Open Domain Question Answering over Virtual Documents: A Unified Approach for Data and Text

Anonymous ACL submission

Abstract

Due to its potential for a universal interface over both data and text, data-to-text generation is becoming increasingly popular. However, few prior work has focused on its application to downstream tasks, *e.g.* using the converted data for grounding or reasoning. In this work, we bridge this gap and use the data-to-text method as a means for encoding structured knowledge for knowledge-intensive applications, *i.e.* open-domain question answering (ODQA). Specifically, we propose a *verbalizer-retriever-reader* framework for ODQA over data and text where verbalized tables from Wikipedia and graphs from Wikidata are used as augmented knowledge sources. We show that our **U**nified **D**ata and **T**ext **QA**, `UDT-QA`, can effectively benefit from the expanded knowledge index, leading to large gains over text-only baselines. Notably, our approach sets the single-model state-of-the-art on Natural Questions. Furthermore, our analyses indicate that verbalized knowledge is preferred for answer reasoning for both adapted and hot-swap settings.

1 Introduction

Data-to-text generation verbalizes structured knowledge, e.g. tables and knowledge base (KB) graphs, into natural language and has a broad range of applications such as dialog response generation (Moon et al., 2019) and multi-document summarization (Fan et al., 2019). Given its potential in providing a universal interface for data and text, it has became increasingly popular (Gardent et al., 2017; Parikh et al., 2020; Nan et al., 2021) with various methods developed recently (Wang et al., 2020; Ribeiro et al., 2020; Chen et al., 2020b). Nevertheless, most existing work has focused on *intrinsic evaluations* exclusively, i.e. the quality of generated text measured by metrics like BLEU (Papineni et al., 2002), leaving its usefulness on downstream tasks largely unknown. Moreover, it remains unclear whether a single data-to-text model is able to verbalize heterogeneous structured data effectively. In this work, we aim to investigate the feasibility of using a unified data-to-text verbalizer as the means for enriching the knowledge source for open-domain question answering (ODQA).

Based on the typical *retriever-reader* framework for ODQA, recent work (Oguz et al., 2020) has demonstrated that expanding the textual knowledge source with more structured tables and KBs is beneficial. However, most existing work either only considers limited size/type of data or uses different knowledge retrieval methods for various sources (Oguz et al., 2020; Agarwal et al., 2021). Here, we propose a simple and unified *verbalizer-retriever-reader* framework, `UDT-QA`, as an extension for ODQA over data and text.

To bridge the gap between existing data-to-text approaches and ODQA, we develop a novel data-to-text generation paradigm for our *verbalizer-retriever-reader* framework. First, both tables and KB graphs are converted into the same format such that a single data-to-text model can handle both cases. Moreover, we design a method consisting of data filtering and beam selection to maximize the faithful coverage of the input information. To remedy the lack of in-domain training data, we further propose an iterative training approach to augment the existing data-to-text training set with selected high quality outputs from the target domain. With this verbalizer, we convert all tables from Wikipedia and sub-graphs from Wikidata into virtual documents as the additional knowledge source for answering open-domain questions.

We first validate our data-to-text method based on the existing intrinsic data-to-text metrics on DART (Nan et al., 2021) and additional faithfulness promoting evaluation on the target ODQA data. Remarkably, our data-to-text generation approach can effectively improve the target-domain faithful metric without compromising the intrinsic metrics. To further validate the effectiveness of the proposed UDT-QA , we carry out experiments on the ODQA task using a recent state-of-the-art (SOTA) retriever-reader pipeline, including DPR (Karpukhin et al., 2020) for dense retrieval and UnitedQA (Cheng et al., 2021) for answer reasoning over the retrieved context. Consistent with previous work, our results also suggest that additional knowledge source from data is beneficial for the ODQA task. Notably, we find that the verbalized knowledge is more favored by the reader compared to the raw format (linearization), especially when the structured data size is comparable to text, leading to more pronounced end-to-end improvements. Overall, UDT-QA shows large improvements over text-only baselines and performs competitively with recent more complicated methods on both Natural Questions (NQ) (Kwiatkowski et al., 2019) and WebQuestions (WebQ) (Berant et al., 2013). In particular, our UDT-QA achieves new STOA performance on NQ under the single-model open-book setting.

The main contribution is summarized below. First, a simple and unified *verbalizer-retriever-reader* framework, UDT-QA, is proposed for ODQA over data and text. Second, a novel data-to-text approach is developed that enables building a large-scale collection of knowledge by verbalizing all tables from Wikipedia and sub-graphs from Wikidata. Last, our proposed method achieves remarkable improvements on both NQ and WebQ with additional knowledge from data, and sets the new single-model SOTA on NQ.

2 Overview of UDT-QA

In this section, we present the overall pipeline of our UDT-QA framework for ODQA over data and text (Figure 1). The major difference between our approach and the popular *retriever-reader* ODQA systems (Min et al., 2021, *inter alia*) is the use of a data-to-text verbalizer (§3) for converting structured data into natural language text, *i.e.* virtual documents, as the universal knowledge source. Here, we consider two types of structured knowledge (§4.2) — tables and KB sub-graphs. After verbalizing the structured knowledge, a subsequent pipeline consisting of a DPR retriever and a UnitedQA-E reader is used for answer inference. Since the retriever and reader are not the main focus of this work, we only briefly describe them below.

