Generative retrieval as multi-vector dense retrieval

Wu, S.; Wei, W.; Zhang, M.; Chen, Z.; Ma, J.; Ren, Z.; ... ; Ren, P.

Generative Retrieval as Multi-Vector Dense Retrieval

Shiguang Wu
Shandong University
Qingdao, China
shiguang.wu@mail.sdu.edu.cn

Wenda Wei
Shandong University
Qingdao, China
weiwenda@mail.sdu.edu.cn

Mengqi Zhang
Shandong University
Qingdao, China
mengqi.zhang@sdu.edu.cn

Zhumin Chen
Shandong University
Qingdao, China
chenzhumin@sdu.edu.cn

Jun Ma
Shandong University
Qingdao, China
majun@sdu.edu.cn

Zhaochun Ren
Leiden University
Leiden, The Netherlands
z.ren@liacs.leidenuniv.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

Pengjie Ren*
Shandong University
Qingdao, China
jay.ren@outlook.com

ABSTRACT

For a given query generative retrieval generates identifiers of relevant documents in an end-to-end manner using a sequence-to-sequence architecture. The relation between generative retrieval and other retrieval methods, especially those based on matching within dense retrieval models, is not yet fully comprehended. Prior work has demonstrated that generative retrieval with atomic identifiers is equivalent to single-vector dense retrieval. Accordingly, generative retrieval exhibits behavior analogous to hierarchical search within a tree index in dense retrieval when using hierarchical semantic identifiers. However, prior work focuses solely on the retrieval stage without considering the deep interactions within the decoder of generative retrieval.

In this paper, we fill this gap by demonstrating that generative retrieval and multi-vector dense retrieval share the same framework for measuring the relevance to a query of a document. Specifically, we examine the attention layer and prediction head of generative retrieval, revealing that generative retrieval can be understood as a special case of multi-vector dense retrieval. Both methods compute relevance as a sum of products of query and document vectors and an alignment matrix. We then explore how generative retrieval applies this framework, employing distinct strategies for computing document token vectors and the alignment matrix. We have conducted experiments to verify our conclusions and show that both paradigms exhibit commonalities of term matching in their alignment matrix.

Our findings apply to many generative retrieval identifier designs and provide possible explanations on how generative retrieval can express query-document relevance. As multi-vector dense retrieval is the state-of-the-art dense retrieval method currently, understanding the connection between generative retrieval and multi-vector dense retrieval is crucial for shedding light on the underlying mechanisms of generative retrieval and for developing, and understanding the potential of, new retrieval models.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Generative Retrieval; Dense Retrieval; Multi-Vector Dense Retrieval

ACM Reference Format:
Shiguang Wu, Wenda Wei, Mengqi Zhang, Zhumin Chen, Jun Ma, Zhaochun Ren, Maarten de Rijke, and Pengjie Ren. 2024. Generative Retrieval as Multi-Vector Dense Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657697

1 INTRODUCTION

In recent years, the advent of pre-trained language models has catalyzed the popularity of neural-based retrieval models within the information retrieval community [13, 14, 29, 32, 38].

Neural-based retrieval models. One family of effective neural-based retrieval methods, *dense retrieval* (DR), has achieved the state-of-the-art ranking performance on multiple benchmarks [13, 14, 29]. Several approaches have been proposed to use multiple vectors to represent documents or queries, a.k.a., *multi-vector dense retrieval* (MVDR) [14, 32, 48].

Recently, *generative retrieval* (GR) has emerged as a new paradigm in information retrieval. It aims to generate identifiers of relevant documents for a given query directly and parametrizes the indexing, retrieval, and ranking process in dense retrieval systems

*Corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657697

into a single model. GR adopts a sequence-to-sequence architecture model and is trained to directly map queries to their relevant document identifiers.

Generative retrieval vs. dense retrieval. Dense retrieval models typically employ encoders, e.g., BERT [6], for encoding both queries and documents, while the generative retrieval model adopts an encoder for query encoding and a decoder for identifier generation. Despite their superficial differences, dense retrieval and generative retrieval share key characteristics in their query-document relevance computations. When the two methods use document identifiers such as sub-strings, titles, or semantic IDs as representations for documents, both methods compute relevance to a query of a document as the dot product of two vectors. Dense retrieval involves using the direct product of the query vectors and document vectors as the relevance, while generative retrieval leverages the product of the last latent state from the decoder at each position with the prediction head, a.k.a., the word embedding lookup table. Consequently, a natural question that arises in this context:

How is generative retrieval related to dense retrieval?

Although GR has shown promising results in various benchmarks as a new end-to-end retrieval paradigm [18, 24, 36, 39, 42], relatively few publications have closely examined how GR models work. Nguyen and Yates [28] have shown that GR using atomic identifiers can be viewed as a variant of bi-encoders for dense retrieval because the word embedding lookup table in generative retrieval works exactly the same as the flat index in dense retrieval. Thus, we can partially respond to the above question that GR with atomic identifiers *is* single-vector dense retrieval. Although atomic identifiers are considered non-mainstream in GR, it offers an insightful perspective on the matter. Nguyen and Yates [28] also discuss that GR with hierarchical semantic identifiers exhibits behavior similar to hierarchical search within a tree index in dense retrieval. However, their discussion focuses only on the retrieval stage without rigorously considering deep interactions within the decoder.

Generative retrieval as multi-vector dense retrieval. In this work, we connect generative retrieval to a state-of-the-art dense retrieval method, multi-vector dense retrieval, in a rigorous way. We illustrate that these two methods exhibit *commonalities in their training targets and a shared focus on semantic matching*. We first examine the attention layer and the prediction head of GR and show that the logits in the loss function can be reformulated to a product of document word embeddings, query token vectors, and attention matrix in Section 4. This corresponds to the unified MVDR framework introduced in [17, 32]. In Section 5 we explore the distinct document encoding and alignment strategy in GR. Specifically, our discussion includes (i) its simple document encoding and how prefix-aware weight-adaptive (PAWA) decoding [39] and non-parametric (NP)-decoding [16] apply to our framework (Section 5.1), and (ii) the distinct alignment strategy employed by GR compared to MVDR and its properties (Section 5.2).

Our discovery provides reliable explanations of how GR can express query-document relevance. By explaining how the GR method models query-document relevance, we can further understand how GR is fundamentally different from dense retrieval methods and adds to the spectrum of neural-based retrieval models. The connection we present provides the variants of GR methods with a theoretical foundation for further improvement.

Contributions. Our main contributions in this paper are:

(1) We offer new insights into GR from the perspective of MVDR by showing that these methods share the same framework for measuring query-document relevance.
(2) We explore how GR applies this framework, employing distinct strategies for document encoding and the alignment matrix.
(3) We also conduct extensive analytical experiments based on the framework to verify our conclusions and illustrate the term-matching phenomenon and properties of different alignment directions in both paradigms.

2 RELATED WORK

Multi-vector dense retrieval (MVDR) can be seen as a generalization of single-vector dual encoder models [13, 14]. Instead of encoding the complete content of both query and documents into a single low-dimensional vector, MVDR uses fine-grained token-level modeling for scoring. MVDR models such as ColBERT [14] compute query-document relevance by selecting the highest-scoring document token for each query token and aggregating the scores. The postponed token-level interactions allow us to efficiently apply the model for retrieval, benefiting the effectiveness of modeling fine-grained interactions. MVDR overcomes the limited expressivity of single-vector retrieval and achieves significantly better results across various benchmarks [13, 14, 17, 23, 29, 32]. However, due to the cost of storing vectors for each document token, it is challenging to scale the approach to large collections [9, 14, 17, 32].

