I3: Intent-Introspective Retrieval Conditioned on Instructions

Kaihang Pan*
Zhejiang University
Hangzhou, China
kaihangpan@zju.edu.cn

Juncheng Li†
Zhejiang University
Hangzhou, China
junchengli@zju.edu.cn

Wenjie Wang
National University of Singapore
Singapore, Singapore
wenjiewang96@gmail.com

Hao Fei
National University of Singapore
Singapore, Singapore
haofei37@nus.edu.sg

Hongye Song
DAMO Academy, Alibaba Group
Hangzhou, China
hongye.shy@alibaba-inc.com

Wei Ji
National University of Singapore
Singapore, Singapore
jiwei@nus.edu.sg

Jun Lin
DAMO Academy, Alibaba Group
Hangzhou, China
linjun.lj@alibaba-inc.com

Xiaozhong Liu
Worcester Polytechnic Institute
Worcester, United States
xliu14@wpi.edu

Tat-Seng Chua
National University of Singapore
Singapore, Singapore
dcscts@nus.edu.sg

Siliang Tang
Zhejiang University
Hangzhou, China
siliang@zju.edu.cn

ABSTRACT

Recent studies indicate that dense retrieval models struggle to perform well on a wide variety of retrieval tasks that lack dedicated training data, as different retrieval tasks often entail distinct search intents. To address this challenge, in this work we leverage instructions to flexibly describe retrieval intents and introduce I3, a unified retrieval system that performs **I**ntent-**I**ntrospective retrieval across various tasks, conditioned on **I**nstructions without any task-specific training. I3 innovatively incorporates a pluggable introspector in a parameter-isolated manner to comprehend specific retrieval intents by jointly reasoning over the input query and instruction, and seamlessly integrates the introspected intent into the original retrieval model for intent-aware retrieval. Furthermore, we propose **progressively-pruned intent learning**. It utilizes extensive LLM-generated data to train I3 phase-by-phase, embodying two key designs: progressive structure pruning and drawback extrapolation-based data refinement. Extensive experiments show that in the BEIR benchmark, I3 significantly outperforms baseline methods designed with task-specific retrievers, achieving state-of-the-art zero-shot performance without any task-specific tuning.

CCS CONCEPTS

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

KEYWORDS

Intent-introspective retrieval, progressive structure pruning, drawback extrapolation-based data refinement

ACM Reference Format:
Kaihang Pan*, Juncheng Li†, Wenjie Wang, Hao Fei, Hongye Song, Wei Ji, Jun Lin, Xiaozhong Liu, Tat-Seng Chua, and Siliang Tang. 2024. I3: Intent-Introspective Retrieval Conditioned on Instructions. 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, 13 pages. https://doi.org/10.1145/3626772.3657745

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

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657745

1 INTRODUCTION

Information Retrieval (IR) is a fundamental task with widespread applications not only in real-world scenarios such as web search [18] and digital libraries [40], but also extending its significance to retrieval-augmented large language models (LLMs) [12]. Recent dense retrieval models have demonstrated remarkable performance based on the transformer architecture in a manner of dual encoders. Through the dual-encoders, they excel at encoding queries and documents into a shared representation space to facilitate semantic matching after the training with abundant annotated data [2, 19].

Nonetheless, this approach overlooks a critical fact that different retrieval tasks often entail varied search intents. Recent studies [42] indicate that existing dense retrieval models struggle to perform well on a wide variety of retrieval tasks that lack dedicated training data. When encountering a novel retrieval task, sufficient annotated data is necessary for training retrieval models to implicitly grasp the search intent, as demonstrated in Figure 1.a. Given the challenge of obtaining such annotated data, a recent work, Promptagator [8] instructs LLMs to generate task-specific training data by presenting them with sets of 8 examples. Then, it utilizes the generated

* Work done when interning at Alibaba DAMO Academy.
† Corresponding Author.

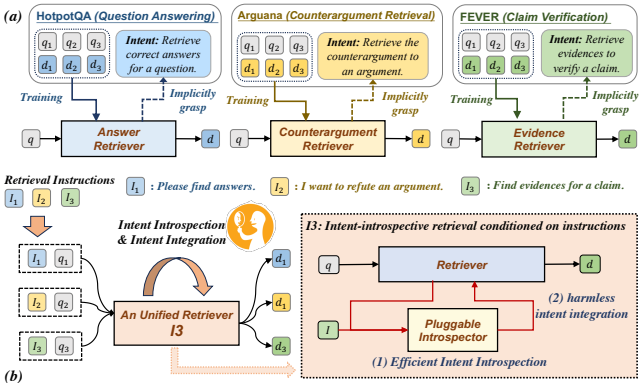

Figure 1: (a) Existing methods require training a distinct retrieval model for each task to implicitly grasp specific retrieval intent. (b) I3 directly handles different tasks through intent-introspective retrieval following user instructions.

pseudo training data to train task-specific retrieval models for each distinct task. While obtaining promising improvements, Promptagator necessitates the training of a distinct model to implicitly grasp the retrieval intent of each task, which limits the flexibility to seamlessly transfer across diverse retrieval tasks.

Instead of training a specific model for each retrieval task, it is desirable to enable pre-trained retrieval models to directly perform different tasks guided by instructions that flexibly describe retrieval intents in natural language [1]. To achieve this, an ideal approach can be divided into two steps: (1) ***effective intent introspection***: deeply comprehending specific retrieval intent by jointly reasoning over the input query and instruction; (2) ***harmless intent integration***: integrating the introspected intent into the pre-trained retrieval model, while avoiding disrupting the model's original capabilities. Based on this insight, we aim to achieve such flexible intent-aware retrieval from both model and learning perspectives.

From the model perspective, we introduce **I3**, a generic approach that enables pre-trained retrieval models to effectively perform **intent-introspective retrieval** across various tasks, while simultaneously maintaining the pre-trained capabilities and avoiding any task-specific training. Specifically, given an arbitrary dense retrieval model with dual-encoders, I3 retains the original encoders and innovatively develops a pluggable introspector in a parameter-isolated manner to preserve the inherent capability of retrieval models. The pluggable introspector subtly interprets the specific retrieval intent by jointly comprehending the given query and instruction. And the introspected intent is then seamlessly integrated into the original query encoder, which is harmless due to the parameter-isolated framework, endowing retrieval models with a new facet of intent-aware retrieval conditioned on instructions.

From the learning perspective, to efficiently perform intent-introspective retrieval across a wide range of retrieval tasks, the pluggable introspector should (1) remain lightweight to avoid significantly affecting the time efficiency of retrieval, (2) effectively understand and perceive various retrieval intents within instructions. Therefore, we further propose **progressively-pruned intent learning** to iteratively train I3 phase-by-phase, incorporating two key designs: (1) *progressive structure pruning*, and (2) *drawback extrapolation-based data refinement*. Specifically, we harness the advancement of LLMs to automatically generate extensive data with instructions as the seed training dataset and then divide the total training into several phases. After each training phase, we prune the pluggable introspector into a sheared module, and simultaneously extrapolate the drawback of the current model to synthesize additional training data for refinement. This allows I3 to progressively comprehend a wide range of retrieval intents with an increasingly streamlined structure.

Benefiting from the advancements in both model and learning perspectives, I3 finally evolves into a unified and lightweight retrieval system, capable of efficiently following instructions to directly perform various retrieval tasks with diverse intents. Remarkably, in the BEIR benchmark [42], feeding with only natural language descriptions of specific intent for each task, I3 significantly outperforms competitive baselines without any task-specific fine-tuning. Overall, our main contributions are three-fold:

- We innovatively propose I3, a generic and efficient approach that endows retrieval models with a new facet of intent-introspective retrieval following instructions, enabling them to directly perform diverse tasks without specific tuning.
- We devise progressively-pruned intent learning, which incorporates progressive structure pruning and drawback extrapolation based data refinement, training I3 phase-by-phase with extensive LLM-generated instruction data.
- Experiments show that I3 achieves state-of-the-art performance on BEIR benchmark under both zero-shot retrieval and reranking scenarios, without any task-specific tuning.