The DPR retriever (Karpukhin et al., 2020) is a bi-encoder model consisting of a question encoder and a context encoder, which is used for data and text retrieval. Following previous work (Karpukhin et al., 2020; Oguz et al., 2020), we use the uncased BERT-base (Devlin et al., 2019) model as the encoder, where the [CLS] token representation is used as the document/question vector. During training, positive and negative pairs of (question, context) are used to update the model. For inference, the entire document index is encoded with context encoder and the encoded question vector is used to retrieve the top documents with highest dot-product scores.

The UnitedQA-E (Cheng et al., 2021) is an extractive reader based on ELECTRA (Clark et al., 2020) for answer inference. Here, a pair of a question and a support passage is jointly encoded into neural text representations. These representations are used to compute scores of possible answer begin and end positions, which are then used to compute probabilities over possible answer spans. Finally, the answer string probabilities are computed based on the aggregation over all possible answer spans from the entire set of support passages.

3 Verbalizer: Data-to-text Generation

Here, we formally describe the data-to-text model developed in this paper, including the input format (§3.1) and the adaptation for ODQA (§3.2).

3.1 Input Format

Given a structured data input D, the data-to-text generator G aims to generate a natural language passage P that faithfully describes the information presented in D. In the literature, the structured data input can be in the form of a set of triples (Nan et al., 2021), a few highlighted cells from a table (Parikh et al., 2020) or a full table (Chen et al., 2020a). Correspondingly, P could a simple surface-form verbalization of D (*e.g.* when D is a triple set) or a high-level summarization in case of a full table or a large KB graph. Since we consider (noisy) tables/KB sub-graphs of arbitrary size in this paper, directly feeding the entire input into the generator is not feasible, likely incurring significant computation challenges. Moreover, it is also desirable to maximize the information coverage of P so that most relevant information in D can be leveraged by the downstream QA retriever and reader. Based on this, we verbalize both tables and KB graphs at a fine-grained level.

In this work, we verbalize tables row by row,

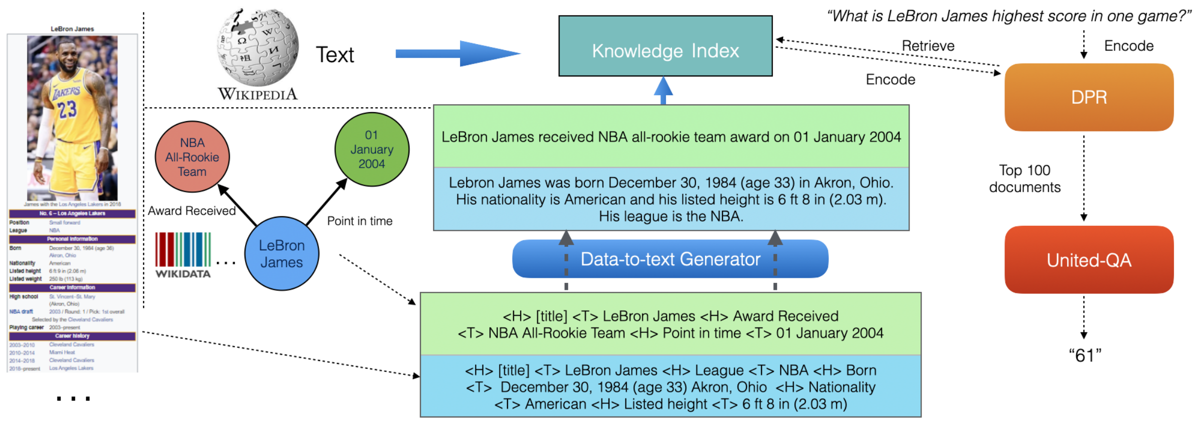

Figure 1: An overview of `UDT-QA` based on the *verbalizer-retriever-reader* pipeline.

i.e. input each table row to G individually, where each row is a set of cells $r = \{c_i\}_{i=1}^{k}$, and k is the number of cells in the corresponding row. Most relevant to our setting, recent work (Nan et al., 2021) represents each cell in a triple. To form such triples, they manually annotate the tree ontology of column headers and then create triples using table title, headers, cell value and header relations, *e.g.* `([TABLECONTEXT], [title], LeBron James)`, `(LeBron James, League, NBA)` where `LeBron James` is the parent cell. Although such triples with fine-grained ordering may help guide the generator, directly applying a such generator to a target domain with no ontology annotation (our case) likely results in degradation. To overcome this, we propose to convert the triple set to pairs, *e.g.* `([title], LeBron James)`, `(League, NBA)`. We find such conversion has little impact on the intrinsic evaluation (§5). After all rows are verbalized, we assemble the text outputs back to form the verbalized table.

For KB, we follow previous work (Agarwal et al., 2021) and break the KB into small sub-graphs based on subject entity. Here, each sub-graph contains one central entity and its neighbors. Although this conversion would inevitably create undesirable artifacts (*e.g.* hurdles for multi-hop reasoning across sub-graphs), this preprocessing allows us to unify the input representations for both table and KB graphs, making it possible for a single verbalizer to convert structured knowledge into text format. Specifically, we convert all KB sub-graphs into the same format as table cell sets above, where the subject entity is treated as the title and all the edges are represented using pairs in the form of `(relation, object)`. Then we verbalize each sub-graph with the generator G. Examples of input and output for table rows and KB sub-graphs are shown in Figure 1.

