$ (a document set) is represented by a short, distinct identifier string $t \in \mathcal{T}$ (the identifier set). Then, an autoregressive model, typically a neural network, is trained to map a query $q \in \mathcal{Q}$ (a query set) to an identifier string t representing a document x. Figure 1(a) schematically illustrates GDR.

GDR thus typically involves two stages, and the main research question has been about how to acquire good identifier strings $\mathcal{T}$ for documents $\mathcal{X}$ in the first stage. Previously, Tay et al. (2022) clustered $\mathcal{X}$ according to vectorial embeddings generated by the BERT model (Devlin et al., 2019), by using the k-means algorithm. Bevilacqua et al. (2022) used a substring of each document as its ID string. All of those works were based on good intuition about how to semantically partition $\mathcal{X}$, but we can theoretically reconsider how best to partition it in terms of $\mathcal{T}$.

For such reconsideration, it is natural to apply the rate-distortion theory initiated by Shannon (Shannon, 1948; 1959; Berger, 1968), which is based on the mutual information. Let X, T, and Q denote random variables defined over $\mathcal{X}$, $\mathcal{T}$, and $\mathcal{Q}$, respectively. As the rate-distortion theory only involves two terms corresponding to X and T, the aim, in short, is to acquire T that minimally distorts X. All previous GDR proposals thus consider its distortion optimality, as illustrated in Figure 1(b-1).

In this paper, we deal with the particularity of the second stage, which involves mapping Q to T. The design of T, which we call *indexing* here, was previously considered only with respect to X, but the best T can theoretically be obtained by considering both X and Q. For this purpose, we apply the information bottleneck theory by Tishby et al.

[*]Equal contribution [1]Waseda Research Institute for Science and Engineering, Waseda University [2]Department of Mathematical Informatics, The University of Tokyo [3]Department of Computer Science and Engineering, Waseda University. Correspondence to: Xin Du <duxin@aoni.waseda.jp>, Kumiko Tanaka-Ishii <kumiko@waseda.jp>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

(2000), an extension of rate-distortion theory.

Under this bottleneck theory, GDR involves information transmission from documents to queries, with the requirement to transmit more bits via T. Obviously this suggests a tradeoff between the two relations $T \leftrightarrow X$ and $T \leftrightarrow Q$, and T then becomes a *bottleneck* of Q in searching X. Our paper's main contribution is to provide a theoretical formalization to evaluate the quality of T in terms of not only X but also Q.

The bottleneck suggests that the optimal T is determined by the probability distribution of queries rather than documents, because Q is posterior to X. This suggests a novel design for T by clustering Q rather than X. We experimentally evaluate this approach's effectiveness in comparison with previous methods, including k-means (Hartigan & Wong, 1979) and locality-sensitive hashing (Datar et al., 2004). The results show that our idea indeed improves the Recall@1 score on the NQ320K and MARCO Lite datasets, by 1.26 and 3.72 points, respectively, with a finetuned `T5-base` model. The margins increase greatly to 7.06 and 6.45 points, respectively, for `T5-mini`, which has fewer parameters. The code is available at `https://github.com/kduxin/Bottleneck-Minimal-Indexing`.

2. Related Works

We start with an overview of the recent progress in GDR in Section 2.1. In Section 2.2, we briefly summarize methods in discrete representation learning, which provide options for indexing methods in GDR. In Section 2.3, we introduce the historic rate-distortion theory initiated by Shannon and the bottleneck theory by Tishby et al. (2000).

2.1. Generative Document Retrieval

Conventional document retrieval methods use a score function to quantify the degree of relevance between a document and a query. Score functions have been manually chosen and include the Hamming distance, (Salakhutdinov & Hinton, 2009), TF-IDF (Manning & Schutze, 1999), BM25 score (Robertson et al., 2009), and cosine similarity between vectoral embeddings of texts (Mikolov et al., 2013; Devlin et al., 2019; Karpukhin et al., 2020; Khattab & Zaharia, 2020). Alternatively, score functions can also be learned from data, e.g., based on a learning-to-rank loss function (Burges et al., 2005; Cao et al., 2007; Li et al., 2023).

In contrast, generative document retrieval aims to directly generate a document x from a query q by using a sequence-to-sequence model. In relation to indexing, Cao et al. (2021) proposed the concept of the entity retrieval task, in which a model generates an *entity* comprising either words, phrases, or article titles. While documents are typically long, with hundreds of words, an entity is short, and their work showed the potential effectiveness of using entities as indexes. Since then, several works have studied generation of short textual summaries to represent long documents (Bevilacqua et al., 2022; Chen et al., 2022). Using short texts as indexes was also explored for recommendation tasks (Geng et al., 2022).

Tay et al. (2022) generalized this idea as a *differentiable search index* (DSI), a more straightforward indexing method to obtain document ID strings that are acquired by applying hierarchical k-means clustering to a set of document vectoral embeddings produced by a BERT model (Devlin et al., 2019). In DSI, a document set $\mathcal{X}$ is partitioned into clusters that are organized hierarchically (by having subclusters in each cluster). An identifier number represents each cluster, and a document is identified by a sequence of identifiers t representing the hierarchical clusters. This simple method was further explored in Wang et al. (2022); Mehta et al. (2023); Tang et al. (2023); Zeng et al. (2023); Rajput et al. (2023). Recent works (Sun et al., 2023; Jin et al., 2023) proposed to learn the index $\mathcal{T}$ by using an autoencoder model with $\mathcal{X}$.

While all these previous works consider organization of $\mathcal{X}$ to design $\mathcal{T}$, our contribution is to also include $\mathcal{Q}$ in the designed approach.

2.2. Discrete Representation Learning

Acquisition of $\mathcal{T}$ can be considered as one kind of representation learning. Learning of discrete representations has been studied in multiple domains, as summarized below.

Vector quantization (VQ). VQ encodes vectors in a Euclidean space as discrete codes with the least distortion. Lloyd (1982); Wu & Yu (2019) recognized that the optimal vector quantization corresponds to the k-means clustering algorithm's results when the distortion is defined via the Euclidean distance. A Euclidean space can be partitioned by assuming different structures, such as a tree (Nister & Stewenius, 2006), in which case the approach corresponds to hierarchical k-means clustering. Deep learning techniques can be combined to learn better vector quantization (van den Oord et al., 2017).

Hash-based methods. A hash function maps vectors to discrete classes. Locality-sensitive hashing (LSH) (Datar et al., 2004) uses a property of p-stable distributions to construct a family of hash functions that is guaranteed to map close vectors to the same class. By using multiple independent hash functions, the dissimilarity of two vectors is measured by the Hamming distance. Semantic hashing (Salakhutdinov & Hinton, 2009) learns hash functions by using an autoencoder to acquire compact discrete representations. Many works since have studied the use of deep neural networks as hash functions (Venkateswara et al., 2017;

Chaidaroon & Fang, 2017; Jin et al., 2019; Lin et al., 2022).

2.3. Rate-Distortion Theory

Rate-distortion (RD) theory (Shannon, 1948) provides an information-theoretic framework for optimal indexing in GDR. RD theory is based on $I(X;T)$, the mutual information for transmitting data between X and T. Such transmission introduces distortion that is measured by a distortion function $d(X,T) \mapsto \mathbb{R}_+$. This function depends on the application, and the choice is sometimes not obvious (Slonim, 2002). The overall distortion $\mathbb{E}_{X,T}d(X,T)$ has a tradeoff with the transmission rate $I(X;T)$. For $p(T|X)$, the probability of T given X, let $p^*(T|X)$ be the probability when T is optimal for X via RD theory. In GDR, this T would be considered the optimal indexing of documents.

The vector quantization problem mentioned in Section 2.2 is closely related to RD theory (Gray, 1984). RD theory can be extended to include a third variable Q via the information bottleneck (IB) theory (Tishby et al., 2000), and the distortion function is implicitly determined via the mutual information $I(T;Q)$. Here, T is called an *information bottleneck* (IB) because it determines the "data transmission rate" between X and Q.

The IB theory was originally proposed for bottom-up hierarchical (i.e., agglomerative) clustering (Slonim & Tishby, 1999) applied to words (Slonim & Tishby, 2000). It has also been applied to explain the superior generalizability of feedforward deep neural networks (Tishby & Zaslavsky, 2015; Saxe et al., 2019).

