CLTR: An End-to-End, Transformer-Based System for Cell Level Table Retrieval and Table Question Answering

Feifei Pan[1], **Mustafa Canim** [2], **Michael Glass** [2], **Alfio Gliozzo**[2], **Peter Fox**[1]
panf2@rpi.edu, mustafa@us.ibm.com,
mrglass@us.ibm.com, gliozzo@us.ibm.com
pfox@cs.rpi.edu
[1] Rensselaer Polytechnic Institute
[2] IBM TJ Watson Research Center

Abstract

We present the first end-to-end, transformer-based table question answering (QA) system that takes natural language questions and massive table corpus as inputs to retrieve the most relevant tables and locate the correct table cells to answer the question [1]. Our system, CLTR, extends the current state-of-the-art QA over tables model to build an end-to-end table QA architecture. This system has successfully tackled many real-world table QA problems with a simple, unified pipeline. Our proposed system can also generate a heatmap of candidate columns and rows over complex tables and allow users to quickly identify the correct cells to answer questions. In addition, we introduce two new open-domain benchmarks, E2E_WTQ and E2E_GNQ, consisting of 2,005 natural language questions over 76,242 tables. The benchmarks are designed to validate CLTR as well as accommodate future table retrieval and end-to-end table QA research and experiments. Our experiments demonstrate that our system is the current state-of-the-art model on the table retrieval task and produces promising results for end-to-end table QA.

1 Introduction

Tables are widely used in digital documents across many domains, ranging from open-domain knowledge bases to domain-specific scientific journals, enterprise reports, to store structured information in tabular format. Many algorithms have been developed to retrieve tables based on given queries (Cafarella et al., 2008, 2009; Sun et al., 2019; Bhagavatula et al., 2013; Shraga et al., 2020a; Chen et al., 2021). The majority of these solutions exploit traditional information retrieval (IR) techniques where tables are treated as documents without considering the tabular structure. However, these retrieval methods often result in an inferior quality due to a major limitation that most of these approaches highly rely on lexical matching between keyword queries and table contents. Recently, there is a growing demand to support natural language questions (NLQs) over tables and answer the NLQs directly, rather than simply retrieving top-k relevant tables for keyword-based queries. Shraga et al. (2020c) introduce the first NLQ-based table retrieval system, which leverages an advanced deep learning model. Although it is a practical approach to better understand the structure of NLQs and table content, it only focuses on table retrieval rather than answering NLQs. Lately, transformer-based pre-training approaches have been introduced in TABERT (Yin et al., 2020), TAPAS (Herzig et al., 2020), and the Row-Column Intersection model (RCI) (Glass et al., 2020). These algorithms are very powerful at answering questions on given tables; however, one cannot apply them over all tables in a corpus due to the computationally expensive nature of transformers. An end-to-end table QA system that accomplishes both tasks is in need as it has the following advantages over separated systems: (1) It reduces error accumulations caused by inconsistent, separated models; (2) It is easier to fine-tune, optimize, and perform error analysis and reasoning on an end-to-end system; and (3) It better accommodates user needs with a single, unified pipeline. Hence, we propose a table retrieval and QA over tables system in this paper, called **C**ell **L**evel **T**able **R**etrieval (*CLTR*). It first retrieves a pool of tables from a large table corpus with a coarse-grained but inexpensive IR method. It then applies a transformer-based QA over tables model to re-rank the table pool and finally finds the table cells as answers. To the best of our knowledge, this is the first end-to-end framework where a transformer-based, fine-grained QA model is used along with efficient coarse-grained IR methods to

[1]System page: https://github.com/IBM/row-column-intersection

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 202–209, August 1st - August 6th, 2021.