3.2 Improved Data-to-Text Model Training

A known problem in data-to-text generation is that the model tends to hallucinate or neglect information in the input (Wang et al., 2020; Agarwal et al., 2021). Faithfulness and information coverage is especially important when we apply the verbalized output to knowledge-intensive downstream tasks like ODQA. To address this, we subsample training data `T` such that the instances are filtered out if they are likely to steer model towards missing information. In particular, we compute ROUGE-1 (Lin, 2004) scores between the input and target of training instances and filter out those whose scores are below a certain threshold. We denote the filtered version as `T-F`. Although filtered examples are mostly valid, we hypothesize that their target sentences may only contain partial input information or high-level summaries, which may bias the model towards unwanted behaviors.

Another challenge we face is that most data-to-text training examples have succinct structured inputs. In other words, the cells in the structured input are usually single words or short phrases with corresponding short target sentences as well. In our case, a number of of tables contain large cells with dozens of words. Models trained with existing data likely have a hard time verbalizing such inputs faithfully. To alleviate this domain-mismatch issue, we propose an iterative training set-up. In the first iteration, we train a generator on `T-F`. Then we apply the generator to our data. We then find high quality verbalized outputs based on the ROUGE-1 score between the input and output, and sample instances with score higher than a threshold for the next-round training. We sample instances up to the same size of `T-F`, and denote this set as `ID-T`. Finally, we mix the `ID-T` with `T-F` and train a second generator for verbalization.

Following recent work (Nan et al., 2021), we use the pretrained T5-Large (Raffel et al., 2020) model as our generator. Given paired training examples consisting of a structured data input and a target sentence, we finetune the T5 model to maximize the log-likelihood of generating the corresponding target sentences. Here, we follow the same experimental setup as (Ribeiro et al., 2020).

4 Experiment Setup

In this section, we describe the data used for experiments and sources of structured knowledge.

4.1 Datasets

In this paper, we use DART (Nan et al., 2021) to train our verbalizer (data-to-text) and two ODQA datasets, NQ and WebQ, to train and evaluate our pipeline, with the same split as in (Lee et al., 2019) provided by (Karpukhin et al., 2020). Below we provide a brief description of each dataset and refer readers to their papers for details.

DART is a data-to-text dataset containing pairs of (triple-set, sentences) collected from WebNLG (Gardent et al., 2017), E2E (Novikova et al., 2017) and crowdsourcing based on tables found in WikiSQL (Zhong et al., 2017) and WikiTableQuestions (Pasupat and Liang, 2015). **Natural Questions** contains questions mined from Google search queries and the answers are annotated in Wikipedia articles by crowd workers. **WebQuestions** consists of questions from Google Suggest API and the answers are annotated as entities in Freebase.

We collect **knowledge-answerable questions** from NQ and WebQ in order to evaluate our verbalizer and construct the retrieval training data. Specifically, we find questions in the original NQ training set that can be answered by a table. For each question, we search through tables in its associated HTML page to locate exact answer matches. In total, we collected 14,164 triples of (question, answer, gold table) from NQ train and dev sets as `NQ-table-Q`. On WebQ, we find questions that can be answered by KB via expanding from question entities and search for their 1-hop neighbors. If an answer entity is matched, we keep this sub-graph. In total, we collected 2,397 triples of (question, answer, sub-graph) from WebQ train and dev set as `WebQ-KB-Q`.

4.2 Structured Knowledge Sources

In addition to regular Wikipedia text passages, we consider two types of structured knowledge — tables from Wikipedia and KB graphs from Wikidata.

For tables from Wikipedia, we follow OTT-QA (Chen et al., 2021b) with slight modifications. Chen et al. (2021b) only consider tables in good format, *i.e.* tables with no empty cell, multi-column or multi-row, and restrict the tables to have at most 20 rows or columns. Instead, we remove such constraints and keep everything with the `<table>` tag, resulting in a larger and noisier table set. We denote this more realistic set of tables as `OTT-tables`.

Note Oguz et al. (2020) only consider tables from the original NQ HTMLs. In addition to the size difference, `OTT-tables` are crawled from a more recent Wikipedia dump than the NQ version. To study the impact of knowledge source size, we also process tables from the NQ HTML pages with the heuristic suggested by (Herzig et al., 2021) to de-duplicate tables and filter lengthy cells (>80 words). We denote this set of tables as `NQ-tables`. To avoid overlap, we remove tables from `OTT-tables` whose page title are in `NQ-tables` set. In total, we have a `All-tables` set with 2.2M tables from `OTT-tables` and 210K tables from `NQ-tables`, respectively.

For KB graphs, we consider using the English Wikidata (Vrandečić and Krötzsch, 2014) as our KB due to its broad coverage and high quality, noting its predecessor Freebase is no longer maintained despite its popularity in research. In order to be comparable with recent work (Agarwal et al., 2021), we directly use their partitioned KB graphs from WikiData in our experiments, which is denoted as `WD-graphs`.