We adopt IB theory to analyze GDR. The relation between documents X and queries Q can be complex, especially in cross-lingual (Grefenstette, 2012) or multimodal (Gabeur et al., 2020) scenarios. Hence, the distortion function $d(X,T)$ must involve the queries Q, which naturally leads to the IB theory.

3. Bottleneck-Minimal Indexing

3.1. Mathematical Setting

As mentioned above, in the GDR context, each document $x \in \mathcal{X}$ is indexed with an ID string $t \in \mathcal{T}$, described by a function $f(x) = t$. It is anticipated that information loss occurs during the application of f, as the semantic associations between documents are simplified into discrete ID strings. To quantify this information loss, the domain of f must be considered within an abstract semantic space of documents, denoted as $\mathscr{X}$. Here, a document is a point in $\mathscr{X}$, and the set $\mathcal{X}$ of all documents in a dataset is a subset of $\mathscr{X}$, i.e., $x \in \mathcal{X} \subset \mathscr{X}$. For convenience, Q is defined over the same semantic vector space $\mathscr{X}$, and T is defined over $\mathcal{T}$, the set of all ID strings.

The indexing function f is formally defined as $f : \mathscr{X} \to \mathcal{T}$. By leveraging $f^{-1} : t \mapsto \mathscr{X}_t \subset \mathscr{X}$, the semantic space $\mathscr{X}$ can be split into multiple regions. According to the standard GDR setting (Tay et al., 2022), it is required that $|\mathcal{T}| = |\mathcal{X}|$; every $t \in \mathcal{T}$ is associated with a distinct document x in $\mathcal{X}$, the only document in $f^{-1}(t) \cap \mathcal{X}$. When $|\mathcal{T}| < |\mathcal{X}|$, indicating that an ID string is associated with multiple documents, this configuration typically aligns with semantic-hash methods where hash collison is desirable (Salakhutdinov & Hinton, 2009).

3.2. Distortion Optimality

Previous works considered the design of $\mathcal{T}$ only with respect to $\mathcal{X}$. As mentioned above, Shannon's rate-distortion theory can be applied to acquire $\mathcal{T}$ as an optimal split of $\mathscr{X}$, formulated as:

$$\min_{p(T|X)} \quad I(X;T), \tag{1}$$

where $I(X;T)$ denotes the mutual information of X and T. This formalization allows for a reconsideration of the optimal T given X for previous works, as will be analyzed in Section 4.

Information retrieval involves another term, Q, and the optimization can be extended as follows:

$$\begin{aligned} \min_{p(T|X)} \quad & I(X;T) \\ \text{s.t.} \quad & I(T;Q) \geq \varepsilon, \end{aligned} \tag{2}$$

where $\varepsilon > 0$ is a threshold. This additional constraint is essential, as $I(X;T) = 0$ when T is constant, but it also decreases $I(T;Q)$. The mutual information $I(T;Q)$ is closely connected to the likelihood $p(T|Q)$ that reflects retrieval accuracy; therefore, having $I(T;Q)$ in the optimization problem facilitates optimization of retrieval accuracy.

Formula (2) characterizes the tradeoff between index conciseness, denoted by $I(X;T)$, and retrieval accuracy, linked to $I(T;Q)$. This dual perspective highlights the superiority of one indexing method over another: for a fixed $I(X;T)$, aiming for a higher $I(T;Q)$ improves retrieval accuracy. Conversely, with a constant $I(T;Q)$, the emphasis shifts towards more compact indexes, encouraging the sharing of semantics across the indexes.

This optimal split is equivalently reformulated via a Lagrangian $\mathcal{L}$ as follows:

$$\mathcal{L}(p(T|X)) = I(X;T) - \beta I(T;Q), \tag{3}$$

where β is the Lagrange multiplier. This coefficient β determines the relative importance of $I(T;Q)$ in $\mathcal{L}$. When $\beta \to 0$, the latter term can be ignored, and T shrinks to a single point so that the former term decreases to 0. Conversely, as β becomes large, T tends to simply copy X.

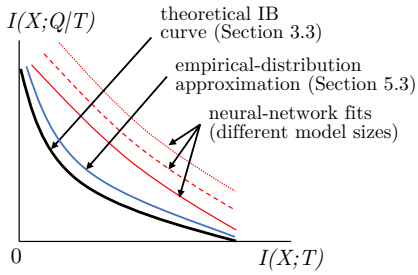


Figure 2: Bottleneck curves

Because the possibilities for X are observed via the subset $\mathcal{X}$, a large β corresponds to observing more documents.

To summarize, β is a hyperparameter related to the document set's size, $|\mathcal{X}|$. This is natural because the optimal indexing $p^*(T|X)$ would change if the number of documents increases. Nevertheless, some indexing method is optimal with respect to different β under certain assumptions, as we will show in Section 4.

3.3. GDR Bottleneck

We now examine the tradeoff, or bottleneck, between $I(X;T)$ and $I(T;Q)$. Tishby & Zaslavsky (2015) showed that the Lagrangian in Formula (3) is equivalent to the following:

$$\mathcal{L}(p(T|X)) = I(X;T) + \beta I(X;Q|T) + \text{constant}. \quad (4)$$

The equivalence requires assuming the Markovian relation $T \leftrightarrow X \leftrightarrow Q$, i.e., $p(T|X,Q) = p(T|X)$, and Appendix A.1 summarizes the proof.

In this formulation, the conditional mutual information $I(X;Q|T)$ quantifies the information distortion due to T. $I(X;Q)$ is constant and unrelated to the indexing T. Hence, a small $I(X;Q|T)$ means a good T for representing the joint distribution $p(X,Q)$.

As mentioned above, Tishby et al. (2000) defined this tradeoff as an *information bottleneck*. The bottleneck can be visualized by a curve in the space of the two terms of Formula (4), as shown in Figure 2. The lower bound of $I(X;Q|T)$ captures the optimal indexing, and it monotonically decreases as $I(X;T)$ increases. We will empirically show in Section 6.1 that this bottleneck curve indeed occurs for GDR.

3.4. A Theoretical Solution to Optimality

Tishby et al. (2000) showed that a stationarity point of the Lagrangian $\mathcal{L}$ in Formula (4) must satisfy the following equation:

$$p^*(T|X) = \frac{p^*(T)}{Z(X,\beta)} \exp\Big(-\beta \mathrm{KL}\big[p(Q|X) \,\|\, p(Q|T)\big]\Big), \quad (5)$$

where $Z(X,\beta)$ is a probability normalization term, $p^*(T) = \mathbb{E}_X[p^*(T|X)]$. The Kullback-Leibler divergence term reflects the information distortion caused by $f : x \mapsto t$ in terms of the discrepancy between $p(Q|x)$ and $p(Q|t)$.

In GDR, we pursue an indexing $f : x \mapsto t$ that is consistent with this optimal solution $p^*(T|X)$. That is, the identifier string $t = f(x)$ for a document x should be drawn from this "best" probability distribution $p^*(T|X = x)$. A natural way to incorporate this intuition is to evaluate f by a likelihood function $p(\mathcal{X}, \mathcal{Q}|f)$ defined as follows.

Definition 3.1. $f : \mathcal{X} \to \mathcal{T}$ is called a *bottleneck-minimal indexing* (BMI) if it maximizes the likelihood function as follows:

$$p(\mathcal{X}, \mathcal{Q}|f) \equiv \prod_{x \in \mathcal{X}} p^*\big(X = x \,\big|\, T = f(x)\big). \quad (6)$$

where p^* is given by Formula (5).

Definition 3.1 is presented in a general manner, intentionally avoiding assumptions about any specific distribution family for $p(Q|X)$ and $p(Q|T)$ as outlined in Formula (5). In Section 4, we will examine some existing indexing methods regarding their relations with BMI. Under this background, a new indexing method will be introduced in Section 4.4.

4. Indexing Methods

In the following, we consider $\mathscr{X}$ as the semantic vector space produced by a pretrained BERT model. In this section we introduce different indexing methods under this perspective and compare them theoretically, while Section 6.2 describes our experimental comparison.