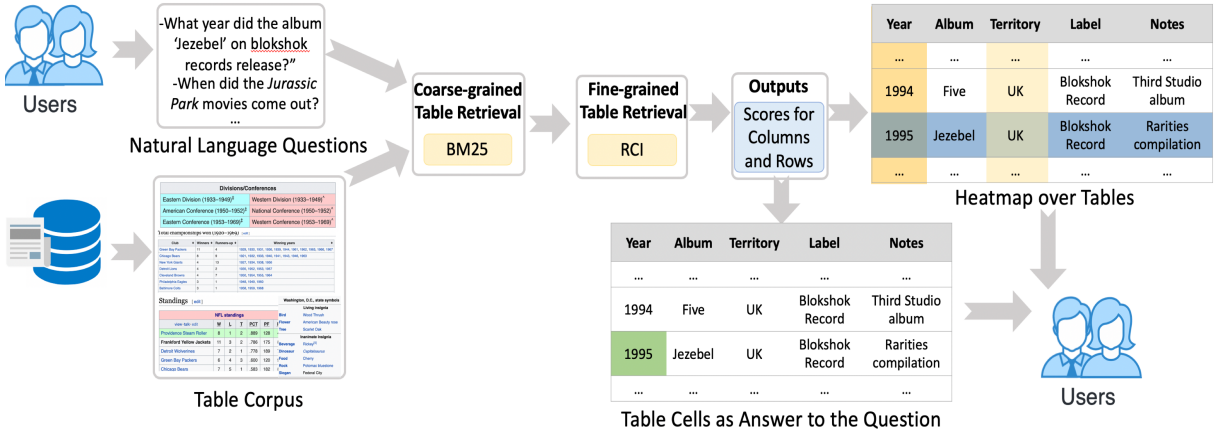

Figure 1: The overview of the end-to-end table QA architecture of CLTR.

retrieve tables and answer questions over them. Our experiments demonstrate that CLTR outperforms current state-of-the-art models on the table retrieval task while further helping customers find answers over returned tables.

To build such a Table QA system, an end-to-end benchmark is needed to evaluate alternative approaches. Current benchmarks, however, are not designed for such tasks, as they either focus on the retrieval task over multiple tables or QA task on a single table. To address the problems, we propose two new benchmarks: E2E_WTQ and E2E_GNQ. The details of these benchmarks and more discussions are provided in Section 4.1.

The specific contributions of this paper are summarized as follows:

- **A transformer-based end-to-end table QA system:** We build a novel end-to-end table QA pipeline by utilizing a transfer learning approach to retrieve tables from a massive table corpus and answer questions over them. The end system outperforms the state-of-the-art approaches on the table retrieval task.
- **Creating heatmaps over complex tables:** To highlight all relevant table columns, rows, and cells, CLTR generates heatmaps on tables. Following a pre-defined color code, the highlighted columns, rows, and cells are ranked according to their relevance to the questions. Using the heatmap, users can efficiently glance through complex tables and accurately locate the answers to the questions.
- **Two new benchmarks for the end-to-end table QA evaluation:** We propose and release two new benchmarks, E2E_WTQ and E2E_GNQ, extending two existing benchmarks, *WikiTableQuestions* and *GNQtables*, respectively. The benchmarks can be used to evaluate systems for table retrieval and end-to-end table QA.

2 Overview

The Architecture The architecture of our end-to-end table QA system, CLTR, is illustrated in Figure 1. This system aims to solve the end-to-end table QA task by generating a reasonable-sized subset of relevant tables from a massive table corpus, and employs the transformer-based approach to re-rank them based on their relevance to the user given NLQs, and finally answer the given NLQs with cells from these tables.

CLTR possess an abundant number of tables generated from documents of various knowledge sources to form a large table corpus. The system has two components: an inexpensive *tf-idf* (Salton and McGill, 1986) based coarse-grained table retrieval component and a fine-grained RCI-based table QA component. CLTR first takes as input any user given NLQs and processes the questions and the table corpus with the inexpensive BM25 algorithm to generate a set of relevant tables, which is relatively large and contains noise (i.e., irrelevant tables). Here we use BM25 to efficiently narrow down the table candidates from a massive table corpus and highly reduce the execution time and computational cost for CLTR. The output of this coarse-grained table retrieval component is later fed into the more expensive but accurate, transformer-based RCI to learn probability scores for table columns and rows, respectively. The scores produced by RCI indicate how likely the given question's final answer exists within a table column or row.

With the probability scores, CLTR re-ranks the tables and produces two outputs to the users: (1) a heatmap over top-ranked tables that highlights the most relevant columns and rows with a color code; (2) the table cells that contain the answers to the NLQs.