5 Experiments: Data-to-Text

In this section, we evaluate our data-to-text model with both intrinsic and extrinsic metrics. Since intrinsic metrics are probably less correlated with the model downstream performance, we focus on using an extrinsic metric for selecting models and include intrinsic metrics as a sanity check for generation quality. During inference, we use beam

		Intrinsic Eval						Extrinsic Eval
Training Set	**# Examples**	**BLEU**	**METEOR**	**TER**	**MoverScore**	**BERTScore**	**BLEURT**	**Ans Cov**
DART (Nan et al., 2021)	62,659	50.66	0.40	0.43	0.54	0.95	0.44	-
DART ours (`T`)	62,628	51.05	0.40	0.43	0.54	0.95	0.43	95.4
DART (`T-F`)	55,115	51.04	0.41	0.43	0.54	0.95	0.43	96.0
DART (`T-F + ID-T`)	110,230	50.59	0.41	0.44	0.54	0.95	0.43	98.4

Table 1: Intrinsic and extrinsic evaluations of verbalization approaches on DART test and `NQ-table-Q` (§4.1), respectively. "Ans Cov" refers to Answer coverage. All metrics are higher the better except for TER.

search with a beam size of 10 and save all completed predictions. To retain as much input information as possible, a re-ranking stage is carried out over these predictions based on the ROUGE-1 score. The highest ranked prediction is then used as the final output.

Intrinsic Evaluation: Since our model is developed mainly on DART, we first conduct the intrinsic evaluation on the DART test set to measure the impact of our improved data-to-text methods, *i.e.* data filtering and iterative training. Following (Nan et al., 2021), we use the official evaluation metrics including BLEU, METEOR (Banerjee and Lavie, 2005), TER, MoverScore (Zhao et al., 2019), BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020). Table 1 summarizes different data-to-text models on DART test. As we can see, the resulting model trained with our data conversion (row 2) performs on par with the model using the original format (row 1). More interestingly, filtering short samples has almost no impact on the verbalizer performance (row 3). Lastly, iterative training with additional target domain data (row 4) slightly hurts on BLEU and TER and achieves similar performances on other metrics. Overall, our verbalizer with the proposed data conversion and improved training remains very effective on DART.

Extrinsic Evaluation: Since we are interested in applying verbalized knowledge for ODQA, the QA model is more likely to predict the correct answer only if the answer still exists after the verbalization. Therefore, we also evaluate each generator using a metric more related with the downstream task performance: **answer coverage**. Specifically, we compute the answer coverage as the percentage of examples that the answer present in the raw structured knowledge is still preserved in the corresponding verbalized output.

First, we compute the answer coverage of different generators discussed in the previous section on `NQ-table-Q` where tables known to contain question-triggering content. The scores are reported in the last column of Table 1. Due to more lengthy tables in `NQ-table-Q`, data filtering improves the answer coverage as expected. Moreover, model trained with our iterative training demonstrates substantial improvements in answer coverage, indicating that our approach is highly effective for converting tables into text. Later, we use this best generator to verbalize `All-tables`.

Lastly, we directly apply our best generator (DART T-F + ID-T) for verbalizing KB graphs. To evaluate the performance, we compare our model with the recent method KELM-verbalizer (Agarwal et al., 2021) using answer coverage on the set `WebQ-KB-Q` where KB sub-graphs are known to contain answer entities. Although never tuned for KB graph inputs, our model achieves 99.6 on answer coverage, outperforming the KELM-verbalizer (97.8 on answer coverage) by a large margin. This suggests that our data-to-text approach is highly effective for both tables and KB sub-graphs.

6 Experiments: QA over Data and Text

Here we present our main experiments on ODQA over data and text. For regular Wikipedia text, we use the same index containing 21M passages as in (Karpukhin et al., 2020). To augment text, two settings are considered, *i.e.* the *single data* setting and the *hybrid data* setting.

In the single data setting for NQ, we augment the text index with tables from the `All-tables` set (§4.2). For comparison, we also experiment with the raw representations using a simple linearization of tables similar to (Oguz et al., 2020). For WebQ, we consider combining text with KB graphs from `WD-graphs` in the single data setting. Different from (Oguz et al., 2020) where a separate entity-linking based retriever is used for KB, we use a single model over the text index with either linearization of raw KB graphs or our verbalized KB graphs. Hence, in our case, both text and data (tables and KB graphs) can be handled

Model	NQ	WebQ
Without Structured Knowledge		
DPR (Karpukhin et al., 2020)	41.5	35.2
UnitedQA (Cheng et al., 2021)	51.8	48.0
With Structured Knowledge		
KEALM (Agarwal et al., 2021)	41.5	43.9
UnitK-QA (Oguz et al., 2020)	54.1	**57.8**
`UDT-QA` *w/* Raw Single Data	54.7	51.4
`UDT-QA` *w/* Verbalized Single Data	**55.2**	52.0
`UDT-QA` *w/* Verbalized Hybrid Data	55.1	52.5

Table 2: End-to-end open-domain QA evaluation of `UDT-QA` in comparison to recent state-of-the-art models on the test sets of NQ and WebQ. Exact match scores are reported (highest scores shown in **bold**).

Source	Format	R20	R100	EM
text	-	80.8	86.1	49.6
+`NQ-tables`	raw	85.2	90.1	51.1
+`NQ-tables`	V	85.5	90.2	51.2
+`All-tables`	raw	85.8	**90.7**	52.1
+`All-tables`	V	**86.0**	**90.7**	**52.5**
text	-	78.9	82.3	52.6
+`WD-graphs-WebQ`	raw	**83.4**	86.1	**57.1**
+`WD-graphs-WebQ`	V	**83.4**	85.0	55.7
+`WD-graphs`	raw	82.8	86.1	54.3
+`WD-graphs`	V	82.8	**86.7**	55.4

Table 3: Impact of knowledge index size over separately trained retriever-reader models (Top for NQ and bottom for WebQ). All metrics are computed on the corresponding dev set.

by a unified retreiver-reader pipeline. In the hybrid data setting for both NQ and WebQ, we use text, `All-tables` and `WD-graphs` for retrieval. The statistics of our knowledge index are shown in Table 6 in Appendix A.