An index $t \in \mathcal{T}$ is represented as a sequence of elements of *alphabet*, denoted as V. An ID string of length m is represented as $t = [t_1, \cdots, t_m]$, where $t_i \in V$ is the string's i-th "digit" for $i = 1, \cdots, m$.

4.1. Hierarchical Random Indexing (HRI)

We consider the most basic method, random indexing. For each document, an ID digit is randomly selected according to some prior distribution $p(T)$. $\mathcal{X}$ is partitioned into $|V|$ subsets. Furthermore, each subset is recursively partitioned for representation by V. The recursive subdivision constitutes a *hierarchy* of subdivisions of depth m. A special case arises when $|V| \geq |\mathcal{X}|$ and $m = 1$; such indexing is termed *atomic* (Tay et al., 2022), since $\mathcal{T}$ lacks hierarchical structure.

A representative partition is obtained when $p(Q|X)$ is uniformly distributed over a compact subset of $\mathscr{X}$. In this case, queries corresponding to a document x are completely unrelated to the semantics of x.

In the experiments described in Section 6.2, we set $V = [1, 2, \cdots, 30]$. The ID string's maximum length m was set to the minimum value such that $|V|^m \geq |\mathcal{X}|$.

4.2. Hierarchical k-Means Indexing (HKmI)

The typical indexing uses a hierarchical k-means clustering algorithm. A document's ID string $t = [t_1, \cdots, t_m]$ is set to the indices of clusters of depth m, such that the document belongs at every level of the hierarchy, where k-means clustering is applied to partition the set of documents.

In a k-means clustering process, the clusters' centroid vectors are optimized to minimize the sum of the squared distances from each centroid to the cluster members. This procedure can be interpreted within a maximum-likelihood formulation by assuming that a document belonging to a cluster is sampled from a Gaussian distribution with the mean vector as the centroid.

Hence, k-means provides a BMI if we assume $p(Q|X) \sim N(\mu_x, \Sigma)$ and $p(Q|T) \sim N(\mu_t, \sigma^2 I)$, where N represents a Gaussian distribution; μ_x and μ_t are mean vectors specific to x and t, respectively; Σ is any covariance matrix; and $\sigma^2 I$ is a diagonal matrix. This conclusion immediately follows from substituting the Gaussian density functions into Formula (5). The optimality holds for any choice of β in Formula (3). A proof is provided in Appendix A.2. The linkage between k-means clustering and information bottleneck theory has been theoretically explored by Still et al. (2003) in depth.

Following Tay et al. (2022), we set μ_x as the vectoral embedding generated by a BERT model (Devlin et al., 2019) for a document x. In our experiments, we used the same settings for V and m as in HRI (Section 4.1). These settings were used as the default for the *neural corpus indexer* (NCI) (Wang et al., 2022).

4.3. Locality-Sensitive Hashing Indexing (LSHI)

Locality-sensitive hashing (LSH) can be considered in the same formulation. In LSH the index t_i is Boolean; that is, $V = \{0, 1\}$, where each element of $t = [t_1, \cdots, t_i, \cdots, t_m]$ is independently generated by a p-stable LSH algorithm (Datar et al., 2004), which is a hyperplane classifier in Euclidean space. The hyperplane's location and direction are randomly determined. Unlike hierarchical indexing methods, LSH-based indexing does not produce "semantic prefixes"; nevertheless, $\mathcal{T}$ encodes location-related information.

In our experiments, LSH indexing was implemented in three steps. First, a standard LSH code, a Boolean vector, was acquired for every document. Second, every fifth entry of the vector was mapped to $V = [1, 2, \cdots, 32]$. Third, LSH code collisions were resolved by appending additional digits acquired by a hierarchical k-means algorithm.

4.4. Our Proposal: Bottleneck-Minimal Indexing (BMI)

Formula (5) indicates that the optimal indexing is dictated by the distributions $p(Q|X)$ and $p(Q|T)$ over the query space, rather than the document space. In other words, any indexing method implemented without considering the distribution of Q would not be able to acquire the optimal indexing. Typical previous works on GDR have this problem. The concept of leveraging query information has been empirically investigated in the contexts of vector quantization (Gupta et al., 2022; Lu et al., 2023) and recommendation systems (Zeng et al., 2023); our contribution extends this exploration by offering a theoretical rationale for the significance of query distribution, grounded in information bottleneck theory.

Hence, we propose a new indexing method that incorporates both the queries and the documents. For the purposes of simplicity in this paper, we assume $p(Q|X = x)$ and $p(Q|T = t)$ to follow Gaussian distributions, enabling the analytical derivation of bottleneck-minimizing indexing. In a manner akin to HKmI discussed in Section 4.2, k-means emerges as the optimal indexing strategy under this Gaussian assumption. However, our approach diverges from HKmI by employing hierarchical k-means clustering on the set $\{\mu_{Q|x} : \forall x \in \mathcal{X}\}$, which comprises the mean vectors of the Gaussian $p(Q|X = x)$ for each document x. We examine k-means here for a fair comparison with HKmI, in addition to its simplicity. However, Definition 3.1 of BMI can be used to analyze more complex clustering algorithms under more sophisticated assumptions on $p(Q|x)$ and $p(Q|t)$.

Unlike μ_x for HKmI, however, it is not straightforward to obtain $\mu_{Q|x}$, and it must be estimated for each document x. In this paper, we estimate $\mu_{Q|x}$ as the mean of the BERT embeddings of queries generated from document x, which is the maximum-likelihood estimator given the previously mentioned Gaussian distribution assumption of $p(Q|x)$. A proof is provided in Appendix A.3. We chose BERT for consistency with previous works, including DSI (Tay et al., 2022) and NCI (Wang et al., 2022); in a future work, we may examine other models that are adapted to short texts.

Let $\mathcal{Q}_x$ denote a set of queries for x. The document x is now represented by mean(BERT($\mathcal{Q}_x$)), the mean vector of the BERT embeddings of queries in $\mathcal{Q}_x$, instead of the document's BERT embedding. The hierarchical k-means algorithm is applied to $\{\mu_{Q|x} = \text{mean}(\text{BERT}(\mathcal{Q}_x)) : x \in \mathcal{X}\}$, which produces a new set of ID strings, $\mathcal{T}$.

We follow Wang et al. (2022) in constructing $\mathcal{Q}_x$ to comprise the following three kinds of queries, as follows:

RealQ: Real queries from the training set;

Table 1: Descriptive statistics of the datasets (upper) and generated queries (lower).

	NQ320K		MS MARCO Lite	
	#	mean # words	#	mean # words
documents	109,739	4902.7	138,457	1210.1
queries (train)	307,373	9.2	183,947	6.0
queries (test)	7,830	9.3	2,792	5.9
Generated queries				
GenQ	1,646,085	5.6	692,285	5.5
DocSeg	1,168,585	62.9	1,393,329	59.0

GenQ: Queries generated by a pretrained model `docT5query` [1](Nogueira et al., 2019);
DocSeg: Random segments of the original documents.

Section 5.1 gives the details of generating these queries.

In a real application, the distributions $p(Q|X = x)$ and $p(Q|T = t)$ may deviate from Gaussian assumptions, potentially exhibiting multi-modal characteristics. Under these conditions, k-means clustering methods become suboptimal and should be replaced by another sophisticated clustering methods that reflect the true distribution of the data. Developing such methods form a future direction.

5. Data and Settings

5.1. Datasets and Metrics

We evaluated different indexing methods on two datasets: NQ320K (Kwiatkowski et al., 2019), and MARCO Lite, which is a subset extracted from the document ranking dataset in MS MARCO (Nguyen et al., 2016). The upper half of Table 1 summarizes the basic statistics of the two datasets.

The entire MS MARCO dataset has 3.2 million documents and 367,013 queries. MARCO Lite was constructed by first randomly selecting half the queries and then extracting those queries' gold-standard documents.

The lower half of Table 1 summarizes our proposed method's generated queries (i.e., $\mathcal{Q}_x$) for creating document ID strings, as described at the end of Section 4.4. For NQ320K, we followed the settings in Wang et al. (2022) to produce 15 **GenQ** queries; for MARCO Lite, we produced 5 **GenQ** queries per document. As for **DocSeg** queries, we randomly selected 10∼12 segments (depending on the document length) per document as queries for both datasets; each segment had around 60 words.