Generative retrieval (GR) is an emerging paradigm in information retrieval [18, 31, 34, 38, 39, 45]. It leverages generative models to directly generate identifiers of relevant documents. This approach originated with [2, 38] and has garnered considerable attention [see, e.g., 37]. Currently, all implementations of the generative retrieval paradigm adhere to an encoder-decoder transformer architecture, e.g., T5 [33] and BART [19]. In this method, documents are initially associated with a concise token sequence that serves as an identifier. The model is then trained to predict this token sequence autoregressively, using conventional cross-entropy loss.

One notable advantage of the generative retrieval model is its streamlined end-to-end architecture, which requires significantly less disk storage space compared to other retrieval methods. However, it is important to note that due to the limited supervision of each token, the generative retrieval may not achieve comparable performance when compared to dense retrieval [31, 44].

Connecting dense retrieval and generative retrieval. Nguyen and Yates [28] show that GR with atomic identifiers is equivalent to single-vector dense retrieval. They compare the inferential processes of DR with a tree index and GR with hierarchical identifiers. However, the former is just an optimized version of the original DR without changing the semantic matching method, while the latter also considers the predicted IDs and the query in each generation step, which greatly affects how GR would express the relevance, but this is ignored in [28]. Yuan et al. [44] empirically analyze the error rate at each generation step of GR and identify the problem

of poor memory accuracy for fine-grained features compared with DR. They integrate GR and DR into a new coarse-to-fine retrieval paradigm, combining their respective strengths, but circumvented an in-depth discussion of the connection.

In this work, we address the limitations listed above by showing that GR expresses query-document relevance in the same way as MVDR. This connection is rigorously derived from the decoder of GR and can be applied to many identifiers.

3 PRELIMINARIES

In this section, we formulate our task and introduce key notation and the mainstream framework of MVDR models.

Task definition. We formulate the retrieval task as ranking by relevance score. Given a query q, we aim to retrieve relevant documents d in $\mathcal{D}$ by ranking them by their relevance $\text{rel}(d, q)$ to q.

Notation. Table 1 lists the main notation used in the paper. We denote the word embedding lookup table in the decoder as $\boldsymbol{E}$ and the vocabulary set as $\mathcal{V}$. Each document d comprises M tokens. To ensure uniform length, padding tokens are added or excess tokens are truncated from each document. The word embedding matrix of d is denoted as $\boldsymbol{E}_d \coloneqq [\boldsymbol{e}_{d_1}, \ldots, \boldsymbol{e}_{d_M}] \in \mathbb{R}^{d \times M}$, and the latent token vector matrix after encoding is $\boldsymbol{D} \coloneqq [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_M] \in \mathbb{R}^{d \times M}$. Each query q with N tokens has token vectors $\boldsymbol{Q} \coloneqq [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N] \in \mathbb{R}^{d \times N}$ after encoding, similar to the documents.

Table 1: Main notation used in this work.

Symbol	Description
$\boldsymbol{E}$	word embedding lookup table
$\boldsymbol{E}_d$	document word embedding matrix
$\boldsymbol{D}, \boldsymbol{Q}$	document / query token vector matrix
d_i, q_j	document / query tokens
$\boldsymbol{d}_i, \boldsymbol{q}_j$	encoded document / query token vector

Framework for MVDR. Following [17, 32], MVDR methods can be represented as a unified framework, in which the relevance to query q of document d is given by:

$$\text{rel}(d, q) = \frac{1}{Z} \text{sum}(\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A}) = \frac{1}{Z} \sum_{i,j} \boldsymbol{d}_i^\top \boldsymbol{q}_j A_{ij}, \tag{1}$$

where $\boldsymbol{A}$ is the alignment matrix that controls whether a document and query token pair can be matched and contribute to the relevance, and $Z = \sum_{ij} A_{ij}$ is used for normalization and is dropped in many MVDR methods.

Alignment strategy. Different MVDR models adopt different alignment strategies, and, thus, a different alignment matrix $\boldsymbol{A}$. It is often computed using heuristic algorithms, such as lexical exact match [11], top-1 relevant token match [14], single-vector alignment [13, 25], or sparse unary salience [32].

Contrastive loss used in MVDR. MVDR methods usually use contrastive loss as the training target, where negative documents are used. For a query q and target document d, the loss is computed as

$$\mathcal{L}(d, q) = -\log \frac{\exp \text{rel}(d, q)}{\sum_{d^- \in \mathcal{D}^-} \exp \text{rel}(d^-, q)}, \tag{2}$$

where $\mathcal{D}^-$ is the collected negative set.

4 IN-DEPTH ANALYSIS OF GENERATIVE RETRIEVAL

To address the question posed in Section 1, this section conducts a detailed analysis of GR. Specifically, we first illustrate the model architecture and training loss of the GR (Section 4.1). Subsequently, we derive that the training target of GR falls into the framework of MVDR (Section 4.2):

$$\mathcal{L}(d, q) \propto \text{sum}\left(\tilde{\boldsymbol{E}}_d^\top \boldsymbol{Q} \odot \boldsymbol{A}\right), \tag{3}$$

where $\tilde{\boldsymbol{E}}_d$, $\boldsymbol{Q}$ and $\boldsymbol{A}$ correspond to $\boldsymbol{D}$, $\boldsymbol{Q}$ and $\boldsymbol{A}$ in Eq. (1).

4.1 Model architecture and training loss

Model architecture. We focus on the transformer sequence-to-sequence architecture used in GR, more precisely, the encoder-decoder structure. Within this framework, the encoder primarily targets processing the input query, while the decoder is tasked with predicting document identifiers.

The decoder component consists of stacks of self-attention, cross-attention, and feed-forward layers. We particularly underscore the significance of the cross-attention layers, as they facilitate interaction between query tokens and document tokens.

To predict the document token d_i at the i-th position, we compute the cross attention weights between query token vectors $\boldsymbol{Q}$ and $\hat{\boldsymbol{d}}_{i-1}$ from the previous attention layers at position $(i-1)$ as follows:

$$\boldsymbol{\alpha}_i = \text{softmax}(\boldsymbol{Q}^\top \boldsymbol{W} \hat{\boldsymbol{d}}_{i-1}), \tag{4}$$

where $\text{softmax}(\cdot)$ denotes the column-wise softmax function, $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ is the product of the attention matrices $\boldsymbol{W}_K$ and $\boldsymbol{W}_Q$, i.e., $\boldsymbol{W} = \boldsymbol{W}_K^\top \boldsymbol{W}_Q$, and $\boldsymbol{\alpha}_i \in \mathbb{R}^N$.

Consequently, the output of the cross-attention layer is

$$\boldsymbol{h}_i = \boldsymbol{W}_V \boldsymbol{Q} \boldsymbol{\alpha}_i \in \mathbb{R}^d. \tag{5}$$

For simplicity, we ignore the non-linear activation functions, and the linear maps in the feedforward layers can be absorbed in attention weights, e.g., $\boldsymbol{W}_V$. Therefore, $\boldsymbol{h}_i$ serves as the prediction head for generating the next token.

Training loss. The loss function to minimize at position i is formulated as:

$$\mathcal{L}_i(d, q) = -\log \left(\frac{\exp \boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i}{\sum_{v \in \mathcal{V}} \exp \boldsymbol{e}_v^\top \boldsymbol{h}_i} \right) \tag{6}$$

$$= -\boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i + \log \sum_{v \in \mathcal{V}} \exp \boldsymbol{e}_v^\top \boldsymbol{h}_i. \tag{7}$$

4.2 GR has the same framework as MVDR

Next, we demonstrate that GR shares a similar framework with MVDR, namely, that the logits within the loss function can be reformulated as a product of document word embeddings, query token vectors, and attention matrix. This formulation corresponds to Eq. (1).

