Task-aware Retrieval with Instructions

Akari Asai[†‡], **Timo Schick**[†], **Patrick Lewis**[†], **Xilun Chen**[†], **Gautier Izacard**[†♠], **Sebastian Riedel**[♣], **Hannaneh Hajishirzi**[‡♡], **Wen-tau Yih**[†]
[†]Meta AI [‡]University of Washington [♠] ENS, PSL University & Inria
[♡]Allen Institute for AI [♣]University College London

Abstract

We study the problem of *retrieval with instructions*, where users provide explicit descriptions of their intent along with their queries to guide a retrieval system. Our solution is a general-purpose task-aware retrieval system, trained using multi-task instruction tuning and can follow human-written instructions to find relevant documents to a given query. We introduce the first large-scale collection of 37 retrieval datasets with instructions, BERRI, and present TART, a single multi-task retrieval system trained on BERRI with instructions that can adapt to a new task without any parameter updates. TART advances the state of the art on two zero-shot retrieval benchmarks, BEIR and LOTTE, outperforming models up to three times larger. We further introduce a new evaluation setup, $\mathbb{X}^2$-Retrieval, to better reflect real-world scenarios in which diverse domains and tasks are pooled. TART significantly outperforms competitive baselines in this setup, further highlighting the effectiveness of guiding retrieval with instructions.[1]

1 Introduction

Information retrieval (IR) is the task of finding *relevant* documents from a large collection of texts to fulfill a user's information need, typically expressed in the form of a textual query (Singhal et al., 2001). The notion of relevance from the user's perspective (i.e., *intent*) can be amorphous (Mizzaro, 1998), and a query alone may not fully capture user information needs (Ruthven and Lalmas, 2003; Taylor, 1962). As illustrated in Figure 1 (top), given the same query, "*implementing batch normalization*," users' intents can be diverse (e.g., find code snippets or paragraph-length answers).

Most existing work tries to learn those *implicit* intents from labeled data (e.g., pairs of queries and relevant documents), yielding separate models for different intents, as shown in the bottom left of Figure 1. These approaches usually require a vast number of annotated examples to train a model to capture the task-specific notion of relevance, while they could benefit from the abundance of data available from related tasks. Additionally, having separate models leads to complicated pipelines.

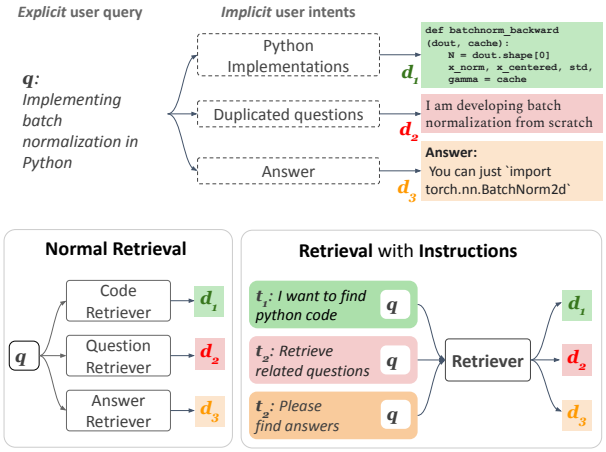


Figure 1: User intents are not fully captured in query q only (top). Conventional approaches (bottom left) take a query and retrieve documents from a closed corpus using a task-specific retriever. *Retrieval with instructions* (bottom right) additionally takes explicit intent.

This paper advocates for a new problem formulation, *retrieval with instructions* (Figure 1 bottom right), to *explicitly* model a user's intent by providing a natural language description of the search task (a.k.a. instruction). The goal of retrieval systems is to retrieve documents that are both relevant to the query *and* well-suited to the instructions (*task-aware*). Explicitly defining the user intent with natural language instructions provides additional flexibility that enables unifying diverse retrieval tasks during training.

Despite active research in language models (LMs), instruction-following has not been systematically explored in retrieval, partly due to the lack of annotated resources. To facilitate research

[1]Code and models are available at https://github.com/facebookresearch/tart.

Findings of the Association for Computational Linguistics: ACL 2023, pages 3650–3675
July 9-14, 2023

in retrieval with instructions, we introduce BERRI (**B**ank of **E**xplicit **R**et**R**ieval **I**nstructions), a collection of approximately 40 retrieval datasets with diverse instructions in a unified format, covering 10 diverse domains. Each task has on average 3.5 diverse instructions annotated by experts, following our novel instruction schema for retrieval tasks.

We showcase the benefit of BERRI to train TART (**T**ask-**a**ware **Re****T**riever), a multi-task retrieval system that learns to follow instructions to perform diverse tasks. We employ two widely explored architectures: TART-dual, a dense dual-encoder architecture that retrieves documents based on the similarity of independently encoded query succeeded by instructions and document embeddings; TART-full, a cross-encoder architecture that calculates probabilities of a document being relevant to the query according to the instruction. We train TART leveraging hard negative samples and new instruction-unfollowing negative samples.

The TART models, particularly TART-full yields state-of-the-art results on two popular zero-shot retrieval benchmarks, BEIR (Thakur et al., 2021) and LOTTE-pooled (Santhanam et al., 2022), outperforming systems using three times more parameters (Nogueira et al. 2020; Ni et al. 2021; Muennighoff 2022) as well as task-specific retrievers trained on millions of automatically generated examples (Dai et al., 2022; Wang et al., 2022a).

We further introduce a new evaluation setup, $\mathbb{X}^2$-Retrieval (**Cross**-task **Cross**-domain Retrieval), where a system needs to handle queries with diverse intents and find relevant documents from a large-scale, cross-domain pooled corpus, simulating challenges in real-world retrieval applications. TART outperforms other state-of-the-art methods, demonstrating its effectiveness in this under-explored setting, leveraging explicit textual intents. In summary, our contributions are as follows:

- *Retrieval with instructions*, a new formulation to model users' intent *explicitly* (Section 3.1).
- BERRI, a new collection of about 40 retrieval datasets with instructions (Section 3.3).
- TART, a task-aware retriever trained on BERRI that advances state of the art on zero-shot and cross-task retrieval (Section 4).

2 Background and Related Work

Zero-shot training of retrievers. Recent neural retrievers (Karpukhin et al., 2020; Lee et al., 2019; Khattab and Zaharia, 2020) show their superiority over term-based retrievers (e.g., BM25; Robertson and Zaragoza 2009) across domains when training data is abundant (Luo et al., 2022; Asai et al., 2021; Petroni et al., 2021). Due to the high annotation cost, improving neural retrievers in zero-shot settings is an active area of study. Pre-training neural retrievers (Izacard et al., 2022) and training a single retriever on large-scale supervised datasets such as MS MARCO (Bajaj et al., 2016) show improvements in transferring to related retrieval tasks (Khattab and Zaharia, 2020; Nogueira et al., 2020; Chen et al., 2022), while they often struggle with tasks unlike those used for training (Dai et al., 2022). To address this, several work (Wang et al., 2022a; Dai et al., 2022) trains customized retrievers for each task using unlabeled corpora, leveraging another model to automatically generate training data (Wang et al., 2022a). It often requires running massive LMs and training separate retrievers, resulting in slow and costly adaptation. Concurrent to our work, Su et al. (2022) trains a single dual-encoder model trained on embedding tasks including retrieval tasks with instructions.

Instruction tuning. Training LMs with instructions or demonstrations on many tasks has proven to be very effective for zero- or few-shot transfer (Wei et al., 2022a; Sanh et al., 2022; Ouyang et al., 2022; Min et al., 2022; Wang et al., 2022b; Mishra et al., 2022; Chung et al., 2022). Yet, such instruction tuning has not been systematically explored in retrieval for several reasons. First, large-scale instruction-annotated datasets (Bach et al., 2022; Wang et al., 2022b) exclude retrieval tasks. Second, instruction-following LMs are encoder-decoder or decoder-only models with tens of billions of parameters, which are difficult to be adapted for retrieval tasks requiring encoding millions of documents. Our work is inspired by this line of work and addresses those challenges.

Retrieval with descriptions. The problem of retrieval with descriptions (e.g., TREC 2004 Robust Track; Voorhees 2005) incorporate query-dependent descriptions that describe information needs for query disambiguation (e.g., desirable documents), unlike query-independent instructions in this work. Early work shows that concatenating descriptions only marginally helps (Walker et al., 1998), while Dai and Callan (2019, 2020) suggests that powerful BERT encoders (Devlin et al., 2019) could better incorporate such rich context.

Dataset	Instruction
NQ	Retrieve a Wikipedia paragraph that answers this question.
QReCC	Find a dialogue response from dialogue history to answer the user's question.
Arguana	Retrieve a paragraph from an argument website that argues against the following argument.
SciFact	Find a sentence from a scientific paper to check if the statement is correct or not.
MultiLexSum	I want to find the one-sentence summary of this legal case.

Table 1: Example instructions for Natural Questions (NQ; Kwiatkowski et al. 2019), QReCC (Anantha et al., 2021), Arguana (Wachsmuth et al., 2018), SciFact (Wadden et al., 2020) and MultiLexSum (Shen et al., 2022). Each instruction defines *intent*, *domain* and *unit*. The full list of instructions are in Appendix A.5.

3 Formulation and Data Collection

3.1 Problem Formulation

This work introduces a new problem formulation, *retrieval with instructions* (Figure 1 bottom right). We are given a large collection of N documents $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_N\}$, a search task instruction $\boldsymbol{t}$ and a query $\boldsymbol{q}$. The problem of retrieval with instructions is to find a document $\boldsymbol{d} \in \mathcal{D}$ that is relevant to $\boldsymbol{q}$ according to the instruction $\boldsymbol{t}$. Compared to the standard retrieval setting (e.g., Figure 1 bottom left), the difference is the explicit definition of *relevance* in the instruction $\boldsymbol{t}$ as additional input to the system and a retrieval system needs to be task-aware—changing their relevance measure by attending to the instruction. This new formulation brings both new research challenges and opportunities. For instance, a retriever is now required to modify its search behavior according to the instructions. On the plus side, different datasets can be naturally grouped to train a single retriever, yielding benefits from cross-task interdependence. Instructions provide extra flexibility and enable zero-shot transfer via natural language instructions, unlike training with fixed task tags (Maillard et al., 2021). A single task-aware retriever obviates the need to host multiple task-specific retrievers.

