# ZEROGR: A GENERALIZABLE AND SCALABLE FRAMEWORK FOR ZERO-SHOT GENERATIVE RETRIEVAL

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Generative retrieval (GR) reformulates information retrieval (IR) by framing it as the generation of document identifiers (docids), thereby enabling an end-to-end optimization and seamless integration with generative language models (LMs). Despite notable progress under supervised training, GR still struggles to generalize to zero-shot IR scenarios, which are prevalent in real-world applications. To tackle this challenge, we propose ZEROGR, a zero-shot generative retrieval framework that leverages natural language instructions to extend GR across a wide range of IR tasks. Specifically, ZEROGR is composed of three key components: (i) an LM-based docid generator that unifies heterogeneous documents (e.g., text, tables, code) into semantically meaningful docids; (ii) an instruction-tuned query generator that generates diverse types of queries from natural language task descriptions to enhance corpus indexing; and (iii) a reverse annealing decoding strategy to balance precision and recall during docid generation. We investigate the impact of instruction fine-tuning scale and find that performance consistently improves as the number of IR tasks encountered during training increases. Empirical results on the BEIR and MAIR benchmarks demonstrate that ZEROGR achieves strong performance across diverse retrieval tasks, for example establishing a new state of the art among generative retrieval methods.

## 1 INTRODUCTION

Dense retrieval (DR) (Karpukhin et al., 2020; Izacard et al., 2021), which encodes documents and queries as embedding vectors, is arguably the most effective and widely adopted paradigm (Thakur et al., 2021; Muennighoff et al., 2022) in information retrieval (IR). Despite its success, DR's expressivity is fundamentally limited by the embedding dimensionality (Cao et al., 2020) and does not fully leverage the capabilities of generative language models (LMs) (Tay et al., 2022). As an alternative, generative retrieval (GR) (Metzler et al., 2021) introduces a paradigm shift that encodes corpus information into the model parameters, enabling document retrieval by generating (relevant) document identifiers (docids). GR has demonstrated competitive performance on various IR tasks when large-scale supervised data is available (Tay et al., 2022; Sun et al., 2023b; Chen et al., 2022), spanning both traditional web search (Campos et al., 2016) and knowledge-intensive retrieval applications (Petroni et al., 2020).

Despite its promising performance on in-domain tasks, GR exhibits limited generalization to out-of-distribution IR tasks. Existing GR models are typically trained on specific corpora and queries, and prior studies has shown that such training leads to poor performance on unseen tasks (Zhang et al., 2025b; Liu et al., 2023b). In contrast, real-world IR models are typically evaluated in a broader setting, characterized by substantial diversity and heterogeneity. These often involve heterogeneous corpora and queries (Thakur et al., 2021), task-specific relevance criteria (Su et al., 2022; Asai et al., 2022), and predominantly zero-shot scenarios where no supervised data is available (Thakur et al., 2021; Muennighoff et al., 2022). Consequently, GR approaches designed for supervised conditions struggle to generalize to such heterogeneous and data-scarce retrieval scenarios.

To address the limitations of GR in zero-shot and heterogeneous IR scenarios, we draw inspiration from recent advancements in instructed DR methods (Su et al., 2022; Asai et al., 2022) and propose ZEROGR, a generalizable framework for **ZERO**-shot **G**enerative information **R**etrieval. ZEROGR is a simple yet effective way to adapt GR to diverse IR tasks in a zero-shot setting by leveraging natural

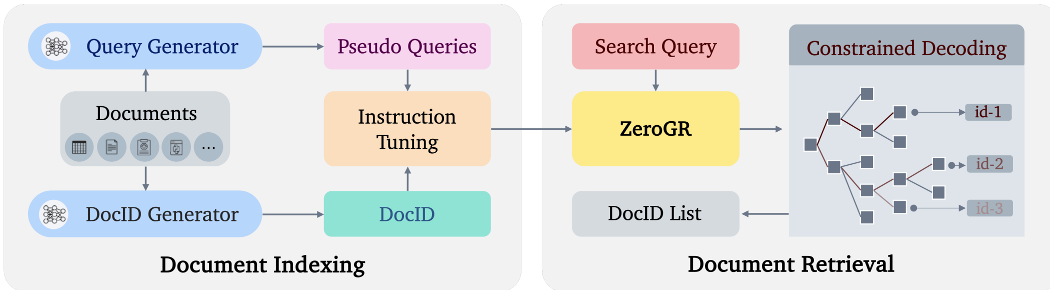


Figure 1: **An overview of ZEROGR.** Given a document collection, ZEROGR converts them into unified DocID representations, generates diverse pseudo-queries, and builds a generative retrieval index. During online retrieval, ZEROGR decodes docids with reverse-annealed temperature scheduling to balance precision and recall.

language task instructions. Specifically, we advance GR along three dimensions: (i) for *docid design*, we propose a docid generator to efficiently convert a document of any format (e.g., paragraph, table, code) into a unified text-based docid representation; (ii) for *corpus indexing*, we propose an instructed query generator to generate diverse types of queries based on different task instructions; (iii) for *docid decoding*, we propose a reverse annealing strategy that more effectively trades off precision and recall of docid decoding than prior work.

Building on ZEROGR, we investigate *instruction fine-tuning scaling* (Chung et al., 2022) in the context of GR along two key axes: the size of instruction tuning data and the size of the underlying model. We find that increasing both the diversity and quantity of training tasks yields substantial improvements in zero-shot retrieval performance on unseen tasks. Beyond training data scaling, we also examine model size scaling and inference-time scaling for corpus indexing, observing consistently promising scaling trends in both cases.

Our best-performing model, based on the Llama-3B LM, outperforms previous generative retrieval methods and narrows the gap to state-of-the-art dense retrieval systems across heterogeneous IR benchmarks, including BEIR (Thakur et al., 2021) and MAIR (Sun et al., 2024). Notably, ZEROGR outperforms OpenAI Embed-v3 on zero-shot MAIR tasks, highlighting its strong generalization to unseen retrieval tasks.

In summary, our contributions are as follows: (i) We propose ZEROGR, a zero-shot GR framework that can construct task-specific GR search indices based on natural language instructions. (ii) Within ZEROGR, we enhance GR by introducing three key components: a unified text-based docid generator, an instruction-conditioned pseudo-query generator, and a reverse annealing decoding strategy. And (iii) ZEROGR achieves competitive performance on heterogeneous IR benchmarks, establishing it as the first GR approach capable of generalizing to diverse tasks in a zero-shot setting.

## 2 RELATED WORK

**Document Retrieval** Document retrieval is a fundamental task in information retrieval, with broad applications in search engines and retrieval-augmented generation systems (Karpukhin et al., 2020; Lin et al., 2020; Chen et al., 2025). It typically follows a two-stage pipeline: an initial retrieval stage that recalls candidate documents, followed by a reranking stage for fine-grained ranking. Traditional sparse retrieval methods (Robertson and Walker, 1997; Lafferty and Zhai, 2001; Robertson and Zaragoza, 2009) rely on lexical overlap but suffer from vocabulary mismatch (Lin et al., 2020). Dense retrieval (DR) addresses this issue by embedding queries and documents into dense vectors and comparing them via inner product or cosine similarity (Karpukhin et al., 2020), with subsequent improvements from hard negative mining, late interaction, and pre-training (Xiong et al., 2020; Khattab and Zaharia, 2020; Wang et al., 2022a; Qu et al., 2021; Izacard et al., 2021). The reranking stage is usually performed using cross-encoders or LLM prompting (Nogueira and Cho, 2019; Nogueira et al., 2020; Sun et al., 2023c; Chen et al., 2024; Sun et al., 2023a; Zhang et al., 2025a; Liu et al., 2025; Ma et al., 2023). However, this two-stage pipeline is difficult to optimize end-to-end due to its MIPS-based retrieval component and the objective mismatch with generative language model training (Tay et al., 2022; Bevilacqua et al., 2022).

**Generative Retrieval** Unlike traditional dense retrieval methods (Karpukhin et al., 2020; Xiong et al., 2020), GR formulates information retrieval as a docid generation task, enabling end-to-end optimization of the inference-time search index (Tay et al., 2022; Metzler et al., 2021). Previous research on GR has largely focused on three key aspects: (i) *Docid design*: Early approaches employed rule-based formats such as titles (Cao et al., 2020; Chen et al., 2022), URLs (Zhou et al., 2022), or text spans/summaries (Bevilacqua et al., 2022; Li et al., 2023a). More recent work has shifted toward learning-based docid designs that capture corpus semantics more effectively, including embedding clustering (Tay et al., 2022) and RQ-VAE–based approaches (Wang et al., 2024; Zeng et al., 2023; Wang et al., 2023b). (ii) *Corpus indexing*: Several strategies have been explored to enrich corpus representations, such as document chunking (Tay et al., 2022), pseudo-query generation (Zhuang et al., 2022), rehearsal-based augmentation (Tang et al., 2023), multi-granular indexing (Wen et al., 2025), and continual training for dynamic corpora (Mehta et al., 2022; Chen et al., 2023; Zhang et al., 2025b). (iii) *Docid decoding*: The dominant approach has been constrained beam search (Cao et al., 2020; Tay et al., 2022). More advanced strategies include multi-stage decoding (Ren et al., 2023), multi-docid decoding (Li et al., 2023b), and simultaneous decoding (Zeng et al., 2024). Despite steady progress, existing work primarily remains confined to supervised fine-tuning, relying heavily on training data and failing to generalize to zero-shot retrieval tasks.