In particular, as we employ teacher-forcing supervision, ground-truth document identifiers are directly fed into the decoder, and token vectors at all positions are computed simultaneously. Based

on this configuration, the overall loss is given by:

$$\mathcal{L}(d,q) = \sum_{i\in[M]} \log p(d_i \mid d_{i-1},\dots,d_0,q) \tag{8}$$

$$= \sum_{i\in[M]} \mathcal{L}_i(d,q) \tag{9}$$

$$= \sum_{i\in[M]} \left(-\boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i + \log \sum_{v\in\mathcal{V}} \exp \boldsymbol{e}_v^\top \boldsymbol{h}_i\right), \tag{10}$$

where d_0 could be some special token such as [BOS] or the [CLS] token vector from the query.

When using the sampled softmax loss, which involves employing several negative tokens instead of the entire set of tokens in the lookup embedding table, the loss exhibits a similar structure to the contrastive loss used in DR and MVDR. Consequently, we treat the dot product of embedding $\boldsymbol{e}_{d_i}$ and token vector $\boldsymbol{h}_i$, i.e., $\boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i$, as the final relevance score at position i. Further insights into the reason are elaborated in Appendix A. We plug in the dot product with the computation of $\boldsymbol{h}_i$ from Eq. (5) and obtain:

$$\text{rel}(d,q) = \sum_{i\in[M]} \boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i \tag{11}$$

$$= \sum_{i\in[M]} \boldsymbol{e}_{d_i}^\top \boldsymbol{W}_V \boldsymbol{Q} \boldsymbol{\alpha}_i \tag{12}$$

$$= \sum_{i\in[M]} \sum_{j\in[N]} \tilde{\boldsymbol{e}}_{d_i}^\top \boldsymbol{q}_j \alpha_{ij} \tag{13}$$

$$= \text{sum}\left(\tilde{\boldsymbol{E}}_d^\top \boldsymbol{Q} \odot \boldsymbol{A}\right), \tag{14}$$

where $\tilde{\boldsymbol{e}}_{d_i}^\top = \boldsymbol{e}_{d_i}^\top \boldsymbol{W}_V$, $\tilde{\boldsymbol{E}}_d^\top = \boldsymbol{E}_d^\top \boldsymbol{W}_V$, $\boldsymbol{A} = [\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_M]^\top \in \mathbb{R}^{M\times N}$, and $\odot$ is the element-wise matrix product operation.

Further, we have a more detailed computation

$$\text{rel}(d,q) = \text{sum}\left(\tilde{\boldsymbol{E}}_d^\top \boldsymbol{Q} \odot \boldsymbol{A}\right) \tag{15}$$

$$= \text{sum}\left(\tilde{\boldsymbol{E}}_d^\top \boldsymbol{Q} \odot \text{softmax}\left(\hat{\boldsymbol{D}}_{-1}^\top \boldsymbol{W}^\top \boldsymbol{Q}\right)\right), \tag{16}$$

where $\hat{\boldsymbol{D}}_{-1} = \left[\hat{\boldsymbol{d}}_0, \hat{\boldsymbol{d}}_1, \dots, \hat{\boldsymbol{d}}_{M-1}\right]$ is the output from the previous layer with the right-shifted document tokens as model input.

In conclusion, for GR we observe a similar framework as for MVDR, $\text{rel}(d,q) = \text{sum}(\tilde{\boldsymbol{E}}_d^\top \boldsymbol{Q} \odot \boldsymbol{A})$, where relevance is represented by an interaction of multiple "token vectors," i.e., $\tilde{\boldsymbol{E}}_d$ and $\boldsymbol{Q}$, from both query and document and aligned by a matrix $\boldsymbol{A}$. We summarize our derivation and conclusion in Figure 1.

5 COMPARISON BETWEEN MVDR AND GR

To further explore how GR is related to MVDR, we build upon the unified framework of relevance computation for GR and MVDR derived in the previous section. We conduct a comprehensive analysis of both methods, focusing specifically on their similarities and disparities in terms of the document encoding and the design of the alignment matrix. A summary of the comparison between the two methods is shown in Table 2.

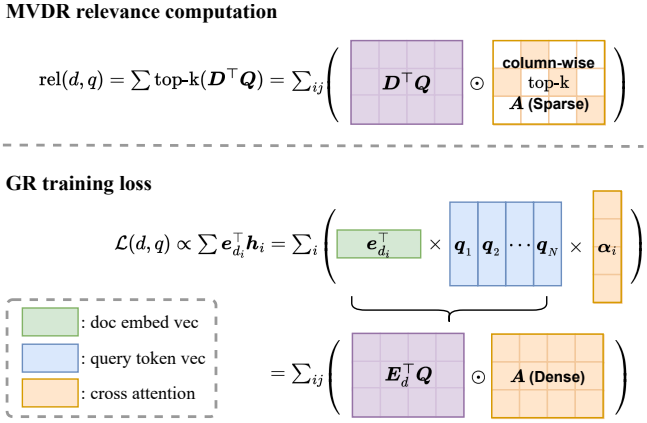


Figure 1: Summary of our derivation and conclusion. The logits of GR can be reformulated as $\text{sum}(\boldsymbol{E}_d^\top \boldsymbol{Q} \odot \boldsymbol{A})$, which corresponds to the framework $\text{sum}(\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A})$ of MVDR.

Table 2: Summary of our comparison between MVDR and GR.

Component in $\text{sum}(\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A})$	Model: MVDR (Sect. 3)	Model: GR (Sect. 4)	Comparison
$\boldsymbol{D}$ doc token	$\boldsymbol{D}$ (token vec.)	$\tilde{\boldsymbol{E}}_d$ (embed. vec.)	Sect. 5.1
$\boldsymbol{Q}$ query token	$\boldsymbol{Q}$ (token vec.)	$\boldsymbol{Q}$ (token vec.)	—
$\boldsymbol{A}$ alignment matrix	sparse query-to-doc	dense and learned doc-to-query	Sect. 5.2

5.1 Document encoding

One of the noticeable differences between GR and MVDR is in the document encoding. As depicted in Figure 1, MVDR uses more expressive contextualized token vectors $\boldsymbol{D} = [\boldsymbol{d}_1,\dots,\boldsymbol{d}_M]$ for each position. In contrast, GR only attends each query token to a simple word embedding $\boldsymbol{e}_{d_i}$ that does not hold any contextual information about the document. This was considered a severe compromise for the extremely lightweight modeling and storage of GR. To address this imbalance in modeling capacity, several studies [16, 39] have proposed novel decoding methods. Wang et al. [39] introduce the prefix-aware weight-adaptive (PAWA) decoding method, while Lee et al. [16] propose the non-parametric (NP) decoding. We incorporate these methods into our framework and show how they fundamentally improve the encoding compared with MVDR in Table 3.

Table 3: Document encoding comparison between GR and MVDR. PAWA and NP-decoding either multiply or replace the simple embedding vectors $\tilde{\boldsymbol{E}}_d$ with contextualized token vectors $\tilde{\boldsymbol{D}}$.

Model	Document encoding
MVDR (Sect. 3)	$\boldsymbol{D}$ (token vec.)
GR (Sect. 4)	$\tilde{\boldsymbol{E}}_d$ (embed. vec.)
– w/ PAWA	$\tilde{\boldsymbol{E}}_d \rightarrow \tilde{\boldsymbol{E}}_d \tilde{\boldsymbol{D}}'$ (embed. & token vec.)
– w/ NP-dec.	$\tilde{\boldsymbol{E}}_d \rightarrow \boldsymbol{D}$ (token vec.)