The applications Figure 2 presents the user interface of an application of the CLTR system. In this example, we apply the system to table QA over an aviation-related dataset, a domain-specific dataset on tables in aviation companies' annual reports. This user interface consists of two major sections, with *Tag A* and *Tag B* point to the user input and the system output sections, respectively. Under *Tag A* and *B*, the CLTR pipeline is employed to support multiple functionalities. Users can input any NLQs, such as "When is the purchase agreement signed between Airbus and Virgin America?" in this example, into the text box at *Tag D* and click the *Search* button at *Tag C* to query the preloaded table corpus. Users may select to reset the system for new queries or re-train a new model with a new corpus. In the system output sections, a list of tables similar to the table at *Tag F* is generated and presented to users. For each table, the system output includes: (a) the surrounding text of the table from the original PDF (*Tag E*); (b) the pre-processed table in a clean, human-readable format with a heatmap on it, indicating the most relevant rows, columns, and cells (*Tag F*); (c) an annotation option, where the users can contribute to refining the system with feedback (*Tag G*). In addition, the CLTR architecture has been widely applied to datasets from many other domains, varying from finance to medical. The system is also validated with open-domain benchmarks, with more details discussed in Section 4.

3 The RCI-based Table QA

Traditional approaches solve the table QA problem with two consecutive steps: retrieval of the most relevant tables for a given NLQ and locating the correct answers out of the cells with the help of a QA over tables model. These steps are usually studied separately. Our proposed system, CLTR, unifies the two-step table QA with a single pipeline by leveraging the novel RCI model. RCI is the state-of-the-art approach for locating answers over tables (Glass et al., 2020); however, it is not designed to retrieve tables out of large table corpus. In this section, we describe how we build an end-to-end table QA system combining the strength of inexpensive IR methods and the RCI model.

3.1 The Row-Column Intersection Model

We first briefly introduce the Row-Column Intersection model (RCI), which supports the fine-grained table retrieval component of our system. The RCI model decomposes table QA into its two components: projection, corresponding to identifying columns, and selection, identifying rows. Every row and column identification is a binary sequence-pair classification. The first sequence is the question and the second sequence is the row or column textual sequence representation. We use the interaction model of RCI that concatenates the two sequences, with standard separator tokens, as the input to a transformer.

The RCI interaction model uses the sequence representation which is later appended to the question with standard $[CLS]$ and $[SEP]$ tokens to delimit the two sequences. This sequence pair is fed into a transformer encoder, ALBERT (Lan et al., 2020). The final hidden state for the $[CLS]$ token is used in a linear layer followed by a softmax to classify if the column or row containing the answer or not. Each row and column is assigned with a probability of containing the answer. The RCI model outputs the top-ranked cell as the intersection of the most probable row and the most probable column.

Figure 3 gives a sample question fed into the transformer architecture along with the column and row representation of a table.

3.2 The End-to-End Table QA with RCI

To tackle the table retrieval problem, we exploit an inexpensive IR method together with the state-of-the-art RCI model. Unlike the traditional methods treating tables as free text, a set of features, or multi-modal objects, CLTR treats tables as a set of columns and rows and re-rank the tables based on cell-level RCI scores.

As we previously mentioned in Section 2, CLTR first processes the question and table corpus with the inexpensive BM25 algorithm to generate a pool of highly relevant tables. Later, the RCI model is used to produce probability scores for every column and row for tables in the pool. Therefore, for every table t with n columns and m rows in the table pool T, we have two set of scores, $P_{column} = \{p_{c_1}, p_{c_2}, p_{c_3}, ..., p_{c_n}\}$ for columns and $P_{row} = \{p_{r_1}, p_{r_2}, p_{r_2}, ..., p_{r_m}\}$ for rows. We calculate the overall probability score for each ta-

