Open-domain Question Answering via Chain of Reasoning over Heterogeneous Knowledge

Kaixin Ma[♣†*], **Hao Cheng**[♠*], **Xiaodong Liu**[♠], **Eric Nyberg**[♣], **Jianfeng Gao**[♠]
[♣] Carnegie Mellon University [♠] Microsoft Research
{kaixinm,ehn}@cs.cmu.edu {chehao,xiaodl,jfgao}@microsoft.com

Abstract

We propose a novel open-domain question answering (ODQA) framework for answering single/multi-hop questions across heterogeneous knowledge sources. The key novelty of our method is the introduction of the intermediary modules into the current retriever-reader pipeline. Unlike previous methods that solely rely on the retriever for gathering all evidence in isolation, our intermediary performs a chain of reasoning over the retrieved set. Specifically, our method links the retrieved evidence with its related global context into graphs and organizes them into a candidate list of evidence chains. Built upon pretrained language models, our system achieves competitive performance on two ODQA datasets, OTT-QA and NQ, against tables and passages from Wikipedia. In particular, our model substantially outperforms the previous state-of-the-art on OTT-QA with an exact match score of 47.3 (45 % relative gain).

1 Introduction

The task of open-domain question answering (ODQA) typically involves multi-hop reasoning, such as finding relevant evidence from knowledge sources, piecing related evidence with context together, and then producing answers based on the final supportive set. While many questions can be answered by a single piece of evidence (Joshi et al., 2017; Kwiatkowski et al., 2019), answering more complex questions are of great interest and require reasoning beyond *local* document context (Yang et al., 2018; Geva et al., 2021). The problem becomes more challenging when evidence is scattered across heterogeneous sources, *e.g.,* unstructured text and structured tables (Chen et al., 2020a), which necessitates hopping from one knowledge modality to another. Consider the question in Fig. 2.

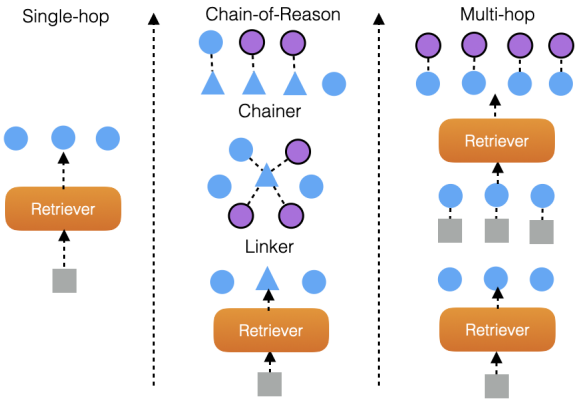

Figure 1: Our CORE vs. previous retriever-only methods for evidence discovery. The grey square is the question, the blue dots are 1st-hop passages, the blue triangles are 1st-hop tables, and the purple dots are 2nd-hop passages.

To form the final answer, a model needs to find the entry list (a table) of the mentioned touring car race, look up the driver name with the correct rank, search for the corresponding driver information, and extract the birthplace from the free-form text.

Existing *retriever-reader* methods (Min et al., 2021, *inter alia*) for ODQA mainly customize the retriever model for tackling individual question types, *i.e.,* exclusively relying on the retriever to gather all necessary context in a query-dependent fashion. As shown in Fig. 1, the single-hop model (Karpukhin et al., 2020) only retrieves a list of isolated passages (blue dots). For multi-hop cases, an iterative retriever looks for a query-dependent path of passages (blue-purple dot chains) (Xiong et al., 2020), *i.e.,* the later hop passages are retrieved using expanded queries including the original question and previous hop passages. Although those retrieval-only methods achieve promising results on their targeted cases, the customized retrievers are unable to generalize well. For example, an iterative passage retriever trained with both multi-hop and single-hop questions performs poorly over both types (Xiong et al., 2020). For real-world applica-

† Part of this work is done during an internship at Microsoft Research
∗ Equal contribution

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5389 - 5403

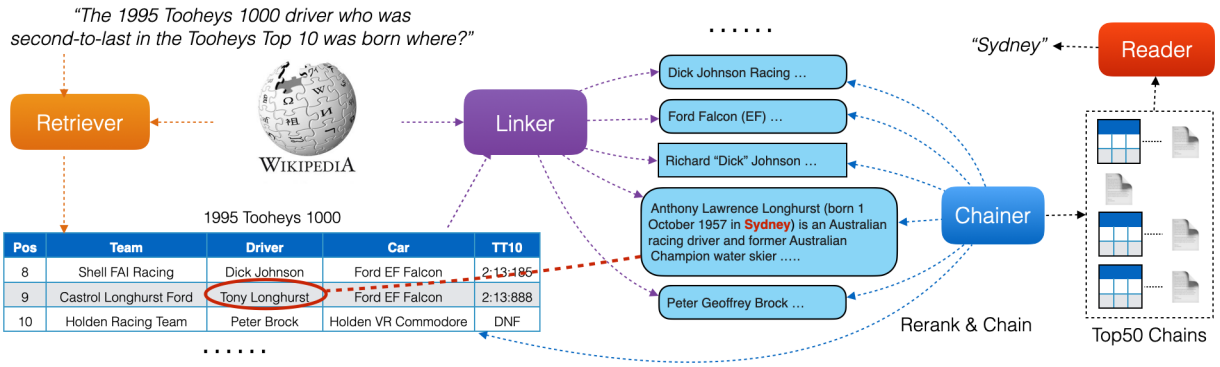

Figure 2: An illustration of CORE for ODQA. Given a question, the *retriever* first finds hop-1 evidence from the entire Wikipedia (orange arrows). Then the *linker* gathers relevant documents for the hop-1 evidence (purple arrows), which are treated as hop-2 evidence. Next, the *chainer* reranks all hop-1 and hop-2 evidence and splices them together into evidence chains (blue arrows). Finally, the *reader* takes in the top-50 chains and produces the answer (black arrows). The gold evidence chain is marked in red.

tions with heterogeneous knowledge sources, it is desirable for an ODQA system to handle both cases well, and the retrieval-only methods are unlikely to succeed.