We create additional retriever training data from `NQ-Table-Q` and `WebQ-KB-Q` in a similar fashion as in the text-only setting, so that DPR can better handle additional knowledge. Followig (Oguz et al., 2020), we also use the iterative training setup for retriever training. More training details can be found in Appendix B.

To evaluate the effectiveness of our `UDT-QA` for ODQA, we first include recent state-of-the-art ODQA models using text as the only knowledge source, *i.e.* DPR (Karpukhin et al., 2020) and UnitedQA (Cheng et al., 2021). We also compare our `UDT-QA` with recent models using additional structured knowledge, *i.e.* KEALM (Agarwal et al., 2021) and UnitK-QA (Oguz et al., 2020). Following the literature, we report the exact match (EM) score for evaluation. The results are in Table 2.

As we can see, models with additional structured knowledge achieve better performance than text-only models. This indicates that both KB graphs and tables contain complementary knowledge which is either absent in text or harder to be reasoned over. For NQ, although we consider a significantly larger structured knowledge source which is likely to be more challenging, all our models substantially outperform UnitK-QA. As for WebQ, our model achieves competitive performance, although worse than UnitK-QA. We attribute this gap to two possible reasons. First, UnitK-QA uses a separate entity-linking based retriever for KBs which might lead to higher retrieval recall. Second, since WebQ is fully based on FreeBase, using WikiData only in our models likely suffers from mismatch (Pellissier Tanon et al., 2016). Nevertheless, our verbalizer-based models achieve better performances than the corresponding raw format models on both datasets, indicating that the proposed verbalizer is highly effective for tables and KB graphs.

7 Analysis

In this section, we present analyses over the impact of knowledge index size, the use of additional structured knowledge in a hot-swap setting, comparison to a recent KB-only data-to-text approach in an end-to-end fashion, and manual exam of the verbalized/raw tables for their impact on ODQA.

How does the size of knowledge index affect retriever and reader performance? More knowledge is likely to have better coverage of relevant information. On the other hand, larger and noisier index also increases the reasoning complexity. To understand the impact of the increased knowledge index size, we conduct experiments with a restricted setting where only relevant subset of knowledge to the corresponding dataset (a prior) is used for retrieval. Similar to (Oguz et al., 2020), we experiment with the combined knowledge index of text and `NQ-tables` for NQ. As for WebQ, we keep documents from `WD-graphs` that contain any of the question entity in WebQ to build `WD-graphs-WebQ`, and experiment with using text + `WD-graphs-WebQ`. In addition to EM, we report R20 and R100, evaluating the retrieval accuracy of gold passages in the top-20 and top-100 documents, respectively. The results are reported

Knowledge	Format	R20	R100	EM
Text-only		81.3	87.3	51.8
+NQ-tables	raw	83.9	90.3	51.7
+NQ-tables	V	84.3	90.4	52.5
+All-tables	raw	84.0	**90.6**	51.7
+All-tables	V	**84.5**	**90.6**	**52.7**

Table 4: Hot-swap evaluation of raw vs verbalized table using a text-only retriever-reader model on NQ test.

Knowledge	R20	R100	EM
KELM	78.2	85.3	51.5
WD-graphs (Ours)	**78.5**	**85.5**	**52.0**

Table 5: Comparison of verbalized knowledge from our verbalizer and KELM for retriever and reader on WebQ test. Dev results can be found in Table 8 in Appendix D.

in Table 3.

For NQ, in spite of being more challenging, we see that using All-tables yield substantial improvement in both recall and answer exact match compare to using NQ-tables. This indicates that, with proper training, ODQA models are likely to benefit from enriched knowledge. Although the larger raw form index brings in decent improvement (+1 EM) in terms of reader performance (+All-tables vs+NQ-tables), our verbalized knowledge is more friendly for answer reasoning leading to a more notable QA improvement (+1.3 EM). Different from NQ, we observe that on WebQ the restricted setting with WD-graphs-WebQ achieves better results. We hypothesize that this is likely due to the scale of WebQ dataset. The small amount of WebQ training makes the retriever insufficient to handle large-scale knowledge index. We leave the verification of this hypothesis for future work.

Does a text-only retriever-reader model benefit more from verbalized knowledge compare to raw format (hot-swap)? Since both retriever and reader are based on pretrained language models, we hypothesize that they would probably benefit more from the verbalized knowledge due to its similar style as text. This can be particularly useful for a hot-swap setting where both retriever and reader have only seen textual knowledge during training. To verify that verbalized knowledge is more amenable, we carry out a hot-swap experiment here. Specifically, we directly use a DPR model trained on NQ text-only data for additionally indexing both NQ-tables and All-tables. Then, the inference retrieval is performed on the augmented knowledge index for an input question, and a text-only United-QA-E reader trained on NQ is applied for answer inference afterwards. The results are summarized in Table 4. Similar to the previous fully fine-tuned settings, we see that additional knowledge still provide substantial improvements for text-only retriever using either raw or verbalized knowledge. However, the improvement in recall is not reflected in the later reader performance for the raw format, whereas the hot-swap answer inference performance is notably improved with verbalized knowledge. This observation further validates our hypothesis that verbalized knowledge is more beneficial, especially for reader.