2 RELATED WORK

Zero-shot Dense Retrieval. Dense retrieval [16, 17, 20, 50, 51] is widely adopted in information retrieval that demonstrates strong advantages over sparse retrieval methods [36]. However, Thakur et al. [42] has shown that dense retrieval models still struggle to generalize to out-of-domain data and do not perform well in zero-shot settings, where no task-specific signals are available. To improve zero-shot dense retrieval, some existing works [28, 47] bring in domain adaptation techniques and utilize documents from the target domain to generate corresponding pseudo queries for task-specific training. Other efforts [15, 49, 54] focus on designing more efficient unsupervised contrastive learning paradigm, while some studies [31, 32] attempt to enhance generalizability by scaling up retrieval models, among other approaches.

Nevertheless, expecting retrieval models to perform well solely based on the query in a zero-shot setup can be inherently challenging as different tasks entail different retrieval intents. To reveal the retrieval intent, for each task Promptgator [8] samples 8 query-document pairs and then generates pseudo retrieval data for training task-specific retrievers. In comparison, our method proposes a single unified retrieval model to perform various tasks following instructions without any task-specific tuning.

Another recent work TART [1], leverages instructions to flexibly describe the retrieval intents. It collects 40 existing retrieval datasets and manually annotates them with instructions. Then TART directly inputs instructions into the query encoder, simply concatenating them ahead of the queries. On this basis, it trains

a multi-task retrieval model to perform retrieval tasks following instructions. However, TART faces the following challenges: (1) the human-annotated training datasets are costly and suffer from limited diversity, with many of them sharing common retrieval intent (*e.g.*, retrieve correct answer to a question). (2) Given that retrieval models typically lack advanced in-context understanding capabilities, TART does not assure that the query encoder effectively understands the retrieval intents within the instruction. And the addition of extra instructions also disrupts the original input format of the query encoder, potentially diminishing its inherent abilities. In contrast, I_3 leverages LLMs to automatically generate instruction data comprising a wide range of retrieval intents with iterative drawback extrapolation-based refinement. It also preserves the inherent capability of retrieval models and efficiently empowers them with a new facet of intent-aware retrieval conditioned on task-specific instructions through a parameter-isolated architecture.

3 METHODOLOGY

In this section, we first give the preliminaries of dense retrieval (§3.1). Then, we introduce I3, which performs intent-introspective retrieval with a parameter-isolated architecture (§3.2). Finally, we elaborate on the progressively-pruned intent learning (§3.3). It capitalizes the advancement of LLMs to automatically generate extensive data (§3.3.1). And the LLM-generated data is then leveraged to train I3 phase-by-phase, which incorporates two key designs: progressive structure pruning (§3.3.2) and drawback extrapolation-based data refinement (§3.3.3).

3.1 Preliminaries

Dense retrieval leverages a dual-tower architecture, consisting of a query encoder E_Q and a document encoder E_D, to encode query q and document d into dense vectors. After obtaining the representations of both query and document, a similarity function (*e.g.*, dot product) $s(q, d)$ is leveraged to calculate the relevance score between them:

$$s(q, d) = \langle E_Q(q; \Theta_q), E_D(d; \Theta_d) \rangle \tag{1}$$

To train the dual encoders, given a query and the relevant (positive) document d^+, a common approach is to sample irrelevant (negative) documents d^-, which can be either in-batch negatives, BM25 negatives [16], or hard negatives mined by dense retrieval models [48]. The objective is to maximize the probability of selecting the positive document over other negative documents via contrastive learning [6, 58]:

$$p(d^+|q, d^-) = \frac{e^{s(q,d^+)}}{e^{s(q,d^+)} + \sum e^{s(q,d^-)}} \tag{2}$$

Furthermore, to realize retrieval with instructions that describe the retrieval intents in natural language, the instruction $\mathcal{I}$ is incorporated into the query encoding. Then the query encoder can be formulated as $E'_Q(q, \mathcal{I}; \Theta'_q)$ with the similarity function defined as:

$$s(q, \mathcal{I}, d) = \left\langle E'_Q(q, \mathcal{I}; \Theta'_q), E_D(d; \Theta_d) \right\rangle \tag{3}$$

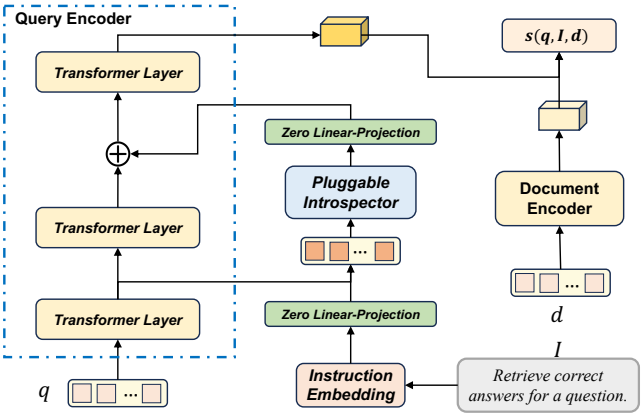

Figure 2: Parameter-isolated framework of I3, with a pluggable introspector enabling retrieval models to efficiently integrate the introspected intents for better query encoding.

3.2 Intent-Introspective Retrieval Conditioned on Instructions

In this section, we introduce the framework of I3. As a generic approach, given an arbitrary dense retrieval model, I3 fully retains its document encoder and encompasses two indispensable components within the query encoder: (1) a pluggable introspector, deeply comprehends the retrieval intent by jointly reasoning over the input query and instruction; (2) the original query encoder, which is frozen, faithfully integrates the introspected intent for better query encoding. These two components integrate seamlessly to construct a parameter-isolated architecture. This design not only preserves the inherent capability of retrieval models, but also efficiently empowers them with a new facet of intent-aware retrieval following task-specific instructions.

Specifically, we fully keep the original document encoder to eliminate the substantial time expanse of re-encoding all documents from a large corpus. For the query encoder, we first retain and fix its original parameters Θ_q to preserve the existing capabilities. Additionally, we construct a pluggable introspector (with its parameters denoted as Θ_p) to explicitly introspect for specific retrieval intents. Then the query encoding can be conceptualized as a two-step process involving ***intent introspection*** and ***intent-integrated encoding***. We first derive the query embedding from the output of the early layer within the original query encoder. The introspector subtly interprets the specific retrieval intent by jointly comprehending the derived query embedding and an additional instruction embedding (denoted as c). The introspected intent is then seamlessly re-integrated into the late layer of the original encoder via a skip connection, enhancing subsequent encoding.

To facilitate efficient intent introspection and integration, we construct two instances of "zero linear-projection" ($\mathcal{ZP}$), Θ_{zp1} and Θ_{zp2}, which involve unique fully connected layers with weights and biases initialized as zeros. Firstly, the instruction embedding $c \in \mathcal{R}^{1\times d}$ is projected to compose with the query embedding derived from the early layer's output of the original query encoder: $h_{q,c} = \mathcal{ZP}(c, \Theta_{zp1}) + h_q^{l_{early}}$, where $h_q^{l_{early}} \in \mathcal{R}^{n\times d}$ and

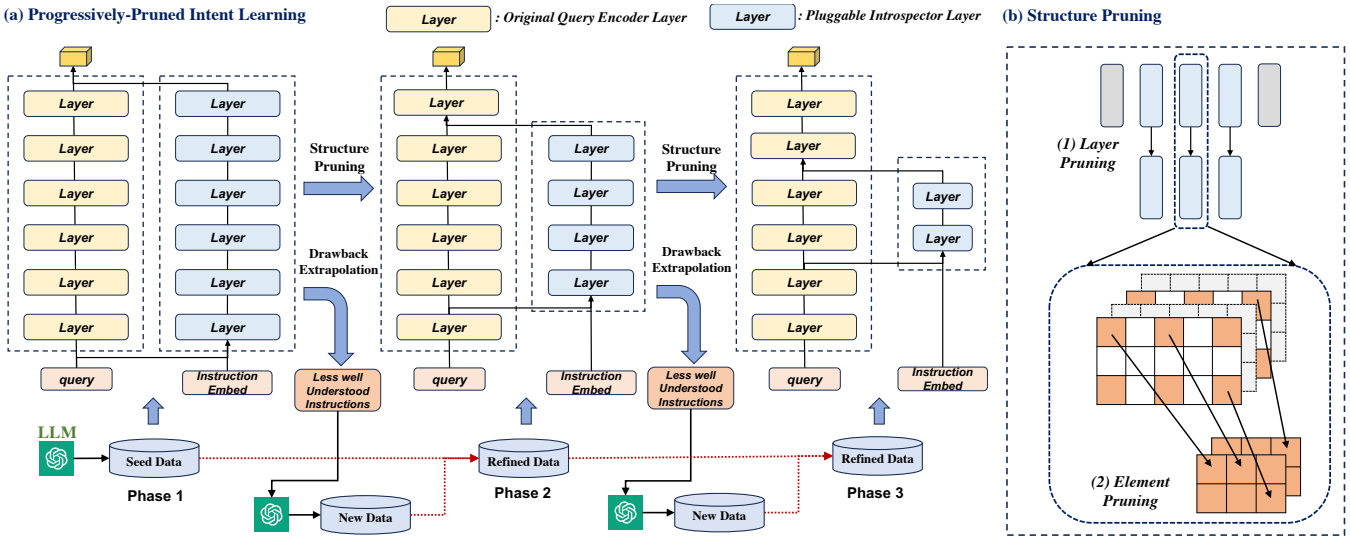

Figure 3: progressively-pruned intent learning with structure pruning and drawback extrapolation-based data refinement.