Multi-task training with instructions has not been studied in the area of retrieval due to the lack of resources and dedicated models. To facilitate the research on retrieval with instructions, we introduce BERRI, a large-scale retrieval benchmark with expert-written annotations (Section 3.3) in a unified format (Section 3.2), and subsequently the multi-task instruction-following retrievers (Section 4).

3.2 Unified Task and Instructions Schema

Task format. Each task $\mathcal{T}$ in BERRI consists of a corpus $\mathcal{D}$, queries $\mathcal{Q} = \{\boldsymbol{q_1}, \ldots, \boldsymbol{q}_K\}$, and an instruction $\boldsymbol{t}$, where K is the number of the queries included in the task. An instance of each task includes a query $\boldsymbol{q}$, gold (relevant) documents $\boldsymbol{d}^+$, and negative (irrelevant) documents $\boldsymbol{d}^-$. For each task, an explicit intent $\boldsymbol{t}$ is given.

Instruction schema for retrieval. We introduce a novel schema to define an informative instruction for retrieval tasks, which have not been studied in prior instruction-following literature. An instruction that sufficiently describes an arbitrary retrieval task should include: *intent*, *domain* and *unit*. Specifically, *intent* describes how the retrieved text relates to the query, such as whether the text answers a question in the query or paraphrases it. *Domain* is the expected source or type of retrieved text, such as Wikipedia or PubMed articles. *Unit* defines the text block to retrieve, such as a sentence or a paragraph. Table 1 shows examples of instructions, and Appendix A.5 shows the full list.

3.3 Dataset: BERRI

Dataset selection and unification. We manually collect datasets from (1) KILT (Petroni et al., 2021), (2) the Sentence-Transformers Training Data for Text Embedding Models[2], and (3) manual searches in ACL anthologies and huggingface datasets[3] to cover diverse tasks and domains. Except for a few domains (e.g., Wikipedia) many domains do not have retrieval datasets while there are a few datasets for other NLP tasks that can be cast as retrieval (e.g., sentence paraphrasing). Re-purposing those non-retrieval tasks as retrieval tasks enables the diversity of the domains as well as the instructions in BERRI. From initial collections of more than 60 datasets, we conduct manual dataset inspection and select 37 datasets (Figure 2) covering diverse domains (e.g., Wikipedia, scientific papers) and tasks (e.g., fact verification, dialogue response retrieval, QA). See Appendix A.1 for more details.

[2] https://huggingface.co/datasets/sentence-transformers/embedding-training-data

[3] https://huggingface.co/docs/datasets/index

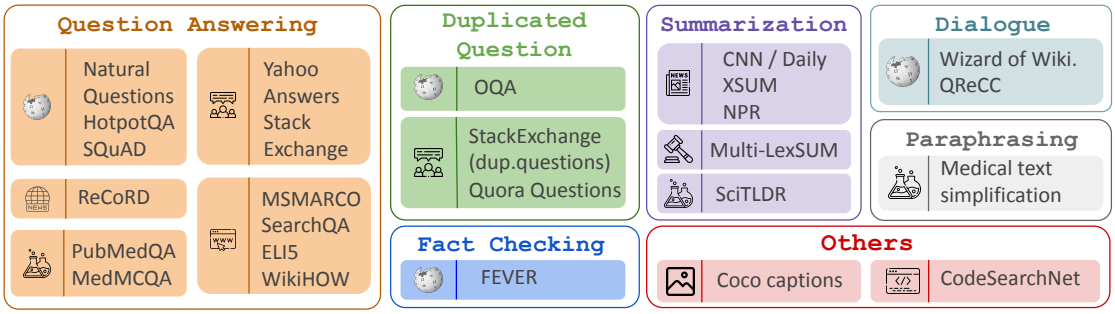


Figure 2: Examples of datasets included in BERRI. Table 5 shows the full dataset list.

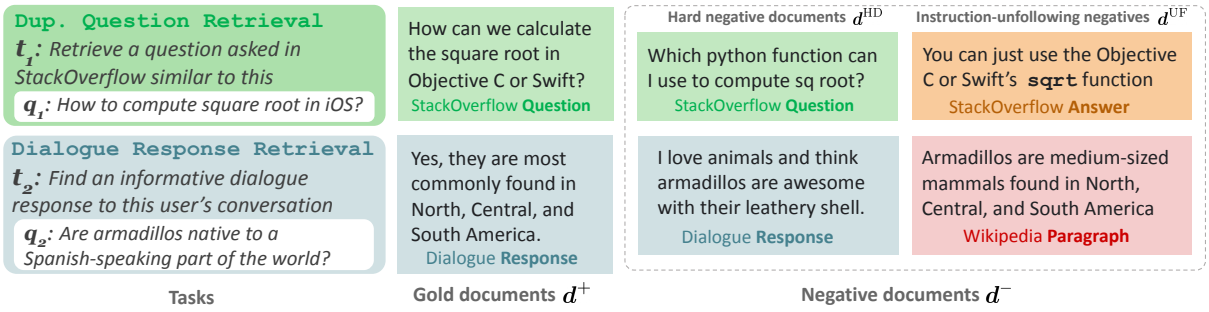


Figure 3: Examples of documents that are considered gold documents d^+, and two types of negative documents d^-: hard negatives d^{HD} and instruction-unfollowing negatives d^{UF} for two different query and instruction pairs.

Negative documents selection. Negative samples are crucial for training retrieval systems (Zhan et al., 2021; Qu et al., 2021). In addition to randomly sampled negative samples (random negative documents), we introduce two types of challenging negative samples: denoised hard negative documents d^{HD} and instruction-unfollowing negative documents d^{UF}. Figure 3 shows examples of gold documents and those negative samples.

For hard negatives d^{HD} we run Contriever (Izacard et al., 2022) and then filter out false negative documents by running an off-the-shelf reranker[4] and keeping passages with low scores (smaller than 0.1). We further introduce a new negative sampling strategy, instruction-*unfollowing* negative samples d^{UF}, to make systems learn to retrieve documents that are well-suited to the instructions. As shown in Figure 3, given an instruction "find an informative dialogue response", a system should not retrieve a Wikipedia paragraph about armadillos, even though that is highly relevant to the query. To obtain such negative documents, we retrieve documents from a different task's target corpus using Contriever and consider all those documents to be negatives since they do not satisfy the instruction. Details are in Appendix Section C.3.

[4] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

4 TART: Multi-task Instructed Retriever

We now present TART (**TA**sk-aware **Re**Triever) trained on BERRI via multi-task instruction-tuning, leveraging our unified task-aware schema.

4.1 Model Architecture

TART-dual. TART-dual adopts a dual-encoder architecture to independently encode queries with instructions and documents. It uses maximum inner product search (MIPS) over the embeddings (Karpukhin et al., 2020). The similarity between a query q and a document d, given an instruction t, is calculated as follows:

$$s(t, q, d) = \mathbf{E}([t; q])^T \mathbf{E}(d), \quad (1)$$

where $\mathbf{E}(\cdot)$ is the embedding function[5] and $[t; q]$ is the concatenation of the instruction and query. For this model, document embeddings can be computed offline, improving inference efficiency at the cost of storage space (Yamada et al., 2021).

TART-full. The dual-encoder architecture is known to be less expressive due to its limited query-document interactions (Khattab and Zaharia, 2020). To address this issue, we also explore a cross-encoder architecture (Nogueira and Cho, 2019),

[5] We use a shared encoder since having separate encoders gave no additional gains in preliminary experiments.

which computes the relevance between a query and each document by jointly encoding them with cross-attention. A cross-encoder model is often prohibitively expensive to scale up to millions of documents, so we first run a lightweight off-the-shelf dual-encoder retriever to retrieve the top documents. For each of these documents, TART-full computes the similarity score as:

$$\mathrm{s}(\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{d}) = \mathrm{FFN}(\mathbf{E}([\boldsymbol{t}; \boldsymbol{q}; \boldsymbol{d}])), \quad (2)$$

where FFN represents an additional feed-forward network that predicts whether the document follows the instruction and is related to the query.

We initialize TART-full with encoders of T5-based instruction-following pretrained models, namely T0-3B (Sanh et al., 2022) and FLAN-T5-3B (Chung et al., 2022) for their empirical competitiveness, as found in prior work (Sachan et al., 2022). We follow the EncT5 approach (Liu et al., 2021) and prepended each sequence with a start-of-sequence token. The token representation is then fed to a newly initialized feed-forward network. Unlike MonoT5 (Nogueira et al., 2020), we use their encoders only for parameter efficiency, reducing the number of the parameters to half.

4.2 Training TART

We train TART-dual and TART-full using the positive documents and three types of negative documents in BERRI with instructions (Figure 3).

Training TART-dual. We train TART-dual using annotated positive and negative documents in BERRI as well as in-batch negatives as follows:

$$\mathcal{L} = -\log \frac{e^{s(\boldsymbol{t},\boldsymbol{q},\boldsymbol{d}^+)}}{\sum_{\boldsymbol{d}\in\mathcal{B}} e^{s(\boldsymbol{t},\boldsymbol{q},\boldsymbol{d})}},$$

where $\mathcal{B}$ denotes all documents in the same mini-batch (Karpukhin et al., 2020).

Training TART-full. Following prior work (Nogueira and Cho, 2019), TART-full is trained with the cross entropy loss as:

$$\mathcal{L} = -\sum_{\boldsymbol{d}\in\boldsymbol{d}^+} \log s(\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{d}) - \sum_{\boldsymbol{d}\in\boldsymbol{d}^-} \log(1 - s(\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{d})).$$

Knowledge distillation from TART-full to TART-dual. The default hard negatives in BERRI rely on off-the-shelf models fine-tuned on MS MARCO; for some domains, the hard negative samples mined by those models can be less reliable. For a smaller dual-encoder model, those false positive and negative samples can diminish performance (Qu et al., 2021). We apply hard knowledge distillation with TART-full (Qu et al., 2021). We first train TART-full on the annotated gold documents and the negative documents in BERRI, and then update hard negative documents and positive documents as in Section 3.3 with TART-full, with instructions.