We propose a novel **C**hain **O**f **RE**asoning (CORE) framework that generalizes well on both single-hop and multi-hop question answering. The main contribution of CORE is the introduction of two intermediary modules, the linker and the chainer, that play the bridging role between the retriever and the reader, *i.e.,* piecing together related evidence with necessary context for single/multi-hop questions. These two modules work in a forward-backward fashion. In the forward pass, the linker, a novel table entity linking model (§3.1), links the raw evidence with its related context (*e.g.,* table-passage graphs in Fig. 1). The chainer, a new unsupervised reranker (§3.2), then prunes all linked evidence using the corresponding question generation scores from a pretrained language model (Sanh et al., 2021) to form a shortlist of relevant evidence chains in a backward noisy channel fashion (*e.g.,* table-passage paths in Fig. 1). By delegating the *hopping* operation to the intermediary, our formalization can potentially gather evidence more effectively over different question types.

To demonstrate the effectiveness of CORE, we evaluated the proposed model on two popular ODQA datasets, the multi-hop dataset OTT-QA (Chen et al., 2020a) and the single-hop dataset NQ (Kwiatkowski et al., 2019). Empirically, we show that our approach is general for both types of reasoning in ODQA. In particular, CORE substantially outperforms the previous state-of-the-art (SOTA) on OTT-QA by 14+ points on exact match scores (45%+ relative gain), and it is competitive with SOTA models on NQ. Moreover, we show that one single unified model can learn to solve both tasks under our framework. From our analysis, we also find that our evidence chains can potentially help answer single-hop questions by enriching the evidence with more supportive context. [1]

2 Overview of the CORE Framework

The CORE framework is designed to answer questions where the answer is a contiguous span from a table t or a passage p. Here neither t nor p is given, so they need to be retrieved from the table corpus $\mathbb{C}_t$ and the passage corpus $\mathbb{C}_p$, respectively. For single-hop questions, a single t or p may be sufficient, whereas for multi-hop questions, one or more t and p are required to find the answer.

As shown in Fig. 2, CORE consists of a *retriever*, a *linker*, a *chainer* and a *reader*. We adopt the DPR model (Karpukhin et al., 2020) as our retriever. We only briefly describe the retriever here as it is not the main focus of our work. The DPR is a bi-encoder model that consists of a question encoder and a context encoder. For our setting, the questions and passages/tables are represented by the [CLS] embedding produced by their respective encoder, and the retrieval is done by maximum inner product search in the vector space. For a given question, we use DPR to retrieve the initial evidence set which includes tables and passages.

Given the initial evidence set (*e.g.,* the car race

[1] Data and code available at https://github.com/Mayer123/UDT-QA

entry list table in Fig. 2), our intermediary module produces a list of query-dependent evidence chains (*e.g.,* the red line linked evidence chain consisting of the car race entry list and the driver's Wikipedia page). We first propose a linker model (§3.1) to expand the candidate evidence set by including extra passages related to tables in the initial set (purple arrows in Fig. 2). This step allows the model to enrich the evidence context, especially including reasoning chains needed for multi-hop questions. Since there could be many links between a piece of evidence and others (*i.e.,* a densely connected graph), considering all links is computationally infeasible for the downstream reader. Thus, we develop a chainer model (§3.2) to prune the evidence graph with the corresponding question and then chain the evidence across hops together to form query-dependent paths. Here, we only keep top-K scored chains for reading so that the reader can work on a fixed computation budget.

Finally, the Fusion-in-Decoder (FiD) (Izacard and Grave, 2021), a T5-based generative model (Raffel et al., 2019), is used as the reader for generating the final answer. The model first encodes each top-K evidence chain independently along with the question. During decoding, the decoder can attend to all chains, thus fusing all the input information.

3 Intermediary Modules

In this part, we present the two key components of `CORE` for supporting multi-hop reasoning, *i.e.,* the linker for building evidence graphs and the chainer for forming query-dependent paths.

3.1 Linker

In this work, we mainly focus on linking an entity mention in the retrieved evidence to the corresponding Wikipedia page for building evidence graphs. This setup is related to the recent entity linking work (Wu et al., 2020). However, there are important modifications for ODQA. In particular, instead of assuming the entity mention as a prior, we consider a more realistic end-to-end scenario for ODQA: the linker model has to first *propose candidate entity mentions (spans)* for a given evidence (*e.g.,* "Tony Longhurst" in Fig. 2), and then *links the proposed mention* to its Wikipedia page. Another major difference is that we study entity mentions in tables instead of text. As tables contain more high-level summary information than text, using tables as pivots for constructing evidence graphs can potentially help improve the recall of evidence chains for QA. In the meanwhile, this task is challenging due to the mismatch between the lexical form of the table cells and their linked passage titles. For example, the table of "NCAA Division I women's volleyball tournament" contains the cell *VCU*, which refers to *VCU Rams* instead of *Virginia Commonwealth University*. Thus simple lexical matching would not work.

In the following, we first describe the model for entity mention proposal and then present a novel entity linking model for mentions in tables. Both models are based on a pretrained language model, BERT (Devlin et al., 2019). Following previous work (Oguz et al., 2020), we flatten the table row-wise into a sequence of tokens for deriving table representations from BERT. In particular, we use $x_1, \ldots, x_N$ to denote an input sequence of length N. Typically, when using BERT, there is a prepended token `[CLS]` for all input sequences, *i.e.,* $\texttt{[CLS]}, x_1, \ldots, x_N$. Then the output is a sequence of hidden states $\mathbf{h}_{\texttt{[CLS]}}, \mathbf{h}_1, \ldots, \mathbf{h}_N \in \mathbb{R}^d$ from the last BERT layer for each input token, where d is the hidden dimension.