**Instruction Fine-tuning in IR** Inspired by the studies in LLM instruction tuning (Chung et al., 2022; Wang et al., 2022b), instruction fine-tuning for retrieval has gained increased attention to improve zero-shot IR performance (Su et al., 2022; Asai et al., 2022). Instruction-tuned models are able to adapt to various tasks based on natural language instructions that specify the relevance criteria. Recent studies in this direction include multi-task fine-tuning (Lee et al., 2024a), LLM-generated instruction data (Wang et al., 2023a; Lee et al., 2024b; Oh et al., 2024), and instruction-negatives (Weller et al., 2024). These efforts have primarily focused on dense retrieval or cross-encoder rerankers (Sun et al., 2024). To the best of our knowledge, we are the first to investigate instruction fine-tuning for GR and to conduct a systematic study of the factors that influence instruction fine-tuning in IR models.

## 3 PRELIMINARIES

**Zero-shot document retrieval.** We formulate the task of zero-shot document retrieval as follows. Given a corpus $\mathcal{D} = (d_1, \ldots, d_n)$ containing $n$ documents, a *corpus indexing* function $\mathcal{I}$ takes $\mathcal{D}$ as input and constructs a search index $m = \mathcal{I}(\mathcal{D})$. Then, a *retrieval* function $\mathcal{F}$ takes the index $m$ and a query $q$ as input, and returns a list of relevant documents: $(d_i, \ldots) = \mathcal{F}(m, q)$. Note that in a typical zero-shot document retrieval setting, no training data is available. However, a natural language task instruction $instr_t$ specifying the retrieval task is generally assumed to be available, as it is usually easier to obtain (Muennighoff et al., 2022).

**Generative retrieval.** GR aims to retrieve the document $d_i$ by generating the corresponding document identifier (docid) given the query $q$. To this end, GR assigns an identifier (docid) to each document in the corpus, e.g. $(z_1, \ldots, z_n)$, where each $z_i$ is a sequence of tokens $z_i = \{z_i^{(1)}, \ldots, z_i^{(T)}\}$ with a maximum length of $T$. Based on this, the indexing function $\mathcal{I}(\mathcal{D})$ of GR is to train a language model (LM) $\mathcal{M}$ on the corpus $\mathcal{D}$, encoding the corpus information and also document-docid mapping. The retrieval function $F$ is instantiated by the same $\mathcal{M}$, and it generates the relevant document identifiers (docids) $(z_1, \ldots, z_n)$ given the query $q$: $(z_i, \ldots) = \mathcal{M}(q)$.

## 4 ZEROGR

We propose ZEROGR, a zero-shot GR framework that can adapt LMs into task-specific generative search indexes based on task instructions. As shown in Figure 1, the proposed ZEROGR framework consists of three key components: (i) a docid generator $G_\psi$, which takes a document $d_i$ as input and outputs its docid $z_i$; (ii) an instructed query generator, which takes a task instruction *instr* and a document $d_i$ as input and outputs multiple pseudo-queries; (iii) a generative retriever $\mathcal{M}$, which takes the instruction and a query as input and generates a list of docids.

The ZeroGR pipeline proceeds as follows: (i) given a new corpus $\mathcal{D}$ and its associated task instruction *instr*, the docid generator assigns each document $d_i$ a docid $z_i$; (ii) the instructed query generator $G_\theta$ samples $B$ queries $\{q_{i,1}, \ldots, q_{i,B}\}$ for each document $d_i \in \mathcal{D}$, thereby creating $\langle q_{i,j}, z_i \rangle$ pairs; and (iii) the generative retriever is trained to predict the corresponding docid $z_i$ given the concatenation of *instr* and a sampled query $q_{i,j}$. After training, the generative retriever $\mathcal{M}(z \mid q, \textit{instr})$ serves as the search index $m$. For a given query $q$, a newly proposed reverse annealing decoding strategy is employed to generate a ranked list of docids as retrieval results.

### 4.1 Unified Docid Representation

Documents in downstream IR tasks can be heterogeneous, e.g., financial tables (Zhu et al., 2022), code files (Liu et al., 2023a), meeting transcripts (Golany et al., 2024), or legal cases (Bhattacharya et al., 2019). Existing simple docid strategies, such as using document titles, URLs, or spans (Cao et al., 2020; Bevilacqua et al., 2022), often fail to generalize to user-customized data. ZeroGR therefore introduces a model-based **docid generator** $G_\psi$ that maps any document to a short, keyword-rich sentence (typically 6–8 words) ranked by coverage. Formally, for a document $d_i$ we define

$$z_i \;=\; G_\psi(d_i) \;=\; \arg\max_{t \in \mathcal{V}^{\leq L}} G_\psi\big(t \mid d_i\big), \tag{1}$$

where $t$ is a token sequence of length $\leq L$ (with $L = 8$) drawn from the vocabulary $\mathcal{V}$. To instantiate $G_\psi$, we first prompt a powerful LM (e.g., GPT-4o) to create a training set of $\langle d_i,\, z_i \rangle$ pairs (see Appendix A for the detailed prompt used). A smaller model (Llama-3.2-1B) is then fine-tuned on this data, enabling fast, scalable generation of unified docids across diverse IR tasks. See Section 5.1 for details of training data.

### 4.2 Instructed Corpus Indexing

Corpus indexing in GR encodes each document $d_i \in \mathcal{D}$ into the model's parameters so that, at inference time, the model can recover $d_i$ by *generating* its document identifier $z_i$. DSI-QG (Zhuang et al., 2022) accomplishes this by pairing every document with a set of pseudo-queries, but its effectiveness diminishes when the pseudo-query distribution diverges from real user queries (Pradeep et al., 2023; Dai et al., 2022). This gap is especially large in heterogeneous IR scenarios, such as conversational, code, or multimodal search.

We mitigate the distribution gap with an **instructed query generator** $G_\theta$, obtained by instruction-tuning a 1B-parameter Llama model on diverse IR datasets verbalized through task-specific instructions. Given a document $d_i$ and a task instruction *instr*, the generator produces a pseudo-query $q_{i,j}$ from the conditional distribution

$$q_{i,j} \;\sim\; G_\theta\big(\cdot \mid d, \textit{instr}\big). \tag{2}$$

For each document we draw $B$ queries with temperature of 1:

$$\mathcal{Q}_i = \{\, q_{i,1}, \ldots, q_{i,B} \,\}. \tag{3}$$

These $\langle d_i,\, z_i \rangle$ pairs are used to train the generative retriever $\mathcal{LLM}$ by minimizing the cross-entropy loss

$$\mathcal{L}(\phi) = -\sum_{d_i \in \mathcal{D}} \sum_{q_{i,j} \in \mathcal{Q}_i} \log \mathcal{M}\big(z_i \mid q_{i,j}, \textit{instr}\big), \tag{4}$$

thereby embedding the corpus into the model's parameters. Appendix D summarizes the instruction-tuning datasets.

### 4.3 Reverse-Annealed Docid Generation

During inference, a GR model must decode each docid $z_i$ as a *sequence of tokens*. Standard beam search often collapses to a few high-probability sequences, hurting recall. We therefore propose **reverse-annealed sampling**: each $z_i$ is generated token-by-token, while the sampling temperature is gradually *increased* to encourage diversity. Let $f(\cdot)$ denote the trained decoder after corpus indexing, and let $T$ be a prefix tree whose leaves correspond to valid docids. For the $i$-th docid we decode a token sequence $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,L_i})$ using temperature $t_i = g(i)$. At position $j$ we sample

$x_{i,j} \sim \text{Softmax}\Big(\frac{\boldsymbol{\ell}_{i,j}}{t_i}\Big)\Big|_{T_{i,j}}$, where $\boldsymbol{\ell}_{i,j}$ are the logits conditioned on the current prefix $(x_{i,1:j-1})$, and the subscript $T_{i,j}$ masks probabilities to tokens that keep the prefix inside the tree. After the complete sequence $\mathbf{x}_i$ is produced, its leaf is removed from $T$ so no subsequent iteration can repeat the same docid. The per-iteration temperature $t_i$ follows a *normalized sigmoid*:

$$t_i = g(i) = T_{\max} \cdot \frac{\sigma\big(k(\frac{i}{K} - m)\big) - \sigma(-km)}{\sigma\big(k(1-m)\big) - \sigma(-km)}, \quad \sigma(z) = \frac{1}{1+e^{-z}}, \tag{5}$$

where $K$ is the total number of docids to generate, $k > 0$ controls the slope, and $m \in (0, 1)$ sets the midpoint. Starting from a low temperature yields high-precision early selections; increasing $t_i$ over iterations boosts exploration, thereby balancing precision and recall across the final ranked list. See Alg. 1 and Figure 10 for algorithm detail.

# 5 Evaluation Setup

## 5.1 Training Datasets

To support the development of ZeroGR, we collect training data covering a diverse range of IR tasks. Specifically, we use MAIR (Sun et al., 2024), a multi-task IR evaluation benchmark comprising 126 tasks, and extract the training splits of these tasks when available. As shown in Table 1 (and Figure 7 in Appendix D for data example), **ZeroGR-Train** is a dataset spanning 69 IR tasks across 6 domains, containing 41 million query-document pairs. ZeroGR-Train is the largest open-source IR training corpus to date. It offers greater domain and task diversity, includes detailed instructional annotations, and provides reliable relevance labels. See Table 7 for details.

| Domain | #Tasks | #Samples |
|---|---|---|
| Medical | 5 | 421, 430 |
| Financial | 8 | 31, 315 |
| Academic | 18 | 744, 160 |
| Coding | 13 | 1, 969, 586 |
| Legal | 7 | 23, 086, 948 |
| Web-based | 18 | 15, 319, 445 |

Table 1: Statistics of ZeroGR-Train

## 5.2 Evaluation Datasets

To evaluate zero-shot GR on diverse downstream tasks, we use the BEIR and MAIR benchmarks: (i) **BEIR** (Thakur et al., 2021). We evaluate models on all 12 tasks from BEIR collections. (ii) **MAIR** (Sun et al., 2024). As we collect training data from a subset of MAIR tasks, we divide MAIR into seen and unseen subsets, where the unseen subset contains tasks not present in the ZeroGR-Training data, to validate the zero-shot generalization of models. In constructing this benchmark, we curated a diverse set of long-tail tasks across 6 domains, and intentionally omitted redundant tasks (e.g., different years of the same competition) and structurally complex ones (e.g., IFEval) that would introduce evaluation overhead. Given the large size of the MAIR dataset, we also develop a Dev subset of MAIR for model ablation. Note that our current evaluation focus on tasks with moderately sized corpus.