5 Experiments

We evaluate TART on zero-shot retrieval (Section 5.1) and our new more challenging evaluation setup, $\mathbb{X}^2$-Retrieval (Section 5.2).

5.1 Zero-shot Retrieval Evaluations

We run experiments on two popular zero-shot retrieval benchmarks: **BEIR** (Thakur et al., 2021) and **LOTTE** (Santhanam et al., 2022). None of the evaluation datasets overlap with BERRI.

BEIR is a collection of diverse retrieval tasks in multiple domains where the retrieval target is restricted to the target corpus in a single domain. We used publicly available datasets.[6] **LOTTE-Search** samples GooAQ (Khashabi et al., 2021) questions whose answers come from certain forums in StackExchange. We evaluate our model in the pooled setup, where documents come from forums in diverse domains (e.g., cooking, technical). GooAQ is not included in our training set. In LOTTE, our instructions specify which *forum* our system should retrieve evidence from (e.g., "Retrieve a *cooking* StackExchange forum post").

Metrics. Following Thakur et al. (2021), for BEIR, we use NDCG@10 as our primary metric on BEIR. For LOTTE-pooled, we use Success@5 (= Recall@5) as our primary metric, as in the original paper (Santhanam et al., 2022).

5.2 $\mathbb{X}^2$-Retrieval Evaluation

Users' intents can be diverse, requiring searching in an open-domain environment (Piktus et al., 2021), which is currently under-explored. We introduce a more realistic evaluation setup, $\mathbb{X}^2$-Retrieval (Cross-task Cross-domain Retrieval), where several retrieval tasks with different intents are pooled to form a single retrieval target containing diverse documents. This requires a system not only to adapt to a new task in a zero-shot manner but also to model users' intents expressed in natural languages to meet their information needs.

[6] Following Dai et al. (2022), we exclude Natural Questions, MS MARCO, HotpotQA, FEVER, and CQADupStack from our evaluation targets for fair comparison since they are included either in encoders' pretraining or in BERRI.

Task	$\|q\|$	$\|\mathcal{C}\|$	Domain	Query	Gold documents
Ambig QA (Min et al., 2020)	1,172	18,809	Wikipedia	question	duplicated question
WIKIQA (Yang et al., 2015)	369	26,196	Wikipedia	question	answer sentence
SciFact (Wadden et al., 2020)	300	5183	Science	claim	scientific paper paragraph
GooAQ-Technical (Khashabi et al., 2021)	1,000	4,086	Technical	question	StackOverflow answer
LinkSo-Python (Liu et al., 2018)	1,000	485,413	Technical	question	StackOverflow question
CodeSearchNet-Python (Husain et al., 2019)	1,000	457,414	Code	comment	Python code

Table 2: The $\mathbb{X}^2$-Retrieval evaluation. Example pairs of queries and documents are shown in Table 8. In addition to the corpora listed above, we add the Natural Questions corpus data from BEIR (Thakur et al., 2021).

Tasks and queries. Our $\mathbb{X}^2$-Retrieval evaluation covers six datasets across three domains, namely, Wikipedia, Science, and Technical (Table 2) domains. The key challenge here includes datasets with different search intents that may not always be obvious from the queries alone.

A pooled corpus. For the primary *pooled* setup, we combine all documents from different tasks and the BEIR NQ Wikipedia corpus to form a single retrieval corpus, consisting of approximately 3.7 million documents. We also report the simplified *closed* setup performance as an oracle setup, where a system retrieves only from the original corpus.

Metrics. We report NDCG@10 on both pooled and closed setups for each task. In addition, we evaluate the performance gap between the closed and pooled setups and refer to it as *robustness*. A smaller gap means that the model is distracted less by the documents from undesirable corpora.

5.3 Baselines

We compare TART with various state-of-the-art methods. The first group is unsupervised models that are not trained or trained on unlabeled text; these include **Contriever** (Izacard et al., 2022) and **BM25**. We also compare TART with **UPR** (Sachan et al., 2022), which reranks the Contriever results using a pretrained T0-3B. The second group trains retrievers and rerankers on MS MARCO or a few large-scale datasets and directly transfers them to new tasks with no adaptations, including **MonoT5** (Nogueira et al., 2020), **Contriever-MS MARCO** and **Contriever-MS MARCO + Cross Encoder (CE)**, **ColBERT v2** (Santhanam et al., 2022), and **SGPT-6.8B** (Muennighoff, 2022). The final group of models is specialized retrievers trained for each task on automatically generated task data. **Promptagator** (Dai et al., 2022) generates large amount of in-domain data using FLAN (Wei et al., 2022a), and **GPL** (Wang et al., 2022a) generates them using DocT5Query (Nogueira et al., 2019). We also compare TART with their counterparts trained on BERRI and evaluated without instructions, **TART-dual w/o I** and **TART-full w/o I**.

5.4 Experimental Settings

We initialize TART-full from the T0-3B (Sanh et al., 2022) and FLAN-T5 encoder (Chung et al., 2022). We sample positive and negative passages with a 1:4 ratio. We initialize TART-dual from Contriever-MS MARCO (Izacard et al., 2022), which is based on BERT-base.[7] Per-GPU batch size is 16, and for each positive document, we sample in total 5 negative passages, where 90% of them are randomly sampled from $\mathcal{D}$, and 10% are sampled from $\boldsymbol{d}^{\text{HD}}$ and $\boldsymbol{d}^{\text{UF}}$. We use top 100 Contriever-MS MARCO results as the TART-full initial candidates.[8] Table 9 shows instructions for evaluations. More details are in Appendix C.1.

6 Results and Analysis

6.1 Results

Zero-shot evaluation. As shown in Table 3, TART-full and TART-dual largely outperform their counterparts trained and tested without instructions, demonstrating the effectiveness of instruction-tuning for better zero-shot retrieval. TART-full significantly outperforms larger models and customized models trained on millions of synthetically generated in-domain data, advancing the state of the art on BEIR and LOTTE. Unlike prior methods that require additional data generation, TART only requires a single human-written instruction to

[7] We also tried larger models such as SGPT-1.3B (Muennighoff, 2022), but observed large performance drop on some datasets, resulting in lower average than Contriever-based TART-dual. Therefore, we use Contriever-based TART-dual.

[8] We found that combining TART-full with the original Contreiver performs better than combining TART-full with TART-dual, possibly because TART-full uses the hard negative samples retrieved by Contriever's top-retrieved results.

	model size & rerank			BEIR										LOTTE
	Ret.	Gen.	K	TREC	NFC	FQA	ARG	TOU	DBP	SCD	CLI	SCF	Avg.	Search-Pooled
BM 25	0	0	0	65.6	32.5	23.6	31.5	36.7	31.3	15.8	21.3	66.5	36.0	48.3
Contriever	110M	0	0	27.4	31.7	24.5	37.9	19.3	29.2	14.9	15.5	64.9	29.3	55.5
UPR†	3B	0	0	60.4	33.3	45.0	50.3	21.3	33.8	17.3	9.5	69.6	37.8	–
Contriever (MS)	110M	0	0	59.6	32.8	32.9	44.6	23.0	41.3	16.5	23.7	67.7	38.0	66.0
Contriever+CE†	133M	0	100	70.1	34.4	36.7	41.3	29.8	47.1	17.1	25.8	69.2	41.3	73.5
ColBERT-v2	110M	0	0	73.8	33.8	35.6	47.9	26.3	44.6	15.8	17.6	69.3	40.5	71.6
BM25 + MonoT5 (3B)†	3B	0	1000	**79.6**	**38.4**	**51.2**	28.8	20.0	**47.8**	18.4	28.9	**77.7**	43.4	–
SGPT-6.8B	6.8B	0	0	87.3	36.2	37.2	51.4	25.4	39.9	**19.7**	30.5	74.7	44.7	–
GPL	66M×9	220M	0	72.6	–	32.8	–	–	–	–	–	66.4	–	–
Promptagator	110M ×9	175B	0	72.7	33.4	40.4	53.8	26.6	36.4	16.3	21.4	62.3	40.4	–
Promptagator (rank)†	220M ×9	175B	200	76.0	36.0	45.9	53.1	27.8	41.3	19.1	22.6	73.6	43.9	–
TART-dual	110M	0	0	64.9	33.6	34.2	48.6	20.7	41.3	14.1	14.7	70.1	38.1	56.9
TART-full (T0-3B)†	1.5B	0	100	71.7	34.0	42.2	49.8	**31.2**	45.1	17.5	30.0	75.8	44.1	**75.7**
TART-full (FLAN-T5)†	1.5B	0	100	72.8	33.4	41.8	**51.5**	24.9	46.8	18.7	**35.4**	**77.7**	**44.8**	73.1
TART-dual w/o I	100M	0	0	46.3	32.7	28.6	44.7	12.3	33.0	12.8	15.7	67.4	32.6	56.7
TART-full† (T0-3b) w/o I	1.5B	0	100	57.2	37.1	41.3	50.4	18.3	41.3	18.3	32.5	73.2	41.1	71.2

Table 3: Zero-shot retrieval results on BEIR and LOTTE-Search. † indicates the models using cross-encoder-based reranking models. The first group of models use no labeled data during training. The second group uses MS MARCO at training time but has no customized task-specific data. The third group trains individual retrieval systems using automatically generated data. TREC, NFC, FQA, ARG, TOU, DBP, SCD, CLI, SCF indicates TREC-COVID, FIQA, NF Corpus, Arguana, Touche-2020, DBPedia, SciDocs, Climate- Fever, and SciFact, respectively. "×9" of GPL, Promptagator means that those models train customized models for each dataset.