How does the proposed verbalizer compare to recent data-to-text models? Lastly, we compare our verbalizer with the recently proposed data-to-text generator for converting KB graphs only, KELM (Agarwal et al., 2021). Since both KELM generator and our verbalizer are based on the same partitioned Wikidata, this evaluation can fully reflect their corresponding generation impacts on ODQA in an end-to-end fashion. Here, we evaluate using our verbalized WD-graphs and the KELM corpus (Agarwal et al., 2021) as additional knowledge on WebQ. In particular, we follow the same procedure to train and evaluate our retriever and reader except that we swap the WD-graphs with KELM corpus in data construction and retrieval. Both retriever and reader performances are reported in Table 5. Note that the KELM data-to-text model is customized solely for converting KB graphs and trained with a much larger dataset (about 8M training instances), whereas our verbalizer is applicable to both tables and KB graphs with a smaller training data (only 110K instances). Nevertheless, consistent with its better extrinsic performance (§5), our verbalizer again outperforms the KELM generator in both retrieval and reading, which provides further support for the effectiveness of our approach as a unified interface for ODQA over data and text.

What is the impact of verbalized/raw table on ODQA? We also manually analyze examples of verbalized and raw tables, the examples are shown in Table 10 in Appendix E, as well as details of annotation. Overall, we find that verbalized tables help connect the information in the headers with cell values, making it easier for model to reason over. On the other hand, verbalization can suffer from the table structure loss, which may hinder the model from leveraging such shortcuts, *e.g.* answering a ranking question where the model can directly look for answers in the first/last row (see example 3&4 in Table 10). This also suggests a possible direction for future work: to better incorporate the table structure information in verbalization.

8 Related Work

Data-to-Text Generating text from structured data has been a popular task in NLP. Many dataset have been proposed for this task such as Wikibio (Lebret et al., 2016), Rotowire (Wiseman et al., 2017), WebNLG (Gardent et al., 2017) and E2E (Novikova et al., 2017), where each dataset focuses on a particular domain. More recently, large-scale datasets that contains open-domain examples have been proposed including DART (Nan et al., 2021), TOTTO (Parikh et al., 2020), WikiTableT (Chen et al., 2021a) and GenWiki (Jin et al., 2020). On the modeling side, finetuning the pretrained models typically achieves promising performance (Ribeiro et al., 2020). Wang et al. (2020) propose customized loss functions to reduce model hallucination during generation. Muti-task learning is used to improve model's robustness towards input variations (Hoyle et al., 2021). Chen et al. (2020b) introduce a generalized format and a pretrained model that can generate text from both table rows and knowledge graphs. Most previous work on data-to-text generation have only conducted internal evaluation, using typical generation metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), hence the data-to-text is considered the target task. In this paper, we argue that different training strategies and evaluation metrics should be adapted when applying data-to-text models to downstream tasks, i.e. ODQA. Related to our work, Agarwal et al. (2021) convert the entire Wikidata to natural language using a finetuned T5 model (Raffel et al., 2020). In this work, we generalize the data-to-text approach for verbalizing both tables and KB graphs in a unified fashion and study the verbalized knowledge on ODQA.

QA with Data and Text As the knowledge required to answer the questions may not be available in textual corpus, previous studies have sought to incorporate knowledge from difference sources such as tables and knowledge bases. Min et al. (2019) use Wikidata to expand seed passages found by the retriever and enhance encoded passage representations in the reader. Li et al. (2021) propose a hybrid framework that takes both text and tables as inputs to produce answers and SQL queries. Recently, Chen et al. (2021b) develop the OTT-QA dataset containing questions that require joint reasoning over both tables and text, where the tables and text come from entire Wikipedia. There is also a line of work that studies model architectures for tables specifically or joint encoding of tables and text (Yin et al., 2020; Herzig et al., 2020; Zayats et al., 2021; Glass et al., 2021). However, their focus is not on open-domain QA tasks. Most similar to our work is (Oguz et al., 2020), where they use both tables and Wikidata/Freebase knowledge graph along with Wikipedia text to build retriever index. However, their tables are only mined from original NQ HTMLs, hence it is still a constrained setting. In contrast, we consider tables from full Wikipedia which is a much larger set. Additionally, separate retrieval models are used for tables and KB in (Oguz et al., 2020) whereas we develop a unified model over text and data including tables and KB graphs.

9 Conclusion

In this paper, we demonstrated that a unified *verbalizer-retriever-reader* framework, `UDT-QA`, for open-domain QA over data and text. We proposed a novel data-to-text paradigm that can largely improve the verbalization effectiveness for downstream knowledge-intensive applications, *i.e.* open-domain QA, when attaining good intrinsic performances. Leveraging the verbalized knowledge, we achieved a new state-of-the-art result for NQ. Remarkably, we showed that simply augmenting the document index with the verbalized knowledge is able to improve the performance without retraining the model.