$\mathcal{ZP}(c, \Theta_{zp1})$ is added to each token of $h_q^{l_{early}}$. And then $h_{q,c}$ serves as the input of the pluggable introspector, enabling it to perceive both the input query and instruction for intent comprehension. Finally, after introspecting for the specific intents K, the intent embedding after projection is re-integrated with the hidden representations from the late layer of the query encoder, via skip connection: $h_q^{l_{late}} = h_q^{l_{late}} + \mathcal{ZP}(K, \Theta_{zp2})$, which is taken as the input to the next, *i.e.*, $(l_{late}+1)$-th layer in the original encoder. It achieves a cohesive integration between the query encoding and the introspected intents.

Training Objectives. To achieve the desired outcome of intent-introspective retrieval, the two key steps, *i.e.*, intent introspection, and intent-integrated encoding, should be well adapted to each other. In this regard, we design training objectives in two aspects for this mutual adaptation.

On the one hand, the intent introspection should effectively empower the subsequent query encoding. So we train the model to accurately match related queries and documents after explicitly introspecting for retrieval intents. For a given query $q_i$and corresponding instruction $\mathcal{I}_i^+$ from the training data, we select mismatched documents $\{d_{i,j}^-\}_{j=1}^m$ as negative samples, minimizing the negative log-likelihood of the relevant document d_i^+:

$$\mathcal{L}_1 = \sum_{i=1}^{n} -log \frac{e^{s(q_i, \mathcal{I}_i^+, d_i^+)}}{e^{s(q_i, \mathcal{I}_i^+, d_i^+)} + \sum_{j=1}^{m} e^{s(q_i, \mathcal{I}_i^+, d_{i,j}^-)}} \tag{4}$$

On the other hand, the intent-integrated encoding should also be capable of understanding the introspected intents in order to utilize the knowledge therein to the fullest extent. To achieve this, we explicitly optimize I3 to recognize what constitutes correct retrieval intents in different scenarios. For each training group of instruction, query, and relevant document, we sample some irrelevant instructions $\{\mathcal{I}_{i,j}^-\}_{j=1}^m$ as negative examples, misleading the introspector to produce incorrect retrieval intents. We optimize the negative log-likelihood of the positive instruction:

$$\mathcal{L}_2 = \sum_{i=1}^{n} -log \frac{e^{s(q_i, \mathcal{I}_i^+, d_i^+)}}{e^{s(q_i, \mathcal{I}_i^+, d_i^+)} + \sum_{j=1}^{m} e^{s(q_i, \mathcal{I}_{i,j}^-, d_i^+)}} \tag{5}$$

Finally, the training loss is represented as $\mathcal{L} = \mathcal{L}_1 + \alpha \mathcal{L}_2$, with α as the hyper-parameter. During training, we fix the parameters of the original dual-encoders (*i.e.*, Θ_q and Θ_d), only optimizing parameters of the pluggable introspector and the two instances of zero linear-projection (*i.e.*, Θ_p, Θ_{zp1} and Θ_{zp2}). This realizes efficient training while simultaneously effectively preserving the original capabilities of retrieval models.

Analysis of Harmless Intent Integration. Because the weight and bias of the fully connected layers in "zero linear-projection" are both initialized as zeros, it ensures harmless intent integration, providing better optimization than training from scratch for the introduced parameters. Specifically, in the initial training step, the instruction embedding c does not introduce any additional noise to the query embedding. Moreover, the integration of intent embedding also does not alter the hidden representations within the original query encoder, effectively maintaining the original output as if the pluggable introspector did not exist. It indicates that before any parameter optimization, integrating the pluggable introspector into the retrieval model is completely harmless to the original encoding process. With the parameters of the original dual-encoders fixed, we *ensure the preservation of previously learned knowledge, while also facilitating efficient intent introspection and integration.* Consequently, I3 effectively enhances retrieval models, controlling them to perform intent-aware retrieval following instructions.

3.3 Progressively-Pruned Intent Learning

To efficiently perform intent-introspective retrieval following instructions, the pluggable introspector should be trained to adeptly perceive various retrieval intents within instructions, while also maintaining a lightweight architecture to guarantee minimal impact on the time efficiency of query encoding. To achieve this goal, we propose **progressively-pruned intent learning**, as shown

in Figure 3. It harnesses the advancement of LLMs to automatically generate extensive data with instructions as the seed training dataset, and then divides the total training into several phases. In each phase, we leverage the LLM-generated data to train I3 with the objectives mentioned in §3.2, and also incorporate two key designs: (1) ***progressive structure pruning*** that streamlines the pluggable introspector into a more lightweight module; (2) ***drawback extrapolation-based data refinement*** that extrapolates the drawback of the current model to synthesize additional data for enhancing the initial seed training dataset.

3.3.1 ***LLM-guided Instruction Data Synthesizing.*** To effectively optimize I3 for intent introspection conditioned on instructions, first it is crucial to construct a diverse set of training data comprising various retrieval instructions. We develop a generation pipeline that utilizes an LLM to automatically synthesize a large amount of query-document pairs together with instructions as the seed training dataset.

(1) Instruction Generation. First, we prompt the LLM to generate a diverse range of retrieval instructions. To ensure comprehensive expression of different retrieval tasks, when generating instructions, we require the LLM to specify the topic (*e.g.*, scientific, legal) and organizational formats (*e.g.*, sentence, paragraph, dialogue) of the retrieved text, while also incorporating a clear definition of relevance (*i.e.*, search intent) for the retrieval task. Moreover, To further elevate the quality of instruction generation, we also incorporate some instruction examples (selected from previously generated instructions) into the prompt template as in-context samples. **(2) Query-Document Pair Generation.** For each generated instruction, we subsequently ask the LLM to write some appropriate documents and their associated queries. The generated documents and queries should align with the designated topic, organizational formats and the relevance definition outlined in the instructions. **(3) Query Self-check.** However, the query and document simultaneously generated in the preceding step may not always correctly capture the retrieval intents expressed in the instructions. To tackle this issue, we ask the LLM to verify if the query-document relationship matches the relevance criteria set by the instruction. And the LLM is further required to rewrite the query that fails to meet the relevance criteria, ensuring a cohesive association among the instruction, query, and document.

Through the above three steps, we have generated extensive seed data, encompassing a diverse range of retrieval instructions. For each instruction, we select a small subset of corresponding query-document pairs to form the validation set, while the remainder is utilized as the seed training dataset for fine-tuning I3.

3.3.2 ***Progressive Structure Pruning.*** To design the specific structure of the pluggable introspector for efficient intent introspection, a straightforward approach entails copying an extra query encoder as the introspector and training it to understand diverse instructions[55]. However, the introspector only needs to perceive specific intents over the input query and instruction, without the necessity to match the model size of the query encoder. Moreover, it is essential for the introspector to maintain a lightweight design, ensuring minimal impact on the time efficiency of query encoding.

To address these considerations, we duplicate an additional query encoder as the initial pluggable introspector, and propose ***progressive structure pruning*** that prunes the introspector into a sheared structure after each training phase. Only a subset of the weights from the larger introspector is selected to initialize the smaller version. This facilitates the transfer of knowledge learned by the larger introspector to the smaller counterpart, ensuring that the model becomes increasingly lightweight with almost no performance degradation.