	AMB		WQA		SCF		GAT		LSO		CSP		Avg.		Δ
	cl	pl	cl	pl	cl	pl	cl	pl	cl	pl	cl	pl	cl	pl	cl−pl
Contriever	**96.8**	93.8	80.9	54.1	67.7	57.4	73.2	59.8	28.0	26.7	36.7	36.1	63.9	54.6	9.3
Contriever+CE	96.6	47.4	78.2	58.4	69.1	61.7	75.4	66.0	**32.1**	**31.4**	42.0	40.2	65.5	50.9	13.4
TART-dual	96.3	**95.3**	80.2	**63.1**	70.1	66.2	75.0	65.0	23.0	23.4	31.3	31.3	60.5	53.6	**6.9**
TART-full (T0)†	91.1	90.5	82.1	52.5	74.7	66.2	**80.5**	**68.6**	25.1	24.9	51.4	**51.4**	67.5	**59.1**	8.4
TART-full (FLAN)†	94.0	89.6	**86.9**	55.9	**77.4**	**66.3**	78.3	66.7	18.1	18.4	**51.8**	50.1	**67.7**	57.8	9.9

Table 4: $\mathbb{X}^2$-Retrieval results. Δ shows the gap of the average performance in the pooled and closed settings. AMB, WQA, GAT, LSO, CSP denote AmbigQA, WikiQA, GooAQ-Technical, LinkSO, and CodeSearchNet-Python.

adapt to a new task. Compared to other methods using cross-encoder-based reranking models (e.g., BM25+MonoT5), TART-full uses a much smaller number of paragraphs to be re-ranked, which significantly reduces latency caused by reranking at test time. The large performance gain from Contriever (MS) to TART-dual on six out of the nine BEIR tasks (e.g., SciFact, Arguana) shows the effectiveness of instructions and knowledge distillations. However, for the other three datasets (e.g., Touche-2020), TART-dual shows large performance deterioration. We hypothesize that model capacity (i.e., BERT-base) and limited interactions between the query and document embeddings could be major bottlenecks. Prior work on instruction training in large LMs has shown that smaller models often do not get as much benefit as larger ones from instructions and increasing dataset size, possibly due to their limited model capacities (Chung et al., 2022). Su et al. (2022) also observe more significant gain from instruction tuning when they use larger encoder models (i.e., GTR-base v.s. GTR-XL), reporting performance deterioration in retrieval tasks when they instruction tune 335 million parameter base model. Future work can investigate efficient architectures that enable more rich interaction between queries with instructions and documents.

$\mathbb{X}^2$-Retrieval evaluation. Table 4 shows the models' $\mathbb{X}^2$-Retrieval performance. Contriever and Contriever+CE show competitive closed performance in the closed setup, as in BEIR, but they struggle in the pooled setup due to their inability to handle human instructions. Especially Contriever+CE shows a large performance drop on AmbigQA-pooled by retrieving documents instead of queries due to the biases from fine-tuning on a

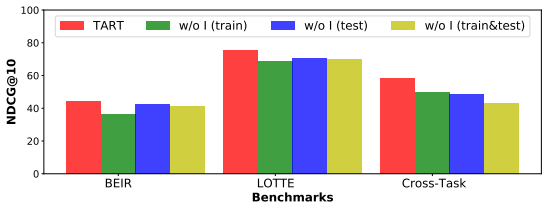
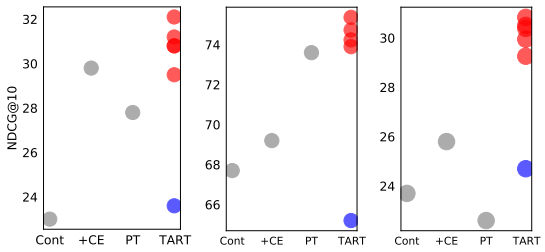
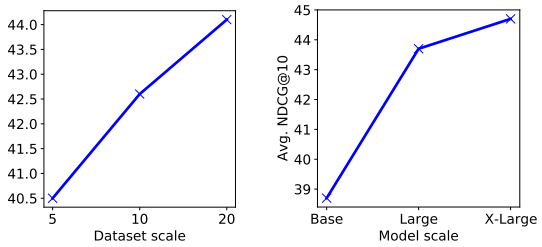


Figure 4: Ablations of instructions. w/o I (train), w/o I (test), and w/o (train & test) indicate (a), (b) and (c).



(a) Touche-2020 (b) SciFact (c) CLI-FEVER

Figure 5: Performance with different instructions (red circles). Blue circles show results without instructions. PT, Cont, +CE denote Promptagator, Contriever and Contriever+CE.



(a) Dataset scale (b) Model scale

Figure 6: Analysis of dataset and model scale.

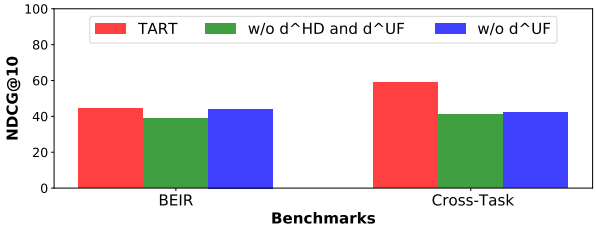


Figure 7: Ablations of negative samples. "w/o $\boldsymbol{d}^{\text{HD}}$ and $\boldsymbol{d}^{\text{UF}}$" denotes a model trained without hard and instruction-unfollowing negative documents while "w/o $\boldsymbol{d}^{\text{UF}}$" ablates instruction-unfollowing documents only.

QA dataset (i.e., MS MARCO) only.

TART-full shows the best-closed performance and pooled performance, indicating its strong zero-shot adaptation and cross-task abilities. We found that a model can flexibly change its behavior based on the instructions, as shown in Table 11. TART-dual shows strong performance on the pooled setup, indicating that smaller models can be also guided by explicit instructions.

6.2 Analysis

Ablating instructions. We compare TART-full with three variants: (a) *train without instructions, test with instructions* prepends instructions at test time only to test if the models just exploit keyword matching only at test time; (b) *train with instructions, test without instructions* uses TART-full without instructions at test time; (c) *train without instructions, test without instructions* does not use instructions at all during training and test time. Figure 4 shows the performance of those baselines. On all benchmarks, ablating instructions during training or test time causes a notable performance drop. We also see that a model trained with instructions but given no instruction at test time still yields a few performance improvements over the model trained completely without instructions, indicating the effectiveness of multi-task instruction tuning.

Robustness toward instructions. Figure 5 shows the performance variance given multiple different instructions. Instructions significantly improve model performance without instructions (the blue circles). Although different instructions give small performance variance, TART often outperforms other baselines when informative instructions are given. See Table 15 for individual instructions.

Dataset scale. Following prior work on instruction tuning for LMs (Wang et al., 2022b; Wei et al., 2022a), we conduct dataset ablation, where we reduce the number of training datasets. Figure 6a shows the average BEIR performance of TART-full trained on randomly sampled 5, 10, and 20 datasets. Increasing the number of the training datasets helps TART to perform better. In addition to domain and task diversity, the diversity of instructions observed during training may also improve performance, as in Appendix Section E.3.

Effects of negative sampling. We analyze the effectiveness of negative samples by ablating them during training. Figure 7 shows the performance of the models trained without negative samples on BEIR and $\mathbb{X}^2$-Retrieval. Adding more challenging negative documents (i.e., $\boldsymbol{d}^{\text{HD}}$ and $\boldsymbol{d}^{\text{UF}}$) during training largely improves the model performance on BEIR. Moreover, the model trained without

instruction-following samples (w/o d^{UF}) results in lower $\mathbb{X}^2$-Retrieval performance, although this model performs on par with the original TART-full on BEIR. This indicates that our new instruction-unfollowing negative documents largely contribute to improving the ability to distinguish instructions and are thus crucial to build a robust task-aware retrieval system.

Model scale. We test different TART-full sizes to see how model scale affects final performance. Prior work has shown that scaling up re-ranking models often improves reranking performance (Rosa et al., 2022), and models' instruction-following abilities improve as models get larger (Wang et al., 2022b; Sanh et al., 2022; Wei et al., 2022b). We investigate how model scale affects the ability to generalize to new tasks *and* follow instructions. For a fair comparison, we train TART-full using different T5 LM-Adapt (base, large, and XL) and evaluate performance using them to rerank the top 100 Contriever results. Figure 6b shows TART-full's average performance across different model scales. We observe clear performance improvements by increasing model size as observed in prior work on large LM.

7 Conclusion

This paper lays the foundation for building a general-purpose task-aware retriever that can follow natural language instructions. We introduced a new setup, *retrieval with instructions*, to model users' intents explicitly. We presented BERRI, the first large-scale retrieval dataset with expert-written annotations. Building upon BERRI, we trained the first instruction-following retrieval system by massive multi-task instruction-tuning, TART advances the state of the art on two zero-shot retrieval benchmarks BEIR and LOTTE as well as on our newly introduced challenging evaluation setup.

Limitations

Although our TART-full model shows the effectiveness of instruction-tuning for retrieval, on some datasets TART-dual shows large performance degradation from its non-instruction-following counterpart. We hypothesize that a smaller model size (i.e., 110 million parameters) and limited interactions between query and document embeddings are the main factors. We conduct primarily experiments training larger dual-encoder models such as SGPT (Muennighoff, 2022) on BERRI but still observe some notable performance drop on some datasets, which indicate only scaling up encoders may not significantly improve instruction-following retrieval systems. Future work can study the better approach to train larger-scale dual-encoder models as well as explore modeling architectures that enable rich interactions but are still more efficient than the cross-encoder, such as ColBERT-v2 (Santhanam et al., 2022).

Retrieval tasks are excluded in prior work on instruction-following of LLMs. This work is the first to explore instruction tuning in the area of retrieval, and we annotate more than 100 instructions for approximately 40 tasks, and we demonstrate the effectiveness of the dataset scale in retrieval. Yet, recent work (Wang et al., 2022b; Chung et al., 2022) show that scaling up the number of the training datasets improves LLMs' ability to adapt to new task via instructions, and the current dataset scale might not be optimal. We open-source our instruction data and call for community efforts to collect more retrieval tasks and human-written instructions as in instruction-following for LMs (Wang et al., 2022b; Bach et al., 2022), to investigate whether further increasing the number of the datasets lead to improvements.