## 5.3 Evaluation Metric

We evaluate models using the following metrics: (i) *Top-1 accuracy*, which measures retrieval precision by checking whether the top-ranked document is relevant to the query; (ii) *nDCG@10*, a popular metric that evaluates the quality of the top-10 ranked results by considering both the relevance and position of retrieved documents; and (iii) *Recall@100*, which assesses recall by calculating the percentage of relevant documents retrieved within the top-100 ranked list.

## 5.4 Implementation details

We implement the three components of ZeroGR, i.e., query generator, docid generator, and final generative retriever, all with Llama-based LMs. For the docid generator, a Llama-1B-Instruct model is trained on our curated document-docid pairs for 5 epochs with a constant learning rate of 5e-5. Similarly, for the query generator, a Llama-1B-Instruct model is trained on the ZeroGR-Training set

| Model | MAIR (38 Tasks) | | | | | | | BEIR (11 Tasks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg | Web. | Aca. | Legal | Med. | Fin. | Cod. | Avg | Web. | Aca. | Med. | Fin. |
| BM25 | 36.1 | 34.3 | 39.2 | 34.5 | 42.4 | 40.0 | 17.3 | 42.4 | 45.4 | 38.8 | 32.7 | 41.6 |
| Contriever | 33.6 | 39.8 | 33.4 | 26.8 | 30.8 | 37.3 | 17.7 | 47.6 | 51.5 | 43.0 | 33.9 | 47.6 |
| GTR-T5-base | 32.5 | 36.0 | 33.6 | 25.3 | 31.9 | 37.4 | 18.7 | 45.3 | 50.7 | 35.3 | 32.7 | 45.3 |
| GTR-T5-large | 35.4 | 39.8 | 39.6 | 27.8 | 31.8 | 38.5 | 24.0 | 48.0 | 53.3 | 37.4 | 33.4 | 50.0 |
| E5-Base | 37.2 | 36.2 | 48.6 | 28.5 | 35.3 | 44.9 | 26.7 | 48.9 | 51.8 | 46.1 | 35.0 | 50.2 |
| E5-Large | 38.2 | 38.6 | 51.0 | 25.0 | 35.6 | 46.6 | 25.7 | 49.2 | 51.7 | 47.9 | 37.4 | 48.8 |
| BGE-Base | 37.0 | 38.6 | 40.2 | 25.8 | 37.6 | 42.2 | 29.0 | 50.5 | 52.5 | 47.1 | 36.0 | 55.2 |
| BGE-Large | 39.4 | 39.4 | 46.2 | 36.0 | 37.2 | 45.1 | 29.0 | 51.8 | 53.8 | 47.9 | 38.1 | 56.5 |
| OpenAI-Embed | 40.6 | 40.6 | 48.2 | 31.0 | 39.7 | 49.4 | 28.7 | 54.2 | 56.3 | 47.2 | 37.6 | 63.4 |
| E5-mistral-7B | 46.8 | 45.4 | 55.4 | 42.3 | 43.1 | 55.3 | 40.0 | 55.7 | 56.4 | 48.6 | 39.6 | 68.8 |
| GritLM-7B | 47.0 | 44.1 | 58.2 | 43.3 | 42.6 | 57.6 | 40.0 | 45.0 | 47.7 | 48.2 | 36.9 | 37.8 |
| ZeroGR-3B | 41.1 | 42.7 | 47.4 | 40.0 | 38.3 | 39.2 | 36.3 | 48.1 | 49.2 | 45.8 | 34.7 | 53.8 |

Table 2: **Combined Domain-wise Results on MAIR (Acc@1) and BEIR (nDCG@10).** Performance of different retrieval models across various domains. See Tab 4 and Tab 5 for details.

for 5 epochs with a constant learning rate of 5e-5. For the generative retriever, the model is trained for each evaluated task on data generated by the query generator and docid generator, based on our "Document Indexing" workflow described in Fig 1.

### 5.5 Baselines

We evaluate ZeroGR against several representative IR baselines, spanning different retrieval paradigms to provide a comprehensive comparison. (i) For sparse retrieval, we adopt the classical term-based model **BM25**, implemented using the BM25S package (Lù, 2024), which remains a strong baseline in many IR tasks due to its simplicity and effectiveness. (ii) For traditional dense retrieval models trained on a single task, we include **Contriever-MARCO**, **GTR-base**, and **GTR-Large**, all of which are pretrained or fine-tuned on the MS MARCO dataset (Ni et al., 2021; Izacard et al., 2021), representing a common practice in dense retrieval pipelines. (iii) For multi-task-trained dense retrievers, we incorporate **E5-Base** and **E5-Large** (Wang et al., 2022a), **BGE-base** and **BGE-Large** (Xiao et al., 2023), as well as **OpenAI-Embedding-v3-Small**, all of which use supervision from multiple tasks to enhance generalization across diverse domains. (iv) For instruction-tuned dense retrieval models, which aim to align the retriever with human instructions, we include **E5-Mistral-7B-instruct** (Wang et al., 2023a), and **GritLM-7B** (Muennighoff et al., 2024), which are trained on large-scale, diverse instruction datasets to follow task-specific intents effectively.

## 6 Experiments

Our experiments address the following research questions:

1. **How does ZeroGR compare with dense retrieval methods?**
   We evaluate ZeroGR against leading models on the MAIR benchmark (Section 6.1) and conduct additional analysis on the BEIR datasets (Section 6.2).
2. **How do model design and training strategies influence the performance of ZeroGR?**
   To answer this, we conduct a systematic study on the development set, investigating key factors in generative retrieval. Specifically, we analyze how instruction tuning task diversity (Section 6.3), docid design (Section 6.4), corpus indexing strategy, model size (Section 6.5), and decoding strategy (Section 6.6) affect performance.

### 6.1 Evaluation Results on MAIR

As shown in Table 2 (MAIR), our proposed ZeroGR framework demonstrates strong performance across a wide range of retrieval tasks. It achieves an average score of 41.1 (Acc@1), substantially

| Method | Training Data | Avg | Argu. | SciF. | NFC. | FiQA | SciD. | Covid |
|---|---|---|---|---|---|---|---|---|
| GENRE (Cao et al., 2020) | GPL | 23.0 | **42.5** | 42.3 | 20.0 | 11.6 | 6.8 | 14.7 |
| GENRET (Sun et al., 2023b) | GPL | 41.1 | 34.3 | 63.9 | 31.6 | 30.2 | 14.9 | 71.8 |
| GLEN (Lee et al., 2023) | NQ320k | – | 17.6 | – | 15.9 | – | – | – |
| TIGER (Rajput et al., 2023) | ZeroGR-Train | 31.0 | 14.0 | 37.0 | **39.5** | 16.0 | 14.0 | 65.7 |
| ZeroGR (Ours) | ZeroGR-Train | **44.9** | 35.4 | **72.8** | 34.7 | **34.1** | **18.7** | **73.5** |

Table 3: Performance of different generative retrieval models across various datasets on BEIR.

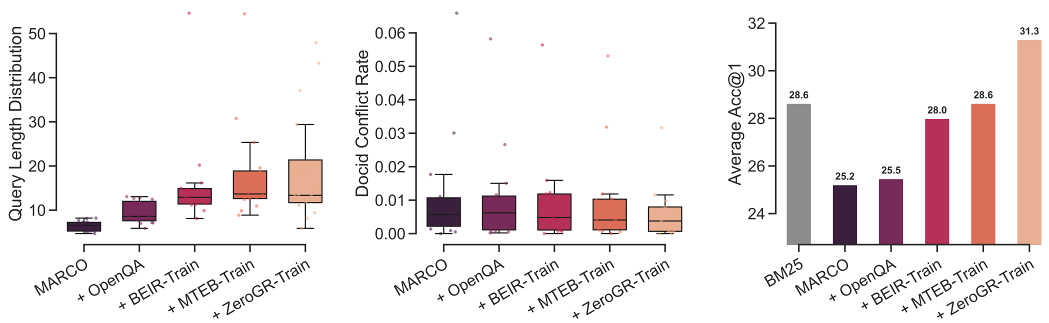


Figure 2: **Model performance on unseen-dev tasks as a function of the number of training tasks.** We increase the number of training tasks, starting from MS MARCO, and incrementally add open-domain QA datasets (e.g., NQ), BEIR-Train sets (e.g., NFC), MTEB-Train data (e.g., NLI), and finally the ZeroGR-Train collection, which includes 60 tasks across 6 domains. **Left:** More instruction-tuning tasks lead to more diverse queries. **Middle:** More instruction-tuning tasks reduce docid conflicts. **Right:** More instruction-tuning tasks improve the Acc@1 score.

outperforming traditional sparse retrieval methods like BM25 and widely adopted dense retrieval models such as Contriever, GTR, E5, BGE, and even the strong instruction-tuned OpenAI-Embedding-v3-Small. These results highlight the effectiveness of our instruction-based generative retrieval approach in capturing deeper semantic relevance.