PAWA enhances the document encoding from $\tilde{\boldsymbol{E}}_d$ to $\tilde{\boldsymbol{E}}_d\tilde{\boldsymbol{D}}'$. PAWA [39] aims to improve the embedding modeling for distinguishing different semantics of a token ID at different positions. Unlike the standard transformer, which uses a static embedding lookup table for every position, PAWA generates different embedding tables at each generation step. PAWA consists of a transformer decoder and an adaptive projection layer $\mathbf{E} \in \mathbb{R}^{M\times|\mathcal{V}|\times d\times d}$. The projection matrix of token v at the i-th position is

$$\boldsymbol{E}_{i,v} = \mathbf{E}[i, v, :, :] \in \mathbb{R}^{d\times d}.$$

Here, $\mathbf{E}$ can be seen as a generalized version of the embedding lookup table that uses a matrix $\boldsymbol{E}_{i,v}$ to represent each token v. To get the generated embedding vector for token v at the i-th position, PAWA decoder first uses the transformer decoder to process the document into a set of latent vectors $\boldsymbol{D}' = [\boldsymbol{d}'_1, \ldots, \boldsymbol{d}'_M] \in \mathbb{R}^{d\times M}$. Then it multiplies the projection matrix $\boldsymbol{E}_{i,v}$ with the latent vector $\boldsymbol{d}'_i$ and gets the final embedding vector $\boldsymbol{e}_{i,v} = \boldsymbol{E}_{i,v}\boldsymbol{d}'_i$.

Therefore, we have the logit $\boldsymbol{e}_v^\top \boldsymbol{h}_i$ in loss Eq. (10) replaced by $\boldsymbol{e}_{i,v}^\top \boldsymbol{h}_i = \boldsymbol{d}_i'^\top \boldsymbol{E}_{i,v}^\top \boldsymbol{h}_i$:

$$\mathcal{L}(d, q) = \sum_{i\in[M]} \left[-\boldsymbol{d}_i'^\top \boldsymbol{E}_{i,d_i}^\top \boldsymbol{h}_i + \log \sum_{v\in\mathcal{V}} \exp \boldsymbol{d}_i'^\top \boldsymbol{E}_{i,v}^\top \boldsymbol{h}_i \right]. \quad (17)$$

With a similar derivation as in Section 4.2, the relevance can be established as

$$\mathrm{rel}(d, q) = \mathrm{sum}\left(\tilde{\boldsymbol{D}}'^\top \tilde{\boldsymbol{E}}_d^\top \boldsymbol{Q} \odot \boldsymbol{A}\right), \quad (18)$$

where

$$\tilde{\boldsymbol{D}}' = \begin{bmatrix} \boldsymbol{d}'_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{d}'_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{d}'_M \end{bmatrix} \in \mathbb{R}^{(d\times M)\times M}, \quad (19)$$

and $\tilde{\boldsymbol{E}}_d = \boldsymbol{W}_V^\top [\boldsymbol{E}_{1,d_1}, \ldots, \boldsymbol{E}_{M,d_M}] \in \mathbb{R}^{d\times(d\times M)}$.

As we can see, PAWA multiplies the term $\tilde{\boldsymbol{E}}_d^\top \boldsymbol{Q}$ with contextualized document token vectors $\tilde{\boldsymbol{D}}'$, which greatly improves the expressivity of document encoding.

NP-decoding directly replaces $\tilde{\boldsymbol{E}}_d$ with $\boldsymbol{D}$. Lee et al. [16] employ an approach akin to contextualized sparse retrieval methods, leveraging token vectors encoded by the Contextualized Embedding Encoder [CE Encoder, 16], referred to as contextualized token embeddings. This set of vectors, written as $\boldsymbol{D}$ in our notation, serves as the embedding table for the decoder. Both the Base and Async non-parametric decoding methods in [16] can be reformulated within our framework as:

$$\mathrm{rel}(d, q) = \mathrm{sum}(\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A}), \quad (20)$$

where $\boldsymbol{D}$ are the token vectors of documents either pre-computed (as done by the pre-trained T5 model and frozen in the Base method) or gradually updated (by the encoder of the GR model every N epochs in the Async method) during the training of the GR model.

While the NP-decoding method shares the same document encoding with MVDR, two significant differences exist:

(1) In NP-decoding, $\boldsymbol{D}$ is mostly frozen and detached from training, causing training imbalance compared to MVDR. $\boldsymbol{Q}$ and $\boldsymbol{A}$ in NP-decoding are fully trained, while $\boldsymbol{D}$ remains frozen.
(2) Due to GR computing logits for the entire vocabulary in each generation step, there is a need to reduce the large storage footprint of $\boldsymbol{D}$ to save computation. NP-decoding methods address this by using clustering to compress token vectors. MVDR, on the other hand, achieves lower inference time through a sparse alignment strategy.

5.2 Alignment strategy

In addition to document encoding, the alignment matrix represents a crucial distinction between GR and MVDR methods. This matrix plays a decisive role in shaping the divergent inference procedures employed in retrieval. In this section, we analyze the alignment matrix, denoted as $\boldsymbol{A}$ within our unified framework, in terms of sparsity, alignment direction, and some common properties.

5.2.1 The concept of "alignment" in both methods. The concept of "alignment" has garnered significant attention in MVDR [8, 17, 22, 32]. We will briefly introduce the alignment problem in MVDR models, and claim that the alignment matrix of the GR method, as asserted in our framework Eq. (16), indeed exhibits similar alignment functionality to MVDR models.

Token alignment involves determining whether tokens from the query and document should be matched lexically or semantically. It essentially represents another formulation of the "term mismatch problem" [9, 14, 32, 47]. The prevailing strategy considered optimal at present is the all-to-all soft alignment strategy in MVDR models [14, 32], which eliminates the lexical form match restriction.

GR methods leverage the transformer architecture that originated in NLP, and the concept of "alignment" has been extensively discussed in the domain of neural machine translation [3, 5, 20, 21], focusing on the alignment between tokens in source and target sentences. The attention mechanism, as a core component, computes the alignment matrix and proves highly effective in capturing alignment between source and target sentences [5, 12, 43, 46]. Theoretical work [7, 30] has validated the phenomenon of copying behavior, forming a foundational basis for the alignment ability.

We conclude that, in GR methods, the attention matrix is able to capture the alignment between the query and the document, akin to the alignment matrix observed in MVDR methods.

5.2.2 Different sparsity and learnability: sparse vs. dense and learned alignment matrices. The alignment matrices of MVDR and GR differ in sparsity and learnability. MVDR typically employs a sparse alignment matrix for maximum efficiency during inference. In contrast, GR uses a dense and fully learnable alignment matrix derived from the computationally intensive attention mechanism.

For MVDR methods, the sparse alignment matrix is often computed using heuristic algorithms [11, 14, 32]. Taking ColBERT [14] as an example, it selects the most relevant document token for each query token. The relevance score between document d and query

q is computed as

$$\begin{aligned}\mathrm{rel}(d,q) &= \mathrm{sum}\left(\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A}\right) = \sum_{i,j} \boldsymbol{d}_i^\top \boldsymbol{q} A_{ij} \\ &= \sum_{j\in[N]} \max_{i\in[M]} \boldsymbol{d}_i^\top \boldsymbol{q}_j = \sum_{j\in[N]} \mathrm{rel}(d,q_j), \end{aligned} \tag{21}$$

where $\boldsymbol{A}$ is a sparse alignment matrix with only one non-zero element for each column ($A_{ij} = 1$ if $\boldsymbol{d}_i^\top \boldsymbol{q}_j = \max_{i\in[M]} \boldsymbol{d}_i^\top \boldsymbol{q}_j$; $A_{ij} = 0$ otherwise). The sum-max operation is highly parallelizable, ensuring efficiency during inference.