We followed previous works by using the recall (Rec@N) and mean reciprocal rank (MRR) for evaluation, where a higher Rec@N or MRR indicates a better GDR system. For each query, GDR produces a ranking of documents with respect to their degrees of relevance to the query, via a beam-search procedure as suggested in Tay et al. (2022). In our evaluation across two datasets, each query was associated with a single gold-standard document. The Rec@N measures the percentage of queries for which the gold-standard document is among the top N documents in the ranking, while the MRR is equal to the gold-standard document's MRR. We used values of $N = 1, 10, 100$.

5.2. Model and Training Settings

We used the same neural-network architecture as in NCI (Wang et al., 2022). For sequence-to-sequence generation of an index for a query, the NCI architecture combined a standard transformer (Vaswani et al., 2017) with a *prefix-aware weight-adaptive* (PAWA) decoder. The Transformer encoder's parameters were initialized with T5 weights (Raffel et al., 2020) acquired by large-scale pretraining, while the decoder weights were randomly initialized.

We tested models of different sizes, which were initialized from the weights of T5-tiny[2], T5-mini[3], T5-small[4], and T5-base[5], where the string after "T5" indicates the weights. The PAWA decoder had four transformer layers. All models were trained using the default hyperparameters of NCI, as provided in its official GitHub repository[6].

For ablation tests, we considered different $\mathcal{Q}_x$ variants. Apart from setting $\mathcal{Q}_x = \text{GenQ} + \text{RealQ} + \text{DocSeg}$ to include all three kinds of queries, we also tested GenQ alone and GenQ + RealQ. In addition, the previous method of clustering documents by their embeddings also constitutes an ablated version, denoted here as **Doc**, because it is almost equivalent to DocSeg, which uses an average of random document segments.

5.3. Estimation of Mutual Information

To verify the information bottleneck in GDR, the mutual information $I(X;T)$ and $I(X;Q|T)$ must be calculated. They are defined as follows:

$$I(X;T) = \mathbb{E}_{X,T} \log \frac{p(X|T)}{p(X)}, \tag{7}$$

$$I(X;Q|T) = \mathbb{E}_{X,T,Q} \log \frac{p(X,Q|T)}{p(X|T)p(Q|T)} \tag{8}$$

$$= \mathbb{E}_{X,T,Q} \log \frac{p(X|Q)}{p(T|Q)} \frac{p(T)}{p(X)}. \tag{9}$$

[1] huggingface.co/castorini/doc2query-t5-base-msmarco

[2] https://huggingface.co/google/t5-efficient-tiny

[3] https://huggingface.co/google/t5-efficient-mini

[4] https://huggingface.co/t5-small

[5] https://huggingface.co/t5-base

[6] github.com/solidsea98/Neural-Corpus-Indexer-NCI

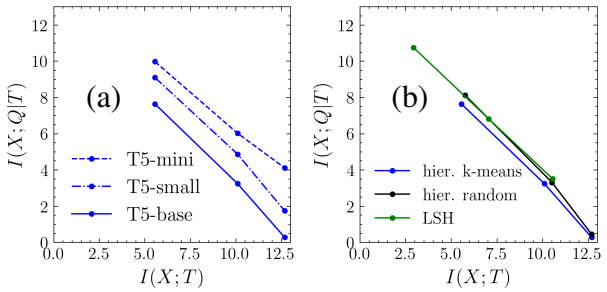


Figure 3: Experimental information bottleneck curves corresponding to Figure 2. (a) Empirical information curves for HKmI, measured with T5 models of different sizes. (b) Empirical information curves for different indexing methods, estimated on the NQ320K dataset with the T5-base model.

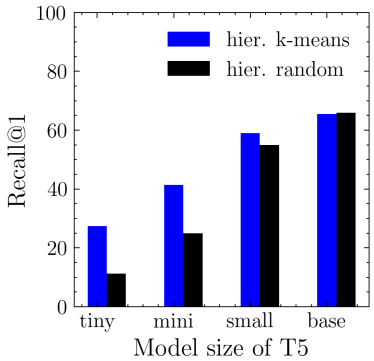


Figure 4: Rec@1 scores on the test set of NQ320K for document IDs generated by hierarchical k-means (blue) or random (red) clustering, for different sizes of the finetuned language model T5.

Appendix A.1 gives the detailed derivation of Formula (9). The two definitions above are calculated as further detailed in Appendix A.4. To acquire the mutual information values for $|\mathcal{T}| < |\mathcal{X}|$, we considered a generalized GDR task of predicting the prefixes of ID strings rather than the whole strings, for a prefix length l. In addition to the default setting $l = m$ (i.e., standard GDR), we also tested $l = 2, 3$.

6. Experiments

6.1. Quantification of GDR Bottleneck

Figure 3(a) shows $I(X;Q|T)$ with respect to $I(X;T)$, as measured on the NQ320K dataset's training set. The ID strings were acquired by HKmI. A curve was obtained for every neural network model of a different size. The points on each curve were acquired with different l values, as detailed at the end of Section 5.3. As seen in the figure, a larger model corresponded to a curve closer to the graph's lower-left corner, thus suggesting the existence of a bottleneck as anticipated by the IB theory.

Figure 3(b) compares several existing indexing methods. Here, HKmI was closer to the lower-left corner than the other two methods. Hence, the optimal condition for k-means indexing better fits the reality of data.

Next, Figure 4 compares the random (black) and k-means (blue) hierarchical indexing methods in terms of the Rec@1 on the NQ320K test set. As the model size is reduced, Rec@1 shows a general decline for both methods. With the reduction in size, the superiority of the k-means method over the random method became apparent. While the performance of the two methods was similar for the T5-base model, the Rec@1 of the k-means method was more than twice that of the random method in the T5-tiny model.

These observations suggests that the IB curve indicates the quality of GDR, specifically in terms of its distance from the graph's lower-left corner. While the Rec@1 merely evaluates the case of $|\mathcal{T}| = |\mathcal{X}|$, the IB curve evaluates the case of $|\mathcal{T}| < |\mathcal{X}|$, thus enabling recognition of overfitting in GDR.

6.2. Comparison Among Indexing Methods

We also evaluated our proposed indexing method in comparison with the existing methods. Table 2 summarizes our experimental results on the two datasets. The table's three blocks correspond to training models of different sizes: T5-mini, T5-small, and T5-base, from top to bottom. Each row represents an indexing method, and each column corresponds to a metric. The scores were acquired on the test sets. A higher score indicates a better indexing method.

As seen in the table, our method (bottom row in each block) achieved the best scores under most settings. In particular, our method consistently outperformed hierarchical k-means (third row in each block), often with a large difference. On MARCO Lite and with T5-base (right side, bottom block), our method had a Rec@1 of 45.20, 3.72 points higher than the score for the original hierarchical k-means indexing. This margin was even larger than that between k-means and random clustering (2.48 points).

With the smaller model T5-mini (top block), the improvement achieved by our method was dramatic. Specifically, compared with the original hierarchical k-means, our method improved the Rec@1 by 7.06 points (17%) on NQ320K and 6.45 points (91%) on MARCO Lite. This improvement is corroborated by the observation that our method's IB curve is located closer to the lower-left corner, as shown in Figure 6 in Appendix B.

These large improvements indicate that there is a better indexing than with the previous methods, but this indexing must involve Q. Recall that our method is also based on hierarchical k-means, and the only difference from the original method is application of the clustering algorithm to queries

Table 2: Performance of different indexing methods.