The performance gains of ZeroGR are not limited to familiar tasks but also generalize well to unseen domains. Notably, the model performs better than all baselines on several previously unseen datasets, including Apple, MB, PM.A, DD, and NCL (see Table 4). This demonstrates the robustness and transferability of the approach, as it adapts effectively to new retrieval settings without requiring additional task-specific supervised data. See Figure 6 for a comparison on the MAIR unseen subset, where ZeroGR achieves competitive performance against recent dense retrieval methods.

Using a 3B LLM, ZeroGR can achieve strong performance across different tasks compared to baselines, though it still underperforms large embedding models such as GritLM-7B and E5-Mistral-7B. This indicates that our design is highly parameter-efficient, achieving strong performance across diverse tasks without relying on massive model scaling. See Appendix C for per-task performance.

### 6.2 Evaluation Results on BEIR

As shown in Table 2 (BEIR), ZeroGR outperforms several baselines such as BM25, Contriever, GTR, and GritLM-7B, but still underperforms other dense retrieval methods. Table 5 compares ZeroGR with previous generative retrieval baselines on BEIR, which we can see our method achieves best performance among most datasets.

### 6.3 Scaling Instruction Fine-tuning

A key factor in enhancing the performance of LM-based tasks is scaling, i.e., increasing model size or data volume. The effectiveness of ZeroGR stems from instruction fine-tuning on multi-task IR datasets, which improves the instruction-following abilities of both the query generator and the title generator models. To investigate the impact of multi-task training, we curate training data with varying numbers of tasks: (a) *MS MARCO*, which contains a single task (i.e., MS MARCO (Campos

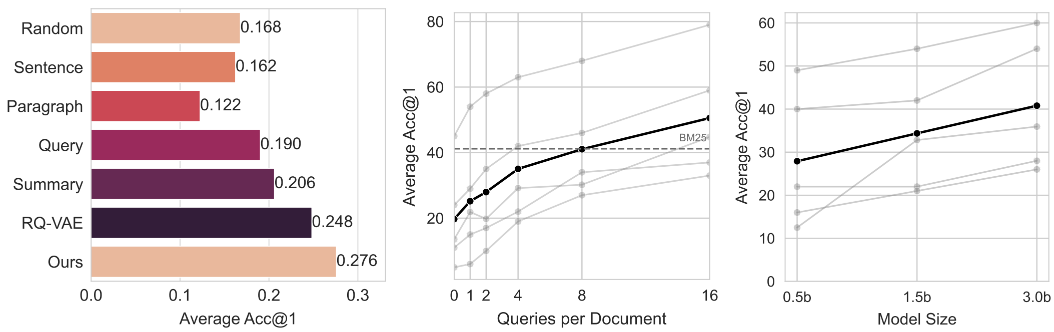


Figure 3: **Left:** Comparison of different docid designs. **Middle:** Acc@1 vs. generated queries per document. **Right:** Acc@1 vs. model size. Gray curves are per-task score.

et al., 2016)) and is commonly used in previous GR work; (b) + *OpenQA*, which adds popular open-domain question answering datasets, including NQ (Kwiatkowski et al., 2019) and HotpotQA; (c) + *BEIR-Train*, which incorporates the training splits of BEIR (Thakur et al., 2021), such as NFCorpus and Quora; (d) + *MTEB-Train*, which includes additional tasks from MTEB (Muennighoff et al., 2022) that are not covered in BEIR, such as NLI (we use the public BGE training split to collect these data); and (e) + *ZeroGR-Train*, which includes the data we collected from the training split of the MAIR (Sun et al., 2024) task collection, comprising 69 tasks from 6 domains (Figure 2).

Figure 2 shows the evaluation results of models (both query generator and docid generator) trained with different levels of task diversity, evaluated on the unseen task subset (i.e., tasks not included in any training set) of MAIR. The left plot in Figure 2 shows the distribution of average query length across tasks. We observe that models trained on more IR tasks generate queries with greater length diversity, indicating task-aware query generation strategies. In contrast, the baseline model trained only on MS MARCO produces short queries, averaging 8 words. The middle plot shows the docid conflict rate, i.e., the percentage of documents in the corpus assigned the same docid by the docid generator. Models trained on diverse tasks exhibit lower conflict rates, suggesting a stronger ability to process heterogeneous corpora. The MS MARCO baseline shows higher conflict on several diverse tasks. Finally, the right plot reports retrieval performance (top-1 accuracy) for different models. We observe consistent performance improvements on unseen tasks as training data diversity increases.

### 6.4 Comparisons of Different Docid Designs

Figure 3 compares our proposed unified docid with previous GR docid designs, while keeping all other factors (e.g., query generator, model choice, optimization strategy) constant to ensure an apple-to-apple comparison of docid effectiveness. The compared docid designs include: (i) **Random** (Tay et al., 2022), a baseline that assigns each document a random string as its docid; (ii) **Sentence** (Bevilacqua et al., 2022), which uses all sentences of each document as its docid; (iii) **Paragraph** (Tay et al., 2022), which takes the first paragraph of each document as its docid; (iv) **Query** (Tang et al., 2023), which uses a query generator to produce a single query per document as its docid; (v) **Summary**, as introduced in (Li et al., 2024), which uses the output of a summarization model as the docid; (vi) **RQ-VAE** (Zeng et al., 2023), which trains a RQ-VAE model on document embeddings produced by the `BGE-Large` model, enabling quantization of document embeddings into a sequence of tokens. This is a widely adopted docid representation in competitive GR systems.

From the results, we observe that among the various docid designs, our proposed docid generator consistently achieves the best performance on unseen development tasks. In particular, it significantly outperforms other text-based approaches such as *Summary* and *Query*, highlighting its superior ability to encode meaningful and discriminative document representations. This suggests that our design not only captures richer document semantics but is also better aligned with the generative retrieval objective, enabling more accurate and robust document retrieval. We further find that the performance of the *RQ-VAE* method is relatively unstable across different tasks, often requiring longer training to converge effectively. In contrast, our text-based docid benefits from the pretrained LM's inherent understanding of natural language, which facilitates more efficient learning and faster convergence.

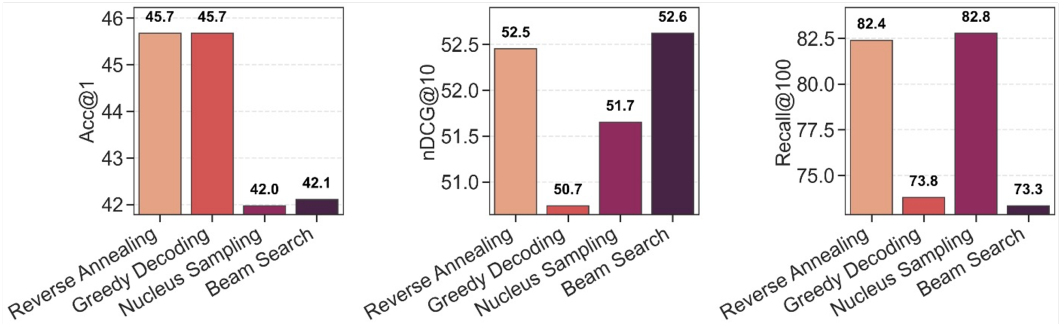


Figure 4: Ablation study of decoding algorithms across different metrics. Our proposed reverse annealing decoding achieves a good balance between precision and recall. Note that the y-axis is rescaled based on the model gap.

This synergy between instruction-driven docid generation and LM capabilities underpins the strong performance and generalization ability observed in our experiments.

### 6.5 Scaling Queries Number and Model Size

The middle section of Figure 3 illustrates the impact of the number of queries generated per document on the average top-1 accuracy of ZeroGR. We observe a clear upward trend: as the number of queries increases, the retrieval performance improves steadily. This highlights the importance of diverse query views for better semantic coverage during indexing. Notably, when using eight queries per document, ZeroGR already reaches performance on par with the strong sparse baseline BM25. Further increasing the query count to sixteen enables ZeroGR to surpass BM25, suggesting that high query diversity provides richer signals for matching user queries to relevant documents.

The right section of Figure 3 examines how the size of the backbone language model affects retrieval performance. For this analysis, we adopt a series of Qwen2.5 (Qwen et al., 2025) models with varying parameter scales. The results demonstrate a consistent gain in top-1 accuracy on unseen IR tasks as the model size grows, implying that larger models benefit from enhanced generalization and better understanding of the instruction-based retrieval formulation. This finding underscores the value of scaling up model capacity in generative retrieval frameworks, particularly in zero-shot settings.

### 6.6 Analysis of Decoding Strategies

In Figure 4, we compare our reverse annealing decoding with other popular decoding algorithms, including greedy decoding (i.e., greedily sampling from the GR model without replacement), nucleus sampling with a top-p of 0.9, and beam search. All methods decode the top-100 docids for evaluation. From the results, we observe that greedy decoding achieves the best performance in terms of Acc@1, but lacks diversity and yields low recall. Nucleus sampling performs poorly on Acc@1 but achieves high recall. In contrast, reverse annealing strikes a good balance between precision and recall, achieving competitive results across all metrics.

## 7 Conclusion

This work presents ZeroGR, an instruction-driven framework that extends generative retrieval to zero-shot scenarios. By unifying three key components, viz. a model-based docid generator, an instruction-conditioned query generator, and a reverse-annealed decoding algorithm, ZeroGR transforms a corpus and a natural-language task description into a task-specific generative index without requiring supervision. Systematic scaling studies along task diversity, query volume, and model size reveal consistent performance improvements. Empirical evaluations on MAIR tasks and BEIR datasets demonstrate the effectiveness of ZeroGR. The limitations of this work include the lack of evaluation on large-scale corpora (e.g., those with over 1M documents) and the use of relatively small LLMs (our largest model is only 3B). We believe further work is required to scale both the corpus size and the model size.