Ethical Considerations

Although instruction-tuning using many datasets enable better zero-shot transfer, TART does not always retrieve documents that perfectly align with users' expectations. Applying TART to safety-critical domains requires extra attention. BERRI includes approximately 40 tasks covering diverse domains. Although the data has been automatically filtered, and we have examined the data, there may still be harmful or privacy-sensitive contents. We will release all of the data and preprocessing scripts for follow-up work to inspect those dataset issues and the effects of those data.

Acknowledgements

We thank Allen School NLP and Meta AI researchers for their insightful discussions and Jeff Dalton, Mike Lewis, Sheng-Chieh Lin, Sandy Kaplan, and Yizhong Wang for their helpful feedback on this paper and discussions.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. 2021. One question answering model for many languages with cross-lingual dense passage retrieval. *Proceedings of Advances in Neural Information Processing Systems*.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A human generated machine reading comprehension dataset.

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 *Proceedings of Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association*.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Proceedings of the 38th European Conference on Information Retrieval*.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2022. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server.

Hyung Won Chung, Le Hou, Shayne 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, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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*.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhuyun Dai and Jamie Callan. 2020. Context-aware document term weighting for ad-hoc search. In *Proceedings of The Web Conference 2020*.

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

Ashwin Devaraj, Iain Marshall, Byron Wallace, and Junyi Jessy Li. 2021. Paragraph-level simplification of medical texts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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*.

Thomas Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. CLIMATE-FEVER: A dataset for verification of real-world climate claims. In *Proceedings of Tackling Climate Change with Machine Learning Workshop at NeurIPS*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *Proceedings of International Conference on Learning Representations*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new q&a dataset augmented with context from a search engine.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Antonio Gulli. 2004. Ag's corpus of news articles.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*.

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*.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Codesearchnet challenge: Evaluating the state of semantic code search.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

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*.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. GooAQ: Open question answering with diverse answer types. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

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*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

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*.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*.

Frederick Liu, Siamak Shakeri, Hongkun Yu, and Jing Li. 2021. EncT5: Fine-tuning t5 encoder for non-autoregressive tasks.

Xueqing Liu, Chi Wang, Yue Leng, and ChengXiang Zhai. 2018. LinkSO: A dataset for learning to retrieve similar question answer pairs on software development forums. In *Proceedings of the 4th ACM*

SIGSOFT International Workshop on NLP for Software Engineering.

Man Luo, Arindam Mitra, Tejas Gokhale, and Chitta Baral. 2022. Improving biomedical information retrieval with neural retrievers. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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 Proceedings of the The Web Conference 2018*.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wentau Yih, Barlas Oguz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Stefano Mizzaro. 1998. How many relevances in information retrieval? *Interacting with Computers*.

Niklas Muennighoff. 2022. SGPT: Gpt sentence embeddings for semantic search.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, and Sebastian Riedel. 2021. The web is your oyster–knowledge-intensive nlp against a very large web corpus.

pushshift. 2021. Npr corpus.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Soumik Rakshit. 2019. Yahoo answers dataset.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*.

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.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Ian Ruthven and Mounia Lalmas. 2003. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*.

Devendra Singh 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*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *Proceedings of International Conference on Learning Representations*.

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*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Kornrel Csernai Shankar Iyer, Nikhil Dandekar. 2012. First quora dataset release: Question pairs.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. 2022. Multilexsum: Real-world summaries of civil rights lawsuits at multiple granularities. In *Proceedings of the 36th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*.

Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Robert S. Taylor. 1962. The process of asking questions. *American Documentation*.

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 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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*.

Ellen Voorhees. 2005. Overview of the trec 2004 robust retrieval track. In *Proceedings of the Thirteenth Text REtrieval Conference*.

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*.

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*.

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*.

S. Walker, Stephen Robertson, M. Boughanem, G. J. F. Jones, and K. Sparck Jones. 1998. Okapi at trec-6: Automatic adhoc, vlc, routing, filtering and qsdr. In *The Sixth Text REtrieval Conference*.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022a. 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*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *Proceedings of International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

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*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. ReCoRD: Bridging the gap between human and machine commonsense reading comprehension.

Appendix

A Further BERRI Details

A.1 Detailed Dataset Creation Process

Manual dataset selection. From an initial list of more than 60 datasets, we assess whether it is suitable for repurposing as a retrieval task. Specifically, we sample 20 instances from the candidate dataset and check if the queries are self-contained.[9] If the majority of queries fail this test, we exclude the corresponding dataset. Consequently, we use 37 datasets, including more than 5 million instances in total. For datasets that are orders of magnitude larger than other datasets (e.g., PAQ; Lewis et al. 2021), we randomly sample up to 300k instances, except for MS MARCO.[10] As a result, BERRI covers diverse domains (e.g., Wikipedia, scientific papers) and tasks (e.g., fact verification, dialogue response retrieval, QA). See Appendix A.3 for more details.

Unification and instruction annotations. For retrieval datasets such as MS MARCO, we use the annotated gold documents as positive documents $\boldsymbol{d}^+$ to a given query $\boldsymbol{q}$. Regarding non-retrieval tasks, we use the original input sequence as a query $\boldsymbol{q}$ and the original output or given context as $\boldsymbol{d}^+$. For instance, given a summarization dataset we use a source text and a summary as a query and a gold document, respectively. More details about the dataset unification are available in Section A.2.

For datasets without preprocessed retrieval targets,[11] we gather all positive and negative documents provided by the original dataset to build a single task-specific retrieval corpus $\mathcal{D}$.

A.2 Details of Dataset Unification

As shown in Table 5, some datasets were not originally retrieval datasets (e.g., summarization datasets). We describe how we convert these into the unified retrieval task format.

QA. For QA datasets, where each instance consists of a query, a gold context, and answers, we assume the original gold context is the gold document used as a positive sample during training.

[9] For examples, finding a corresponding review text for the review title "*I love this!*" is under-specified.

[10] Prior work has shown that MS MARCO can be beneficial to many downstream retrieval tasks (Izacard et al., 2022).

[11] For example, KILT datasets such as FEVER or NQ use the unified Wikipedia corpus.

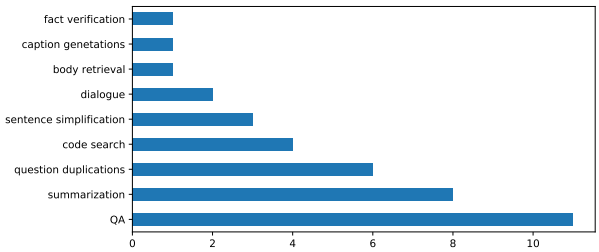


Figure 8: The task distributions of the datasets included in BERRI.

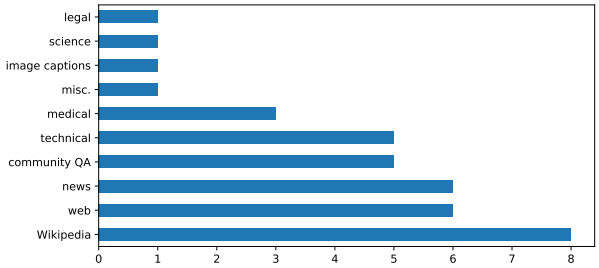


Figure 9: The domain distributions of the datasets included in BERRI.

For some exceptional datasets, we performed additional preprocessing. We found that ReCoRD instances are occasionally self-containing due to the nature of the cloze-style QA; therefore, for ReCoRD, we replace the original placeholder with the gold answer and use this original question with the answer as the query and the original context as a gold document. For MedMCQA, we use the source exam question as the query and the answer evidence as the positive document.

Summarization. For summarization datasets, we use target summarizations as the gold document and source text as the query.

Text simplifications. For text simplification datasets, we use source (often more complex) sentences as the query and simplified sentences as the gold document.

Code search. We use the source comment as the query and the corresponding implication as the gold document. We exclude the python subset from BERRI as we use it for $\mathbb{X}^2$-Retrieval.

A.3 BERRI Statistics

We conduct analyses on BERRI to understand its domain and intent diversities.

Intents. Open-ended intents are diverse and hard to classify into fixed sets of categories. As a proxy

for intents, Figure 8 shows the distributions of the source task categories. QA is the most representative category, while summarization and question duplication detection is also common due to their abundance in large-scale datasets. On the other hand, around 50 % of the tasks do not belong to those top three categories, such as code search or caption generations, which contribute to the diversity of BERRI. We also find that traditional non-retrieval tasks, such as sentence simplification or dialogue, can be repurposed as retrieval tasks.

Domains. Our dataset covers diverse domains. Figure 9 shows that Wikipedia (e.g., NQ), web (e.g., MS MARCO), Community QA (e.g., Quora), News(e.g., CNN/Daily) dominate, while we also have some expert domains (e.g., medical, legal, technical). We found that although many expert domain datasets are smaller than the ones in general domains like Wikipedia, adding those high-quality expert domain datasets helps the system learn to adapt to those domains or unseen expert domains with a similar writing style (e.g., scientific papers).

A.4 Dataset List

Table 5 shows all datasets we used in BERRI. Table 6 provides references for these datasets.

A.5 Instructions for BERRI

Table 7 shows the full list of the instructions in BERRI. Note that we present only one instruction for each dataset. A full list of the instructions will be released in our repository.

B Further Detail about the $\mathbb{X}^2$-Retrieval

Query and corpus creations. For AmbigQA, we use the official development split, including 1,172 queries, as the official test split annotations are not publicly available. We use all paraphrased questions for all train and development sets to form the retrieval corpus. For WIKIQA, we combine the development split and test split available at the huggingface datasets,[13] and we use the question and answer sentence pairs that are labeled as `1` as the queries for evaluations, and use the answer sentences as the gold documents. Regarding the retrieval target, we use all sentences available in the WIKIQA dataset, including the sentences that are labeled as `0`. For LinkSO, we use the original datasets' test split for the python domain and sample 1,000 queries.[14] We find questions that are labeled as duplicated and use their corpus as our retrieval target. For GooAQ-technical, we sample 1,000 GooAQ questions whose answers are from `stackoverflow.com`. As 20% of the sampled GooAQ tech queries share the same answer posts, we remove the duplicated paragraphs. For CodeSearchNet-Python, we use the comments describing the codes as queries and the corresponding python codes as positive documents. We sample 1,000 queries from the test split.