For GR methods, the alignment matrix is computed through the attention mechanism, considering all possible pairs of query and document tokens, as shown in Eq. (16).

The dense alignment matrix is highly expressive and trainable. While not suitable for exact relevance score computation in inference for each query-document pair, efficient approximate algorithms such as greedy search or beam search can be used to retrieve the top-k documents. These decoding algorithms rely on the following decomposition:

$$\mathrm{rel}(d,q) = \sum_{i\in[M]} \mathrm{sum}\left(\tilde{\boldsymbol{e}}_{d_i}^\top \boldsymbol{Q}\boldsymbol{\alpha}_i\right) = \sum_{i\in[M]} \mathrm{rel}(d_i,q), \tag{22}$$

where $\mathrm{rel}(d_i,q)$ is conditioned on $d_0,\ldots,d_{i-1}$, approximating the search for the most relevant document d to finding the most relevant token d_i at each position i.

5.2.3 Different alignment directions: query-to-document vs. document-to-query alignment. Beyond differences in the sparsity and learnability of the alignment matrix, MVDR and GR exhibit distinctions in their alignment directions.

Eq. (21) reveals that MVDR's relevance score can be decomposed into the sum of relevance scores for each query token and its aligned document token. In this context, we consider the alignment matrix in MVDR as **query-to-document** alignment. Each query token individually aligns to a document token, seeking the optimal match during retrieval. Mathematically, the alignment matrix is computed *column-wise* and represents a one-hot vector for each column.

Conversely, the relevance score of GR, as depicted in Eq. (22), is the sum of relevance scores for each document token and its softly aligned query token. Here, we categorize the alignment matrix in GR as **document-to-query** alignment. Each document token is considered individually to focus attention on the most relevant query token. The alignment matrix is computed *row-wise* with a softmax$(\cdot)$ operation to normalize attention weights in each row.

Document-to-query alignment may seem counter-intuitive for a retrieval task, as we do not know the target documents while predicting. As a solution, GR pre-computes the alignment strategy for the document token d_i (to be predicted) using *previous* document tokens $d_0,\ldots,d_{i-1}$ and thus can retrieve the next token that best aligns with the desired *next* alignment strategy.

5.2.4 Low-rank nature of both alignment matrices. In analyzing the shared characteristics of the two alignment matrices, it is demonstrated that both matrices exhibit a low-rank property.

MVDR models, e.g., AligneR [32], integrate the pairwise alignment matrix with unary salience, given by

$$\boldsymbol{A} = \tilde{\boldsymbol{A}} \odot \boldsymbol{u}_d \boldsymbol{u}_q^\top. \tag{23}$$

Here, $\tilde{\boldsymbol{A}} \in \mathbb{R}^{M\times N}$ signifies the pairwise alignment matrix, determining the alignment of query and document tokens. The sparse token weights, $\boldsymbol{u}_d \in \mathbb{R}^M$ and $\boldsymbol{u}_q \in \mathbb{R}^N$, decide whether a token requires alignment. Notably, the alignment matrix $\boldsymbol{A}$ contains a low-rank component $\boldsymbol{u}_d\boldsymbol{u}_q^\top$ that influences the alignment strategy.

In the case of GR methods, the alignment matrix is computed using an attention mechanism, which inherently results in a low-rank matrix. A lemma provides evidence of this low-rank property and is presented briefly here, with a detailed proof delegated to Wu et al. [41, Appendix B] due to space limitations.

Lemma 5.1. *For a matrix $\boldsymbol{A} = \mathrm{softmax}\left(\boldsymbol{D}^\top \boldsymbol{W}\boldsymbol{Q}\right)$, there exists a rank-one matrix $\boldsymbol{R}$ such that*

$$\|\boldsymbol{A} - \boldsymbol{R}\| \le 4\gamma \|\boldsymbol{W}\|, \tag{24}$$

where the term γ depends on the matrix entries.

From this lemma, we can conclude that both MVDR and GR methods reveal a rank-one component in their alignment matrices.

5.2.5 Decomposition of both relevance scores. In this subsection, we show that the relevance score computation in both MVDR and GR models can be decomposed into query and document components.

The MVDR method employs a bi-encoder architecture, wherein query and document tokens are modeled separately. This architecture can easily be regarded as a decomposition of the relevance score between the query and document:

$$\mathrm{rel}(d,q) = \mathrm{sum}\left(\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A}\right) = \mathrm{sum}\left(\text{top-1}(\boldsymbol{D}^\top \boldsymbol{Q})\right),$$

where top-1$(\cdot)$ is the operator that selects the maximum value in each column of the matrix.

In the subsequent lemma, we establish that the relevance score of GR cannot only be decomposed but also be kernelized, implying the existence of a kernel function capable of processing both query vectors and document vectors to compute the score (further details are delegated to Wu et al. [41, Appendix C] due to space limitations):

Lemma 5.2. *For simplicity, let* $\mathrm{rel}(d,q) = \mathrm{sum}(\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A})$, *where* $\boldsymbol{A} = \mathrm{softmax}(\boldsymbol{D}^\top \boldsymbol{Q})$. *It can be kernelized as*

$$\begin{aligned}\mathrm{rel}(d,q) &= \sum_{i,j} \boldsymbol{d}_i^\top \boldsymbol{q}_j A_{ij} = \sum_{i,j} \boldsymbol{d}_i^\top \boldsymbol{q}_j \,\mathrm{softmax}\left(\boldsymbol{d}_i^\top \boldsymbol{Q}\right)_j \\ &= \sum_{i,j} \frac{1}{p_{ij}} \mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{d}_i)^\top \boldsymbol{F}(\boldsymbol{q}_j)\right), \end{aligned} \tag{25}$$

where $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{x}\phi(\boldsymbol{x})^\top$, and p_{ij} is a term that depends on $\boldsymbol{d}_i$ and $\boldsymbol{q}_j$. We choose elu$(\cdot)$ *as the kernel function $\phi(\cdot)$.*

Furthermore, by applying the trace inequality, we can approximately decompose the relevance score as

$$\mathrm{rel}(d,q) \le \sum_{i,j} \frac{1}{\hat{p}_i \hat{p}_j} \sqrt{\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{d}_i)^\top \boldsymbol{F}(\boldsymbol{d}_i)\right)} \sqrt{\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{q}_j)^\top \boldsymbol{F}(\boldsymbol{q}_j)\right)}.$$

From this lemma, we can conclude that both relevance scores in MVDR and GR methods can be decomposed. The decomposition of MVDR is more straightforward, and the kernelization of GR is more complicated. Both kernelizations would provide possibilities for new retrieval strategies.

5.3 Upshot

In summary, our findings indicate that certain studies enhance the modeling capacity of GR by employing more expressive document encoding, akin to MVDR. Furthermore, GR employs a distinct alignment direction, but it also exhibits similar low-rank and decomposition properties as MVDR.

6 EXPERIMENTAL SETUP

Next, we seek experimental confirmation that generative retrieval and multi-vector dense retrieval share the same framework for measuring relevance to a query of a document, as derived in Section 4.

6.1 Datasets

We conduct experiments on two well-known datasets, NQ [15] and MS MARCO [27]. We use the same settings and processed datasets as Sun et al. [36], and we summarize the statistics of the datasets in Table 4.

Table 4: Statistics of datasets used in our experiments.