# References

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen tau Yih. Task-aware retrieval with instructions. *ArXiv*, abs/2211.09260, 2022.

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

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. Overview of the fire 2019 aila track: Artificial intelligence for legal assistance. In *Fire*, 2019.

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268, 2016.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. *ArXiv*, abs/2010.00904, 2020.

Jiangui Chen, Ruqing Zhang, J. Guo, Y. Liu, Yixing Fan, and Xueqi Cheng. Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022.

Jiangui Chen, Ruqing Zhang, J. Guo, M. de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. Continual learning for generative retrieval over dynamic corpora. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023.

Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Weiwei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. Tourrank: Utilizing large language models for documents ranking with a tournament-inspired strategy. *Proceedings of the ACM on Web Conference 2025*, 2024.

Yiqun Chen, Ling Yan, Weiwei Sun, Xinyu Ma, Yi Zhang, Shuaiqiang Wang, Dawei Yin, Yiming Yang, and Jiaxin Mao. Improving retrieval-augmented generation through multi-agent reinforcement learning. *ArXiv*, abs/2501.15228, 2025.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416, 2022.

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

Lotem Golany, Filippo Galgani, Maya Mamo, Nimrod Parasol, Omer Vandsburger, Nadav Bar, and Ido Dagan. Efficient data generation for source-grounded information-seeking dialogs: A use case for meeting transcripts. *ArXiv*, abs/2405.01121, 2024.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2021.

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

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR 2020*, 2020.

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 V. Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

John Lafferty and ChengXiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR 2001*, 2001.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. *ArXiv*, abs/2405.17428, 2024a.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernández Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha R. Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models. *ArXiv*, abs/2403.20327, 2024b.

Sunkyung Lee, Minjin Choi, and Jongwuk Lee. Glen: Generative retrieval via lexical index learning. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Alan Li, Daniel Cheng, Phillip Keung, Jungo Kasai, and Noah A. Smith. Summarization-based document IDs for generative retrieval with language models. In Lucie Lucie-Aimée, Angela Fan, Tajuddeen Gwadabe, Isaac Johnson, Fabio Petroni, and Daniel van Strien, editors, *Proceedings of the First Workshop on Advancing Natural Language Processing for Wikipedia*, pages 126–135, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.wikinlp-1.18.

Haoxin Li, Phillip Keung, Daniel Cheng, Jungo Kasai, and Noah A. Smith. Summarization-based document ids for generative retrieval with language models. *Proceedings of the First Workshop on Advancing Natural Language Processing for Wikipedia*, 2023a.

Yongqing Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. Multiview identifiers enhanced generative retrieval. *ArXiv*, abs/2305.16675, 2023b.

Jimmy J. Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2020.

Tianyang Liu, Canwen Xu, and Julian McAuley. Repobench: Benchmarking repository-level code auto-completion systems. *ArXiv*, abs/2306.03091, 2023a.

Wenhan Liu, Xinyu Ma, Weiwei Sun, Yutao Zhu, Yuchen Li, Dawei Yin, and Zhicheng Dou. Reasonrank: Empowering passage ranking with strong reasoning ability. *ArXiv*, abs/2508.07050, 2025.

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Wei Chen, and Xueqi Cheng. On the robustness of generative retrieval models: An out-of-distribution perspective. *arXiv preprint arXiv:2306.12756*, 2023b.

Xing Han Lù. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring. *ArXiv*, abs/2407.03618, 2024.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. Dsi++: Updating transformer memory with new documents. In *Conference on Empirical Methods in Natural Language Processing*, 2022.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. Rethinking search: Making experts out of dilettantes. *ArXiv*, abs/2105.02274, 2021.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2022.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. *ArXiv*, abs/2402.09906, 2024.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. *ArXiv*, abs/2112.07899, 2021.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *ArXiv*, abs/1901.04085, 2019.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy J. Lin. Document ranking with a pretrained sequence-to-sequence model. *ArXiv*, abs/2003.06713, 2020.

Hanseok Oh, Hyunji Lee, Seonghyeon Ye, Haebin Shin, Hansol Jang, Changwook Jun, and Minjoon Seo. Instructir: A benchmark for instruction following of information retrieval models. *ArXiv*, abs/2402.14334, 2024.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktaschel, and Sebastian Riedel. Kilt: a benchmark for knowledge intensive language tasks. In *North American Chapter of the Association for Computational Linguistics*, 2020.

Ronak Pradeep, Kai Hui, Jai Gupta, Ádám Dániel Lelkes, Honglei Zhuang, Jimmy J. Lin, Donald Metzler, and Vinh Q. Tran. How does generative retrieval scale to millions of passages? *ArXiv*, abs/2305.11841, 2023.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *NAACL 2021*, 2021.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Hieu Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. *ArXiv*, abs/2305.05065, 2023.

Ruiyang Ren, Wayne Xin Zhao, J. Liu, Huaqin Wu, Ji rong Wen, and Haifeng Wang. Tome: A two-stage approach for model-based retrieval. *ArXiv*, abs/2305.11161, 2023.

Stephen E. Robertson and Steve Walker. On relevance weights with little relevance information. In *SIGIR 1997*, 1997.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389, 2009.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. *ArXiv*, abs/2212.09741, 2022.

Weiwei Sun, Zheng Chen, Xinyu Ma, Lingyong Yan, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Instruction distillation makes large language models efficient zero-shot rankers. *ArXiv*, abs/2311.01555, 2023a.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, M. de Rijke, and Zhaochun Ren. Learning to tokenize for generative retrieval. *ArXiv*, abs/2304.04171, 2023b.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agent. *ArXiv*, abs/2304.09542, 2023c.

Weiwei Sun, Zhengliang Shi, Jiulong Wu, Lingyong Yan, Xinyu Ma, Yiding Liu, Min Cao, Dawei Yin, and Zhaochun Ren. Mair: A massive benchmark for evaluating instructed retrieval. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

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

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

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663, 2021.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv*, abs/2212.03533, 2022a.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *ArXiv*, abs/2401.00368, 2023a.

Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, Zhumin Chen, and Xin Xin. Content-based collaborative generation for recommender systems. *CIKM*, 2024.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, M. Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Conference on Empirical Methods in Natural Language Processing*, 2022b.

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

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. Promptriever: Instruction-trained retrievers can be prompted like language models. *ArXiv*, abs/2409.11136, 2024.

Haoyang Wen, Jiang Guo, Yi Zhang, Jiarong Jiang, and Zhiguo Wang. On synthetic data strategies for domain-specific generative retrieval. *ArXiv*, abs/2502.17957, 2025.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding. *ArXiv*, abs/2309.07597, 2023.

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

Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. Scalable and effective generative information retrieval. *Proceedings of the ACM on Web Conference 2024*, 2023.

Hansi Zeng, Chen Luo, and Hamed Zamani. Planning ahead in generative retrieval: Guiding autoregressive generation through simultaneous decoding. *ArXiv*, abs/2404.14600, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *ArXiv*, abs/2506.05176, 2025a.

Zhen Zhang, Xinyu Ma, Weiwei Sun, Pengjie Ren, Zhumin Chen, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Replication and exploration of generative retrieval over dynamic corpora. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3325–3334, 2025b.

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

Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat seng Chua. Towards complex document understanding by discrete reasoning. *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

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

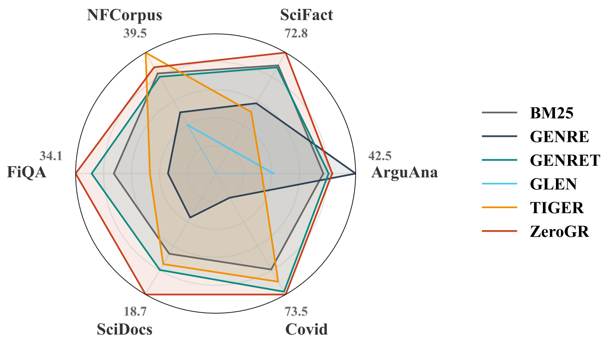


Figure 5: Performance (nDCG@10) of different generative retrieval models across various datasets on BEIR.

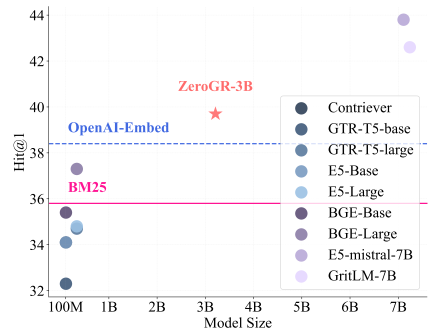


Figure 6: Performance (Acc@1) on unseen subset of MAIR.

## A Prompts

```
1. **Length**: Strictly 6-8 words (terms/words)
2. **Term Inclusion**: Must include 3-5 core terms directly from
the document
3. **Term Positioning**: Rank by relevance and importance (highest
→ lowest, general → specific)
4. **Formatting**:
   - Use lowercase letters, numbers, and spaces only
   - Preserve special terms/symbols (e.g., PD3.1)
   - **No articles** (a, the), **linking verbs**, or auxiliary
verbs
   - **No verbs** (use nouns/adjectives only)
5. **Requirements**:
   - Terms must be derivable from the document
   - Ensure uniqueness and precise core content representation
```

## B Decoding

**Algorithm 1** DocID Generation with Reverse Annealing

**Require:** $T$ (total number of docids), model, query, max_temperature
**Ensure:** List of generated docids

- **for** $t = 1, 2, \ldots, T$ **do**
  - # Compute normalized decoding temperature (Eq. 5)
  - $\text{temperature}_t \leftarrow \text{reverse-annealing}(t, T, \text{max\_temperature})$
  - # Generate next tokens with temperature control
  - $\text{docid}_t \leftarrow \text{model}(\text{query}, \text{temperature}_t)$
- **end for**
- **return** List of generated docids

## C Experimental Results on MAIR and BEIR

Our experimental results on MAIR and BEIR are shown in Table 4 and Table 5.