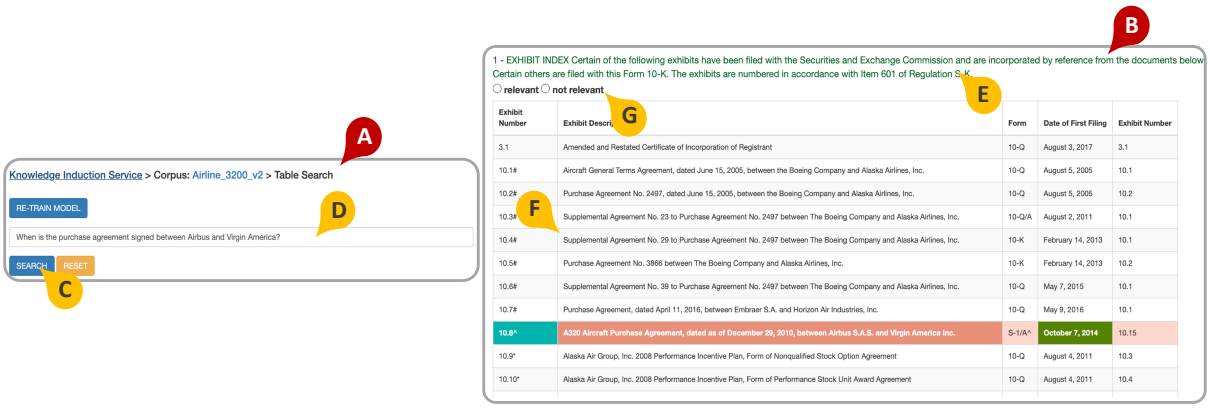

Figure 2: The application of CLTR on an aviation corpus

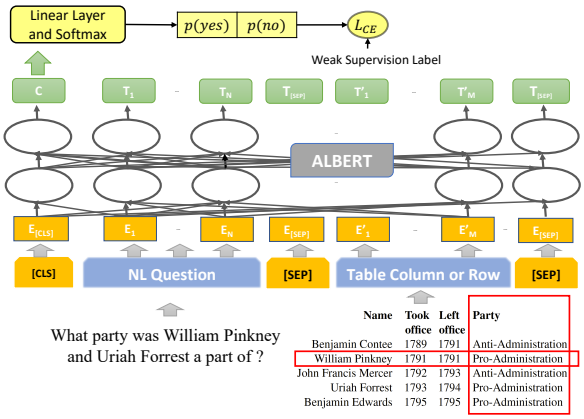

Figure 3: The RCI Table QA Model

ble by taking the maximum cell-level score, using $P_t = max(P_{col}) + max(P_{row})$. Our experiments prove the advantages of this method over the other algorithms (e.g., taking the averaged cell-level scores).

CLTR re-ranks the tables within the table pool T using the maximum cell-level scores. Once the re-ranking is done, the top-k tables out of T are returned to the users. The correct cells on the top-k tables are later identified by locating the intersection of the most relevant columns and rows discovered by the RCI model.

4 Experiments

4.1 Data

Proposed Benchmarks: Existing table retrieval and QA benchmarks focus on either answering NLQs on a single table or the retrieval of multiple tables for a keyword query. A comprehensive comparison of existing benchmarks with their limitations is listed in Table 1. WikiSQL (Zhong et al., 2017) and WikiTableQuestions (Pasupat and Liang, 2015) are widely used to evaluate table QA systems. More recently, they have been used by TAPAS (Herzig et al., 2020) and TABERT (Yin et al., 2020) where transformer-based models for QA over tables have been introduced. However, these benchmarks are not created to be used as part of an end-to-end table retrieval and QA pipeline. On the other hand, WikiTables was created based on the corpus introduced by Bhagavatula et al. (2015) and used in many recent table retrieval studies (Zhang and Balog, 2018a; Deng et al., 2019; Shraga et al., 2020b,c). Despite its popularity, the WikiTables benchmark has two major limitations. First, the query set is fairly limited, containing only 100 keyword-based queries. Many recent studies use this small set of queries for a learning-to-rank (LTR) task with 5-fold cross-validation, potentially causing overfitting issues for the proposed table retrieval models. Second, the query set includes only keyword-based queries, which do not represent the NLQs customers are expected to ask to get answers over tables. To solve the aforementioned issues and create an end-to-end table QA benchmark with NLQs, we introduce two new benchmarks, E2E_WTQ and E2E_GNQ, inspired by *WikiTableQuestions* and *GNQtables*.

The *WikiTableQuestions* (Pasupat and Liang, 2015) benchmark is originally designed for finding answer to questions from given tables. It consists of complex NLQs and tables extracted from Wikipedia. We filter the benchmark following Glass et al. (2020) to generate a subset of 1,216 questions with 2,108 tables.