Dataset	# Docs	# Test queries	# Train pairs
NQ320K	109,739	7,830	307,373
MS MARCO	323,569	5,187	366,235

NQ320k. NQ320K is a popular dataset for evaluating retrieval models [13, 24, 38, 39]. It is based on the Natural Questions (NQ) dataset [15]. NQ320k consists of 320k query-document pairs, where the documents are gathered from Wikipedia pages, and the queries are natural language questions.

MS MARCO. The MS MARCO document retrieval dataset is a collection of queries and web pages from Bing searches. Like NQ320k and following [36], we sample a subset of documents from the labeled documents and use their corresponding queries for training. We evaluate the models on the queries of the MS MARCO dev set and retrieval on the sampled document subset.

6.2 Base models

As we aim to provide a new perspective on GR as MVDR, we consider representative models from both paradigms, i.e., SEAL [1] for GR and ColBERT [14] for MVDR. For a fair comparison, we reproduce both methods using the T5 architecture [33]. We have made several changes to adapt ColBERT and SEAL to their T5 variants:[1]

- **T5-ColBERT.** We use in-batch negative samples instead of the pair-wise samples in the official ColBERTv1 implementation. We set the batch size to 256 and train 5 epochs. Due to space limitations, details of our T5 variant ColBERT are delegated to Wu et al. [41, Appendix D].
- **T5-SEAL.** We use the Huggingface transformers library [40] to train the model. We use the constructed query-to-span data for training and each span has a length of 10 sampled according to Bevilacqua et al. [1]. The learning rate is set to 1e-3 and the batch size is 256.

[1]Our code link is https://github.com/Furyton/GR-as-MVDR.

6.3 Inference settings

We consider two inference settings: end-to-end and re-ranking.

End-to-end retrieval setting. Both methods can perform an end-to-end retrieval on the corpus for a given query.

- **T5-ColBERT** maintains a large vector pool of all document token vectors after training. During inference, it first retrieves for each query token vector, the k-nearest document token vectors in the vector pool, resulting in $N \times k$ retrieved vectors. These vectors are from at most $N \times k$ different documents which are used as candidates. It then computes the exact relevance score for each candidate document and performs the final re-ranking.
- **T5-SEAL** directly uses its generative style inference with the help of constrained beam search to predict valid document identifiers, i.e., n-grams from the documents.

Re-ranking setting. Since we are focusing on relevance computing in the training target, we introduce a re-ranking setting that removes the influence of different approximated inference strategies. As stated in some previous work [17, 24], both MVDR and GR have discrepancies between training and inference. The approximated retrieval methods are largely different from the training target and may decrease the performance of the trained retrievers. In the re-ranking setting, we collect 100 documents retrieved by BM25 [35] together with the ground-truth document as the candidate set for each query. As in the training stage, we take both the query and each candidate document as the input of the model and use the relevance computing in Section 3 and 4.

7 EXPERIMENTAL ANALYSES

7.1 Performance of different alignment directions

As described in Section 5.2.3, MVDR and GR exhibit different alignment directions, i.e., query-to-document and document-to-query alignment. We aim to look at how alignment directions affect retrieval performance. We first conduct experiments in the re-ranking setting to show the performance gap between MVDR and GR. As shown in Table 5, MVDR with the original alignment strategy, which is indicated as MVDR (q→d), has a much better performance than GR. To compare the alignment directions of MVDR and GR, we have designed a model MVDR (q←d) that integrates the features of both, i.e., expressive document encoding from MVDR and document-to-query alignment strategy from GR. From Table 5, we can see that the performance of the new model is roughly intermediate between the other two. Note that the designed experimental model MVDR (q←d) can only be used in a re-ranking setting. We conclude that query-to-document alignment is preferred for re-ranking.

7.2 Term matching in alignment

As we have discussed in Section 5.2.1, alignment is essentially a term-matching problem. In this section, we design an experiment to observe the extent of term matching in the two methods, and we find that both methods exhibit a preference for exact term matching in their alignment.

Exact match of MVDR in query-to-document direction. We calculate the exact matching rate between document token IDs and

Table 5: Comparison of MVDR and GR in the re-ranking setting. MVDR and GR are our reproduced T5-SEAL and T5-ColBERT. "R" denotes Recall, and "M" denotes MRR.

	NQ320K			MS MARCO		
Model	R@1	R@10	M@10	R@1	R@10	M@10
MVDR (q→d)	61.3	91.9	72.0	46.5	84.5	58.9
MVDR (q←d)	53.2	90.1	65.7	34.8	78.8	48.4
GR	47.4	87.0	60.5	35.3	77.1	48.3

each query token ID during the alignment process, which we refer to as the "hard exact match rate." We also define a "soft exact match rate" which is the alignment score corresponding to the exact match query-document token pairs. The alignment score is defined as the element in the alignment matrix. As MVDR uses a sparse alignment matrix, we apply column-wise softmax$(\cdot)$ to $\boldsymbol{A}$ and use the element as the alignment score. We average the rate over candidate documents for each query token and categorize the query tokens according to their IDF. We assume that IDF approximates the term importance as is done in [10]. From the results in Figure 2, we can see that MVDR chooses exactly matched document tokens in 11.4% on average. Also, we notice that rare query tokens have not received much attention during alignment. This observation suggests that MVDR may prioritize commonly occurring query tokens in its alignment process, potentially overlooking or underemphasizing the importance of rare query tokens.

Exact match of MVDR and GR in document-to-query direction. We have devised an experiment to investigate the alignment in the opposite direction, i.e., document-to-query, in Figure 3. As GR is not trained with hard alignment, we only examine the soft exact match rate of both methods. The computation of the exact match rate is similar except that it is computed for each document token. From the results, we have discerned a consistent trend: as the importance of tokens increases, the rate of exact matches also tends to rise. We think this is because it is hard for the rare query token to match among many common tokens since the document is much longer than the query. When we look at each document token, it will be easier to match among fewer query tokens. We also conduct experiments on MS MARCO and have similar results.

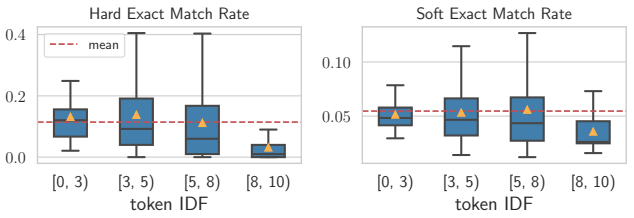


Figure 2: Exact match rate of MVDR on NQ320k dataset in the query-to-document direction.

7.3 Improved document encoding

In Section 5.1, we include two popular document encoding methods, PAWA and NP-decoding, into our framework. To demonstrate the improvement of these two methods, we compare the performance of GR with and without them in Table 6. PAWA is typically used in GR with short semantic identifiers due to its high computational

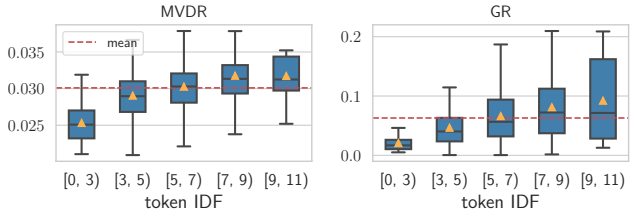


Figure 3: Soft exact match rate of MVDR and GR on NQ320k dataset in the document-to-query direction.

complexity during generation. Thus, we compare it with DSI [38] with semantic identifiers. Note that all these models use the same architecture (T5-base [33]) and similar training procedures without data augmentation, e.g., synthetic query-doc pairs generation, etc. *-PAWA and *-NP can be seen as a naive approach to using the two enhancing methods to the base GR models. From Table 6, we see that both PAWA and NP-decoding can greatly improve the performance and achieve similar results compared with T5-ColBERT on Recall@1. However, there is still a large gap in terms of Recall@10. The implementation of the additional decoding modules is only an approximation for reducing the cost of time and storage as discussed in Section 5.1. This, together with the alignment direction, may be a cause of the performance gap between GR equipped with these document encodings and MVDR.