Examples. Examples of $\mathbb{X}^2$-Retrieval are shown in Table 8. As shown, queries themselves often do not fully indicate the users' intents. By specifying users' intents as explicit textual instructions, our model can effectively perform multi-task retrieval over a single pooled corpus.

Human evaluations of quality. To access the possibility of having false negative passages, we run an off-the-shelf retrieval system to retrieve the top 10 documents for randomly sampled 20 questions for each task, and we evaluate if any of the negative passages, especially from the non-target corpus, are indeed positive. We found that the false negative ratio is less than 10%.

C Modeling Details

C.1 Hyperparameters of TART

TART-dual. We set the learning rate to be 1×10^{-5} and warm-up steps to be 1,000. The softmax temperature is set to 0.05. The batch size is 1024. We use 7 negative samples per instance; 10% of the time we use hard negative or instruction-unfollowing negatives, while 90% of the time we use negative documents that are randomly sampled from the same target corpus. The maximum document chunk length is set to 256.

TART-full. To train a cross-encoder using the T0-3B encoder, we set the maximum sequence length to 512 and the batch size to 1, increasing the gradient accumulation steps to 8. We set the dropout rate to 0.1 and the learning rate to 1×10^{-5}.

C.2 Instructions for Evaluations

Table 9 lists the instructions used for the BEIR and $\mathbb{X}^2$-Retrieval evaluation.

[13] https://huggingface.co/datasets/wiki_qa

[14] https://sites.google.com/view/linkso

dataset	domain	task	unit
1. Altlex	Wikipedia	sentence paraphrase	sentence
2. StackExchange (title → title)	community forum	duplicated questions	title
3. StackExchange (query → answer)	community forum	QA	answer body
4. Yahoo Answers (title → answers)	community forum	QA	answer body
5. MS MARCO	web	QA	paragraph
6. ELI5	web	QA	answer paragraph
7. WikiHow	community forum	QA	answer paragraph
8. SearchQA	web	QA	search snippets
9. AGNews	News	summarization	news summary
10. NPR	News	summarization	news summary
11. CodeSearchNet (java)	code	code search	Java code
12. CodeSearchNet (ruby)	code	code search	Ruby ode
13. CodeSearchNet (JavaScript)	code	code search	Java Script code
14. CodeSearchNet (Go)	code	code search	Go code
15. PAQ	Wikipedia	QA	paragraph
16. Sentence Compression	misc.	sentence compression	sentence
17. CNN Daily Mail	news	summarization	news summary
18. XSUM	news	summarization	news summary
19. Coco captions	image captions	caption generations	captions
20. Quora Duplicated Questions	community forum	duplicated questions	questions
21. CCNews	news	summarization	news summary
22. FEVER (KILT)	Wikipedia	fact verification	paragraph
23. HotpotQA (KILT)	Wikipedia	QA	paragraph
24. NQ (KILT)	Wikipedia	QA	paragraph
25. TriviaQA (KILT)	Wikipedia	QA	paragraph
26. WoW-KILT (knowledge)	Wikipedia	knowledge-grounded dialogue	paragraph
27. WoW-KILT (response)	Wikipedia	knowledge-grounded dialogue	dialogue response
28. medical simplification	medical	sentence simplification	sentence
29. SciTLDR	science	summarization	paper summarization
30. PubMedQA	medical& science	QA	abstract
31. MedMCQA	medical	QA	answer explanation
32. Gigaword	web	headline retrieval	headline
33. ReCoRD	news	QA	news summary
34. MultiLexSum	legal	summarization	legal case summary
35. Qrecc	Wikipedia	conversational QA	response
36. OQA	Wikipedia	duplicated questions	question
37. SQuAD	Wikipedia	QA	paragraph

Table 5: The complete list of datasets included in BERRI. Table 6 shows references for them.

datasets used in BERRI
Altlex* (Hidey and McKeown, 2016), StackExchange (duplicate questions, question-title, question-question) (Reimers and Gurevych, 2019), Yahoo Answers* (Rakshit, 2019), MSMARCO* (Bajaj et al., 2016), ELI5* (Fan et al., 2019), WikiHow* (Koupaee and Wang, 2018), SearchQA* (Dunn et al., 2017), AG News* (Gulli, 2004), NPR* (pushshift, 2021), CodeSearchNet* (Husain et al., 2019), PAQ* (Lewis et al., 2021), Sentence Compression* (Filippova and Altun, 2013), CNN Daily Mail* (See et al., 2017), XSUM* (Narayan et al., 2018), COCO captions* (Chen et al., 2015), Quora Duplicated Questions (Shankar Iyer, 2012), CC News* (Hamborg et al., 2017), SQuAD* (Rajpurkar et al., 2016), FEVER† (Thorne et al., 2018), HotpotQA† (Yang et al., 2018), Natural Questions† (Kwiatkowski et al., 2019), TriviaQA† (Joshi et al., 2017), Wizard of Wikipedia† (Dinan et al., 2019), Medical Simplification Dataset (Devaraj et al., 2021), SCITLDR (Cachola et al., 2020), PubMedQA (Jin et al., 2019), MedMCQA (Pal et al., 2022), Gigaword (Rush et al., 2015), ReCoRD (Zhang et al., 2018), MultiLexSum (Shen et al., 2022), Qrecc (Anantha et al., 2021), OQA (Fader et al., 2014).
datasets used during evaluations
TREC-COVID (Voorhees et al., 2021), FIQA (Maia et al., 2018), NF Corpus (Boteva et al., 2016), Arguana (Wachsmuth et al., 2018), Touche-2020 (Bondarenko et al., 2020), DBPedia (Hasibi et al., 2017), SciDocs (Cohan et al., 2020), Climate-Fever (Diggelmann et al., 2020), SciFact (Wadden et al., 2020), GooAQ (Khashabi et al., 2021), LinkSO (Liu et al., 2018), AmbigQA (Min et al., 2020), WIKIQA (Yang et al., 2015).

Table 6: References for datasets used in BERRI and evaluations. We use the preprocessed versions available on the SentenceTransformers (Reimers and Gurevych, 2019) embedding data page [12] for the datasets with *. We use the preprocessed versions from KILT (Petroni et al., 2021) for the datasets with †.

Dataset	Instruction
1. Altlex	`Retrieve a sentence from Wikipedia that simplifies the following`
2. SE (title → title)	`I want to find a related question asked in StackExchange. Can you find one for me?`
3. SE (title → title)	`StackExchange is a community QA forum for diverse topics including technical or science. Help me to find a question body that duplicates my question`
4. YahooAnswers	`Retrieve the most voted answer for this question from Yahoo Answers.`
5. MSMARCO	`I want to know the answer to the question. Can you find good evidence on the web?.`
6. ELI5	`You have to answer a why / how question from users. Retrieve a Wikipedia paragraph that provides a piece of good evidence for the answer.`
7. WikiHow	`Find a detailed paragraph from WikiHow that explains how-to to achieve`
8. SearchQA	`Pick up the top web search results snippets for the following question.`
9. AGNews	`Find a news summary sentence corresponding to the following header.`
10. NPR	`Given a news article headline published at npr.org, find a corresponding summary of the news`
11. CodeSearchNet (Java)	`Match the following natural language instruction to Java codes`
12. CodeSearchNet (ruby)	`Retrieve ruby codes from GitHub commit history that implements this feature`
13. CodeSearchNet (JavaScript)	`Find a javascript code implementation on GitHub for the following natural language instructions`
14. CodeSearchNet (Go)	`Can you find a Go implementation of this?`
15. PAQ	`Can you answer my question by finding an article on the web?`
16. Sentence Compression	`You have to match this long sentence to a shorter compressed one`
17. CNN Daily Mail	`The following sentences are the summaries of a news article. Find the source news article.`
18. XSUM	`Retrieve a news article that is summarized as following.`
19. Coco captions	`Can you find an image caption talking about the same image as.`
20. Quora Dup. Questions	`Check if a Quora question is duplicated with this question.`
21. CC News	`I want to know the details of this news. Can you find a detailed news article on this for me?`
22. FEVER	`Retrieve a Wikipedia paragraph to verify this claim`
23. HotpotQA	`Find a paragraph that provides useful information to answer this question`
24. NQ	`Retrieve passages from Wikipedia to answer`
25. TriviaQA	`I want to find an answer for this Trivia question. Can you find some paragraphs that provide evidence from Wikipedia?`
26. WoW-Knowledge	`Find a Wikipedia paragraph related to the following conversation topic.`
27. WoW-Response	`Find a meaningful dialogue response to answer the user's question`
28. Medical Simplification	`Please retrieve a medical paper summary that is written in a simple language so that my patient can understand`
29. SciTLDR	`Find a sentence-length summary of this paper.`
30. PubMedQA	`Help me to find a highly related PubMed paper to answer this question.`
31. MedMCQA	`Find the explanation for the correct answer of this medical question.`
32. Gigaord	`Retrieve an extremely short summary of the following Gigaword article.`
33. Record	`Find a News article to verify the following sentence`
34. MultiLexSum	`Map this legal case summary to a sentence-long summary`
35. Qrecc	`You need to find a good response from a collection of previous responses and help users to know this topic more`
36. OQA	`Find a question that is paraphrased of this`
37. SQuAD	`Find a Wikipedia paragraph that answer the question`

Table 7: Full list of the instructions for the BERRI datasets. We present one instruction per dataset. All of the instructions are available at our GitHub repository.