Specifically, prior to the first training phase, the pluggable introspector possesses the same architecture and parameters as the original query encoder. In this setup, the introspector and the original query encoder are connected only at their input and output spaces: the input embedding of the query encoder is derived as the input for the introspector, and the introspected intent embedding is directly integrated with the output of the query encoder, producing the final query representation. In subsequent phases, before training we first perform structure pruning on the introspector from the previous phase, streamlining it into a sheared module. And the weights of the sheared introspector are initialized from the larger counterpart, resembling the process of knowledge distillation. We refer to the larger introspector before pruning as the *teacher introspector*, and the target sheared one as the *student introspector*. And structure pruning involves two aspects: layer pruning and element pruning, as shown in Figure 3.b.

(1) Layer pruning: The student introspector comprises a reduced number of transformer layers compared to its teacher counterpart. And each layer in the student introspector is initialized using a corresponding layer from the teacher. Specifically, we omit several initial and final layers while retaining the intermediate ones from the teacher introspector. Consequently, the early derivation of query embedding which serves as the input for the introspector, and the late integration of specific retrieval intents, both take place within the more intermediate layers of the original query encoder. This not only enables the query encoder to provide enhanced query embeddings for the introspector, but also allows more subsequent layers in the query encoder to effectively integrate the introspected intents for improved encoding.

(2) Element pruning: After layer pruning, we need to initialize each component within the student introspector's layers, using the corresponding larger counterpart from the teacher introspector. To streamline the model architecture, targets for element pruning may include the number of attention heads, and the hidden or intermediate dimension within the transformer layer, among others. We employ a uniform selection strategy, wherein evenly-spaced elements are selected from the teacher's tensor to initialize the student's corresponding component. For example, when leveraging weight tensor $W_t \in \mathcal{R}^{t_1 \times t_2 \times \ldots \times t_n}$ from the teacher introspector to initialize the student's weight tensor $W_s \in \mathcal{R}^{s_1 \times s_2 \times \ldots \times s_n}$ which is of the same component type ($s_i \leq e_i$), we evenly-spaced select s_i slices out of t_i for each dimension i of W_t to facilitate initialization of W_s. And previous research [52] has demonstrated that such a uniform selection strategy is likely to yield benefits of knowledge transfer from the teacher model to the student.

Besides pruning the introspector into a sheared structure, at the beginning of each training phase, we also reinitialize the weights and biases within the two "zero linear-projection" instances to zeros.

Throughout each phase of training, we consolidate and strengthen the model capability based on that inherited from the larger introspector in the preceding phase. Ultimately, we streamlined the introspector into a more lightweight module, which improves the time efficiency of query encoding while maintaining the performance with minimal degradation.

3.3.3 ***Drawback Extrapolation-Based Data Refinement.*** We then describe the drawback extrapolation-based data refinement. It extrapolates the drawbacks of the trained model from each training phase by detecting instructions that the model struggles to understand, and then specifically generates new training data to enhance the original seed dataset.

In particular, after each training phase, we evaluate the current model's performance on the synthesized validation set for each instruction. Instructions that are not fully comprehended by the model inevitably result in sub-optimal performance. We then utilize these challenging instructions as in-context samples for the LLM to generate new instructions, which often bear a resemblance to the original in-context instructions and may similarly pose comprehension challenges to the current model. Moreover, we follow the data-generation pipeline to synthesize corresponding query-document pairs that align with these new instructions, and incorporate them into the original training dataset for refinement. In subsequent training phases, these data can specifically target the identified weaknesses of the current model, thereby enhancing its ability to understand a diverse range of retrieval instructions.

4 EXPERIMENTAL SETUP

To illustrate the effectiveness of `I3`, we evaluate our approach in two settings: (1) zero-shot evaluation on retrieval scenarios and (2) zero-shot evaluation on reranking scenarios after cooperating `I3` with reranking models.

4.1 Benchmark

Our experiments are conducted on the BEIR [42] benchmark for zero-shot evaluation on retrieval and reranking scenarios. Following prior works[15, 42], we leverage Normalized Discounted Cumulative Gain 10 (nDCG@10) as the evaluation metric and use the 15 publicly available datasets in BEIR for evaluating retrieval models, including 14 out-of-domain datasets (*i.e.*, **TREC-COVID** [44], **NFCorpus** [4], **NQ** [19], **HotpotQA** [53], **FiQA-2018** [29], **ArguAna** [45], **Touché-2020** [3], **Quora** [42], **DBPedia-entity** [13], **SCIDOCS** 7, **Fever** [43], **Climate-Fever** [10], **SciFact** [46], **CQADupStack** [14]) and one in-domain dataset (*i.e.*, **MS MARCO** [2]). Besides, the evaluation instructions for all datasets, which describe the retrieval intents, are in alignment with those employed in [1]. Moreover, When evaluating reranking models, we conduct experiments across 11 of the above datasets following [8].

4.2 Baselines

Retrieval Baselines. We compare `I3` with various competitive retrieval methods, which can be categorized into four groups. *The first group* employs a sparse retrieval method known as **BM25**.

The second group trains retrievers on a few supervised datasets (*e.g.*, MS MARCO [2]) and directly transfer them to new tasks, including **Contriever** [15], **GTR** [32], **ColBERT-v2** [39], **CPT** [31], **LaPraDoR** [49], **COCO-DR** [54], and **SGPT** [30]. These baseline models have varying parameter sizes. Notably, GTR comprises 4.8B parameters, SGPT consists of 5.8B parameters, and CPT boasts an impressive 175B parameters, almost directly utilizing LLMs for retrieval tasks.

The third group of models train a task-specific retriever for each downstream task with pseudo generated training data, including **GenQ** [42], **GPL** [47], **Promptgator** [8]. GenQ and GPL utilize a specifically trained T5 for data generation. And Promptgator is a few-shot LLM-enhanced method that prompts LLMs to synthesize training data.

And *the final group* includes **TART-dual** [1] and **InstructOR** [41]. TART-dual collects a large number of supervised datasets and augments them with human-annotated instructions. The collected datasets, known as *Berri*, are then employed to train a single retriever to solve different tasks with instructions. And InstructOR adopts a similar approach, focusing on instruction-based retrieval.

Reranking Baselines. In contexts where speed is not critical, the reranking model with a cross-encoder is often used to compute the query-document relevance by jointly encoding them with cross-attention. Under the reranking settings, a retrieval model is first utilized to retrieve the Top-K documents, followed by the application of a reranking model to reorder these retrieved documents. In our experimental setup, we first leverage `I3` to retrieve the Top-K documents (where K=100) and then employ a competitive reranking model, monoT5 (3B) [33], to reorder these retrieved documents.

We compare with the following state-of-the-art Retriever+Reranker combinations: **UPR (3B)** [38] that initially retrieves 1000 documents with Contriever, **Contriever+CE** that initially retrieves 100 documents with Contriever, **BM25+monoT5** (3B) [33, 37] that initially retrieves 100 or 1000 documents with BM25, **COCO-DR+monoT5** (3B) [33, 37] that initially retrieves 100 documents with COCO-DR, **Promptgator++** (zero-shot version & few-shot version) [8] that initially retrieves 200 documents with Promptgator, **TART-full** (T0-3B version & Flant5-XL version) [1] that initially retrieves 100 documents with Contriever.

4.3 Implementation Details

To automatically generate retrieval data with diverse instructions, we leverage ChatGPT [34] (OpenAI `gpt-3.5-turbo`) as the LLM. We set the temperature to 1 and generate approximately 100 instructions with different retrieval intents. Concurrently, we synthesize about 140K query-document pairs in total, including 100K pairs in the seed training dataset and 20K pairs created during data refinement. Furthermore, we rigorously eliminate the generated instructions that exhibit similar retrieval intents as downstream tasks, thereby ***preventing any overlap or similarities between the generated training data and the test data.***

With LLM-generated training data, we leverage COCO-DR$_{\text{Large}}$ [54], a dual-encoder consisting of 335M parameters, as the backbone model to implement `I3`. Specifically, COCO-DR$_{\text{Large}}$ has the same architecture as BERT$_{\text{Large}}$ [9] and takes the [CLS] pooling of the top encoder layer as the query/document embeddings. We retain the original dual-encoders and duplicate an extra query encoder as the pluggable introspector. And the instruction embedding c is

Table 1: nDCG@10 on the BEIR Benchmark. We compare I3 with other retriever models. Avg CPT Sub is the average performance on 11 BEIR tasks used in [31]. Avg TART Sub is the average performance on 9 BEIR tasks used in [1].