Entity Mention Proposal In realistic settings, the ground truth entity mention locations are not provided. Directly applying an off-the-shelf named entity recognition (NER) model can be sub-optimal as the tables are structured very differently from the text. Thus, we develop a span proposal model to label the entity mentions in the table. Specifically, we use BERT as the encoder (BERT^{m}) and add a linear projection to predict whether a token is part of an entity mention for all tokens in the table,

$$\mathbf{h}^{\text{m}}{}_1, \ldots, \mathbf{h}^{\text{m}}{}_N = BERT^{\text{m}}(t_1, \ldots, t_N), \quad (1)$$

$$\hat{\mathbf{y}} = W\mathbf{h}^{\text{m}}, \quad (2)$$

where $\mathbf{h}^{\text{m}} \in \mathbb{R}^{N \times d}$ and $W \in \mathbb{R}^{2 \times d}$. The model is trained with a token-level binary loss

$$-\frac{1}{N}\sum_{N}^{n=1}(y_n \log P(\hat{\mathbf{y}})_1 + (1 - y_n) \log P(\hat{\mathbf{y}})_0), \quad (3)$$

where y_n is the 0-1 label for the token at position n, and $P(\cdot)$ is the softmax function.

Table Entity Linking Once the candidate entity mentions are proposed, we follow Wu et al. (2020) to use a bi-encoder model for linking. Similarly, two BERT models are used to encode tables (BERT^{t}) and passages (BERT^{p}), respectively.

In contrast, as there are multiple entity mentions for each table, we want to avoid repetitively inserting additional marker tokens and re-computing representations for each mention occurrence. Accordingly, we cannot simply take the `[CLS]` embeddings for linking as previous work (Wu et al., 2020). Inspired by Baldini Soares et al. (2019), to represent each entity mention, we propose a new entity representation based on the entity start and end tokens. For an entity mention with a start position i and end position j in the table, we compute our proposed entity embedding $\mathbf{e} \in \mathbb{R}^d$ for linking by

$$\mathbf{e} = (\mathbf{h}^{\mathrm{t}}{}_i + \mathbf{h}^{\mathrm{t}}{}_j)/2, \tag{4}$$

For passages, we directly take the `[CLS]` hidden state $\mathbf{p} = \mathbf{h}^{\mathrm{p}}{}_{\texttt{[CLS]}} \in \mathbb{R}^d$ as the passage representation. Following the literature, the table entity linker is trained based on a contrastive learning objective

$$L_{\text{sim}} = -\frac{\exp(\text{sim}(\mathbf{e}, \mathbf{p}^+))}{\sum_{\mathbf{p}' \in \mathcal{P} \cup \{\mathbf{p}^+\}} \exp(\text{sim}(\mathbf{e}, \mathbf{p}'))}, \tag{5}$$

where $\mathbf{p}^+$ is the corresponding (positive) passage and $\mathcal{P}$ is the irrelevant set of negative passages.

Training To train our linker including entity mention proposal and table entity linking, we leverage a small set of Wikipedia tables with ground truth hyperlinks, where each linked mention and the first paragraph of the linked page constitute a positive pair (see Appendix C). Similar to Wu et al. (2020), we use BM25 (Robertson and Walker, 1994) to mine hard negative pairs for each entity mention.

Inference During inference, we first use the entity span proposal model to label entity mentions in the tables and then run the table entity linking model to link predicted mentions to passages via maximum inner product search (Johnson et al., 2019). Here, we allow searching over all passages rather than the first paragraph of each Wikipedia page for recall purposes. For efficient inference, the passage corpus can be pre-computed and stored as done in dense retrieval (Karpukhin et al., 2020). Then we can build links for all tables in the corpus (inducing graph-structured corpus), and directly use the graph to enrich the isolated evidence items from the retriever.

3.2 Chainer

Although the linker model can effectively enrich the table with all relevant passages and provide possible hopping paths for reasoning, the amount of resulting information is too overwhelming if sent directly for reading, *i.e.,* a table could contain multiple entity mentions, resulting in a densely connected evidence graph. To make it easy for the reader, we use the chainer model to prune graphs and extract the most relevant paths. Since the linker (§3.1) builds the evidence graphs in a query-agnostic fashion, an important function of chainer here is to incorporate the question when selecting the top-K evidence paths for answer inference.

Motivated by recent work (Sachan et al., 2022) in using pretrained generative language models for passage reranking, we also build the chainer on T0 (Sanh et al., 2021) in a zero-shot fashion, *i.e.,* no training is required. Different from their approach, we design a novel relevance scoring for query-dependent hybrid evidence path reranking rather than isolated passages.

Given a question q, we model the relevance of a question-table-passage path using the following conditional

$$P(t, p|q) = P(p|t, q)P(t|q), \tag{6}$$

where $t \in \mathbb{C}_{\mathbb{T}}$, $p \in \mathbb{C}_{\mathbb{P}}$. The second term $P(t|q)$ is modeled by our retriever. Given that our linker is query-agnostic (*i.e.,* only modeling $P(p|t)$), we do not have a good estimation for $P(p|t, q)$ on the right-hand side. To remedy this, we apply the Bayes rule to Equation 6

$$P(t, p|q) \approx P(q|t, p)P(p|t)P(t|q). \tag{7}$$

To estimate $P(q|t, p)$, we use the question generation likelihood as in Sachan et al. (2022).

Different from Sachan et al. (2022), here we have two conditional variables. Naively computing question generation scores on all pairs results in quadratic complexity which is very computation intensive for T0.[2] To reduce the inference cost, we further decompose $P(q|t, p)$ into two question generation scores $S_{T0}(q|p)$ and $S_{T0}(q|t)$, both based on the question generation likelihood from T0. In this way, we can reuse $S_{T0}(q|t)$ for corresponding linked passages with a linear cost. To compute $S_{T0}(q|p)$ and $S_{T0}(q|t)$, we append the instruction *"Please write a question based on this passage."*[3] to every passage/table and use the same mean log-likelihood of the question tokens conditioned on the passage/table (Sachan et al., 2022).