C.3 Negative Sampling

Mining hard negatives. To mine hard negative documents for BERRI, we retrieve top documents

Dataset	q	d^{gold}
WIKIQA	Who plays henry tudor in the white princess?	Jacob Collins-Levy as Henry VII, the King of England, Elizabeth's husband
Ambig	Who played lead guitar for the rolling stones?	Who played lead guitar for the rolling stones since 1962?
SciFact	The risk of male prisoners harming themselves is ten times that of female prisoners.	5-6% of male prisoners and 20-24% of female inmates self-harmed every year (scientific paper).
GooAQ-tech	project facet java version 1.8 is not supported eclipse mars?	You can remove and create it again, or just update it. It is because the Java version in your Project Facet is 1.8 make it 1.7. Go to Project Properties -> Project Facets and on right side checkboxes, select the java checkbox(It might be already selected) and select the version as 1.7.
LinkSO	could use batch normalization tensorflow	trying implement batch normalization layer tensor flow problem running train step using tf moments get mean variance test time
CodeSearch	Create a Basilisp function, setting meta and supplying a with_meta	def _basilisp_fn(f): assert not hasattr(f, "meta") f._basilisp_fn = True f.meta = None f.with_meta = partial(_fn_with_meta, f) return f

Table 8: $\mathbb{X}^2$-Retrieval examples data.

Dataset	Instruction
TREC-COVID	`Retrieve Scientific paper paragraph to answer this question`
NF Corpus	`Retrieve Scientific paper paragraph to answer this question`
FIQA	`Find financial web article paragraph to answer`
Arguana	`Retrieve an argument that counter argues the following paragraph`
Touche	`You have to retrieve an argument to this debate question`
DBPedia	`Retrieve a Wikipedia introduction paragraph of the following entity`
SCIDOCS	`Find scientific paper titles that are related to the following`
Climate-Fever	`I want to know if the following claim is true or not. Retrieve a Wikipedia paragraph on climate change for this.`
SciFact	`Retrieve a scientific paper sentence to verify if the following claim is true`
WIKIQA	`Retrieve an answer sentence from Wikipedia`
AmbigQA	`Retrieve a question that is similar to this`
SciFact	`Retrieve scientific evidence to verify this claim`
GooAQ-technical	`Find a StackExchange forum that answers this question`
Codesearchnet-py	`Retrieve a python code that implements the following feature.`
LinkSO-Py	`You have to find a python implementation of this`

Table 9: Full list of the instructions used for evaluations.

from the target corpus using Contriever (Izacard et al., 2022) and then add new documents whose normalized scores predicted by a cross-encoder model, `ms-marco-MiniLM-L-12-v2`[15] are below 0.1 as hard negative documents.

Mining instruction-unfollowing samples. To sample instruction-unfollowing samples, given a query from a target dataset, we retrieve the top 20 documents from another task's corpus using Contriever-MS MARCO. For instance, given a PubMedQA, a system should not retrieve a document from a Wikipedia paragraph. A list of source target task and retrieval corpus combinations is shown in Table 10.

Sampling d^- for TART-full training. Challenging negative samples help a system to effectively learn the task. On the other hand, prior work also shows that it can lead to large performance drops in out-of-domain datasets, and having both randomly sampled negative documents and carefully

[15] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

dataset	expected output	instruction-unfollowing corpus
Gigaword	article summary	Wikipedia paragraph
Medical Paragraph Simplification	simplified text of medical cases	Wikipedia paragraph
MS MARCO	web answers	OQA questions
OQA	similar questions	Yahoo Answers answer
PubMedQA	medical paper abstract	Wikipedia paragraph
Qrecc	dialogue responses	Wikipedia paragraph
Quora	duplicated questions	Wikipedia paragraph
sentence compression	simplified sentence	Wikipedia paragraph
StackExchange (question→answer) title	StackExchange answer	StackExchange title
StackExchange (title →title) title	StackExchange title	StackExchange answer
Yahoo Answers	Yahoo Answers answer	Wikipedia paragraphs

Table 10: The list of the combinations of the dataset and corresponding instruction-unfollowing corpora to mine instruction-unfollowing negative documents.

designed negative documents is a key to building a system that is competitive in both in-domain and out-of-domain retrieval (Ni et al., 2021). To effectively combine the negative documents during training, we first combine random samples and hard negative samples, and then we randomly sample 4 negative documents per one positive document. The number of instruction-unfollowing documents, if applicable, is limited to less than 20% of the negative documents, and we set the maximum number of instruction-unfollowing samples from certain combinations listed in Table 10 up to 10k.

D More Experimental Details

in addition to the in-batch negative documents. We use 8 GPUs to train TART-full and 64 GPUs to train TART-dual. We train TART-full up to 10k steps and TART-dual up to 30k steps and take the checkpoint with the best development performance. We use 64 GPUs to train TART-dual and 8 GPUs to train TART-full.

E Further Results and Analyses

E.1 Qualitative Results on $\mathbb{X}^2$-Retrieval

Table 11 shows the qualitative examples given different instructions on $\mathbb{X}^2$-Retrieval, and Table 12 compares TART-full with Contriever MS MARCO.

E.2 Analysis of Instruction Effectiveness

Full results of instruction ablations. Table 13 shows the full BEIR results of ablating instructions and Table 14 shows the ones on LOTTE and $\mathbb{X}^2$-Retrieval. On all of the benchmarks, removing instructions at training or test time largely hurts the performance, indicating the effectiveness of instructions.

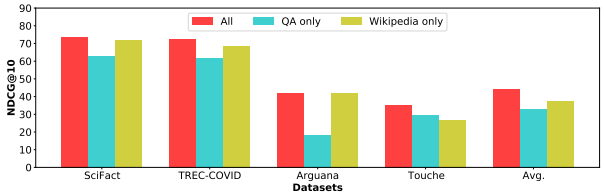


Figure 10: Dataset ablation results. Wikipedia-only denotes TART-full performance trained on Wikipedia-based datasets only. QA-only denotes the model trained on QA datasets only.

Examples of prompts with performance. Table 15 shows the instructions and TART-full performance on three BEIR datasets. We also provide a comparison of the model performance when uninformative instructions are given in Table 16. We see that more informative and related instructions often result in a strong performance, while irrelevant instructions degrade it.

E.3 Analysis on Model and Dataset Scale

Task diversity. As shown in Figure 10, task diversity is a key to improving models' zero-shot transfer performance. QA only struggles on Arguana, where the tasks significantly differ from QA.

Domain diversity. Figure 10 shows that having more diversity in training datasets' domains is also crucial, especially when the target datasets are in non-general domains. For instance, a model trained only on Wikipedia datasets struggles on Touche-2020 or SciFact, where documents come from argument websites and scientific papers, respectively.

Per-dataset performance breakdown. Table 17 shows the NDCG@10 across different model scales. We compare the TART-full initialized with

Query: how to calculate the distance between two points using longitude and latitude	
Instruction	**Top document**
Retrieve an answer post from StackOverflow to this question	SELECT getDistance(lat1,lng1,lat2,lng2) as distance FROM your_table. Here's a MySQL function that will take two latitude longitude pairs, and give you the distance in degrees between the two points. It uses the Haversine formula to calculate the distance.
Find a similar question asked in StackOverflow	tried implementing formula good two points testing yet code working distance returns.
Query: When did the kim family come to power?	
Instruction	**Top document**
find an answer sentence	Kim came to lead the Soviet-backed North's provisional government, becoming the first premier of its new government, the Democratic People's Republic of Korea (commonly known as North Korea), in 1948. He started the Korean War in 1950 with hopes to reunify the region. (Wikipedia)
Find a similar question	When did the kim family come to power in North Korea? (Ambig QA)
Query: 10% of sudden infant death syndrome (SIDS) deaths happen in newborns aged less than 6 months	
Instruction	**Top document**
retrieve a scientific paper paragraph to verify this	Despite declines in prevalence during the past two decades, sudden infant death syndrome (SIDS) continues to be the leading cause of death for infants aged between 1 month and 1 year in developed countries. Behavioral risk factors identified in epidemiological studies include prone and side positions for infant sleep, smoke exposure, soft bedding, and sleep surfaces, and overheating. (Scientific paper)
Find a Wikipedia paragraph to verify this	By definition, SIDS deaths occur under the age of one year, with the peak incidence occurring when the infant is at 2 to 4 months of age. (Wikipedia)

Table 11: Examples of the model's predictions given different instructions with the same query. The queries and documents are from $\mathbb{X}^2$-Retrieval.

different sizes of T5-LM-adapt for a fair comparison. We see in general that larger models perform better.

Table 18 shows the full BEIR results of TART-full trained on varying numbers of datasets. We see that as we increase the number of datasets used during training, model performance often improves, which is consistent with previous work on instruction-tuning in LLMs (Wang et al., 2022b).

E.4 Analysis on Different Pre-trained Models

Our TART-full is initialized with the T0-3B encoder. We experiment with more recent pre-trained instruction-following models: FLAN-T5-XL (Chung et al., 2022) and Tk-Instruct (Wang et al., 2022b), which are trained on the order of magnitude of more datasets. We analyze TART-full performance when we initialize encoders using different pre-trained encoder models, including the ones that are released recently. Table 19 shows the results of TART-full, when the encoder is initialized with three different recent instruction-following pretrained models, T0-3B, FLAN-T5-XL (Chung et al., 2022) and Tk-Instruct-3B (Wang et al., 2022b). FLAN-T5 shows the best average BEIR performance, outperforming TART-full by 0.7 NDCG@10. Tk-Instruct shows a notable performance drop on some datasets (e.g., TREC COVID), resulting in slightly lower performance than the original TART-full (T0-3B).

Query: 10% of sudden infant death syndrome (SIDS) deaths happen in newborns aged less than 6 months.
Instructions: Retrieve a scientific paper abstract to verify this

Contriever	**TART-full**
✗ By definition, SIDS deaths occur under the age of one year, with the peak incidence occurring when the infant is at 2 to 4 months of age. This is considered a critical period because the infant's ability to rouse from sleep is not yet mature (Wikipedia paragraph)	✓ Despite declines in prevalence during the past two decades, sudden infant death syndrome (SIDS) continues to be the leading cause of death for infants aged between 1 month and 1 year in developed countries. Behavioral risk factors identified in epidemiological studies include prone and side positions for infant sleep, smoke exposure, soft bedding, and sleep surfaces, and overheating. (paper)

Query: Which city will host the next winter Olympics?
Instructions: find an answer from Wikipedia

Contriever	**TART-full**
✗ Where will the next winter Olympics be held 2018? (Ambig question)	✓ The host city for the 2022 Winter Olympics, is Beijing in northern China, elected on 31 July 2015, at the 128th IOC Session in Kuala Lumpur. Beijing will be the first city ever to have hosted both the Summer and Winter Olympics. The 2022 Winter Olympics will take place between 4 and 20 February 2022. (Wikipedia paragraph)

Query: use batch normalization tensorflow
Instructions: Can you find python code implementing this?