Indexing Methods	NQ320k Rec@1	NQ320k Rec@10	NQ320k Rec@100	NQ320k MRR@100	MARCO Lite Rec@1	MARCO Lite Rec@10	MARCO Lite Rec@100	MARCO Lite MRR@100
T5-mini								
Hier. random clustering	24.90	54.89	76.78	34.72	5.98	10.57	19.59	7.69
Locality-sensitive hashing	27.94	57.28	78.07	37.63	7.06	20.42	41.26	11.28
Hier. k-means clustering	41.43	72.16	87.75	52.30	7.09	23.17	50.50	12.51
Our method (BMI)	**48.49**	**76.73**	**88.84**	**58.48**	**13.54**	**42.77**	**71.49**	**22.94**
T5-small								
Hier. random clustering	54.93	77.20	87.79	62.75	12.64	43.88	71.91	22.42
Locality-sensitive hashing	52.16	77.01	87.87	61.05	15.62	47.46	76.00	25.71
Hier. k-means clustering	**58.94**	**82.94**	**92.08**	**67.67**	22.35	56.30	83.02	33.04
Our method (BMI)	58.61	82.76	91.42	67.38	**26.71**	**61.00**	**85.46**	**38.02**
T5-base								
Hier. random clustering	65.85	84.16	91.51	72.59	39.00	70.45	88.65	49.54
Locality-sensitive hashing	62.82	83.27	91.42	70.37	39.83	70.77	87.00	50.29
Hier. k-means clustering	65.43	85.20	92.64	72.73	41.48	74.39	89.65	52.57
Our method (BMI)	**66.69**	**86.17**	**93.23**	**73.91**	**45.20**	**76.04**	**90.62**	**55.47**

rather than documents. These improvement results indicate that the information bottleneck theory applies well to GDR. In other words, it is important to pursue "bottleneck-minimal" rather than "distortion-minimal" indexing.

6.3. Ablation Study

Table 3: Ablation study results on the test set of MARCO Lite. GQ and RQ are abbreviations of GenQ and RealQ, respectively.

	Rec@1	Rec@10	Rec@100	MRR@100
		T5-mini		
Doc	7.09	23.17	50.50	12.51
GenQ	9.24	27.51	56.27	15.55
GenQ+RealQ	10.03	30.91	56.59	16.70
GQ+RQ+DocSeg	**13.54**	**42.77**	**71.49**	**22.94**
		T5-small		
Doc	22.35	56.30	83.02	33.04
GenQ	22.74	52.90	79.55	32.29
GenQ+RealQ	25.25	58.92	82.84	36.52
GQ+RQ+DocSeg	**26.71**	**61.00**	**85.46**	**38.02**
		T5-base		
Doc	41.48	74.39	89.65	52.57
GenQ	39.04	70.70	88.93	49.88
GenQ+RealQ	39.43	72.31	88.93	50.62
GQ+RQ+DocSeg	**45.20**	**76.04**	**90.62**	**55.47**

Furthermore, we compared our proposed method with ablated versions of it, as detailed in Section 5.2. Table 3 summarizes the results on the MARCO Lite dataset. Each row besides the last one represents an ablated version used a certain subset of the generated queries as $\mathcal{Q}_x$ to estimate $\mu_{Q|x}$, as mentioned in Section 4.4.

With the smaller models (T5-mini and T5-small), even the ablated versions outperformed Doc. For example, GenQ achieved a Rec@1 of 9.24 with T5-mini, which was higher than the Doc result of 7.09. This is quite surprising because GenQ (or GenQ+RealQ) only used short queries. As seen in the lower half of Table 1, the GenQ queries had a mean length of only 5.5 words, much shorter than the document lengths.

Nevertheless, there was a significant margin between the results with and without DocSeg. With T5-base, GenQ+RealQ+DocSeg had a Rec@1 of 45.20, 5.77 points higher than that for GenQ+RealQ.

The addition of DocSeg is effective for adding rich information existing in documents. Compared with the DocSeg queries, the GenQ and RealQ queries had limited variety: many of them were duplicates that compromised the retrieval performance.

6.4. Comparison with SOTA

Finally, we compare our proposed method against state-of-the-art (SOTA) methods beyond the DSI/NCI framework utilizing the NQ320K dataset. To show the power of our method at full strength, we enhanced the GenQ queries by finetuning the document-to-query model (`docT5query`) on the training set of NQ320K, thus producing higher-quality ID strings. Details of this finetuning process are left to Appendix C.

Figure 5 shows the results of the finetuning process. Solid curves shows the performance using enhanced GenQ queries, while dashed curves represent the baseline results reported in the second column of Table 2. The HKmI method was also examined for its use of enhanced GenQ

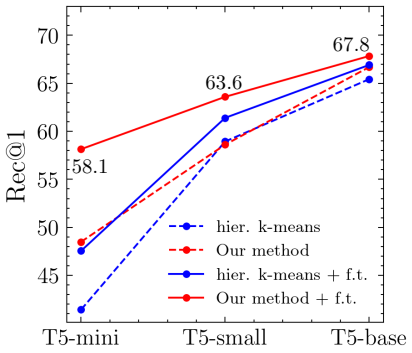


Figure 5: Performance enhancement by finetuning the `docT5query` model for generating improved GenQ documents.

Table 4: Performance of BMI on NQ320K, with improved GenQ queries by finetuning `docT5query` on the training set of NQ320K, compared with existing GDR methods.

GDR Method	Index Type		Rec@1
DSI (Tay et al., 2022)	static	number	27.4
Ultron-PQ (Zhou et al., 2022)	static	number	25.6
NCI (Wang et al., 2022)	static	number	66.9
Tied-At. (Nguyen & Yates, 2023)	static	number	65.3
SEAL (Bevilacqua et al., 2022)	static	textual	26.3
Ultron-URL (Zhou et al., 2022)	static	textual	33.8
Our method	static	number	**67.8**
LMIndexer (Jin et al., 2023)	learned	textual	66.3
GENRET (Sun et al., 2023)	learned	textual	68.1
NOVO (Wang et al., 2023)	learned	textual	**69.3**

queries as augmented training data, although it, unlike our method, did not utilize these enhanced GenQ queries for indexing.

An improvement in Rec@1 is observable across all model sizes for both our method (in red) and HKmI (in blue), with our approach showing a more significant enhancement, particularly with smaller models such as T5-mini or T5-small. After fine-tuning, our method surpassed HKmI across all model sizes. This substantial enhancement underscores the critical role of accurately estimating $\mu_{Q|x}$, the mean of query vectors, in improving the quality of document indexes.

Table 4 compares our new results with previous methods beyond the DSI/NCI scope, using the T5-base retrieval model for consistency with most prior studies. Excluding NCI (i.e., HKmI) and our method—whose results are shown in Figure 5—the performance scores for all other methods are taken directly from their original papers.

The methods are categorized based on whether indexes are determined before or during the training phase of the retrieval model, labeled as "static" and "learned," respectively, as detailed in Table 4's second column. Moreover, the third column specifies whether an index is a numeric string of text (a sequence of words). Designing text-based indexing is inherently more complex and extends beyond number-based methods. This analysis includes only methods which reported their performance on the NQ320K dataset, which means we focus on document retrieval. Related works in vector quantization or recommendation systems are not compared here.

Our approach achieved a Rec@1 of 67.8, surpassing all models that relied on static indexes. This performance closely competes with SOTA methods, such as GENRET (Sun et al., 2023) at 68.1 and NOVO (Wang et al., 2023) at 69.3, which employ textual ID strings and design sophisticated ways of learning indexes concurrent with the retrieval model's training.

These findings underscore our method's effectiveness and simplicity, utilizing static, number-based indexes generated through a simple k-means process. Our model's training procedure is the same as that of NCI, without necessitating additional computational resources, with the sole distinction being the production of document indexes by incorporating query distribution.

7. Conclusion

We introduced a new way to formulate generative document retrieval (GDR) by applying the information bottleneck theory. In GDR, a document $x \in \mathcal{X}$ is indexed by $t \in \mathcal{T}$, and queries $\mathcal{Q}$ are trained by neural autoregressive models for mapping to indexes $\mathcal{T}$. GDR can thus be treated as the transmission of information from documents $\mathcal{X}$ to queries $\mathcal{Q}$, with the requirement to transmit more bits via the indexes $\mathcal{T}$. While previous methods considered how to acquire the best $\mathcal{T}$ with respect to $\mathcal{X}$, this work also incorporated $\mathcal{Q}$.

This reformulation naturally suggested a new document indexing method to design GDR by using $\mathcal{Q}$. Our method, which applies the hierarchical k-means algorithm to queries instead of document vectors, achieved a significant improvement on two datasets in terms of the recall and MRR scores. The margin was especially large when a smaller neural network was used, with an improvement of up to 6.45 points, or 91% relative improvement.