The *GNQtables* dataset, introduced in Shraga et al. (2020c), extends the Google Natural Questions (NQ) benchmark (Kwiatkowski et al., 2019). It contains 789 NLQs and a large table corpus of 74,224 tables. For each question, the ground truth

	# of tables	# of queries	Retrieval task	QA task	Reference
WikiSQL	24,241	80,654	✗	✓	(Zhong et al., 2017)
TabMCQ	68	9,092	✗	✓	(Jauhar et al., 2016)
WikiTableQuestions	2,108	22,033	✗	✓	(Pasupat and Liang, 2015)
WikiTables	1.6M	100	✓	✗	(Bhagavatula et al., 2015)
GNQtables	74,224	789	✓	✗	(Shraga et al., 2020c)
E2E_WTQ	2,108	1,216	✓	✓	
E2E_GNQ	74,224	789	✓	✓	

Table 1: Comparison of table QA and retrieval benchmarks

only points to the most relevant table (with a binary grade 1 indicates *relevant*), while all other tables in the table corpus are considered *irrelevant* (grade 0). *GNQtables* is the only table retrieval benchmark using NLQs, which makes it possible to adapt it to end-to-end table QA. To create the E2E_GNQ, we manually annotate and enhance *GNQtables* with additional ground truth data for each question: (1) the table cells containing the correct answers; (2) the index of the target columns; (3) the index of the target rows.

Experimental Data: We experiment with E2E_WTQ to test the portability of CLTR, in which we fine-tune the RCI model with two other table QA benchmarks. We utilize an open-domain benchmark, WikiSQL (Zhong et al., 2017), and a domain-specific benchmark, TabMCQ (Jauhar et al., 2016). The WikiSQL dataset has 80,654 questions on 24,241 Wikipedia tables, while the TabMCQ is a much smaller dataset, with only 68 hand-crafted tables and 9,092 multiple-choice questions.

4.2 Experimental Setup

Overall Setup: We test our system under two experimental settings for table retrieval: (1) We test CLTR without task-specific training on E2E_WTQ and fine-tune the RCI model with WikiSQL and TabMCQ; (2) To fairly compare against the state-of-the-art, we follow the experimental setup in Shraga et al. (2020c) and fine-tune CLTR with E2E_GNQ. We implement 5-fold cross-validation on E2E_GNQ, where 80% of data is used for fine-tuning and 20% is used for validation. For both E2E_GNQ and E2E_WTQ, we use BM25 as our baseline model, which is widely used in industry-scale IR systems. We test the end-to-end table QA capability of CLTR with our newly proposed benchmarks. Since we are the first publicly accessible end-to-end table QA system, we do not have a baseline to fairly compare to for our end-to-end table QA experiments.

We implement the coarse-grained table retrieval with the BM25 algorithm embedded in the ElasticSearch python API for all of our experiments. This API can be accessed at *https://elasticsearch-py.readthedocs.io/en/master/*. Each table is indexed as a single text document with the embedded English analyzer. For each question, we generate a pool of 300 tables with the highest BM25 similarity scores. Following the current state-of-the-art model in Shraga et al. (2020c), we set $k1 = 1.2$ and $b = 0.7$. The tables in the pool are later processed with the RCI model.

Our experiments employ the RCI model with ALBERT XXL version (Lan et al., 2020). The RCI model is fine-tuned for different benchmarks with the following configurations: (1) training batch size = 128; (2) Number of epochs = 2; (3) Learning rate = 2.5e-5; and (4) maximum sequence length = 512.

The model and data for the experiments with CLTR are available at *https://github.com/IBM/row-column-intersection*.

Evaluation metrics: For table retrieval evaluation, we use the three metrics from previous work (Zhang and Balog, 2018b; Shraga et al., 2020c) for the top-k retrieved tables, namely precision (P) with $k \in \{5, 10\}$, normalized discounted gain (NDCG) with $k \in \{5, 10, 20\}$, and the mean average precision (MAP). For the end-to-end table QA tasks, we evaluate our proposed model following Glass et al. (2020) with two commonly used metrics in the IR community, accuracy at top 1 retrieved answer (Hit@1) and the mean reciprocal rank (MRR).