[2] Even the smallest T0 model has 3B parameters.

[3] Changing "passage" to "table" in the prompt does not make much difference.

As our pilot study suggests that the query-agnostic linker scores are not so informative for query-table-passage paths, we only combine the retriever score with two question generation scores from Equation 7 as the final chainer score for reranking evidence paths (*i.e.,* $P(p|t)$ is dropped)

$$S_R(t,q) + \alpha S_{T0}(q|t) + \beta S_{T0}(q|p), \quad (8)$$

where $S_R(t,q) \sim P(t|q)$, and is defined as

$$S_R(t,q) = -\log(\frac{\exp(\text{sim}(t,q))}{\sum_{t_i \in \mathcal{T}} \exp(\text{sim}(t_i,q))}), \quad (9)$$

$\text{sim}(\cdot,\cdot)$ is the unnormalized retriever score, $\mathcal{T}$ is the first-hop evidence set, α and β are hyper-parameters. For singleton cases (first-hop table/passage without link), we modify the last two terms of Equation 8 to $2\alpha S_{T0}(q|t)$ and $2\alpha S_{T0}(q|p)$ for tables and passages, respectively. This can help ensure that the chainer scores for singletons and table-passage paths are on the same scale. Then we sort all singletons and chains using the chainer score and keep the top-k results. We also use heuristics to reduce potential duplication and details are in Appendix A.

4 Experiments

In this section, we first describe datasets and knowledge sources (§4.1). Then we discuss single-set and joint-set experiment setups (§4.2) and baselines for comparison (§4.3). Finally, we present experiment results on OTT-QA and NQ (§4.4).

4.1 Datasets

OTT-QA (Chen et al., 2020a) is an open-domain QA dataset that contains mostly multi-hop questions. These questions require joint reasoning over evidence from tables and text passages.

Natural Questions (NQ) (Kwiatkowski et al., 2019) contains real user queries submitted to the Google search engine and the questions are mostly single-hop and considered solvable using either a single text passage or a table. We adopt the open-domain setting proposed by Lee et al. (2019).

For OTT-QA, we adopt its official release of text passages and tables. The passages are first paragraphs from all Wikipedia pages, whereas tables are well-formed tables from full Wikipedia (*i.e.,* no infoboxes, no multi-column/multi-row tables, etc.). For NQ, we adopt the Wikipedia passage splits from Karpukhin et al. (2020) and we use the processed table sets of full Wikipedia released by Ma et al. (2022). Statistics can be found in Appendix B.

	Dev		Test	
	EM	**F1**	**EM**	**F1**
HYBRIDER (Chen et al., 2020b)	10.3	13.0	9.7	12.8
FR+CBR(Chen et al., 2020a)	28.1	32.5	27.2	31.5
CARP (Zhong et al., 2022)	33.2	38.6	32.5	38.5
Oracle Link + FR+CBR	35.2	39.9	35.0	39.5
Oracle Link&table + HYBRIDER	44.1	50.8	43.0	49.8
CORE (single)	**49.0**	**55.7**	46.7	53.5
CORE (joint)	**49.0**	**55.7**	**47.3**	**54.1**

Table 1: End-to-end QA results on OTT-QA.

4.2 Experimental Settings

We train a single **linker** on the OTT-QA training set, and directly apply it for both tasks. For **chainer**, we apply the off-the-shelf T0-3B (Sanh et al., 2021) for reranking on both tasks, *i.e.,* no updates to the model. Hence both models are task-independent.

For both **retriever** and **reader**, we consider both *single-set* setup where separate models are trained for individual tasks and the *joint-set* setting where a single model is trained to solve both tasks. During inference, the retriever searches over task-specific knowledge sources. For OTT-QA, since most of its questions have tables as the first-hop evidence, we run the retriever only on the table set in the first hop to find top-100 tables. For NQ, the retriever searches over the joint index of all text and tables and keeps the top-100 items in the first hop. After retrieval, we use the linker to expand tables in the top-100 set into graphs and use the chainer to select the top-50 chains. As the chained evidence is typically longer, unless otherwise specified, we only use the top-50 chains and set the maximum sequence length to 500 for the FiD reader. More training details are in Appendix C.

4.3 Baselines

We briefly describe the baselines for both tasks. For OTT-QA, the HYBRIDER (Chen et al., 2020b) is a reading comprehension model for joint reasoning over tables and passages. This baseline uses BM25 to retrieve relevant tables and passages. Instead, Fusion Retriever + Cross-Block Reader (FR+CBR) (Chen et al., 2020a) first links table rows to passages using BM25 to build an index of linked documents (blocks). Then it trains a bien-coder dense retriever (further enhanced by Inverse

	Tables	EM
DPR (Karpukhin et al., 2020)	N	41.5
FiD (Izacard and Grave, 2021)	N	51.4
UnitedQA (Cheng et al., 2021b)	N	51.8
Unik-QA (Oguz et al., 2020)	Y	54.1
UDT-QA (Ma et al., 2022)	Y	**54.7**
CORE (single)	Y	54.6
CORE (joint)	Y	53.9

Table 2: End-to-end QA results on NQ test.

Cloze Task pretraining (Lee et al., 2019)) to find the most relevant blocks. Then they use the ETC (Ainslie et al., 2020) as the reader to process up to 4K tokens returned by the retriever. CARP (Zhong et al., 2022) employs similar retriever and reader models as FR+CBR, and it additionally extracts hybrid knowledge chains to facilitate the reader's reasoning process. All approaches are dataset-specific, hence are unlikely to handle other question types from NQ. We note that both FR+CBR and CARP's reader components adopt models specifically designed for handling long sequences. Since our overall goal is to build a unified system for both single-hop and multi-hop questions, we choose FiD based on its ability to handle different cases. We also compare against two oracle settings from Chen et al. (2020a). The first one uses gold hyperlinks in fusion retriever instead of BM25 to link table rows and passages. The second setting adopts gold hyperlinks and gold tables for the HYBRIDER reader, *i.e.,* no retrieval is involved. This setting is previously considered as an estimated upper bound for this task (Chen et al., 2020a).