## D The ZeroGR-Train Dataset

We show the statistics of ZeroGR-Train Dataset in the Table 7 and Figure 7.

| Dataset | Seen Subset | | | | | | | | | | | | | | | | Unseen Subset | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Avg** | FiQA | NFC.s | SciD. | SciF. | ToQA. | TAT | CoF. | LeetC. | LitSe. | BiSum | CodeSe. | Math | ConvF. | Conala | StMath | Apple | FinBen | AILAC | AILAS |
| BM25 | 36.1 | 24.0 | 45.5 | 16.0 | 53.0 | 11.0 | 67.1 | 47.9 | 12.0 | 66.0 | 69.0 | 33.0 | 41.0 | 47.9 | 7.0 | 20.0 | 52.1 | 9.0 | 14.0 | 10.0 |
| Contriever | 33.6 | 33.0 | 43.5 | 17.0 | 62.0 | 11.0 | 54.3 | 42.7 | 7.0 | 39.0 | 54.0 | 37.0 | 36.0 | 42.7 | 9.0 | 13.0 | 50.7 | 6.0 | 12.0 | 6.0 |
| GTR-T5-base | 32.5 | 33.0 | 43.0 | 12.0 | 50.0 | 16.0 | 58.6 | 37.5 | 6.0 | 41.0 | 47.0 | 41.0 | 49.0 | 37.5 | 9.0 | 16.0 | 47.9 | 10.0 | 4.0 | 10.0 |
| GTR-T5-large | 35.4 | 45.0 | 43.5 | 14.0 | 55.0 | 12.0 | 40.0 | 42.7 | 10.0 | 43.0 | 59.0 | 51.0 | 63.0 | 42.7 | 11.0 | 23.0 | 50.7 | 14.0 | 6.0 | 6.0 |
| E5-Base | 37.2 | 41.0 | 40.0 | 17.0 | 63.0 | 16.0 | 61.4 | 52.1 | 11.0 | 49.0 | 61.0 | 59.0 | 78.0 | 52.1 | 10.0 | 36.0 | 52.1 | 18.0 | 6.0 | 12.0 |
| E5-Large | 38.2 | 45.0 | 45.5 | 20.0 | 67.0 | 13.0 | 70.0 | 57.3 | 9.0 | 49.0 | 52.0 | 57.0 | 75.0 | 57.3 | 11.0 | 44.0 | 47.9 | 13.0 | 12.0 | 6.0 |
| BGE-Base | 37.0 | 43.0 | 43.5 | 20.0 | 62.0 | 13.0 | 58.6 | 41.7 | 10.0 | 44.0 | 67.0 | 64.0 | 50.0 | 41.7 | 13.0 | 25.0 | 47.9 | 20.0 | 8.0 | 8.0 |
| BGE-Large | 39.4 | 51.0 | 46.5 | 22.0 | 65.0 | 13.0 | 60.0 | 44.8 | 13.0 | 56.0 | 68.0 | 66.0 | 66.0 | 44.8 | 8.0 | 22.0 | 46.6 | 23.0 | 8.0 | 8.0 |
| OpenAI-Embed | 40.6 | 51.0 | 51.0 | 22.0 | 60.0 | 19.0 | 62.9 | 51.0 | 6.0 | 53.0 | 59.0 | 67.0 | 73.0 | 51.0 | 13.0 | 33.0 | 52.1 | 30.0 | 10.0 | 10.0 |
| GTE-Qwen2-1.5B | 44.4 | 54.0 | 50.0 | 24.0 | 69.0 | 25.0 | 65.7 | 65.6 | 41.0 | 63.0 | 79.0 | 70.0 | 84.0 | 65.6 | 20.0 | 40.0 | 47.9 | 33.0 | 12.0 | 10.0 |
| E5-mistral-7B | 46.8 | 60.0 | 50.5 | 17.0 | 67.0 | 14.0 | 67.1 | 64.6 | 36.0 | 68.0 | 74.0 | 54.0 | 78.0 | 64.6 | 30.0 | 47.0 | 43.8 | 41.0 | 12.0 | 38.0 |
| GritLM-7B | 47.0 | 63.0 | 49.5 | 29.0 | 69.0 | 17.0 | 85.7 | 62.5 | 46.0 | 60.0 | 74.0 | 53.0 | 87.0 | 62.5 | 21.0 | 46.0 | 43.8 | 33.0 | 12.0 | 42.0 |
| **ZeroGR-3B** | 41.1 | 37.0 | 36.5 | 24.0 | 51.0 | 13.0 | 38.6 | 57.3 | 36.0 | 41.0 | 81.0 | 61.0 | 81.0 | 57.3 | 12.0 | 40.0 | 52.1 | 11.0 | 12.0 | 22.0 |

| | Unseen Subset | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACOR. | CPCD | CORE | MB. | PM. | PM.A | CliDS | CliT23 | DD | Table | QuanT | PoRec | Monant | NCL. | NCL.T | Legal | Geno. | Touche | CliT21 | News21 |
| BM25 | 32.8 | 1.0 | 37.5 | 83.8 | 53.9 | 6.5 | 28.3 | 51.4 | 15.6 | 10.0 | 86.9 | 24.5 | 67.4 | 50.7 | 22.2 | 45.0 | 52.8 | 59.2 | 33.3 | 10.9 |
| Contriever | 40.4 | 1.0 | 52.5 | 89.2 | 32.9 | 0.0 | 6.7 | 37.8 | 21.3 | 8.3 | 76.8 | 46.7 | 65.0 | 60.0 | 34.2 | 35.0 | 27.8 | 52.0 | 32.7 | 23.8 |
| GTR-T5-base | 31.3 | 1.0 | 47.5 | 89.2 | 36.8 | 1.6 | 13.3 | 36.5 | 13.6 | 12.5 | 77.8 | 37.7 | 70.0 | 48.0 | 22.2 | 40.0 | 25.0 | 55.1 | 29.3 | 15.6 |
| GTR-T5-large | 34.8 | 3.0 | 60.0 | 91.9 | 31.6 | 1.6 | 8.3 | 39.2 | 16.2 | 10.0 | 78.8 | 48.7 | 68.0 | 53.3 | 23.9 | 40.0 | 25.0 | 65.3 | 37.3 | 19.1 |
| E5-Base | 40.4 | 3.0 | 42.5 | 81.1 | 43.4 | 6.5 | 11.7 | 39.2 | 13.2 | 5.8 | 78.8 | 54.2 | 72.0 | 55.3 | 27.4 | 35.0 | 33.3 | 37.8 | 36.0 | 14.5 |
| E5-Large | 38.4 | 3.0 | 45.0 | 86.5 | 36.8 | 4.8 | 15.0 | 40.5 | 12.3 | 7.5 | 80.8 | 52.5 | 71.0 | 55.3 | 47.9 | 30.0 | 38.9 | 41.8 | 32.0 | 17.2 |
| BGE-Base | 39.4 | 0.0 | 45.0 | 91.9 | 48.7 | 0.0 | 30.0 | 31.1 | 16.4 | 8.3 | 81.8 | 44.4 | 70.0 | 57.3 | 42.7 | 20.0 | 36.1 | 41.8 | 41.3 | 19.9 |
| BGE-Large | 36.9 | 0.0 | 52.5 | 94.6 | 42.1 | 3.2 | 23.3 | 28.4 | 17.8 | 5.0 | 81.8 | 50.4 | 74.0 | 54.0 | 40.2 | 60.0 | 38.9 | 51.0 | 41.3 | 15.2 |
| OpenAI-Embed | 32.8 | 1.0 | 55.0 | 86.5 | 46.1 | 3.2 | 25.0 | 44.6 | 13.5 | 12.5 | 86.9 | 47.3 | 76.0 | 57.3 | 49.6 | 45.0 | 33.3 | 52.0 | 38.0 | 13.7 |
| GTE-Qwen2-1.5B | 37.9 | 5.1 | 70.0 | 81.1 | 14.5 | 8.1 | 20.0 | 17.6 | 14.8 | 14.2 | 85.9 | 58.6 | 74.2 | 61.3 | 51.3 | 40.0 | 61.1 | 65.3 | 31.3 | 23.0 |
| E5-mistral-7B | 41.9 | 5.0 | 60.0 | 83.8 | 43.4 | 1.6 | 46.7 | 48.6 | 19.4 | 11.7 | 83.8 | 66.1 | 71.0 | 62.7 | 58.1 | 45.0 | 36.1 | 58.2 | 46.7 | 25.4 |
| GritLM-7B | 35.4 | 7.0 | 65.0 | 70.3 | 59.2 | 0.0 | 28.3 | 45.9 | 14.8 | 10.8 | 86.9 | 72.2 | 77.0 | 63.3 | 50.4 | 45.0 | 33.3 | 57.1 | 47.3 | 22.7 |
| **ZeroGR-3B** | 26.7 | 4.0 | 55.0 | 89.2 | 23.9 | 12.9 | 25.0 | 37.9 | 44.4 | 12.0 | 79.8 | 60.9 | 69.7 | 65.3 | 36.2 | 45.0 | 58.3 | 46.9 | 42.0 | 21.5 |

Table 4: Model performance (top-1 retrieval accuracy) on seen and unseen subset of MAIR.