Datasets	BM25	Contriever	GTR	ColBERTv2	CPT	LaPraDoR	COCO-DR	SGPT	GenQ	GPL	Promptgator	TART-dual	InstructOR	I3
MS MARCO	22.8	40.7	**44.2**	—	—	36.6	42.4	39.9	40.8	—	—	—	41.6	41.8
TREC-COVID	65.6	59.6	50.1	73.8	64.9	77.9	80.4	**87.3**	61.9	70.0	75.6	62.6	71.4	81.6
NFCorpus	32.5	32.8	34.2	33.8	**40.7**	34.7	35.4	36.2	31.9	34.5	33.4	33.7	36.0	37.1
NQ	32.9	49.8	56.8	56.2	—	47.9	54.7	52.4	35.8	48.3	—	—	57.3	**57.4**
HotpotQA	60.3	63.8	59.9	66.7	**68.8**	66.6	64.1	59.3	53.4	58.2	61.4	—	55.9	63.3
FiQA-2018	23.6	32.9	46.7	35.6	**51.2**	34.3	32.9	37.2	30.8	34.4	46.2	33.7	47.0	35.7
ArguAna	41.4	44.6	54.0	46.3	43.5	50.8	51.5	51.4	49.3	55.7	59.4	48.9	55.7	**59.8**
Touché-2020	**36.7**	23.0	25.6	26.3	29.1	33.3	26.3	25.4	18.2	25.5	34.5	20.1	23.4	23.7
Quora	78.9	86.5	89.2	85.2	63.8	87.5	87.2	84.6	83.0	83.6	—	—	88.9	**89.3**
DBPedia-entity	31.3	41.3	40.8	**44.6**	43.2	39.1	40.7	39.9	32.8	38.4	38.0	41.5	40.2	41.8
SCIDOCS	15.8	16.5	16.1	15.4	—	18.4	17.8	19.7	14.3	16.9	18.4	14.2	17.4	**19.9**
Fever	75.3	75.8	74.0	78.5	77.5	76.3	79.3	78.3	66.9	75.9	77.0	—	70.0	**80.8**
Climate-Fever	21.3	23.7	26.7	17.6	22.3	26.1	24.7	30.5	17.5	23.5	16.8	13.8	26.5	**31.1**
SciFact	66.5	67.7	66.2	69.3	75.4	68.7	72.2	74.7	64.4	67.4	65.0	69.0	64.6	**79.9**
CQADupStack	29.9	34.5	39.9	—	—	29.0	39.3	38.1	34.7	35.7	—	—	**43.0**	40.0
Avg CPT(TART) Sub	48.5	50.2	51.6	52.5	52.8	54.1	54.1	55.0	46.4	51.6	—(43.0)	—(37.4)	52.7	**56.7(45.6)**
Avg	43.7	46.6	48.6	—	—	49.3	50.5	51.1	42.5	47.7	—	—	49.8	**52.9**

Table 2: We cooperate I3 with monoT5 [33] and compare with other reranking baselines on 11 datasets in BEIR.

Methods	K	arg	touché	covid	nfc	hotpot	dbp	climate	fever	scifact	scidocs	fiqa	Avg TART Sub	Avg
UPR (3B)	1000	50.3	21.3	60.4	33.3	72.2	33.8	9.5	57.3	69.6	17.3	45.0	37.8	42.7
Contriever+CE	100	41.3	29.8	70.1	34.4	—	47.1	25.8	—	69.2	17.1	36.7	41.3	—
BM25+monoT5 (3B)	100	32.3	21.1	82.0	39.4	73.1	44.1	27.0	83.9	76.2	19.4	46.2	43.0	49.5
BM25+monoT5 (3B)	1000	38.0	30.0	79.5	38.4	**75.9**	47.8	28.0	85.0	77.7	19.7	51.4	45.6	51.9
COCO-DR+monoT5 (3B)	100	40.6	28.4	85.9	39.8	71.4	47.7	29.1	84.6	76.4	20.0	48.5	46.3	52.0
Promptgator++ (zero-shot)	200	52.1	27.8	76.0	36.0	71.2	41.3	22.6	83.8	73.2	19.1	45.9	43.8	49.9
Promptgator++ (few-shot)	200	**63.0**	38.1	76.2	37.0	73.6	43.4	20.3	86.6	73.1	20.1	49.4	46.7	52.8
TART-full (T0-3B)	100	49.8	**31.2**	71.7	34.0	—	45.1	30.0	—	75.8	17.5	42.2	44.1	—
TART-full (FlanT5-XL)	100	51.5	24.9	72.8	33.4	—	46.8	35.4	—	77.7	18.7	41.8	44.8	—
I3+monoT5(3B)	100	41.4	29.0	**86.3**	**40.0**	72.8	**48.5**	**35.6**	**86.7**	**77.9**	**20.4**	**51.4**	**47.8**	**53.6**

pre-encoded by the original query encoder. Moreover, we encompass three phases within progressively-pruned intent learning and perform structure pruning after each of the first two phases. In the final version of the introspector, the number of layers is reduced from the original 24 to 12, the hidden dimension decreases from 1024 to 768, the intermediate dimension decreases from 4096 to 3072, and the number of attention heads decreases from 16 to 12.

During training, we set the batch size as 64, with the maximum sequence length as 256, the learning rate as 5e-6, and α as 0.5. When conducting contrastive learning, we randomly sample 4 mismatched instructions as negative examples in $\mathcal{L}_2$. Moreover, in each phase we leverage the model from the previous phase (COCO-DR in the initial phase) to retrieve documents with high similarity scores yet unrelated to the query from generated documents as hard negatives, together with in-batch negatives for $\mathcal{L}_1$. Furthermore, when evaluating on BEIR, we follow the same hyperparameters in COCO-DR [54] to ensure a fair comparison.

5 EXPERIMENTAL RESULTS

5.1 Main Results on Retrieval Scenarios

Table 1 lists the zero-shot evaluation results of retrieval models on BEIR. The results for baseline methods are derived from their respective papers. As some baselines employ datasets from BEIR for training, their downstream test excludes these datasets to ensure a fair comparison, leading to results missing for some baselines on specific datasets. When compared with these baselines, we compute the average performance for I3 exclusively on the datasets they test. We have the following observations from the experimental results.

First of all, I3 outperforms all previous retrieval models on the average nDCG@10 metric of BEIR tasks. It shows significant improvements over its backbone model, *i.e.*, COCO-DR, improving the average nDCG@10 by 2.3 points and achieving superior performance on 12 of 14 out-of-domain tasks. The significantly higher zero-shot performance of our model demonstrates the stronger generalizability of I3 across various retrieval tasks.

Second, while some approaches (*e.g.*, Promptgator) leverage LLMs to generate pseudo training data and train task-specific models for each downstream task, our model achieves superior transferability across diverse retrieval intents of tasks. For example, I3 yields 2.6 point nDCG improvement over the few-shot method Promptgator. This indicates that our I3 can efficiently and effectively transfer to different retrieval tasks using only instructions without any task-specific training.

Third, compared with models that have significantly more parameters,*e.g.*, SGPT (**5.8B** parameters), CPT (**175B**), our method still achieves stronger average performance with much fewer parameters (**445M**). It highlights smaller models can still achieve remarkable effectiveness when designed with an ingenious architecture to well understand the retrieval intents.

Table 3: Results of ablation study to illustrate the effect of individual components.

	0 backbone COCO-DR	**1** w/o *intro-spection*	**2** w/o *instr*	**3** w/o *progressive*	**4** w/o *refine*	**5** w/o *pruning*	**I3**
MS MARCO	42.4	37.9	40.3	40.9	41.7	41.8	41.8
TREC-COVID	80.4	77.9	76.1	80.3	80.8	**82.1**	81.6
NFCorpus	35.4	34.1	36.2	36.3	36.4	36.5	**37.1**
NQ	54.7	51.0	55.3	56.0	56.3	57.2	**57.4**
HotpotQA	64.1	58.6	61.0	62.9	63.2	**64.5**	63.3
FiQA-2018	32.9	31.8	34.1	34.8	35.2	35.3	**35.7**
ArguAna	51.5	56.4	55.9	57.4	56.4	58.8	**59.8**
Touché-2020	**26.3**	25.5	22.8	22.9	24.3	24.5	23.7
Quora	87.2	86.7	87.6	87.7	88.2	88.6	**89.3**
DBPedia-entity	40.7	32.3	41.2	41.4	41.0	**42.1**	41.8
SCIDOCS	17.8	16.7	18.2	18.8	19.2	19.6	**19.9**
Fever	79.3	77.7	79.7	80.1	80.4	**81.3**	80.8
Climate-Fever	30.4	25.3	30.9	30.6	30.9	**32.0**	31.1
SciFact	72.2	72.0	77.4	78.5	78.8	79.0	**79.9**
CQADupStack	39.3	38.3	39.5	39.1	39.3	**40.1**	40.0
Avg	50.5	48.9	51.1	51.9	52.2	**53.0**	52.9

Table 4: The average retrieval time efficiency across each dataset within BEIR.