Table 6: Performance of GR models with different document encoding methods on NQ320k in end-to-end setting. The results of DSI-PAWA, DSI, and T5-GENRE-NP are from [16, 36, 39]. T5-GENRE is the T5 variant of GENRE [2] used by [16].

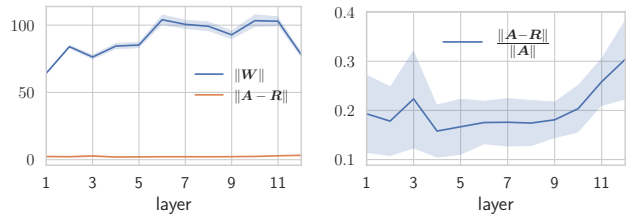
Model	R@1	R@10
T5-ColBERT	61.1	88.4
DSI [38]	55.2	67.4
DSI-PAWA [39]	60.2	80.2
T5-SEAL	44.7	75.5
T5-GENRE [2]	53.7	64.7
T5-GENRE-NP [16]	62.2	78.8

7.4 Low-rank nature of alignment matrix

In Section 5.2.4, we show that the alignment matrix in GR also has a low-rank property in Lemma 5.1. As MVDR using alignment matrix (23) already contains a low-rank component, we only conduct experiments to verify GR. Since the γ in Lemma 5.1 is hard to attain, we illustrate the relation between $\|\boldsymbol{W}\|$ and $\|\boldsymbol{A}-\boldsymbol{R}\|$ in Figure 4(a). We can see that the inequality is loose and $\|\boldsymbol{A}-\boldsymbol{R}\|$ is much lower than $\|\boldsymbol{W}\|$. We also show the relative error of the approximation of $\boldsymbol{R}$ in Figure 4(b). The error is relatively low on average, which indicates the low-rank nature of the alignment matrix of GR.

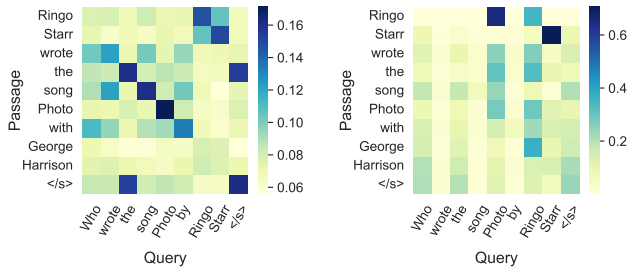
7.5 Case study of the alignment matrix

We chose a specific case from the dataset NQ320k to show what the alignment matrix looks like in Figure 5. Since the document is too long for demonstration, we simplify and extract a sub-sentence containing the answer to the query. In Figure 5(a), we have observed a pronounced phenomenon of exact matches in MVDR. The song name and people's names are completely matched with high scores.



(a) Relation of $\|W\|$ and $\|A-R\|$ (b) Relative error rate of R

Figure 4: Low-rank approximation of R to alignment matrix A in GR in MS MARCO dataset.



(a) MVDR document-to-query direction (b) GR query-to-document direction

Figure 5: An example of alignment matrix in MVDR and GR.

In Figure 5(b), this phenomenon is less obvious, but each document token has more attention on the people's name and song name.

7.6 Upshot

We verify the existence of "exact term match," a specific alignment scenario, in both paradigms. We also show the superiority of the alignment direction in MVDR. The improved document encoding and the low-rank nature of the alignment matrix are validated.

8 LIMITATIONS

We have examined the training target of GR and have connected it with MVDR, but we have not discussed whether relevance computing can be generalized to the generative style inference. We have not considered the multi-layer interactions in the cross-attention between query and document for simplicity.

Our framework does not discuss how query-generation augmentation reduces the discrepancy between training and inference [50]. We aim to study how different architectures and identifier designs will affect the alignment and generalization during inference in future work.

9 CONCLUSION

In this paper, we have offered new insights into GR from the perspective of MVDR that both paradigms share the same frameworks for measuring the relevance between a query and a document. Both paradigms compute relevance as a sum of products of query and document vectors and an alignment matrix. We have explored how GR applies this framework and differs from MVDR. We have shown that GR has simpler document encoding and an alignment strategy with different sparsity and direction. They also share a low-rank property and can be decomposed into query and document components. We have conducted extensive experiments to verify our conclusions and found that both methods have commonalities of term matching in the alignment. We also found that query-to-document alignment direction has better performance than document-to-query.

Based on our findings, practitioners in the field may consider leveraging the shared frameworks highlighted in this study to understand and develop new GR methods, and pay more attention to the classic term matching problem underlying GR models.

As to future work, we will continue to study how multi-layer attention may affect the framework. The difference in the generalization properties for new documents between DR and GR [4, 26, 28, 49] base on our framework is also an important aspect deserving further investigation. We will continue to discover new relations in the GR paradigm and provide more insights into the methodology.

ACKNOWLEDGMENTS

This research was (partially) funded by the Natural Science Foundation of China (62102234, 62372275, 62272274, 62202271, T2293773, 62072279), the National Key R&D Program of China with grant No.2022YFC3303004, the Natural Science Foundation of Shandong Province (ZR2021QF129), the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl, project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO), project ROBUST with project number KICH3.LTP.-20.006, which is (partly) financed by the Dutch Research Council (NWO), DPG Media, RTL, and the Dutch Ministry of Economic Affairs and Climate Policy (EZK) under the program LTP KIC 2020-2023, and the FINDHR (Fairness and Intersectional Non-Discrimination in Human Recommendation) project that received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101070212.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

A FURTHER EXPLANATION OF THE RELEVANCE SCORE IN GR

Further insights into the reason for using $\sum_{i\in[M]} \boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i$ as the relevance can be elaborated as follows. Suppose we treat all other token embeddings $\boldsymbol{e}_k$, where $k \neq d_i$, as fixed with respect to $\boldsymbol{e}_{d_i}$, then at the early stage of the training, the loss can be expressed as:

$$\mathcal{L}_i(d,q) = -\boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i + \log\left(\exp \boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i + \sum_{v\neq d_i} \exp \boldsymbol{e}_v^\top \boldsymbol{h}_i\right) \tag{26}$$

$$= -\boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i + \log\left(\exp \boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i + C\right) \propto -\boldsymbol{e}_{d_i}^\top \boldsymbol{h}_i. \tag{27}$$

Due to space limitations, details of Appendix A, B, C, and D can be found online in [41].

REFERENCES

[1] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive Search Engines: Generating Substrings as Document Identifiers. *Advances in Neural Information Processing Systems* 35 (2022), 31668–31683.

[2] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive Entity Retrieval. arXiv:2010.00904 [cs.CL]

[3] Guanhua Chen, Yun Chen, and Victor O. K. Li. 2021. Lexically Constrained Neural Machine Translation with Explicit Alignment Guidance. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 12630–12638. https://doi.org/10.1609/AAAI.V35I14.17496

[4] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Continual Learning for Generative Retrieval over Dynamic Corpora. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 306–315. https://doi.org/10.1145/3583780.3614821

[5] Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate Word Alignment Induction from Neural Machine Translation. In *EMNLP (1)*. Association for Computational Linguistics, 566–576.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[7] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A Mathematical Framework for Transformer Circuits. Transformer Circuits Thread. https://transformer-circuits.pub/2021/framework/index.html.