All experimental results are evaluated with the TREC standard evaluation tool (Voorhees and Harman, 2005). The source code of the TREC evaluation tool can be found at *https://trec.nist.gov/trec_eval/*.

4.3 Experimental Results

We experimentally compare CLTR against the BM25 baseline and the current state-of-the-art model on table retrieval in this section. Furthermore, we test CLTR with our proposed benchmarks on the end-to-end table QA task.

	P@5	P@10	N@5	N@10	N@20	MAP
BM25	0.5938	0.6587	0.5228	0.5356	0.5359	0.4704
CLTR	**0.7437**	**0.8735**	**0.6915**	**0.7119**	**0.7321**	**0.5971**

(a) E2E_WTQ

	P@5	P@10	N@5	N@10	N@20	MAP
BM25	0.0413	0.0242	0.1650	0.1764	0.1852	0.1601
MTR_{point}	0.1460	0.0767	0.6227	0.6349	0.6359	0.5920
MTR_{pair}	0.1826*	0.0990*	0.6945*	0.7198*	0.7220*	0.6328*
CLTR	**0.2203**	**0.1660**	**0.7235**	**0.7402**	**0.7458**	**0.7176**

(b) E2E_GNQ

Table 2: A comparison of CLTR and the baselines (* indicates the current state-of-the-art numbers).

Table Retrieval: We present the experimental results for table retrieval without task specific training on E2E_WTQ in Table 2a. Since the MTR model (Shraga et al., 2020c) is not available to us and this dataset has never been used in any published table retrieval work, we only compare our results to the coarse-grained BM25 baseline. The results indicate our proposed model outperforms the BM25 baseline with average improvements of 29.12%, 33.94% and 26.93% on precision, NDCG, and MAP, respectively. The results on E2E_WTQ also indicate that pre-trained CLTR can be adapted to new datasets without task-specific training.

The experimental results for E2E_GNQ are shown in Table 2b, comparing against BM25 and the current state-of-the-art, the two MTR models, MTR_{point} (with point-wise training) and MTR_{pair} (with pair-wise training) in Shraga et al. (2020c). The comparison shows that our proposed model outperforms the current best MTR_{pair} model on all metrics, with an average improvement of 28.73% on precision, 3.43% on NDCG, and 13.40% on MAP. The experimental results indicates CLTR is the new state-of-the-art system for table retrieval. Moreover, CLTR can further locate cell values to answer NLQs after table retrieval.

	MRR	Hit@1
E2E_WTQ	0.5503	0.4675
E2E_GNQ	0.4067	0.2699

Table 3: Model evaluation for end-to-end table QA

End-to-End Table QA: To further validate CLTR, we implement the end-to-end Table QA evaluation with E2E_WTQ and E2E_GNQ. The only existing end-to-end table QA model, Sun et al. (2016), and its dataset are not publicly available. Therefore, we do not have any baseline models to compare to. Our experimental results are reported in Table 3. As the first attempt for an end-to-end table QA system with transformer-based architecture on complex table benchmarks, we show that our approach is able to achieve promising and consistent performance. Our results indicate CLTR performs better for the first benchmark, E2E_WTQ, where the table corpus mainly contains well-structured tables. On the other hand, we expect the results for E2E_GNQ to be worse due to the amount of poorly formatted tables in the table corpus.

Qualitative Analysis: The experiments indicate CLTR outperforms all baselines, as well as the current state-of-the-art models on table retrieval. It also produces promising results for the end-to-end table QA task. We further demonstrate the high-portability of CLTR with pre-trained models using unseen benchmarks.

The system performance is much better for E2E_WTQ based on the experimental results. After a thorough investigation, we notice that the original *GNQtables* contains a large amount of noisy tables which do not have tabular structures. A considerable amount of tables in *GNQtables* are Wikipedia *InfoBoxes*, which may have multiple column/row headers and are difficult to process by machines accurately. Although table quality is crucial for table QA models, CLTR proves its advantageous by producing state-of-the-art results with noisy table corpus. Furthermore, the example shown in Figure 2 demonstrates the effectiveness of CLTR when applied to real-world data.