On NQ, we compare with text-only baselines: DPR (Karpukhin et al., 2020) which applies a BERT-based reader to first select the best passage from top-k returned by the retriever and then extracts the answer span in it; FiD (Izacard and Grave, 2021); and UnitedQA (Cheng et al., 2021b)) which is an extractive model based on ELECTRA (Clark et al., 2020) and enhanced with additional training objectives (Cheng et al., 2021a, 2020). We also compare with models that consider tables as knowledge sources: Unik-QA (Oguz et al., 2020) which augments the document index with NQ tables and uses FiD reader to generate the answer; UDT-QA (Ma et al., 2022) which incorporates the tables from full Wikipedia and adopts UnitedQA as its reader.

4.4 Results

The end-to-end results on OTT-QA are shown in Table 1. Our CORE models substantially outperform all baselines by a large margin, illustrating the effectiveness of our proposed framework. It is worth noting that our model also outperforms two oracle settings proposed in Chen et al. (2020a). This is potentially because the capacity of their reader models is quite limited compared to ours. For FR+CBR, the model can only read up to 4K tokens, whereas in our case, the FiD reader can process up to 25K tokens. We also observe that evidence to questions in OTT-QA can be found in many different reasoning chains. In other words, tables that are not annotated as gold may still provide valuable information for reasoning. However, only one table is considered in their oracle experiment with HYBRIDER. Also, it is worth noting that the joint model outperforms the single model on OTT-QA, indicating that our framework can effectively leverage NQ data to benefit OTT-QA.

Table 2 summarizes the results on NQ. Similar to previous work using tables as the extra knowledge source, we also find our method to be consistently better than text-only baselines. Overall, CORE achieves competitive performance compared to SOTA models. It is also worth noting that both Unik-QA and UDT-QA use the iterative training strategy for the retriever, which leads to higher retriever performance. In particular, UDT-QA achieves 91.9 recall@100 on the NQ test set, whereas we only get 90.3. This difference in retriever recall likely explains the gap in the end-to-end QA performance. Since iterative training of the retriever is not used by baselines on OTT-QA, we leave that out in our experiments. For consistency, we only train our retriever for one round. Also, we note that the joint model performs slightly worse than the single-dataset model, which is different from the trend observed on OTT-QA. We hypothesize that this is due to the distribution difference of the evidence chains in the two datasets. Most of the top-ranked evidence are chained cases on OTT-QA, whereas that reduces to only 30% on NQ (the rest 70% are singleton cases). The current FiD reader may have a hard time reasoning over both singleton and chained cases simultaneously. We leave further exploration for future work.

	#Docs	Max Length	EM
CORE (single)	50	500	49.0
no QGS of hop1	50	500	45.1
no QGS of hop1	100	300	44.8
no QGS of hop1&2	100	300	38.7
no Chainer	100	300	29.1

Table 3: Chainer ablation on OTT-QA dev.

# Docs	Avg # Tokens	EM	F1
50	11,897	**49.0**	**55.7**
20	4,757	44.9	51.8
10	2,351	40.8	47.0

Table 4: Reader ablation with different number of documents on OTT-QA dev.

5 Analysis

In this section, we conduct ablation studies and analyze different components of our framework.

5.1 Ablation Study

We start by ablating our chainer on the OTT-QA task using CORE (single) from Table 1. The results are summarized in Table 3. First, we experiment with **removing question generation score (QGS) for hop1 evidence** (row 2). In this case, Equation 8 will not have the second term and a large performance drop is observed, suggesting that it is important to apply chainer to reweight the retrieved items. Under the same setting, we try to increase the number of items and decrease the max sequence length for reading (row 3). Here we are interested in seeing the impact of reading more chains while constraining the length of individual chains. The slight performance drop indicates that potential information loss from longer chains is detrimental to the final performance.

Then we test **removing QGS entirely** (row 4). As there is no reranking at all, we simply take all linked passages for each table based on the initial retrieved order. Here, since we exhaustively include all linked passages for each table, the reader budget can be quickly filled by the top few retrieved tables, which probably makes the reader suffer from information loss with too many irrelevant items. Indeed we see another large drop in the QA performance, showing the importance of reranking. Finally, we experiment with **removing chainer** (row 5). It goes one step further by not concatenating the table rows with linked passages and only passing each piece of evidence independently. Here we are interested in whether FiD is able to fuse the information without explicit chaining. The largest performance drop is observed here, suggesting that chaining the items from different hops is vital for the reader.

We also study the effect of the number of retrieved items as input to the reader on its QA performance by varying **the number of chains sent to the reader**. The results on OTT-QA dev set are shown in Table 4. As we can see, when reading only the top 10 chains (much less than 4K tokens on average considered by previous work (Chen et al., 2020a; Zhong et al., 2022)), CORE still outperforms the previous SOTA method by a large margin, further validating the effectiveness of our framework. In summary, the reader performance increases with more items, and we hypothesize that the reader can be further improved when given a larger capacity. We leave this exploration for future work.

5.2 Impact of Linker & Chainer

We measure evidence recall scores of our joint model to better understand the impact of the linker and the chainer. On OTT-QA, since most questions require multi-step reasoning, we evaluate the retriever by both gold cell recall and answer recall. Here the gold cells refer to table cells whose gold linked passage contains the answer. We consider a table chunk to be gold if it contains at least one gold cell. For answer recall, we use top-K chains produced by the chainer. Since OTT-QA official test set is hidden, we only report dev set results in Table 5. As expected, the answer recall scores are extremely low without the linker and chainer, as most OTT-QA questions are multi-hop and the answers are likely to appear in the hop-2 passages. To verify if the retriever itself is able to discover the full reasoning chain, we further allow the retriever to search through the joint index of tables and passages (last row). Compared with the first row, there is some improvement. On the other hand, with the help of the linker and chainer (row 2), AR@20 and AR@50 are substantially improved, which also explains our superior QA performance. We also observe similar trends on NQ, and more results and discussion are in Appendix D.1.