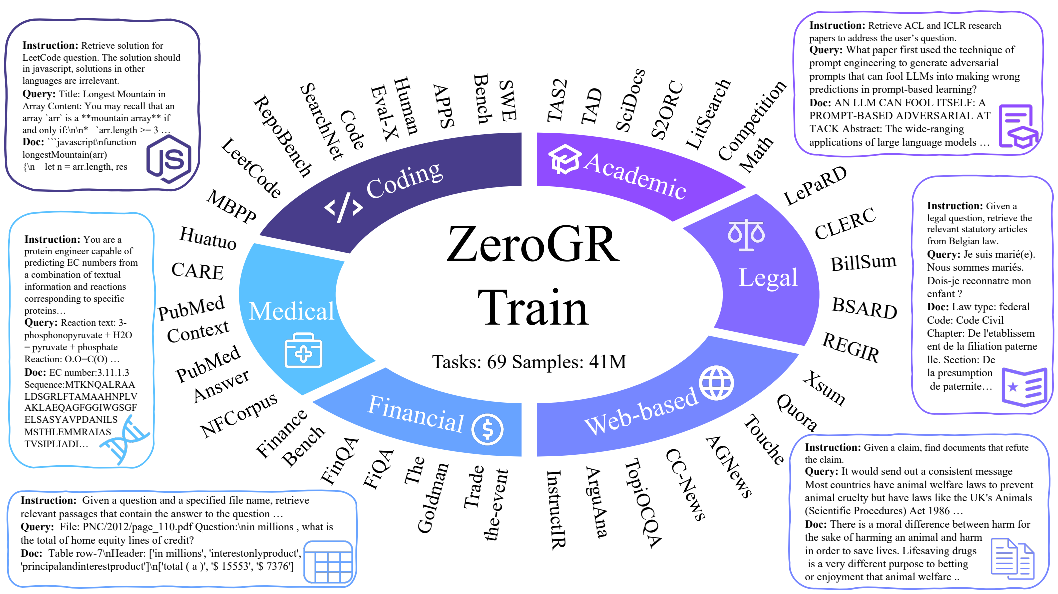


Figure 7: Overview of ZeroGR-Train.

| Category | Method | Avg. | ArguAna | SciFact | NFCorpus | FiQA | SciDocs |
|---|---|---|---|---|---|---|---|
| Sparse | BM25 | 42.3 | 32.7 | 65.1 | 32.7 | 24.8 | 12.4 |
| DR | Contriever | 47.6 | 32.1 | 70.3 | 33.9 | 35.5 | 15.7 |
| DR | GTR-T5-base | 45.3 | 32.7 | 58.6 | 32.7 | 34.5 | 12.1 |
| DR | GTR-T5-large | 48.0 | 34.3 | 61.9 | 33.4 | 43.3 | 12.8 |
| DR | E5-Base | 48.9 | 31.1 | 73.9 | 35.0 | 39.6 | 18.3 |
| DR | E5-Large | 49.2 | 31.7 | 76.3 | 37.4 | 42.3 | 19.6 |
| DR | BGE-Base | 50.5 | 41.8 | 74.3 | 36.0 | 43.4 | 19.8 |
| DR | BGE-Large | 51.8 | 41.6 | 75.2 | 38.1 | 48.5 | 20.6 |
| DR | E5-mistral-7B | 55.7 | 44.1 | 76.6 | 39.6 | 59.7 | 20.7 |
| DR | GritLM-7B | 45.0 | 40.7 | 76.8 | 36.9 | 44.1 | 19.6 |
| DR | OpenAI Embed | 54.2 | 37.1 | 73.1 | 37.6 | 48.5 | 21.2 |
| GR | GENRE | – | 42.5 | 42.3 | 20.0 | 11.6 | 6.8 |
| GR | GENRET | – | 34.3 | 63.9 | 31.6 | 30.2 | 14.9 |
| GR | GLEN | – | 17.6 | – | 15.9 | – | – |
| GR | TIGER (Llama-3B) | – | 14.0 | 37.0 | 39.5 | 16.0 | 14.0 |
| GR | ZeroGR-3B | 48.1 | 35.4 | 72.8 | 34.7 | 34.1 | 18.7 |
| **Category** | **Method** | **Touche** | **TREC-News** | **Fever** | **Quora** | **Covid** | **CQADupStack** |
| Sparse | BM25 | 59.0 | 20.7 | 58.3 | 73.8 | 58.3 | 28.0 |
| DR | Contriever | 42.5 | 27.3 | 90.6 | 86.6 | 59.6 | 29.9 |
| DR | GTR-T5-base | 48.1 | 22.5 | 83.2 | 88.7 | 56.1 | 28.9 |
| DR | GTR-T5-large | 53.1 | 26.6 | 86.8 | 89.1 | 56.7 | 29.8 |
| DR | E5-Base | 41.1 | 22.9 | 91.1 | 86.4 | 60.7 | 38.3 |
| DR | E5-Large | 34.8 | 25.3 | 93.1 | 86.9 | 55.2 | 38.5 |
| DR | BGE-Base | 41.4 | 21.2 | 85.6 | 89.8 | 67.1 | 35.1 |
| DR | BGE-Large | 45.5 | 21.4 | 86.6 | 89.3 | 64.5 | 38.3 |
| DR | E5-mistral-7B | 46.8 | 29.4 | 91.8 | 84.8 | 77.8 | 41.4 |
| DR | GritLM-7B | 21.5 | 34.9 | 68.9 | 84.9 | 31.5 | 35.0 |
| DR | OpenAI Embed | 47.5 | 26.2 | 92.8 | 89.9 | 78.2 | 44.1 |
| GR | GENRE | – | – | – | – | 14.7 | – |
| GR | GENRET | – | – | – | – | 71.8 | – |
| GR | GLEN | – | – | – | – | – | – |
| GR | TIGER (Llama-3B) | 58.1 | 16.4 | – | 59.6 | 65.7 | – |
| GR | ZeroGR-3B | 37.5 | 23.5 | 86.7 | 76.7 | 73.5 | 35.2 |

Table 5: nDCG@10 on BEIR benchmark datasets.

| Domain | MAIR-Full | ZeroGR-Train | MAIR-Test | BEIR |
|---|---|---|---|---|
| Academic | 16 | 18 | 5 | 2 |
| Code | 18 | 13 | 3 | 0 |
| Finance | 8 | 8 | 5 | 1 |
| Legal | 11 | 7 | 4 | 0 |
| Medical | 19 | 5 | 8 | 2 |
| Web | 54 | 18 | 13 | 6 |
| All | 126 | 69 | 38 | 11 |

Table 6: Dataset domain statistics of MAIR-Full, ZeroGR-Train, MAIR-Test, and BEIR.

| Dataset | Samples | Dataset | Samples |
|---|---|---|---|
| **Academic** | | | |
| S2ORC-title-citation | 100,000 | TAD | 208,255 |
| S2ORC-abstract-citation | 100,000 | TAS2 | 107,700 |
| S2ORC-title-abstract | 100,000 | StackMathQA | 47,142 |
| ProofWiki-Proof | 15,520 | ProofWiki-Reference | 2,098 |
| ProofWiki-Proof | 15,520 | ProofWiki-Reference | 2,098 |
| Stacks-Proof | 10,928 | Stacks-Reference | 9,022 |
| Stacks-Reference | 9,022 | Competition-Math | 7,500 |
| Competition-Math | 7,500 | SciDocs | 900 |
| SciFact | 809 | LitSearch | 146 |
| **Code** | | | |
| CodeSearchNet | 1,880,853 | CodeEditSearch | 21,395 |
| SWE-Bench | 18,817 | RepoBench | 16,655 |
| HF-API | 8,191 | TLDR | 6,414 |
| TensorAPI | 6,190 | APPS | 5,000 |
| LeetCode | 2,260 | Conala | 1,794 |
| PyTorchAPI | 837 | HumanEval-X | 720 |
| MBPP | 374 | | |
| **Finance** | | | |
| USnews | 9,999 | FinQA | 6,251 |
| FiQA | 5,500 | HC3Finance | 3,104 |
| ConvFinQA | 3,037 | TheGoldman | 1,512 |
| TAT-DQA | 1,012 | Trade-the-event | 900 |
| **Legal** | | | |
| LePaRD | 22,734,882 | CLERC | 327,414 |
| BillSum | 18,949 | REGIR-UK2EU | 2,100 |
| REGIR-EU2UK | 2,000 | BSARD | 886 |
| CUAD | 717 | | |
| **Medical** | | | |
| PubMedQA-Context | 196,696 | PubMedQA-Answer | 196,696 |
| Huatuo | 25,371 | NFCorpus | 2,590 |
| CARE | 77 | | |
| **Web** | | | |
| Reddit | 12,704,958 | AGNews | 1,157,745 |
| CC-News | 708,241 | Xsum | 204,045 |
| zsRE | 147,909 | ToT | 109,454 |
| Fever | 109,810 | WoW | 63,734 |
| TopiOCQA | 45,450 | AY2 | 18,395 |
| CQADupStack | 13,045 | InstructIR | 9,806 |
| Quora | 9,900 | WnCw | 5,499 |
| TREx | 4,900 | ExcluIR | 3,352 |
| NevIR | 1,896 | ArguAna | 1,306 |

Table 7: Dataset statistics grouped by domain and sorted by sample count.