In addition to our method, there are many recently proposed approaches for open-domain QA that are orthogonal. For example, language models specifically optimized for dense retrieval (Gao and Callan, 2021), pretraining on large-scale QA data (Oğuz et al., 2021) and hybrid system that consists of retriever, reranker, extractive reader and generative reader (Fajcik et al., 2021). Incorporating those methods may further improve the performance for open-domain QA, and we leave that exploration for future work.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Mingda Chen, Sam Wiseman, and Kevin Gimpel. 2021a. WikiTableT: A large-scale data-to-text dataset for generating Wikipedia article sections. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209, Online. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. KGPT: Knowledge-grounded pretraining for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Wenhu Chen, Ming wei Chang, Eva Schlinger, William Wang, and William Cohen. 2021b. Open question answering over tables and text. *Proceedings of ICLR 2021*.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. 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: Pretraining text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

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.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-d2: A modular baseline for open-domain question answering.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Michael Glass, Mustafa Canim, Alfio Gliozzo, Saneem Chemmengath, Vishwajeet Kumar, Rishav Chakravarti, Avi Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2021. Capturing row and column semantics in transformer based question answering over tables. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1212–1224, Online. Association for Computational Linguistics.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, Online. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Alexander Miserlis Hoyle, Ana Marasović, and Noah A. Smith. 2021. Promoting graph awareness in linearized graph-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 944–956, Online. Association for Computational Linguistics.

Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2020. GenWiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2398–2409, Barcelona, Spain (Online). International Committee on 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.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for 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*, pages 6086–6096. 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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. 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.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy. Association for Computational Linguistics.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. 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.

Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen tau Yih, Sonal Gupta, and Yashar Mehdad. 2021. Domain-matched pre-training tasks for dense retrieval.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. 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 1470–1480, Beijing, China. Association for Computational Linguistics.

Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From freebase to wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 1419–1428, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Vicky Zayats, Kristina Toutanova, and Mari Ostendorf. 2021. Representations for question answering from documents with tables and text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2895–2906, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning.

Source	Raw	Verbalized
Text	21M	-
`OTT-tables`	4.0M	6.3M
`NQ-tables`	446K	572K
`WD-graphs`	5.7M	5.8M

Table 6: Statistics of Knowledge Index

Source	Format	R20	R100	EM
text	-	81.3	87.3	51.8
+`NQ-tables`	raw	86.0	91.2	54.8
+`NQ-tables`	V	86.2	91.0	54.2
+`All-tables`	raw	86.9	**91.9**	54.7
+`All-tables`	V	**87.0**	91.7	**55.2**
text	-	73.2	81.4	48.0
+`WD-graphs-WebQ`	raw	**80.2**	**85.8**	51.5
+`WD-graphs-WebQ`	V	79.7	85.3	**52.6**
+`WD-graphs`	raw	78.8	85.1	51.4
+`WD-graphs`	V	78.5	85.5	52.0

Table 7: Impact of knowledge index size over separately trained retriever-reader models (Top for NQ and bottom for WebQ). All metrics are computed on the corresponding test set.

A Knowledge Index Statistics

To be consistent with text passages, we also cut tables and KB sub-graphs (raw or verbalized) into chunks that has about 100 words. Hence the verbalized knowledge will have larger index size than raw format.

B Training Details

To train the retriever to better handle knowledge from tables and KB, we create additional training data from `NQ-Table-Q` and `WebQ-KB-Q`. Given a (question, answer, gold table) from `NQ-Table-Q`, we create a positive passage by concatenating rows containing the answer. Then we randomly sample and concatenate other rows in the table if the passage has less than 100 words. To find negative passages for training, we build a index consists of all the tables and use BM25 to retrieve relevant tables. Ones that do not contain the answer are considered as negative tables. Then we sample rows from the table to build negative passages. For the raw tables, the process is the same except that we also concatenate headers in the beginning to build positive and negative passages. We combine NQ training data with this set to train DPR.

For `WebQ-KB-Q`, we use the verbalized gold sub-graphs as positive passages. For the raw format, this is replaced by flattening the gold sub-graph. Then we build an index with all documents in `WD-graphs` and the top ranked documents by BM25 that do not contain the answer are treated as negatives. Here the documents refer to concatenated triples set for raw setting and sentences produced by the generator in verbalized setting. Additionally, we search through answer entities and their neighbors in the graph to find documents that has word overlap with the question. Then we build training instances in a similar fashion.

As pointed by previous work (Oguz et al., 2020), mining harder negative passages using DPR and iterative training leads to better performance. We also adopted this approach in our experiments. After the first DPR is trained, we used it to retrieve passages from a joint index of text+`structured knowledge`. Then the negative passages are paired with the positive passages from the first round to build new sets of training data. Then we train a second DPR using the iteration1 data combined with the new training sets.

For retriver training, we follow the experiment set-up as specified by (Karpukhin et al., 2020). Specifically, we use the Adam optimizer and a per-gpu batch size of 32 for NQ and 24 for WebQ, respectively. All trainings are done with a fixed learning rate of $2e-5$ and 40 epochs. We select the best model based on the retrieval accuracy on the corresponding dev set.

For reader training, we follow the experiment set-up as described in (Cheng et al., 2021). Specifically, we use the Adam optimizer and a batch size of 16 for NQ and 8 for Webq, respectively. We select the learning rate in $\{3e-5, 5e-5\}$ and number of training epochs in $\{6, 8\}$. The best model is selected based on EM on the corresponding dev set.

C Impact of Knowledge Index Size

We report the test set results of models trained with different knowledge index in table 7 (corresponding to table 3). Overall, we observe similar trends. For NQ, the model benefits more from a larger knowledge index while for WebQ the restricted setting yield better performance.