Contriever	**TART-full**

✗ could use batch normalization tensorflow would like use batch normalization TensorFlow since found source code rel noreferrer core ops nn ops cc however find documented different semantics mlp cnn sure exactly bn find method called either c code copied reference (StackOverflow post)	✓
	`def batch_norm(inputs, training, data_format):`
	`  outputs = tf.layers.batch_normalization(inputs=inputs, axis=1, momentum=_BATCH_NORM_DECAY, epsilon=_BATCH_NORM_EPSILON, center=True, scale=True, training=training, fused=True)`
	`  return outputs`
	(GitHub code)

Query: how many planets is jupiter away from the sun?
Instructions: Can you find an answer sentence to this question for me?

Contriever	**TART-full**
✗ Jupiter is the only planet whose barycenter with the Sun lies outside the volume of the Sun, though by only 7% of the Sun's radius.[80] The average distance between Jupiter and the Sun is 778 million km (about 5.2 times the average distance between Earth and the Sun, or 5.2 AU) (Wikipedia paragraph)	✓ Jupiter is the fifth planet from the Sun and the largest planet in the Solar System. (Wikipedia answer sentence)

Query: Who won the final hoh big brother 20?
Instructions: a question similar to this

Contriever	**TART-full**
• ✓ Who won the Final HoH in the American reality show Big Brother 20? (AmbigQA) • ✓ Who won the final vote in the British reality show Celebrity Big Brother 20? (AmbigQA) • ✗ Caleb Reynolds was a castaway on Survivor: Kaôh Rōng; he was medically evacuated from the game, and placed 15th. Nicole Franzel returned as a HouseGuest on Big Brother 18 where she was crowned the winner and became the first female winner to win against a male in the final 2. (Wikipedia paragraph)	• ✓ Who won the final vote in the British reality show Celebrity Big Brother 20? (AmbigQA) • ✓ Who is left in the American big brother house at the end of season 20? (AmbigQA) • ✓ Who won the Final HoH in the American reality show Big Brother 20? (AmbigQA)

Table 12: We compare TART-full outputs with the Contriever-MS MARCO (Izacard et al., 2022) predictions on $\mathbb{X}^2$-Retrieval. We show the top one prediction for the first four examples, and show the top three predictions for the bottom examples. ✓ mean that the documents follow instructions while ✗ mean that the documents do not satisfy the instructions.

	Using instructions		BEIR									
	at training	at test	TREC	NFC	FQA	ARG	TOU	DBP	SCD	CLI	SCF	avg.	best
TART-full	✓	✓	**72.8**	34.6	**42.0**	**50.0**	**35.3**	46.1	**18.4**	35.2	73.7	**44.4**	5
Ablations	✓		61.1	21.9	38.4	39.8	23.6	36.1	15.0	24.7	65.2	36.2	0
		✓	67.6	34.9	40.6	39.5	20.5	**47.1**	17.5	**39.8**	**75.4**	42.5	3
			57.2	**37.1**	41.3	**50.0**	18.3	41.3	18.3	32.5	73.2	41.1	2

Table 13: The full results of the instruction ablations on BEIR. TREC, NFC, FQA, ARG, TOU, DBP, SCD, CLI, SCF indicate TREC-COVID, FIQA, NF Corpus, Arguana, Touche-2020, DBPedia, SciDocs, Climate-Fever, and SciFact, respectively.

	Using instructions		LOTTE	$\mathbb{X}^2$-Retrieval						
	at training	at test		AMB	WQA	SCF	GAT	LSO	CSP	avg.
TART-full	✓	✓	**75.7**	**90.5**	52.5	**66.2**	**68.6**	24.9	**51.4**	**59.1**
Ablations	✓		68.5	59.3	**54.4**	61.7	62.0	15.1	46.8	49.9
		✓	70.5	40.1	47.2	64.0	69.5	**25.5**	43.7	48.3
			69.9	34.5	32.5	60.8	58.2	24.2	49.3	43.3

Table 14: : Instruction ablations on LOTTE (Search pooled) and $\mathbb{X}^2$-Retrieval (pooled) evaluation. AMB, WQA, SCF, GAT, LSO, CSP denotes AmbigQA, WikiQA, SciFact, GooAQ-Technical, LinkSO-Python, and CodeSearchNet-Python, respectively.

Dataset	**Instruction**	**NDCG@10**
SciFact	`Find a scientific paper sentence to verify this questions`	75.4
	`Retrieve a scientific paper abstract to verify this claim`	75.7
	`can you retrieve reliable scientific evidence to check if the following claim is true or not?`	74.3
	`please retrieve evidence for me to verify the following`	73.8
	`a scientific paper sentence supporting or refuting the following statement`	74.7
Touche-2020	`retrieve an argument paragraph to answer this question`	30.6
	`retrieve a paragraph to answer this debate question`	30.9
	`Find a opinion to this debate question`	29.5
	`retrieve an argument paragraph that supports this debate question to this debate question`	31.2
Climate-FEVER	`Retrieve a scientific paper abstract to verify the following claim`	29.3
	`Retrieve a Wikipedia paragraph to answer this question`	30.4
	`Retrieve a Wikipedia paragraph to verify the following claim about climate change`	30.8
	`I want to know if the following claim is true or not. Can you find Wikipedia evidence?`	30.6
	`Find a Wikipedia paragraph to verify the following claim`	30.8

Table 15: Performance on SciFact, Climate-FEVER and Touche-2020 with different instructions.

Dataset	Instruction	NDCG@10
SciFact	✓ `Retrieve a scientific paper abstract to verify this claim`	75.7
	✗ `Retrieve a Wikipedia paragraph to verify the following claim`	74.0
	`[NULL]`	69.1
Arguana	✓ `Retrieve an article that contradict the following paragraph`	50.6
	✗ `Retrieve a Wikipedia paragraph that answers this question`	47.3
	`[NULL]`	39.8
Touche-2020	✓ `Retrieve an argument for this topic`	29.6
	✗ `retrieve a Wikipedia passage that answers this question`	26.7
	`[NULL]`	22.1

Table 16: Full list of the instructions used for evaluations. `[NULL]` means that at inference time, no instruction is given to TART-full. ✓ means a correct instruction, while ✗ means incorrect instructions.

	model size	BEIR										
pretrained models		TREC	NFC	FQA	ARG	TOU	DBP	SCD	CLI	SCF	avg.	best
T5-LM-base	110M	62.9	29.7	33.9	37.8	30.8	38.6	15.1	29.2	70.7	38.7	0
T5-LM-large	385M	**73.3**	**34.2**	40.2	**47.1**	32.8	45.3	18.2	35.2	74.9	43.7	3
T5-LM-XL	1.5B	71.6	33.1	**41.8**	43.1	**34.0**	**46.0**	**18.5**	**38.3**	**75.5**	**44.7**	**6**

Table 17: Zero-shot retrieval results for different sizes of TART-full on BEIR. TREC, NFC, FQA, ARG, TOU, SCD, CLI, SCF indicate TREC-COVID, FIQA, NF Corpus, Arguana, Touche-2020, DBPedia, SciDocs, Climate-Fever, and SciFact, respectively.

	dataset number	BEIR										
pretrained models		TREC	NFC	FQA	ARG	TOU	DBP	SCD	CLI	SCF	avg.	best
T5-LM-XL	5	63.3	28.3	37.6	47.8	24.3	42.3	17.0	30.8	73.4	40.5	0
T5-LM-XL	10	68.8	30.5	39.5	47.5	29.4	**46.7**	**18.2**	26.9	**76.0**	42.6	3
T5-LM-XL	20	**71.0**	**33.7**	**41.7**	**48.7**	**33.2**	46.1	**18.2**	**29.8**	74.7	**44.1**	**6**

Table 18: Zero-shot retrieval results of TART-full on BEIR when different numbers of the datasets are used for training. TREC, NFC, FQA, ARG, TOU, SCD, CLI, SCF indicate TREC-COVID, FIQA, NF Corpus, Arguana, Touche-2020, DBPedia, SciDocs, Climate-Fever, and SciFact, respectively.

	BEIR									
pretrained models	TREC	NFC	FQA	ARG	TOU	DBP	SCD	CLI	SCF	avg.
T0-3B	71.7	**34.0**	**42.2**	49.8	**31.2**	45.1	17.5	30.0	75.8	44.1
FLAN-T5	72.8	33.4	41.8	51.5	24.9	46.8	18.7	**35.4**	**77.7**	**44.8**
Tk-Instruct	65.4	34.7	32.3	44.5	24.3	42.3	19.2	34.0	76.2	41.4

Table 19: Zero-shot retrieval results for TART-full initialized with different pretrained models' encoders on BEIR. TREC, NFC, FQA, ARG, TOU, SCD, CLI, SCF indicate TREC-COVID, FIQA, NF Corpus, Arguana, Touche-2020, DBPedia, SciDocs, Climate-Fever, and SciFact, respectively.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The limitation section.

☑ A2. Did you discuss any potential risks of your work?
The ethical consideration section.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
The Abstract and Introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3.3

☑ B1. Did you cite the creators of artifacts you used?
Sections A.1 Table 6.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Sections A.1 and 3.3.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Sections A.1 and 3.3.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Sections A.1 and 3.3.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Sections 3.3 and A.5.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Sections 3.3 and A.5.

C ☑ Did you run computational experiments?

Sections 4 and 5.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Sections 4 and 5.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sections 4, 5, C, and D.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Not applicable. Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 3.3.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 3.3.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
The annotators of the instructions are the authors of the papers, so we cannot disclose our basic demographics due to the risk of anonymity violations.