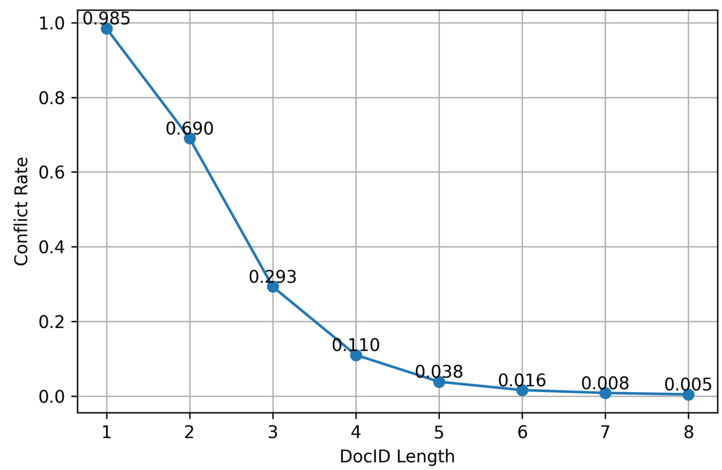


Figure 8: Docid conflict rate wrt docid length.

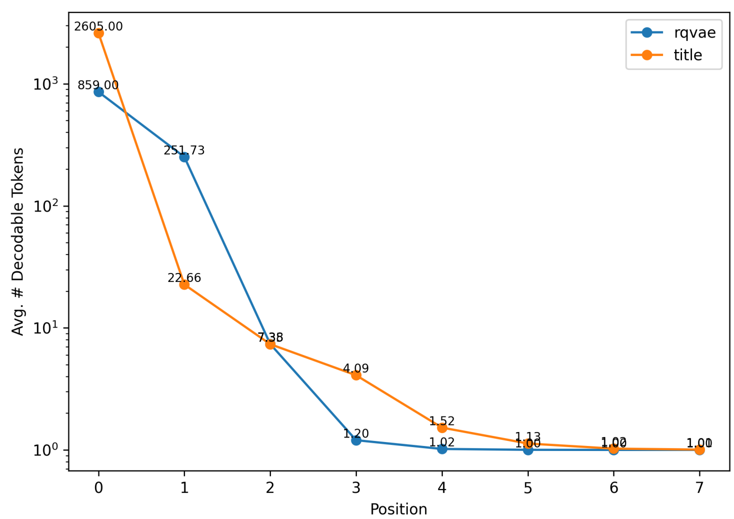


Figure 9: Average number of decodable tokens at each position, for RQ-VAE docid and our title docid.

| Model | Size | Training Data | Link |
|---|---|---|---|
| BM25 | N/A | N/A | https://github.com/cvangysel/BM25S |
| Contriever-MARCO | 110M | MS MARCO | https://github.com/facebookresearch/contriever |
| GTR-base | 110M | MS MARCO | https://huggingface.co/google/gtr-base |
| GTR-large | 335M | MS MARCO | https://huggingface.co/google/gtr-large |
| E5-base | 110M | unknown | https://huggingface.co/intfloat/e5-base-v2 |
| E5-large | 335M | unknown | https://huggingface.co/intfloat/e5-large-v2 |
| BGE-base | 110M | MTEB-Train | https://huggingface.co/BAAI/bge-base-en-v1.5 |
| BGE-large | 335M | MTEB-Train | https://huggingface.co/BAAI/bge-large-en-v1.5 |
| OpenAI-Embed-Small | unknown | unknown | https://platform.openai.com/docs/guides/embeddings |
| E5-Mistral-7B-instruct | 7B | E5 (LLM generated) | https://huggingface.co/intfloat/e5-mistral-7b-instruct |
| GritLM-7B | 7B | E5 (LLM generated) | https://huggingface.co/GritLM/GritLM-7B |

Table 8: Dense retrieval model information.

| Method | Training Data | Model Size | DocID Type | Decoding |
|---|---|---|---|---|
| GENRE (Cao et al., 2020) | GPL | T5-220M | Title | Beam Search |
| GENRET (Sun et al., 2023b) | GPL | T5-220M | RQ-VAE | Beam Search |
| GLEN (Lee et al., 2023) | NQ320k | T5-220M | Keywords | Beam Search |
| TIGER (Rajput et al., 2023) | ZeroGR-Train | Llama-3B | RQ-VAE | Reverse-Annealing |
| ZeroGR (Ours) | ZeroGR-Train | Llama-3B | Title | Reverse-Annealing |

Table 9: Generative retrieval model information.

| **Name** | **Model** | **Task List (with query count)** |
|---|---|---|
| Figure 2, Figure 3 (left) | Llama-1B | {ToolBench (100), AILA2019-Case (50), NFCorpus (100), SciFact (100), ArguAna (100), LitSearch (100), ClinicalTrials_2023 (37), FinanceBench (100), SciDocs (100), News21 (100), TopiOCQA (100), Touche (49), FiQA (100)} |
| Figure 3 (middle, right) | Llama-1B, or Qwen2.5 | {LeetCode (100), Competition-Math (100), TMDB (100), Stein_Proof (64), PytorchAPI (100)} |
| Figure 4 | Llama-1B | {Leetcode (100), Competition-Math (100), BillSum (100), SciFact (100), TAT-DQA (70), ConvFinQA (96)} |

Table 10: Development set for ablation study.

| **Task** | **doc2query** | **our query generator** | **Diff** |
|---|---|---|---|
| AILA2019-Case | 2.00 | 2.00 | +0.00 |
| Apple | 13.70 | 5.48 | -8.22 |
| ArguAna | 12.00 | 11.00 | -1.00 |
| BillSum | 36.00 | 66.00 | **+30.00** |
| ClinicalTrials_2021 | 4.67 | 6.67 | +2.00 |
| ClinicalTrials_2023 | 1.35 | 2.70 | +1.35 |
| CodeEditSearch | 13.00 | 22.00 | +9.00 |
| CodeSearchNet | 33.00 | 56.00 | **+23.00** |
| Competition-Math | 40.00 | 61.00 | **+21.00** |
| Conala | 3.00 | 9.00 | +6.00 |
| ConvFinQA | 22.92 | 37.50 | +14.58 |
| FiQA | 7.00 | 13.00 | +6.00 |
| FinQA | 14.44 | 41.11 | **+26.67** |
| LeetCode | 6.00 | 30.00 | **+24.00** |
| LegalQuAD | 10.00 | 4.00 | -6.00 |
| LitSearch | 12.00 | 31.00 | **+19.00** |
| NFCorpus | 41.00 | 6.50 | -34.50 |
| News21 | 13.67 | 21.88 | +8.20 |
| SciDocs | 16.00 | 14.00 | -2.00 |
| SciFact | 34.00 | 42.00 | +8.00 |
| StackMathQA | 13.00 | 26.00 | +13.00 |
| TAT-DQA | 7.14 | 27.14 | **+20.00** |
| ToT_2023 | 3.00 | 0.00 | -3.00 |
| TopiOCQA | 18.00 | 8.00 | -10.00 |
| Touche | 46.94 | 39.80 | -7.14 |
| **Average** | **16.95** | **23.35** | **+6.40** |

Table 11: Performance comparison between doc2query and our method for the RQ-VAE docID baseline (TIGER (Rajput et al., 2023)).

```
def normalized_sigmoid(t, k=10, m=0.5):
    sigmoid = lambda z: 1 / (1 + np.exp(-z))
    a = sigmoid(k * (0 - m))
    b = sigmoid(k * (1 - m))
    return (sigmoid(k * (t - m)) - a) / (b - a)
```

Figure 10: Normalized sigmoid function. $t$ is the step number.

| **Type** | **Example** |
|---|---|
| Random | asd8xc2c9ma90xj2398 |
| Sentence | LIMASSOL, Cyprus, April 28, 2021 /PRNewswire/ – One of the top financial investment firms of the FX industry, Windsor Brokers .... |
| Paragraph | LIMASSOL, Cyprus, April 28, 2021 /PRNewswire/ – One of the top financial investment firms of the FX industry, Windsor Brokers .... |
| Query | Induction of myelodysplasia by myeloid-derived suppressor cells. |
| Summary | 1. Game of Thrones season 7 2. Plot and storyline 3. New cast members 4. Filming locations 5. Critical reception and ratings |
| RQ-VAE | `<\|g16289\|> <\|g13509\|> <\|g10485\|> <\|g11274\|> <\|g369\|> <\|g3661\|> <\|g13026\|> <\|g8187\|>` |
| IDF | brokerswindsor mt4 brokerswere kontos windsorbrokers |
| Ours | rna folding computational methods thermodynamic optimization model |

Table 12: Examples of different types of docids.

| **Category** | **Query Types** | **Doc Types** |
|---|---|---|
| **Train ∩ Eval** | Question, Dialog, Claim, Function Header, NL Command, Code Problem, Math question, Paper Title, Summary *(9 types)* | Document, Answer, Function, Command Doc, Solution, Article, Articles, Medical Document, Paragraph, Pages, Statute, Passage, Passages, Table & Paragraph *(14 types)* |
| **Only in Eval** | Health Record, Topic, Situation, Request, Patient Data, Medical Case, Patient Description, Medical Claim, Numerical Claim *(9 types)* | Clinical Trials, Prior Case, Communications, Dataset, Music, Tweet, News, POI, Table *(8 types)* |
| **Only in Train** | Math Statement, Entity & Relation, Paper Abstract, Entity Mention, CNL Command, GitHub Issue, Commit, Code Context, Math Question, Title, EU Directive, UK Legislation, Instruction, Reaction, Description *(14 types)* | Entity Page, Citation, Proof, Reference, Duplicate Question, Related File, Code Diff, Next Function, HuggingFace API, Tensor API, PyTorch API, UK Legislation, EU Directive, Highlight, Proteins Documents, Wikipedia Page *(16 types)* |

Table 13: Comparison of Query and Doc Types between Dataset A (38 datasets) and Dataset B (51 datasets)