Method	**COCO-DR**	**I3 w/o pruning**	**I3**
#Query per second	1600	700	**1150**
#Document per second	500	500	500
Query encoding time	2.3s	5.2s	**3.1s**
Document encoding time	4514s	4514s	4514s
Retrieval Latency	7.5s	7.5s	7.5s
Total Time	4523.8s	4526.7s	**4524.6s**

Finally, I3 leverages a large margin improvement compared with TART-dual (45.6 v.s. 37.4 on average) and InstructOR (52.9 v.s. 49.8 on average), which leverage human-annotated training data for instruction-based retrieval. We argue that it can be attributed to two key factors. First, the parameter-isolated architecture of I3 better preserves the original capability of the retrieval model, while also flexibly unleashing its ability to conduct intent-introspective retrieval following instruction. Second, progressively-pruned intent learning explicitly optimizes the transferability of I3 based on extensive automatically LLM-generated retrieval data with diverse search intents.

5.2 Main Results on Reranking Scenarios

To further illustrate that our method can be applied to reranking scenarios and yield performance improvements, we cooperate I3 with a competitive reranking model, monoT5 (3B) [33] and conduct experiments on BEIR. As shown in **Table 2**, the combination of I3+monoT5 has achieved the new state-of-the-art reranking performance in BEIR. Compared to BM25+monoT5 and COCO-DR+monoT5, our approach involves a mere substitution of the retrieval model, yet results in a marked enhancement in performance. And the combination of I3+monoT5 is also notably superior to zero-shot reranking models like TART-full [1] and Promptgator++ [8]. With a zero-shot setup, it even outperforms few-shot Promptgator++ which achieves previous SOTA performance, exhibiting average gains of 0.8 nDCG@10 points.

Furthermore, for certain baseline methods like BM25+monoT5 and Promptgator++, we adhere to the experimental settings in their original papers, and a value larger than 100 is selected for K as the number of documents initially retrieved. Typically, A larger

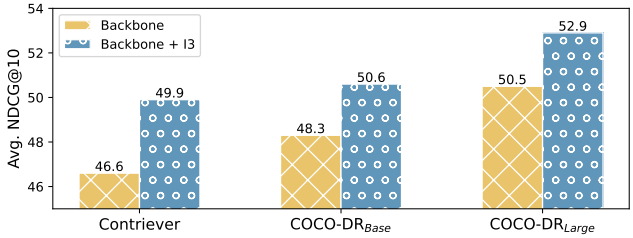

Figure 4: Average nDCG@10 of I3 incorporating different backbones on the BEIR benchmark.

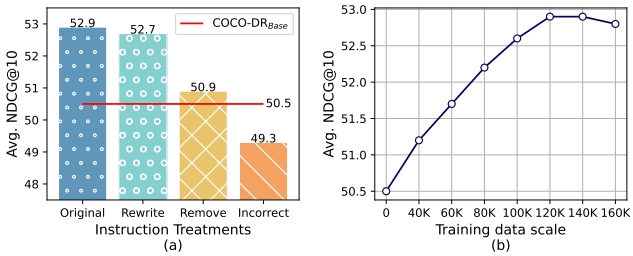

Figure 5: (a): Performance on BEIR with different instruction treatments during testing. (b): Performance of I3 with different training data scales.

K implies a greater reliance on the capabilities of more powerful reranking models that utilize cross-encoders. It tends to improve the reranking performance but also increases the time cost for reranking (the time cost is proportional to the value of K), as reranking is generally more time-consuming compared to retrieval. In contrast to baselines with larger K values, our method still achieves superior performance under more stringent conditions and additionally reduces the time cost of reranking.

5.3 In-Depth Analysis

Effect of Individual Components. We conduct the ablation study to illustrate the effect of each component in **Table 3**. Specifically, we train the following ablation models: (1) ***w/o introspection***: we remove the pluggable introspector and directly tune the COCO-DR$_{\text{Large}}$ backbone with our generated data, following TART-dual. (2) ***w/o instr***: we do not provide any instructions to I3 during both training and evaluation. (3) ***w/o progressive***: Before training, we prune the original query encoder directly to the desired sheared structure to serve as the pluggable introspector, no longer conducting progressive structure pruning during subsequent training phases. (5) ***w/o refine***: We do not perform drawback extrapolation-based data refinement, with the synthesized seed dataset as the only training data. (4) ***w/o pruning***: We do not perform structure pruning and the pluggable introspector always maintains the same structure as the original query encoder.

The result of **Column 1** in Table 3 indicates that directly instruction-tuning the backbone model actually undermines its inherent capabilities, resulting in a decline in performance. This observation underscores the crucial role of the I3 architecture in enhancing zero-shot retrieval. The result of **Column 2** emphasizes the necessity of fine-tuning I3 with instruction-based data. Besides, the result of **Column 3** confirms the superiority of iterative training with progressive structure pruning, which ensures that the knowledge

Table 5: Performance on BEIR with different training data (ChatGPT-Generated, LLaMA2-Generated, from TART-dual).

Datasets	Backbone COCO-DR	I3 TART-Data	I3 LLaMA2-Data	I3 ChatGPT-Data
MS MARCO	**42.4**	—	41.5	41.8
TREC-COVID	80.4	78.2	80.5	**81.6**
NFCorpus	35.4	36.0	36.2	**37.1**
NQ	54.7	—	55.9	**57.4**
HotpotQA	64.1	—	**64.3**	63.3
FiQA-2018	32.9	34.6	34.4	**35.7**
ArguAna	51.5	56.1	56.7	**59.8**
Touché-2020	26.3	23.9	**26.4**	23.7
Quora	87.2	—	87.5	**89.3**
DBPedia-entity	40.7	41.1	41.2	**41.8**
SCIDOCS	17.8	18.7	18.9	**19.9**
Fever	79.3	—	80.2	**80.8**
Climate-Fever	30.4	29.6	30.8	**31.1**
SciFact	72.2	76.7	77.8	**79.9**
CQADupStack	39.3	—	39.5	**40.0**
Avg TART Sub	43.1	43.9	44.8	**45.6**
Avg	50.5	—	52.2	**52.9**

learned by larger models is gradually and effectively transferred to smaller models. Moreover, **Column 4** suggests that the process of data refinement is geared towards identifying and mitigating the drawbacks of the model, thus further boosting the performance. Lastly, the result of **Column 5** demonstrates that structure pruning effectively renders our model more lightweight, while simultaneously leading to almost no performance degradation compared to the un-pruned model.

Retrieval Time Efficiency. **Table 4** presents the average retrieval time efficiency across each dataset within BEIR, specifically comparing I3, COCO-DR, and the un-pruned version of I3 (w/o pruning). The results reveal that the integration of the pluggable introspector, without pruning, significantly slows down the speed of query encoding by over 50%. However, the incorporation of a pruned lightweight introspector only slightly decelerates the query encoding. Furthermore, the extra time incurred by the introspector has a minimal impact on the overall time, which is dominated by document encoding and the retrieval latency.

Incorporating with Different Backbones. As a generic approach, I3 can be seamlessly integrated into different retrieval models with dual-tower architecture. Besides COCO-DR_{Large}, we also leverage COCO-DR_{Base} and Contriever as the backbone to train I3 in the same way. As shown in **Figure 4**, across all three backbone models, our method significantly enhances the zero-shot results, enabling them to achieve stronger performance. Notably, the performance of COCO-DR_{Base}, after instruction tuning within our framework, even surpasses the original COCO-DR_{Large}. This underscores the universality of our proposed approach.