5.3 Alternative Linking Strategy

Since most NQ questions only require single-hop reasoning, one alternative linking strategy for NQ is to skip the linker for single-hop ones. As the

	R@20	R@100	AR@20	AR@50
Joint Retriever	**83.8**	**92.1**	31.8	37.6
+Linker&Chainer	83.5	90.8	**74.5**	**82.9**
Retrieve full index	82.4	90.6	34.6	42.6

Table 5: Evidence recall of the joint retriever on OTT-QA dev set, where R@K evaluates gold table chunk recall and AR@K evaluates answer recall.

	# Link	# No Link	AR@20	AR@50
CORE	2,214	0	74.5	82.9
Classifier	2,133	81	73.3	81.9
CORE	8,757	0	85.7	88.1
Classifier	370	8,387	86.0	88.1

Table 6: Answer recall on OTT-QA (top) and NQ (bottom) dev with different linking strategies.

question type is typically unknown in real cases, we experiment with training a question classifier to predict whether a question requires multi-step reasoning or not. We directly use the encoded question representation produced by our joint retriever, and train a linear classifier for this task. For simplicity, we consider all NQ questions to be single-hop and all OTT-QA questions to be multi-hop. Based on this, the classifier can achieve 95.9% accuracy on the combined dev set of NQ and OTT-QA. We then proceed to compute answer recall for both NQ and OTT-QA using the classifier to decide the linking strategy for each question. The results are shown in Table 6. We observe the difference in answer recall between CORE and that using the question classifier is quite small. Thus, future work can also use this strategy for handling different types of questions.

5.4 Linker Performance

The success of our framework depends on the linking quality, thus we also evaluated our linker model as a standalone module to better understand its performance. Here we use the 789 tables in the OTT-QA dev set with ground truth hyperlinks for evaluation. For metrics, we compute precision, recall and F1 score for finding all the links for each table and only consider the top-1 retrieved results in all settings. We compare the cell linker model proposed by Chen et al. (2020a) as a baseline. In this model, they first train a GPT-2 (Radford et al., 2019) model on OTT-QA training set to generate queries for every cell in the table (empty if the model decides to not link a certain cell), and then use BM25 to retrieve passages. In addition, we also consider a baseline that uses ground truth table cells (cells that have links) as queries and retrieve with BM25. The results are shown in Table 8. Our linker achieves the best F1 score compared to the two baselines, and the advantage on recall is especially prominent. The oracle+BM25 model has the best precision because the information for whether a cell requires linking is given as a prior. However, it cannot retrieve well when the passage title does not overlap significantly with the cell text, as discussed in (§3.1). The GPT2 model can alleviate this issue to some extent by generating additional terms for matching, however, it still lags behind our proposed linker model.

5.5 Case Study

We manually inspect evidence chains found by our model to better understand the benefits of our intermediary modules. Examples where predicted chains contain supportive evidence are in Table 7.

Despite that NQ questions are usually short and single-hop, we posit that some questions can potentially benefit from the proposed chain of reasoning. As illustrated by the first two examples, though the entity mentions from the tables are not directly relevant, the table-passage chains actually contain the supportive information. Therefore, in contrast to considering the evidence in isolation, our way of constructing query-dependent table-passage chains is likely to improve the density of relevant information for single-hop questions.

For the OTT-QA chains, there is little overlap between the question and the target entity passage as expected. Thus, it is relatively difficult for the retriever to succeed in gathering all relevant information alone. On the contrary, our framework can effectively handle it with our proposed linking and chaining operations.

6 Related Work

Multi-hop question answering (QA) has been studied extensively for both knowledge base (KB) setting (Yih et al., 2015; Zhang et al., 2017; Talmor and Berant, 2018) and open-domain setting (Yang et al., 2018; Feldman and El-Yaniv, 2019; Geva et al., 2021). For KB-based setting, the models are trained to parse questions into logical forms that can be executed against KB (Das et al., 2021; Yu et al., 2022) or directly select entities in KB as answers (Sun et al., 2019). On the other hand, the open-domain setting requires the model to retrieve

Q&A	Evidence Chain
Q: who plays ryders mum on home and away **A:** Lara Cox	**Table Title:** List of Home and Away characters ‖ character, actor(s), duration ‖ *ryder jackson*, lukas radovich, 2017- ‖ *Ryder Jackson* is a fictional character from the Australian television soap opera "Home and Away" ... **His mother is Quinn Jackson (Lara Cox)**, who is estranged from Alf. When ...
Q: what was blur 's first number one single in the uk **A:** Country House	**Table Title:** List of UK top-ten singles in 1994 ‖ artist, weeks, singles ‖ blur, 3, *"girls & boys"*, "parklife" ‖ *Girls & Boys* (Blur song) is a 1994 song by British rock band Blur. It was released ... "Girls & Boys" was Blur's first top 5 hit and their most successful single until **"Country House" reached number 1** the following year ...
Q: Who is the dad of the cyclist that placed directly behind Marieke van Wanroij at the 2011 Holland Hills Classic ? **A:** Hans Daams	**Table Title:** Holland Hills Classic ‖ Year, First, Second, Third ‖ 2011, Marianne Vos, Marieke van Wanroij, *Jessie Daams* ‖ *Jessie Daams* Jessie Daams (born 28 May 1990) is a Belgian racing cyclist . She competed in the 2013 UCI women 's road race in Florence . **Her father is the Dutch cyclist Hans Daams** .
Q: What other team did the Cuban player on the 2012 Charlotte Eagles team play for ? **A:** Wichita Wings	**Table Title:** 2012 Charlotte Eagles season ‖ No, Position, Player, Nation ‖ 19, Midfielder, *Miguel Ferrer*, Cuba ‖ *Miguel Ferrer* (footballer) Miguel Ferrer (born March 28 , 1987) is a Cuban footballer who **played for the Wichita Wings** in the Major Indoor Soccer League .