Knowledge	R20	R100	EM
KELM	**83.1**	**86.7**	55.1
WD-graphs (Ours)	82.8	**86.7**	**55.4**

Table 8: Dev set results of models trained on WebQ with verbalized WD-graph and KELM

	V-correct	V-error
Raw-correct	1750	223
Raw-error	242	1395

Table 9: Error matrix of UDT-QA trained with text+All-tables in raw and verbalized format

D Comparison betweeh Our Verbalizer and KELM-verbalizer

We report the dev set results of WebQ models trained with our verbalized WD-graphs in comparison with KELM in table 8 (corresponding to table 5).

E Case Study on Raw vs Verbalized Tables

Here, we showcase the examples of verbalized tables and their raw counterpart and discussion their effect on our UDT-QA system.

We start by computing the error matrix of the NQ models trained with text+All-tables in both format, as shown in table 9. We then manually annotated 100 examples where only 1 format of knowledge successfully answered the question (50 for each format), and we select examples where at least 1 table chunk is marked as positive by the retriever. Out of 50 examples where verbalized tables contain the answer span, 40 of them are true positives that provide direct evidence to the questions. In 35 out of 40 questions, the retriever for the raw model actually find the same table/chunks that provide the answer. However, the model failed to extract answer for those cases and we think it's mainly because the raw format of the noisy tables can be hard for the model to reason over. We identify 2 common patterns of raw table from these 35 examples, as shown in the first 2 rows of table 10. In the first example, **the concatenated numbers in the raw table can be hard to interpret**, and we have to carefully align the row with the header, which is very far away. In the second example, **the raw infobox can be in ill-format and very long**, making it hard to understand. On the other hand, the verbalized row clearly stated the information required by the question, making it straightforward to find the answer.

We then looked at the other group of 50 questions. 37 of them are true positives that contain direct evidence. Then in 30 out of 37 questions, the verbalized retriever is able to find the corresponding verbalized table/chunks that also contain the answer. The remaining cases are all due to retriever failed to find the true positive table chunks. We found that raw tables are better at answering ranking questions, as the examples shown in row 3&4 of table 10. When asked about the top or bottom ranked subject, the model can directly look for evidence from the starting or the end of the table. On the other hand, when the table is verbalized, the model can not rely on such property because the boundary of rows is not clear and **the original structure of the tables are lost**. Thus future work should study how to preserve structure information in verbalized tables.

Q&A	V table	Raw table
Q: star wars the clone wars season 3 episode 1 **A:** Clone Cadets	**TITLE:** List of Star Wars: The Clone Wars episodes .... the theatrical film: "the new padawan" "castle of deception" "castle of doom" "castle of salvation" is no. 3-6 in the series of star wars: the clone wars episodes. **"clone cadets" in season 3 of star wars: the clone wars is number 1 in season and number 7 in series.** "supply lines" is episode 8 in series and 3 in season of star wars: the clone wars game	| no. in series, season, no. in season, title | .... | 3-6, empty, empty, theatrical film: "the new padawan" "castle of deception" "castle of doom" "castle of salvation" | **7, 3, 1, "clone cadets"** | 8, 3, empty, "supply lines" | ....
Q: when was the last time mount ruapehu erupted **A:** 25 September 2007	**TITLE:** Mount Ruapehu .... mount ruapehu is a stratovolcano mountain with an age of 200,000 years. **the last eruption was 25 september 2007** and the volcanic arc/belt is taupo volcanic zone. mount ruapehu was first ascent in 1879 by g. beetham and j. p. maxwell. the easiest route to climb mount ruapehu is hike.	| empty, empty, empty, elevation, prominence, listing, coordinates, empty, translation, empty, empty, empty, age of rock, mountain type, volcanic arc/belt, **last eruption**, empty, first ascent, easiest route | .... 200,000 years, strato-volcano, taupo volcanic zone, **25 september 2007**, climbing, 1879 .... |
Q: who has the most yards per carry in nfl history **A:** Emmitt Smith	**TITLE:** List of National Football League career rushing yards leaders **emmitt smith** of the dallas cowboys (1990-2002) and arizona cardinals (2003-2004) **was the first player on the national football league career rushing yards leaders list**. walter payton of the chicago bears (1975-1987) ranked second	| rank, player, team(s) by season, carries, yards, average | **1, emmitt smith**, dallas cowboys (1990-2002) arizona cardinals (2003-2004), 4,409, 18,355, 4.2 | 2, walter payton, chicago bears ....
Q: which country has the smallest population in europe **A:** Vatican City	**TITLE:** List of European countries by population .... **vatican city ranks 50 on the list of european countries by population with 1,000 current population** and 0.0 % of population. the list of european countries by population has 0.0 average relative annual growth(%) and 0 average absolute annual growth. the source is official estimate and the date of last figure is 2012. The total population	| rank, country, current population, % of population, average relative annual growth(%), average absolute annual growth, estimated doubling time(years), official figure, date of last figure, regional grouping, source | 1 .... 49 .... | **50, vatican city, 1,000**, 0.0, 0.0, 0, -, 0, 2012, empty, official estimate | empty, total, ...

Table 10: Examples of tables/chunks retrieved by our model given the question, where the evidence is bolded. In raw table, | is the row separator and empty is the filler token used by our table parsing heuristic (to make the table in good shape)