Impact of Instructions. To analyze the effectiveness of instructions, we employ different treatments for instructions during downstream zero-shot evaluation, *i.e.*, **Rewrite Instr**, **Remove Instr**, and **Incorrect Instr**, which involve rephrasing the instructions without altering their original intents, entirely omitting the instructions, and providing misleading incorrect instructions, respectively. As shown in **Figure 5.a**, rewriting test instructions without altering their original meaning has minimal impact on performance. This demonstrates that *our model is robust to various instructions*. However, the performance noticeably drops when no instructions are provided during evaluation, and providing instructions with incorrect retrieval intents leads to a more substantial decline in performance. This highlights *the pivotal importance of instructions in intent-introspective retrieval.*

Analysis of Training Data Sources. Besides generating training data with ChatGPT, we also leverage ***LLaMA2*** to synthesize an equivalent amount of training data. Additionally, we also experiment with ***Berri***, the manually-collected data in Tart-dual, as the training data to tune I3 (BEIR datasets contained in *Berri* are excluded in zero-shot evaluation). As shown in **Table 5**, replacing ChatGPT with LLaMA results in a marginal decrease in performance compared to the original I3 due to reduced data quality, but it still surpasses COCO-DR and the model trained on Berri. The results show that our method does not simply rely on ChatGPT and highlight the efficacy of our data generation strategy.

Impact of Training Data Scales. To investigate the impact of the training data scale, we control the total amount of training data by altering the size of the seed training dataset. As shown in **Figure 5.b**, the performance keeps increasing when the total number of training data expands from 40K to 120K. While further increasing the data size from 120K to 160K shows minimal impact on performance. These observations underscore that our iterative training paradigm is data-efficient.

6 CONCLUSION AND FUTURE WORK

In this paper, we present I3, a generic and efficient approach capable of controlling retrieval models to introspect for specific retrieval intent and directly perform varied retrieval tasks without task-specific tuning. Integrating a pluggable introspector in a parameter-isolated manner, I3 effectively preserves the original capability of the retrieval model, meanwhile efficiently empowering it with a new facet of intent-introspective retrieval conditioned on instructions. Furthermore, we also innovatively propose progressively-pruned intent learning, which incorporates progressive structure pruning and drawback extrapolation-based data refinement, training I3 phase-by-phase with extensive LLM-generated instruction data. Extensive experiments demonstrate the superior zero-shot generalizability of I3 on diverse retrieval tasks under both retrieval and reranking scenarios.

In the future, we aim to extend the idea of instruction-based task-intent introspection across various fields to enhance the capabilities of different models [22, 25, 35, 57, 59]. Furthermore, we hope the technology of intent-introspective retrieval technology can emerge as an important tool for augmenting a range of downstream tasks, such as LLM pre-training [21, 56], video comprehension [23, 24, 26, 27], and anomaly detection [5, 11].

ACKNOWLEDGEMENTS

This work was supported by the NSFC (No. 62272411), Key Research and Development Projects in Zhejiang Province (No. 2024C01106), the National Key Research and Development Project of China (2018AAA0101900), Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, and Ant Group.

REFERENCES

[1] Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware Retrieval with Instructions. *arXiv preprint arXiv:2211.09260* (2022).

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[3] Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2020. Overview of Touché 2020: Argument Retrieval: Extended Abstract. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings* (Thessaloniki, Greece). Springer-Verlag, Berlin, Heidelberg, 384–395. https://doi.org/10.1007/978-3-030-58219-7_26

[4] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. In *Advances in Information Retrieval*, Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello (Eds.). Springer International Publishing, Cham, 716–722.

[5] Dong Chen, Kaihang Pan, Guoming Wang, Yueting Zhuang, and Siliang Tang. 2023. Improving vision anomaly detection with the guidance of language modality. *arXiv preprint arXiv:2310.02821* (2023).

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709* (2020).

[7] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2270–2282. https://doi.org/10.18653/v1/2020.acl-main.207

[8] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=gmL46YMpu2J

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *arXiv preprint arXiv:2012.00614* (2020).

[11] Jianhao Guo, Siliang Tang, Juncheng Li, Kaihang Pan, and Lingfei Wu. 2023. RustGraph: Robust Anomaly Detection in Dynamic Graphs by Jointly Learning Structural-Temporal Dependency. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[12] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 368, 10 pages.

[13] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity v2: A Test Collection for Entity Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 1265–1268. https://doi.org/10.1145/3077136.3080751

[14] Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. CQADupStack: A Benchmark Data Set for Community Question-Answering Research. In *Proceedings of the 20th Australasian Document Computing Symposium* (Parramatta, NSW, Australia) *(ADCS '15)*. Association for Computing Machinery, New York, NY, USA, Article 3, 8 pages. https://doi.org/10.1145/2838931.2838934

[15] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised Dense Information Retrieval with Contrastive Learning. https://doi.org/10.48550/ARXIV.2112.09118

[16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[17] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3397271.3401075

[18] Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *ACM computing surveys (CSUR)* 32, 2 (2000), 144–173.

[19] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur 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. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466. https://doi.org/10.1162/tacl_a_00276

[20] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6086–6096. https://doi.org/10.18653/v1/P19-1612

[21] Juncheng Li, Xin He, Longhui Wei, Long Qian, Linchao Zhu, Lingxi Xie, Yueting Zhuang, Qi Tian, and Siliang Tang. 2022. Fine-grained semantically aligned vision-language pre-training. *Advances in neural information processing systems* 35 (2022), 7290–7303.

[22] Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. 2023. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. In *The Twelfth International Conference on Learning Representations*.

[23] Juncheng Li, Siliang Tang, Linchao Zhu, Haochen Shi, Xuanwen Huang, Fei Wu, Yi Yang, and Yueting Zhuang. 2021. Adaptive hierarchical graph reasoning with semantic coherence for video-and-language inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1867–1877.

[24] Juncheng Li, Siliang Tang, Linchao Zhu, Wenqiao Zhang, Yi Yang, Tat-Seng Chua, and Fei Wu. 2023. Variational Cross-Graph Reasoning and Adaptive Structured Semantics Learning for Compositional Temporal Grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[25] Juncheng Li, Xin Wang, Siliang Tang, Haizhou Shi, Fei Wu, Yueting Zhuang, and William Yang Wang. 2020. Unsupervised reinforcement learning of transferable meta-skills for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12123–12132.

[26] Juncheng Li, Junlin Xie, Long Qian, Linchao Zhu, Siliang Tang, Fei Wu, Yi Yang, Yueting Zhuang, and Xin Eric Wang. 2022. Compositional temporal grounding with structured variational cross-graph correspondence learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3032–3041.

[27] Juncheng Li, Junlin Xie, Linchao Zhu, Long Qian, Siliang Tang, Wenqiao Zhang, Haochen Shi, Shengyu Zhang, Longhui Wei, Qi Tian, et al. 2022. Dilated context integrated network with cross-modal consensus for temporal emotion localization in videos. In *Proceedings of the 30th ACM International Conference on Multimedia*. 5083–5092.

[28] Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 1075–1088. https://doi.org/10.18653/v1/2021.eacl-main.92

[29] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 1941–1942. https://doi.org/10.1145/3184558.3192301

[30] Niklas Muennighoff. 2022. SGPT: GPT Sentence Embeddings for Semantic Search. *arXiv preprint arXiv:2202.08904* (2022).

[31] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005* (2022).

[32] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large Dual Encoders Are Generalizable Retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 9844–9855. https://aclanthology.org/2022.emnlp-main.669

[33] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[35] Kaihang Pan, Juncheng Li, Hongye Song, Jun Lin, Xiaozhong Liu, and Siliang Tang. 2023. Self-supervised Meta-Prompt Learning with Meta-Gradient Regularization for Few-shot Generalization. *arXiv preprint arXiv:2303.12314* (2023).

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

[37] Guilherme Moraes Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. No parameter left behind: How distillation and model size affect zero-shot retrieval. *arXiv preprint arXiv:2206.02873* (2022).

[38] Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving Passage Retrieval with Zero-Shot Question Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 3781–3797. https://doi.org/10.18653/v1/2022.emnlp-main.249

[39] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 3715–3734. https://doi.org/10.18653/v1/2022.naacl-main.272

[40] Bruce R Schatz. 1997. Information retrieval in digital libraries: Bringing search to the net. *Science* 275, 5298 (1997), 327–334.