Table 7: Example evidence chains found by our CORE, where ‖ separates tables and passages. The answer evidence is **bold** and the linked entity mention in the table is *italic*. The first two are from NQ and the latter are from OTT-QA.

	Precision	Recall	F1
Oracle cell + BM25	**61.7**	51.0	55.9
GPT2+BM25	59.9	58.3	59.1
Our Linker	60.3	**63.0**	**61.6**

Table 8: Linker variants on OTT-QA Dev set tables. In total, there are 20,064 unique passages attached to these 789 tables

multiple pieces of evidence from a textual corpus and then produce the answer (Nie et al., 2019; Zhu et al., 2021). Our work falls into the latter category.

One stream of open-domain work uses multiple rounds of retrieval, where the later rounds depend on the previous ones. It is typically achieved by some form of query reformulation, *e.g.,* expanding the question with previous passages (Xiong et al., 2020; Zhao et al., 2021) or re-writing the query using relevant evidence keywords (Qi et al., 2019, 2021). The other line of work directly leverages the gold graph structure to expand the initial set of passages for hopping (Ding et al., 2019; Asai et al., 2019; Zhang et al., 2021), assuming the presence of oracle hyperlinks. Different from those customized text-only multi-hop methods, our approach constructs evidence graphs on-the-fly and we show that it can handle single/multi-hop questions with a unified model over heterogeneous knowledge sources. Moreover, we do not presume the existence of gold hyperlinks in the corpus, making our model more applicable in realistic settings.

There are also recent efforts in leveraging structured knowledge for ODQA. Li et al. (2021) proposed to leverage information from both text and tables to generate answers and SQL queries. Oguz et al. (2020) studied the benefits of both tables and knowledge bases on a set of ODQA tasks. Ma et al. (2022) introduced a unified knowledge interface that first verbalizes tables and KB sub-graphs into text and then uses a single retriever-reader model to handle all knowledge sources. Similarly, we also consider heterogeneous knowledge sources for ODQA. Instead of developing task-specific models and considering the evidence in isolation, we focus on finding evidence paths across different knowledge types for single/multi-hop questions.

7 Conclusion

In this paper, we present a new framework, CORE, for ODQA over heterogeneous knowledge sources. With the novel task-agnostic intermediary module, CORE can effectively handle single/multi-hop tasks using a unified model and achieve new SOTA results on OTT-QA. Our analyses show that the intermediary module is necessary to achieve good results. For future work, it would be interesting to apply our framework to other complex reasoning tasks such as fact verification (Thorne et al., 2018; Aly et al., 2021) or commonsense QA (Talmor et al., 2022).

8 Limitations

We identify the limitations of our study below.

Conceptually, our proposed linker model is generic and it should be able to build edges between documents regardless of their type, e.g. passage-to-passage, passage-to-table. However, in this paper, we focus on one instantiation of the linker model, linking tables to passages. It would be interesting to build a more generic linker that is able to handle different edge types.

Although we have experimented with OTT-QA and NQ, which exhibit very different question styles and reasoning types, both tasks are built on Wikipedia. As most pretrained language models used in our work are also trained using Wikipedia, there are potential issues in generalization. It would be interesting to apply our framework to other domains and test its generalizability, *e.g.,* biomedical domain.

One major bottleneck we found in our experiments is computation. Since we use the T0-3B model (which is the cheaper one, T0-11B is the recommended model by Sanh et al. (2021)) as our chainer, it incurs a very large memory footprint and computation cost even just for inference. Moreover, the FiD reader model is built on T5-large. Encoding top-50 chains is still computationally expensive. In our case, even with a per GPU batch size of 1, the model still cannot run on V100-32G GPUs, thus we had to resort to gradient checkpointing, leading to longer running time. It would be interesting to experiment with other alternatives that require less computation.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W. Cohen. 2020a. Open question answering over tables and text.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Hao Cheng, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2020. Probabilistic assumptions matter: Improved models for distantly-supervised document-level question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5657–5667, Online. Association for Computational Linguistics.

Hao Cheng, Xiaodong Liu, Lis Pereira, Yaoliang Yu, and Jianfeng Gao. 2021a. Posterior differential regularization with f-divergence for improving model robustness. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1078–1089, Online. Association for Computational Linguistics.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021b. UnitedQA: A hybrid approach 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 (Volume 1: Long Papers)*, pages 3080–3090, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021. Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9594–9611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309, Florence, Italy. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

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 (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. 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 (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Alexander Hanbo Li, Patrick Ng, Peng Xu, Henghui Zhu, Zhiguo Wang, and Bing Xiang. 2021. Dual reader-parser on hybrid textual and tabular evidence 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 (Volume 1: Long Papers)*, pages 4078–4088, Online. Association for Computational Linguistics.

Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022. Open domain question answering with a unified knowledge interface. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1620, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen tau Yih. 2021. NeurIPS 2020 EfficientQA competition: Systems, analyses and lessons learned.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the importance of semantic retrieval for machine reading at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2553–2566, Hong Kong, China. Association for Computational Linguistics.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering.

Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering complex open-domain questions through iterative query generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2590–2602, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Stephen E. Robertson and Stephen Walker. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proc. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, Dublin, Ireland.

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.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, 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, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2022. Commonsenseqa 2.0: Exposing the limits of ai through gamification.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2020. Answering complex open-domain questions with multi-hop dense retrieval.

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*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Wang, Zhiguo Wang, and Bing Xiang. 2022. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases.