In a comparison with various indexing methods, our method not only outperformed all the other models that are based on static indexes, but also achieved a competitive accuracy against those SOTA models which employ textual ID strings and sophisticated index-learning strategies during the retrieval model's training. These results affirm the effectiveness of our approach, positioning it as a simple yet strong competitor within the field.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR2114.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning and, specifically, generative information retrieval. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Berger, T. Rate distortion theory for sources with abstract alphabets and memory. *Information and Control*, 13: 254–273, 1968.

Bevilacqua, M., Ottaviano, G., Lewis, P., Yih, S., Riedel, S., and Petroni, F. Autoregressive search engines: Generating substrings as document identifiers. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Z4kZxAjg8Y.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.

Cao, N. D., Izacard, G., Riedel, S., and Petroni, F. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=5k8F6UU39V.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pp. 129–136, 2007.

Chaidaroon, S. and Fang, Y. Variational deep semantic hashing for text documents. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 75–84, 2017.

Chen, J., Zhang, R., Guo, J., Liu, Y., Fan, Y., and Cheng, X. Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 191–200, 2022.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pp. 253–262, 2004.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Gabeur, V., Sun, C., Alahari, K., and Schmid, C. Multimodal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pp. 214–229. Springer, 2020.

Geng, S., Liu, S., Fu, Z., Ge, Y., and Zhang, Y. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 299–315, 2022.

Gray, R. Vector quantization. *IEEE Assp Magazine*, 1(2): 4–29, 1984.

Grefenstette, G. *Cross-language information retrieval*, volume 2. Springer Science & Business Media, 2012.

Gupta, G., Medini, T., Shrivastava, A., and Smola, A. J. Bliss: A billion scale index using iterative re-partitioning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 486–495, 2022.

Hartigan, J. A. and Wong, M. A. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

Jin, B., Zeng, H., Wang, G., Chen, X., Wei, T., Li, R., Wang, Z., Li, Z., Li, Y., Lu, H., et al. Language models as semantic indexers. *arXiv preprint arXiv:2310.07815*, 2023.

Jin, S., Yao, H., Sun, X., and Zhou, S. Unsupervised semantic deep hashing. *Neurocomputing*, 351:19–25, 2019.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Khattab, O. and Zaharia, M. 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*, pp. 39–48, 2020.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Li, Y., Yang, N., Wang, L., Wei, F., and Li, W. Learning to rank in generative retrieval. *arXiv preprint arXiv:2306.15222*, 2023.

Lin, Q., Chen, X., Zhang, Q., Cai, S., Zhao, W., and Wang, H. Deep unsupervised hashing with latent semantic components. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7488–7496, 2022.

Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, Z., Chen, J., Lian, D., Zhang, Z., Ge, Y., and Chen, E. Knowledge distillation for high dimensional search index. *Advances in Neural Information Processing Systems*, 36, 2023.

Manning, C. and Schutze, H. *Foundations of statistical natural language processing*. MIT press, 1999.

Mehta, S., Gupta, J., Tay, Y., Dehghani, M., Tran, V., Rao, J., Najork, M., Strubell, E., and Metzler, D. DSI++: Updating transformer memory with new documents. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8198–8213, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.510. URL https://aclanthology.org/2023.emnlp-main.510.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Nguyen, T. and Yates, A. Generative retrieval as dense retrieval. *arXiv preprint arXiv:2306.11397*, 2023.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. MS MARCO: A human generated machine reading comprehension dataset. In Besold, T. R., Bordes, A., d'Avila Garcez, A. S., and Wayne, G. (eds.), *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, 2016. URL https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

Nister, D. and Stewenius, H. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 2161–2168. Ieee, 2006.

Nogueira, R., Lin, J., and Epistemic, A. From doc2query to doctttttquery. *Online preprint*, 6:2, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Rajput, S., Mehta, N., Singh, A., Hulikal Keshavan, R., Vu, T., Heldt, L., Hong, L., Tay, Y., Tran, V., Samost, J., et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36, 2023.

Robertson, S., Zaragoza, H., et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Salakhutdinov, R. and Hinton, G. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7): 969–978, 2009.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Shannon, C. E. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Shannon, C. E. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record*, 4: 142–163, 1959.

Slonim, N. *The information bottleneck: Theory and applications*. PhD thesis, Hebrew University of Jerusalem Jerusalem, Israel, 2002.

Slonim, N. and Tishby, N. Agglomerative information bottleneck. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

URL https://proceedings.neurips.cc/paper_files/paper/1999/file/be3e9d3f7d70537357c67bb3f4086846-Paper.pdf.

Slonim, N. and Tishby, N. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 208–215, 2000.

Still, S., Bialek, W., and Bottou, L. Geometric clustering using the information bottleneck method. In *Advances in neural information processing systems*, 2003.

Sun, W., Yan, L., Chen, Z., Wang, S., Zhu, H., Ren, P., Chen, Z., Yin, D., de Rijke, M., and Ren, Z. Learning to tokenize for generative retrieval. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=UKd6dpVGdu.

Tang, Y., Zhang, R., Guo, J., Chen, J., Zhu, Z., Wang, S., Yin, D., and Cheng, X. Semantic-enhanced differentiable search index inspired by learning strategies. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 4904–4913, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599903. URL https://doi.org/10.1145/3580305.3599903.

Tay, Y., Tran, V., Dehghani, M., Ni, J., Bahri, D., Mehta, H., Qin, Z., Hui, K., Zhao, Z., Gupta, J., Schuster, T., Cohen, W. W., and Metzler, D. Transformer memory as a differentiable search index. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 21831–21843. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/892840a6123b5ec99ebaab8be1530fba-Paper-Conference.pdf.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pp. 1–5. IEEE, 2015.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *ArXiv*, physics/0004057, 2000. URL https://api.semanticscholar.org/CorpusID:8936496.

van den Oord, A., Vinyals, O., and kavukcuoglu, k. Neural discrete representation learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Venkateswara, H., Eusébio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5385–5394, 2017. URL https://api.semanticscholar.org/CorpusID:2928248.

Wang, Y., Hou, Y., Wang, H., Miao, Z., Wu, S., Sun, H., Chen, Q., Xia, Y., Chi, C., Zhao, G., Liu, Z., Xie, X., Sun, H., Deng, W., Zhang, Q., and Yang, M. A neural corpus indexer for document retrieval. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=fSfcEYQP_qc.

Wang, Z., Zhou, Y., Tu, Y., and Dou, Z. Novo: Learnable and interpretable document identifiers for model-based ir. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 2656–2665, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614993. URL https://doi.org/10.1145/3583780.3614993.

Wu, Z.-b. and Yu, J.-q. Vector quantization: a review. *Frontiers of Information Technology & Electronic Engineering*, 20(4):507–524, 2019.

Zeng, H., Luo, C., Jin, B., Sarwar, S. M., Wei, T., and Zamani, H. Scalable and effective generative information retrieval. *arXiv preprint arXiv:2311.09134*, 2023.

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

A. Mathematical Details

A.1. $I(X;Q|T)$ in Information Bottleneck Context

The information bottleneck theory assumes the Markovian condition $T \leftrightarrow X \leftrightarrow Q$, which means that $p(Q|X,T) = p(Q|X)$.

Proposition A.1.

$$I(X;Q|T) = -I(T;Q) + \textit{constant} \tag{10}$$

Proof.

$$\begin{aligned}
I(X;Q|T) &= \mathbb{E}_{X,T,Q} \log \frac{p(X,Q|T)}{p(X|T)p(Q|T)} && (11)\\
&= \mathbb{E}_{X,T,Q} \log \frac{p(Q|X,T)p(X|T)}{p(X|T)p(Q|T)} && (12)\\
&= \mathbb{E}_{X,T,Q} \log \frac{p(Q|X)}{p(Q|T)} \qquad \Big(\because p(Q|X,T) = p(Q|X)\Big) && (13)\\
&= \mathbb{E}_{X,T,Q} \log \frac{p(X|Q)}{p(T|Q)} \frac{p(T)}{p(X)} \qquad (\because \text{Bayes' theorem}) && (14)\\
&= -\mathbb{E}_{X,T,Q} \log \frac{p(T|Q)}{p(T)} + \mathbb{E}_{X,T,Q} \frac{p(X|Q)}{p(X)} && (15)\\
&= -I(T;Q) + I(X;Q). && (16)
\end{aligned}$$

The mutual information $I(X;Q)$ is a constant determined by the dataset. Therefore, $I(X;Q|T) = -I(T;Q)$+constant. □

Proposition A.2. *The Lagrangians in Formulas (3) and (4) are equivalent.*

Proof. Substitution of Formula (10) into (3) immediately produces Formula (4), which implies the two Lagrangians' equivalence. □

A.2. k-Means Produces Bottleneck Minimum

In both the HKmI (Section 4.2) and BMI (Section 4.4), the k-means clustering method is applied hierarchically to the vector representations of documents $\mathcal{X}$. We demonstrate that at each hierarchical level, as $\mathcal{X}$ is assigned identifiers from the alphabet V, k-means clustering effectively minimizes the information bottleneck as defined in Definition 3.1.