[41] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1102–1121. https://doi.org/10.18653/v1/2023.findings-acl.71

[42] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung (Eds.), Vol. 1. Curran. https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/65b9eea6e1cc6bb9f0cd2a47751a186f-Paper-round2.pdf

[43] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 809–819. https://doi.org/10.18653/v1/N18-1074

[44] Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection. *SIGIR Forum* 54, 1, Article 1 (feb 2021), 12 pages. https://doi.org/10.1145/3451964.3451965

[45] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the Best Counterargument without Prior Topic Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 241–251. https://doi.org/10.18653/v1/P18-1023

[46] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609

[47] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2345–2360. https://doi.org/10.18653/v1/2022.naacl-main.168

[48] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*. https://openreview.net/forum?id=zeFrfgyZln

[49] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. LaPraDoR: Unsupervised Pretrained Dense Retriever for Zero-Shot Text Retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 3557–3569. https://doi.org/10.18653/v1/2022.findings-acl.281

[50] Shicheng Xu, Liang Pang, Huawei Shen, and Xueqi Cheng. 2022. MatchPrompt: Improving Multi-task Generalization Ability for Neural Text Matching via Prompt Learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 2290–2300. https://doi.org/10.1145/3511808.3557388

[51] Shicheng Xu, Liang Pang, Huawei Shen, and Xueqi Cheng. 2023. NIR-Prompt: A Multi-task Generalized Neural Information Retrieval Training Framework. *ACM Trans. Inf. Syst.* 42, 2, Article 57 (nov 2023), 32 pages. https://doi.org/10.1145/3626092

[52] Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. 2023. Initializing Models with Larger Ones. *arXiv preprint arXiv:2311.18823* (2023).

[53] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2369–2380. https://doi.org/10.18653/v1/D18-1259

[54] Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. COCO-DR: Combating the Distribution Shift in Zero-Shot Dense Retrieval with Contrastive and Distributionally Robust Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 1462–1479. https://aclanthology.org/2022.emnlp-main.95

[55] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023).

[56] Qi Zhang, Yiming Zhang, Haobo Wang, and Junbo Zhao. 2024. RECOST: External Knowledge Guided Data-efficient Instruction Tuning. *arXiv preprint arXiv:2402.17355* (2024).

[57] Yun Zhu, Jianhao Guo, and Siliang Tang. 2023. Sgl-pt: A strong graph learner with graph prompt tuning. *arXiv preprint arXiv:2302.12449* (2023).

[58] Yun Zhu, Jianhao Guo, Fei Wu, and Siliang Tang. 2022. RoSA: A Robust Self-Aligned Framework for Node-Node Graph Contrastive Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3795–3801. https://doi.org/10.24963/ijcai.2022/527 Main Track.

[59] Yun Zhu, Yaoke Wang, Haizhou Shi, Zhenshuo Zhang, and Siliang Tang. 2023. GraphControl: Adding Conditional Control to Universal Graph Pre-trained Models for Graph Domain Transfer Learning. *arXiv preprint arXiv:2310.07365* (2023).

APPENDIX

A DETAILED PROMPT FOR DATA SYNTHESIZING

The LLM-guided instruction data synthesizing involves three steps as shown in Figure 6: 1) instruction generation; 2) query-document pair generation; 3) query self-check (refinement).

Instruction Generation. We first utilize the following prompt template to generate diverse instructions covering a wide range of retrieval intents:

> The task is to generate some diverse instructions that reveal search intents for retrieval tasks. Here are some generation requirements:
> (1) Each generated instruction must explicitly outline the retrieval intent, which describes how the retrieved text relates to the query, such as whether the text answers a question in the query.
> (2) Within each generated instruction, you must specify the expected source or topic of retrieved text, such as Wikipedia, scientific, or legal.
> (3) Within each generated instruction, you should also define the text block to retrieve, such as a document or a paragraph.
> Here are specific examples of instructions:
> `<GIVE INSTRUCTION EXAMPLES HERE>`
> Now please directly generate instructions without writing any other explanations:

Query-Document Pair Generation. For each generated instruction, we first extract the `TEXT TYPE` of the retrieved text, including the topic (*e.g.*, scientific, legal) and the organizational formats (*e.g.*, document, paragraph). Subsequently, we employ the following prompt template to generate query-document pairs for each instruction:

> I will give you an instruction that describes how the retrieved text relates to the query in a retrieval task. The task is to generate a pair of query and the retrieved text based on the given instruction. Here are some generation requirements:
> (1) The retrieved text you generate should be `<TEXT TYPE>` according to the instruction.
> (2) The connection between your generated query and the retrieved text should correspond to the relationship specified in the instruction.
> Here are examples of generating query and retrieved text with instructions:
> `<EXAMPLES OF QUERY-DOCUMENT PAIR GENERATION>`
>
> Please directly generate the query and the retrieved text without writing any other explanations, based on the following instructions:
> `<INSTRUCTION>`

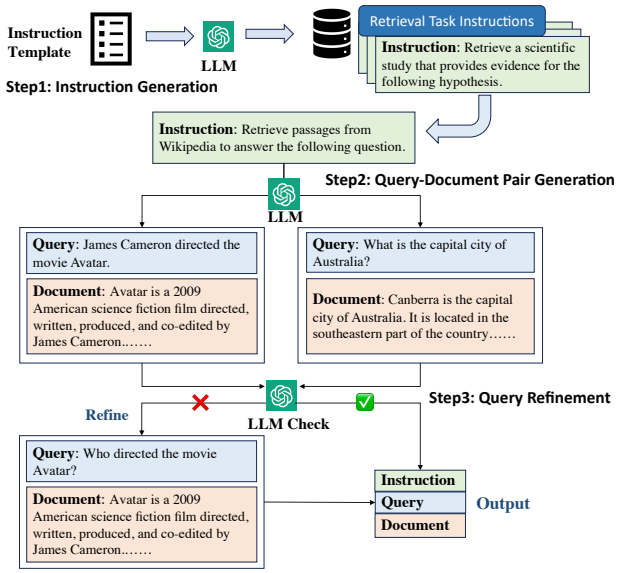

Figure 6: LLM-guided instruction data synthesizing.

Query Self-check. We observed that for certain generated instructions, some of the subsequent generated queries and documents do not always correctly capture the retrieval intents expressed within those instructions. To address this challenge, we finally design the following prompt templates to refine or regenerate these misaligned queries:

> I will give you an instruction that describes how the retrieved text relates to the query in a retrieval task. And then you are provided with the query and the retrieved text. You should assess the query for the following criteria:
> (1) Does the given query adhere to the query format specified in the instruction?
> (2) Does the relationship between the query and the retrieved text align with the relationship specified in the instruction?
> If the existing query satisfies the aforementioned criteria, directly output the existing query without any alterations. Conversely, if the query falls short of meeting the criteria, please revise or rewrite the existing query to make it satisfy the criteria and then directly output the revised query. DO NOT provide any explanations.
> Here is an example:
> `<EXAMPLES OF QUERY REFINEMENT>`
>
> Now I will give you the instruction, the retrieved text, and the query:
> The instruction: `<INSTRUCTION>`
> The retrieved text: `<RETRIEVED TEXT>`
> The query: `<QUERY>`

Table 6: Evaluation instructions for BEIR benchmark.

Dataset	Instruction
MS MARCO	`I want to know the answer to the question. Can you find good evidence on the web?`
TREC-COVID	`Retrieve Scientific paper paragraph to answer this question.`
NFCorpus	`Retrieve Scientific paper paragraph to answer this question.`
NQ	`retrieve passages from Wikipedia that provides answers to the following question.`
HotpotQA	`Find a paragraph that provides useful information to answer this question.`
FiQA-2018	`Find financial web article paragraph to answer.`
ArguAna	`Retrieve an argument that counter argues the following paragraph.`
Touché-2020	`You have to retrieve an argument to this debate question.`
Quora	`Check if a Quora question is duplicated with this question.`
DBPedia-entity	`Retrieve a Wikipedia introduction paragraph of the following entity.`
SCIDOCS	`Find scientific paper titles that are related to the following.`
Fever	`Retrieve a Wikipedia paragraph to verify this claim.`
Climate-Fever	`Retrieve a Wikipedia paragraph to verify this claim.`
SciFact	`Retrieve a scientific paper sentence to verify if the following claim is true.`
CQADupStack	`I want to identify duplicate questions asked in community question answering forums.`

B EVALUATION INSTRUCTIONS

We conduct all experiments on the BEIR, a widely used benchmark for zero-shot evaluation of information retrieval models. We follow the evaluation instructions in Asai et al. [1] to provide an extra instruction for each dataset to reflect specific retrieval intents. The detailed evaluation instructions can be found in Table 6.