Xinyu Zhang, Ke Zhan, Enrui Hu, Chengzhen Fu, Lan Luo, Hao Jiang, Yantao Jia, Fan Yu, Zhicheng Dou,

Zhao Cao, and Lei Chen. 2021. Answer complex questions: Path ranker is all you need. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 449–458, New York, NY, USA. Association for Computing Machinery.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2017. Variational reasoning for question answering with knowledge graph.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. 2021. Multi-step reasoning over unstructured text with beam dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4635–4641, Online. Association for Computational Linguistics.

Wanjun Zhong, Junjie Huang, Qian Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022. Reasoning over hybrid chain for table-and-text open domain qa.

Yunchang Zhu, Liang Pang, Yanyan Lan, Huawei Shen, and Xueqi Cheng. 2021. Adaptive information seeking for open-domain question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3615–3626, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dataset	OTT-QA	NQ
Train	41,469	79,168
Dev	2,214	8,757
Test	2,158	3,610
# Tables	400K	2.4M
# Passages	6.1M	21M

Table 9: Statistics of Datasets

A Chaining Strategy

At the chaining step, a table t can appear multiple times in this evidence chain list, if there are multiple passages linked to it. Also, a single passage can be linked to multiple tables, resulting in potential redundancy. Moreover, concatenating the table with the passage may make the sequence too long for the reader to handle. To reduce the duplication and sequence length, we adopt the following strategy: for a document in the first hop, we only add it to the Top-K list if it has not been included before; similarly, for a linked passage in the second hop, we only added it to the Top-K if it has not been added before, and we only concatenate the table header and its entity mention's row to it, and then include it as a separate document. We iterate over the reranked list of chains until reaching K chains.

B Dataset Statistics

The dataset statistics are shown in Table 9. For both datasets, we follow Ma et al. (2022) to split tables into chunks of approximately 100 words, which results in about two chunks for each table. Thus the size of the table index also increased. (840K table chunks and 4.4M table chunks for OTT-QA and NQ, respectively)

C Training Details

We run all of our experiments once due to computation constraints, and we largely follow previous works' hyper-parameters settings.

C.1 Linker Training

Only tables in OTT-QA train and dev set contain hyperlinks (8K for train, 0.8K for dev), and we only used these tables [4] to train our linker model (§3.1).

To mine hard negatives for linker training, we adopted two strategies using BM25. For the first strategy, we used the entity mentions in the table are queries, and retrieve from an index of passage titles only. In the second one, we used the entity mentions along with the table title as queries, and retrieve from an index of passage titles concatenated with the first sentence from their corresponding pages. We observe very different negatives from these two strategies and we used negatives from both to train our linker model.

During training, we use randomly sample one hard negative for every (mention, positive passage) pair, and also use in-batch negatives to compute the contrastive loss. We train entity mention proposal model for 40 epochs and table entity linking model for 100 epochs, both with batch size 64, learning rate 2e-5, and linear warm-up scheduler.

C.2 Retriever Training

For training data, we directly used the data released by Karpukhin et al. (2020) for NQ [5] and Ma et al. (2022) for NQ-table-answerable set [6]. Note that the NQ-table-answerable data is considered part of NQ in all settings. For OTT-QA, each question is annotated with one or more positive tables and we use BM25 to mine hard negatives from an index of all OTT-QA tables. Following previous work (Karpukhin et al., 2020), we train our retrievers (for both single-dataset and joint setting) for 40 epochs, with batch size 128, learning rate 2e-5 and select the best checkpoint using validation average rank on dev set.

C.3 Chainer Inference

We set $\alpha = 16$ and $\beta = 9$ in Equation 8 for OTT-QA, and $\alpha = 10$ and $\beta = 12$ for NQ. These hyper-parameters are selected based on the respective dev set's answer recall performance. We did a grid search over integer values in the range [1, 20] for both parameters.

C.4 Reader Training

Following Izacard and Grave (2021), we train our reader models for 15000 steps, with batch size 64, learning rate 5e-5.

D Results and Discussion

D.1 Impact of Linker & Chainer

Here we evaluate the evidence recall performance of the OTT-QA single retriever in Table 10. We

[4] https://github.com/wenhuchen/OTT-QA

[5] https://github.com/facebookresearch/DPR

[6] https://github.com/Mayer123/UDT-QA

	R@20	R@100	AR@20	AR@50
OTT-QA Retriever	82.7	92.5	31.2	37.5
+Linker&Chainer	84.1	91.2	74.4	83.5
Retrieve full index	82.3	91.8	32.7	40.2

Table 10: Evidence Recall of OTT-QA single retriever on OTT-QA Dev set, where R@K evaluates gold table chunk recall and AR@K evaluates answer recall

	Dev		Test	
	AR@20	AR@50	AR@20	AR@50
Joint retriever	82.3	87.0	83.6	88.2
+Linker&Chainer	85.7	88.1	86.8	89.4
NQ Retriever	83.2	87.4	84.1	88.4
+Linker&Chainer	85.9	88.4	86.5	89.4

Table 11: Answer Recall on NQ task

can see that compared to results in Table 5, the performance of the single dataset retriever is quite similar for all metrics, suggesting that a joint model is sufficient to learn multiple tasks under our framework. We conduct the same evaluation on NQ in Table 11. With the addition of Linker and Chainer, we see that the answer recall improved over the retriever-only setting. We also observe that the results for NQ-single retriever are quite similar to joint retriever, as seen on OTT-QA.

E Computation Time and Infrastructure

We train our linker and retriever models on two A6000 GPUs, and the training took less than a day to finish. For the Chainer model, running inference for one million question-document pairs takes about 5 hours on one A100-80G GPU. We train our reader model on 16 V100-32G GPUS, which takes about 1 week to finish.

Our linker model has 330M parameters (3 bert-base encoders), the retriever model has 220M parameters (2 bert-base encoders), the chainer model has 3B parameters and the reader model has 770M parameters.