Let $\mathbf{x}$ represent the vector representation of a document x. The goal of k-means clustering is to partition $\{\mathbf{x} : x \in \mathcal{X}\}$ into k clusters, aiming to minimize the sum of the squared distances from each document to its cluster's centroid. This is represented by the partition function $f : x \mapsto i$. The optimization problem is described as follows (Hartigan & Wong, 1979):

$$\min_f S(f) = \sum_{i=1}^{k} \sum_{x \in f^{-1}(i)} \|\mathbf{x} - \bar{\mathbf{x}}_i\|^2, \tag{17}$$

$$\text{where } \bar{\mathbf{x}}_i = \frac{1}{|f^{-1}(i)|} \sum_{x \in f^{-1}(i)} \mathbf{x}. \tag{18}$$

This paper focuses on the optimal solution f^* of Formula (17). We acknowledge various implementations for achieving f^*; however, they are not covered in this paper.

We now aim to establish the following proposition:

Proposition A.3. *Assume that $p(Q|x) \sim N(\mathbf{x}, \Sigma)$ and $p(Q|t) \sim N(\mu_t, \sigma^2 I)$, where Σ is an arbitrary covariance matrix and $\sigma^2 I$ is an isotropic diagonal matrix; μ_t is the vector estimated for every element $t \in V$ in the alphabet V, Then, any optimal solution f^* to Formula (17) also maximizes the likelihood $p(\mathcal{X}, \mathcal{Q}|f)$ as defined in Formula (6).*

Proof. By definition, $p(\mathcal{X}, \mathcal{Q}|f) = \prod_{x \in \mathcal{X}} p^*(X = x \mid T = f(x))$, where p^* is detailed in Formula (5). From Bayes' theorem:

$$p^*(X|T) = \frac{p^*(T|X)}{p^*(T)} p(X), \tag{19}$$

where $p(X)$ is a constant independent of f.

Applying Formula (19) and then inserting Formula (5), we derive:

$$p(\mathcal{X}, \mathcal{Q}|f) \propto \prod_{x \in \mathcal{X}} \frac{p^*(T = f(x)|X = x)}{p^*(T = f(x))} \tag{20}$$

$$= \frac{1}{\prod_{x \in \mathcal{X}} Z(x, \beta)} \exp\left(-\beta \sum_{x \in \mathcal{X}} \mathrm{KL}\Big[p(Q|X = x) \,\Big\|\, p\big(Q|T = f(x)\big)\Big]\right) \tag{21}$$

Assuming $p(Q|x) \sim N(\mathbf{x}, \Sigma)$ and $p(Q|t) \sim N(\mu_t, \sigma^2 I)$, the KL divergence is computed analytically; expanding the expression in Formula (21) leads to:

$$p(\mathcal{X}, \mathcal{Q}|f) = p(\mathcal{X}, \mathcal{Q}|f, \{\mu_t : t \in V\}) \tag{22}$$

$$\propto \exp\left\{-\frac{\beta}{2\sigma^2} \sum_{x \in \mathcal{X}} \|\mathbf{x} - \mu_{f(x)}\|^2\right\} \underbrace{\exp\left\{-\frac{\beta}{2}|\mathcal{X}|\Big(d \log \sigma^2 - d - \log|\Sigma| + \mathrm{tr}\big(\Sigma\big)/\sigma^2\Big)\right\}}_{\text{constant}}, \tag{23}$$

$$\propto \exp\left\{-\frac{\beta}{2\sigma^2} \sum_{x \in \mathcal{X}} \|\mathbf{x} - \mu_{f(x)}\|^2\right\} \tag{24}$$

$$= \exp\left\{-\frac{\beta}{2\sigma^2} \sum_{t \in V} \sum_{x \in f^{-1}(t)} \|\mathbf{x} - \mu_t\|^2\right\}. \tag{25}$$

Since β and σ are constants, maximizing $p(\mathcal{X}, \mathcal{Q}|f)$ equates to minimizing $\sum_{t \in V} \sum_{x \in f^{-1}(t)} \|\mathbf{x} - \mu_t\|^2$. At the maximum, μ_t corresponds to the centroid $\bar{\mathbf{x}}_t$. This directly aligns with the k-means objective in Formula (17) and the selection of centroid as the cluster center in Formula (18).

Therefore, any optimal solution f^* to Formula (17) also optimally maximizes the likelihood $p(\mathcal{X}, \mathcal{Q}|f)$.

□

The effectiveness of k-means in HKmI or BMI is validated through Proposition A.3. In HKmI, assuming $p(Q|x) \sim N(\mu_x, \Sigma)$, where μ_x represents the BERT vector representation for the document, aligns with maximizing $p(\mathcal{X}, \mathcal{Q}|f)$. For BMI, where the assumption is $p(Q|x) \sim N(\mu_{Q|x}, \Sigma)$, we also achieve maximization of $p(\mathcal{X}, \mathcal{Q}|f)$.

However, it is a simplification in HKmI to assume $p(Q|x) \sim N(\mu_x, \Sigma)$ as μ_x describes the document, not necessarily its query distribution. In contrast, BMI's assumption of $p(Q|x) \sim N(\mu_{Q|x}, \Sigma)$ is more suitable.

A.3. Average of Query Vectors as A Maximum-Likelihood Estimator for Gaussian $p(Q|x)$ Mean

In Section 4.4, we used the following approximation

$$\mu_{Q|x} \approx \text{mean}(\text{BERT}(\mathcal{Q}_x)). \tag{26}$$

Here, $\mu_{Q|x}$ represents the mean vector of the distribution $p(Q|x)$ which we assumed to be Gaussian. The expression $\text{BERT}(\mathcal{Q}_x) = \{\text{BERT}(q) : q \in \mathcal{Q}_x\}$ denotes the set of BERT-generated query vector embeddings associated with the gold-standard document x. This section elaborates on the rationale behind the approximation, demonstrating that $\text{mean}(\text{BERT}(\mathcal{Q}_x))$ serves as a maximum-likelihood estimator for $\mu_{Q|x}$.

To demonstrate that mean(BERT($\mathcal{Q}_x$)) is a maximum-likelihood estimator for $\mu_{Q|x}$, it is necessary to assume that the queries in $\mathcal{Q}_x$ represent independent samples drawn from $p(Q|x)$. $p(Q|x)$ is a Gaussian distribution with mean $\mu_{Q|x}$ and covariance matrix Σ that is unknown. Then, the likelihood function $p(\mathcal{Q}_x|\mu_{Q|x})$ can be formulated:

$$p(\mathcal{Q}_x|\mu) = \prod_{q\in\mathcal{Q}_x} p(q|\mu) \tag{27}$$

$$\propto \exp\left(-\frac{1}{2}\sum_{q\in\mathcal{Q}_x}(q-\mu)^\top\Sigma^{-1}(q-\mu)\right) \tag{28}$$

$$= \exp -\frac{1}{2}\left(\underbrace{|\mathcal{Q}_x|\mu^\top\Sigma^{-1}\mu - 2\mu^\top\Sigma^{-1}\sum_{q\in\mathcal{Q}_x} q}_{=:g(\mu)} + \sum_{q\in\mathcal{Q}_x} q^\top\Sigma^{-1}q\right), \tag{29}$$

where $|\mathcal{Q}_x|$ denotes the number of queries. Here, q represents the BERT-generated vector embedding of a query.