[8] Zhen Fan, Luyu Gao, Rohan Jha, and Jamie Callan. 2023. COILcr: Efficient Semantic Matching Contextualized Exact Match Retrieval. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I* (Dublin, Ireland). Springer-Verlag, Berlin, Heidelberg, 298–312. https://doi.org/10.1007/978-3-031-28244-7_19

[9] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *SIGIR*. ACM, 2288–2292.

[10] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. A White Box Analysis of ColBERT. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12657)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 257–263. https://doi.org/10.1007/978-3-030-72240-1_23

[11] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. In *NAACL-HLT*. Association for Computational Linguistics, 3030–3042.

[12] Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, and Lingjun Zhao. 2020. Cross-lingual Information Retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, Kathy McKeown, Douglas W. Oard, Elizabeth, and Richard Schwartz (Eds.). European Language Resources Association, Marseille, France, 26–31. https://aclanthology.org/2020.clssts-1.5

[13] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP*. Association for Computational Linguistics, 6769–6781.

[14] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *SIGIR*. ACM, 39–48.

[15] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Trans. Assoc. Comput. Linguistics* 7 (2019), 452–466.

[16] Hyunji Lee, JaeYoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vladimir Karpukhin, Yi Lu, and Minjoon Seo. 2023. Nonparametric Decoding for Generative Retrieval. In *ACL (Findings)*. Association for Computational Linguistics, 12642–12661.

[17] Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Y. Zhao. 2023. Rethinking the Role of Token Retrieval in Multi-Vector Retrieval. *CoRR* abs/2304.01982 (2023). https://doi.org/10.48550/ARXIV.2304.01982 arXiv:2304.01982

[18] Sunkyung Lee, Minjin Choi, and Jongwuk Lee. 2023. GLEN: Generative Retrieval via Lexical Index Learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 7693–7704. https://aclanthology.org/2023.emnlp-main.477

[19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7871–7880. https://doi.org/10.18653/V1/2020.ACL-MAIN.703

[20] Bryan Li. 2022. Word Alignment in the Era of Deep Learning: A Tutorial. *CoRR* abs/2212.00138 (2022). https://doi.org/10.48550/ARXIV.2212.00138 arXiv:2212.00138

[21] Lei Li, Kai Fan, Hongjia Li, and Chun Yuan. 2022. Structural Supervision for Word Alignment and Machine Translation. In *ACL (Findings)*. Association for Computational Linguistics, 4084–4094.

[22] Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. CITADEL: Conditional Token Interaction via Dynamic Lexical Routing for Efficient and Effective Multi-Vector Retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 11891–11907. https://doi.org/10.18653/V1/2023.ACL-LONG.663

[23] Minghan Li, Sheng-Chieh Lin, Xueguang Ma, and Jimmy Lin. 2023. SLIM: Sparsified Late Interaction for Multi-Vector Retrieval with Inverted Indexes. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1954–1959. https://doi.org/10.1145/3539618.3591977

[24] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Learning to Rank in Generative Retrieval. *CoRR* abs/2306.15222 (2023). https://doi.org/10.48550/ARXIV.2306.15222 arXiv:2306.15222

[25] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Trans. Assoc. Comput. Linguistics* 9 (2021), 329–345.

[26] Sanket Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2023. DSI++: Updating Transformer Memory with New Documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 8198–8213. https://doi.org/10.18653/v1/2023.emnlp-main.510

[27] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*, Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[28] Thong Nguyen and Andrew Yates. 2023. Generative Retrieval as Dense Retrieval. *CoRR* abs/2306.11397 (2023). https://doi.org/10.48550/ARXIV.2306.11397 arXiv:2306.11397

[29] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large Dual Encoders Are Generalizable Retrievers. In *EMNLP*. Association for Computational Linguistics, 9844–9855.

[30] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context Learning and Induction Heads. Transformer Circuits Thread. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[31] Ronak Pradeep, Kai Hui, Jai Gupta, Ádám D. Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q. Tran. 2023. How Does Generative Retrieval Scale to Millions of Passages?. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 1305–1321. https://aclanthology.org/2023.emnlp-main.83

[32] Yujie Qian, Jinhyuk Lee, Sai Meher Karthik Duddu, Zhuyun Dai, Siddhartha Brahma, Iftekhar Naim, Tao Lei, and Vincent Y. Zhao. 2022. Multi-Vector Retrieval as Sparse Alignment. arXiv:2211.01267 (November 2022). http://arxiv.org/abs/2211.01267

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[34] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. *CoRR* abs/2305.05065 (2023). https://doi.org/10.48550/ARXIV.2305.05065 arXiv:2305.05065

[35] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[36] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. 2023. Learning to Tokenize for Generative Retrieval. In *NeurIPS 2023: Thirty-seventh Conference on Neural Information Processing Systems*. https://papers.nips.cc/paper_files/paper/2023/file/91228b942a4528cdae031c1b68b127e8-Paper-Conference.pdf

[37] Yubao Tang, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. 2023. Recent Advances in Generative Information Retrieval. In *SIGIR-AP*. ACM, 294–297.

[38] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer Memory as a Differentiable Search Index. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/892840a6123b5ec99ebaab8be1530fba-Abstract-Conference.html

[39] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2022. A Neural Corpus Indexer for Document Retrieval. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/a46156bd3579c3b268108ea6aca71d13-Abstract-Conference.html

[40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP (Demos)*. Association for Computational Linguistics, 38–45.

[41] Shiguang Wu, Wenda Wei, Mengqi Zhang, Zhumin Chen, Jun Ma, Zhaochun Ren, Maarten de Rijke, and Pengjie Ren. 2024. Generative Retrieval as Multi-Vector Dense Retrieval. arXiv:2404.00684 [cs.IR]

[42] Tianchi Yang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. 2023. Auto Search Indexer for End-to-End Document Retrieval. In *EMNLP (Findings)*. Association for Computational Linguistics, 6955–6970.

[43] Puxuan Yu, Hongliang Fei, and Ping Li. 2021. Cross-lingual Language Model Pretraining for Retrieval. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 1029–1039. https://doi.org/10.1145/3442381.3449830

[44] Peiwen Yuan, Xinglin Wang, Shaoxiong Feng, Boyuan Pan, Yiwei Li, Heda Wang, Xupeng Miao, and Kan Li. 2024. Generative Dense Retrieval: Memory Can Be a Burden. arXiv:2401.10487 [cs.IR]

[45] Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2023. Scalable and Effective Generative Information Retrieval. *CoRR* abs/2311.09134 (2023). https://doi.org/10.48550/ARXIV.2311.09134 arXiv:2311.09134

[46] Fuwei Zhang, Zhao Zhang, Xiang Ao, Dehong Gao, Fuzhen Zhuang, Yi Wei, and Qing He. 2022. Mind the Gap: Cross-Lingual Information Retrieval with Hierarchical Knowledge Enhancement. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 4345–4353. https://doi.org/10.1609/AAAI.V36I4.20355

[47] Le Zhao. 2012. Modeling and Solving Term Mismatch for Full-text Retrieval. *SIGIR Forum* 46, 2 (2012), 117–118.

[48] Giulio Zhou and Jacob Devlin. 2021. Multi-Vector Attention Models for Deep Re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 5452–5456. https://doi.org/10.18653/v1/2021.emnlp-main.443

[49] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2023. DynamicRetriever: A Pre-trained Model-based IR System Without an Explicit Index. *Mach. Intell. Res.* 20, 2 (April 2023), 276–288. https://doi.org/10.1007/s11633-022-1373-9

[50] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the Gap Between Indexing and Retrieval for Differentiable Search Index with Query Generation. *ArXiv* abs/2206.10128 (2022). https://api.semanticscholar.org/CorpusID:249890267