$g(\mu)$ in Formula (29) is a quadratic function, and its derivative is as follows:

$$\frac{\mathrm{d}g(\mu)}{\mathrm{d}\mu} = 2\Sigma^{-1}\left(|\mathcal{Q}_x|\mu - \sum_{q\in\mathcal{Q}_x} q\right) \tag{30}$$

Upon solving $\mathrm{d}g(\mu)/\mathrm{d}\mu = 0$, we derive the maximum-likelihood estimator for $\mu_{Q|x}$ as:

$$\hat{\mu}_{Q|x} = \frac{1}{|\mathcal{Q}_x|}\sum_{q\in\mathcal{Q}_x} q, \tag{31}$$

which precisely corresponds to mean(BERT(q)). This outcome affirms that taking the average of BERT embeddings for the set of queries $\mathcal{Q}_x$ associated with a document x provide the most probable estimation of $\mu_{Q|x}$, under the assumed Gaussian distribution of queries conditioned on documents.

Therefore, the accuracy of this approximation relies on the degree to which the queries in $\mathcal{Q}_x$ adhere to the independent and identically distributed (i.i.d.) assumption underlying $p(Q|x)$. As demonstrated in Appendix C, optimizing the GenQ queries through fine-tuning the document-to-query model (which estimates $p(Q|x)$) results in enhanced retrieval performance. This underscores the practical relevance of our theoretical framework in actual applications.

A.4. Estimation of $I(X;T)$ and $I(X;Q|T)$

Among the three variables, T is a discrete variable, while X and Q are continuous variables in Euclidean spaces and do not follow standard distributions. In this paper, we estimate mutual information values via the empirical distributions $\hat{X}$ and $\hat{Q}$, which are uniform over $\mathcal{X}$ and $\mathcal{Q}$, respectively.

In other words,

$$I(X;T) = \lim_{|\mathcal{X}|\to\infty}\frac{1}{|\mathcal{X}|}\sum_{\hat{X}\in\mathcal{X}}\log\frac{p(\hat{X}|\hat{T})}{1/|\mathcal{X}|}, \tag{32}$$

$$I(X;Q|T) = \lim_{|\mathcal{X}|\to\infty}\frac{1}{|\mathcal{X}|}\sum_{\hat{X}\in\mathcal{X}}\sum_{\hat{Q}\in\mathcal{Q}}\log\frac{p(\hat{X}|\hat{Q})}{p(\hat{T}|\hat{Q})}\frac{1/|\mathcal{T}|}{1/|\mathcal{X}|}. \tag{33}$$

Here, $p(\hat{X}|\hat{T})$ is determined by the indexing. In contrast, $p(\hat{X}|\hat{Q})$ is unrelated to the indexing and is a degenerate distribution: it has a probability mass of 1 if and only if $\hat{X}$ is the gold-standard document for query Q. We estimate $p(\hat{T}|\hat{Q})$ with a transformer neural network after fitting to the dataset.

The estimation of $p(\hat{X}|\hat{T})$ is not straightforward. In standard GDR, a document is assigned a distinct ID string, and $p(\hat{X}|\hat{T})$ is a degenerate distribution. In this case, $I(X;T)$ reaches its maximum, i.e., $H(X) = \log|\mathcal{X}|$, which corresponds to the rightmost points of the IB curves shown in Figure 2, where $\beta = \infty$.

To obtain the rest of the IB curves ($\beta < \infty$) in Figure 2, we used a generalized GDR task that permitted ID collisions. This was implementing by training the transformer neural network to predict prefixes of a document ID string $[t_1, \cdots, t_l]$, instead of the whole string $[t_1, \cdots, t_m]$, where $l < m$. An ID prefix thus corresponds to multiple documents. In this setting,

$$p(\hat{X}|\hat{T} = t) = \begin{cases} 1/|\mathcal{X}_t^l| & \hat{x} \in \mathcal{X}_t^l \\ 0 & \text{otherwise} \end{cases}, \tag{34}$$

where $\mathcal{X}_t^l \subset \mathcal{X}$ is the set of documents for which the ID prefix is identical to the first l digits of t.

B. More Results on Information Bottleneck Curves

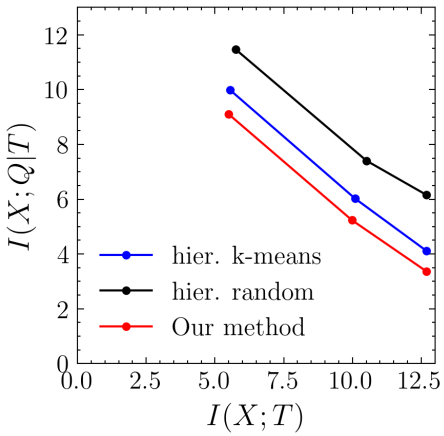


Figure 6: Information bottleneck curves for different indexing methods on the NQ320K training set with the T5-mini model.

Figure 6 compares our proposed method (red) with two other indexing methods on the information bottleneck plane, as measured with the T5-mini model on the NQ320K training set. Our method's curve is consistently located to the lower left of the other two methods' curves. This suggests that our BMI method reduces the bottleneck.

C. Details of Finetuning `docT5query`

The `docT5query` model underwent fine-tuning using the standard next-word prediction task and a cross-entropy loss function. Specifically, given a query q and its corresponding gold-standard document x, the model parameters were adjusted to maximize the likelihood $p(q|x) = \prod_{i=1}^{|q|-1} p(q_i|x; q_1, \cdots, q_{i-1})$, where q_i denotes the i-th word in the query text, and $|q|$ is the query length. Updates to the parameters were implemented using the AdamW optimizer (Loshchilov & Hutter, 2017), with $\beta_1 = 0.9$, $\beta_2 = 0.999$, eps $= 10^{-8}$, and a weight decay of 0.01. The learning rate was set at 5×10^{-5}. This finetuning process was executed 10 epochs on the training set of NQ320K.

D. Visualized Comparison of μ_x and $\mu_{Q|x}$

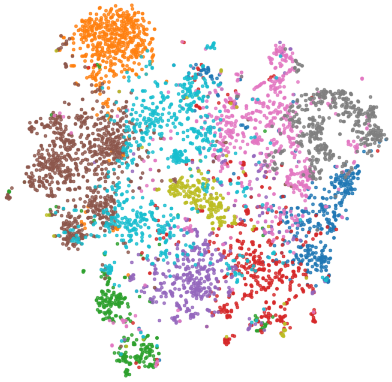
Figure 7 shows a visualization, using the t-SNE method (Van der Maaten & Hinton, 2008), of the vectors on which we applied the hierarchical k-means algorithm to obtain the indexing. 5,000 documents were randomly selected and are each represented by a point in the graphs. Figures 7(a-d) respectively correspond to the four rows in each block of Table 3. (a) was obtained by using the documents' BERT embeddings, i.e., $\{\mu_x : x \in \mathcal{X}\}$, and (b-d) were generated by using the mean vectors of queries, i.e., $\{\mu_{Q|x} = \text{mean}(\text{BERT}(\mathcal{Q}_x)) : x \in \mathcal{X}\}$ where $\mathcal{Q}_x$ consists of GenQ queries, GenQ+RealQ queries, and all queries (i.e., GenQ+RealQ+DocSeg) for (b-d), respectively. In (e), we report the result of (d) when GenQ queries were produced with the finetuned `docT5query` model using the training set of NQ320K.

Documents that were categorized in the same cluster are indicated by the same color. The cluster borders are more evident in (d-e) than in (a-c). This visualization suggests that $\mu_{Q|x}$ better represents "semantics" in the ID prefixes than μ_x does, which explains why our method achieved better retrieval performance.

(a) Doc

(b) GenQ

(c) GenQ+RealQ



(d) GenQ+RealQ+DocSeg

(e) GenQ (finetuned) + RealQ + DocSeg

Figure 7: Visualization of (a) μ_x and (b-d) $\mu_{Q|x}$ obtained with 5,000 documents in the NQ320K dataset. The colors indicate different clusters assigned by k-means clustering in the original high-dimensional space ($k = 10$). Every point represents a document and was obtained as (a) the document's BERT embedding, (b) the mean BERT embedding of GenQ queries, (c) the mean BERT embedding of GenQ+RealQ queries, (d-e) the mean BERT embedding of all queries (GenQ+RealQ+DocSeg). The GenQ queries used to produce (e) were generated with a finetuned version of the model on NQ320K, as detailed in Appendix C, whereas those for (b-d) were not finetuned.