5 Related Work

Table Retrieval A majority of the table retrieval methods proposed in the literature treat tables as individual documents without taking the tabular structure into consideration (Pyreddy and Croft, 1997; Wang and Hu, 2002; Liu et al., 2007; Cafarella et al., 2008, 2009). More recent approaches utilize features generated from queries, tables, or query-table pairs. For example, Zhang and Balog (2018b) introduces an ad-hoc table retrieval method, retrieving tables with features such as #query_term, #columns, #null_values, etc. Similar work includes

Sun et al. (2019), Bhagavatula et al. (2013), and Shraga et al. (2020a). The current state-of-the-art model is introduced in Shraga et al. (2020c), where tables are treated as multi-modal objects and retrieved with a neural ranking model. We compare CLTR with this approach in Section 4.

Table QA Models Early table QA systems typically convert natural language questions into SQL format to answer questions over tables (Yu et al., 2018; Guo and Gao, 2020; Lin et al., 2019; Xu et al., 2018). In Jiménez-Ruiz et al. (2020), the authors promote the idea of matching tabular data to knowledge graphs and create the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), which provide a new solution for table understanding and QA related tasks. Recently, TAPAS (Herzig et al., 2020) and TABERT (Yin et al., 2020) introduce the transformer-based approaches for this task. The RCI (Glass et al., 2020) model is the state-of-the-art model for QA over tables. It utilizes a transfer learning based framework to independently classify the most relevant columns and rows for a given question and further identify the most relevant cells as the intersections of top-ranked columns and rows.

End-to-End Table QA Models To the best of our knowledge, the table cell search framework published in Sun et al. (2016) is the only existing end-to-end Table QA system. This work leverages the semantic relations between table cells and uses relational chains to connect queries to table cells. However, the proposed model only works for well-formatted questions containing at least one highly relevant entity to link tables to the questions. In addition, the model and the data are not publicly available for comparison.

6 Conclusion

This paper proposes an end-to-end solution for table retrieval and finding answers for NLQs over tables. To the best of our knowledge, this is the first system built where a transformer-based QA model is used for locating answers over tables while improving the ranking of tables out of a table pool formed by inexpensive IR methods. To evaluate the efficacy of this system, we introduce two benchmarks, namely E2E_WTQ and E2E_GNQ.

The experimental results indicates that the proposed system, CLTR, outperforms the baselines and the current state-of-the-art model on the table retrieval task. Furthermore, CLTR produces promising results on the end-to-end table QA task. In real-world applications, CLTR can be applied to create a heatmap over tables to assist users in quickly identifying the correct cells on tables.

References

Chandra Bhagavatula, Thanapon Noraset, and Doug Downey. 2013. Methods for exploring and mining tables on wikipedia. *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*.

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Douglas C Downey. 2015. Tabel: Entity linking in web tables. In *The Semantic Web – ISWC 2015 - 14th International Semantic Web Conference, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 425–441. Springer Verlag.

Michael J Cafarella, Alon Halevy, and Nodira Khoussainova. 2009. Data integration for the relational web. *Proceedings of the VLDB Endowment*, 2(1):1090–1101.

Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021. Open question answering over tables and text. In *International Conference on Learning Representations*.

L. Deng, Shuo Zhang, and K. Balog. 2019. Table2vec: Neural word and entity embeddings for table population and retrieval. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Michael Glass, Mustafa Canim, Alfio Gliozzo, Saneem Chemmengath, Rishav Chakravarti, Avi Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2020. Capturing row and column semantics in transformer based question answering over tables. *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT2020)*.

Tong Guo and Huilin Gao. 2020. Content enhanced bert-based text-to-sql generation.

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, Seattle, Washington, United States. Association for Computational Linguistics.

Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. 2016. Tabmcq: A dataset of general knowledge tables and multiple-choice questions.

Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, and Kavitha Srinivas. 2020. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In *ESWC*, pages 514–530.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Kevin Lin, Ben Bogin, Mark Neumann, Jonathan Berant, and Matt Gardner. 2019. Grammar-based neural text-to-sql generation.

Ying Liu, Kun Bai, Prasenjit Mitra, and C Lee Giles. 2007. Tableseer: automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 91–100.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables.

Pallavi Pyreddy and W Bruce Croft. 1997. Tintin: A system for retrieval in text tables. In *Proceedings of the second ACM international conference on Digital libraries*, pages 193–200.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Roee Shraga, Haggai Roitman, Guy Feigenblat, and Mustafa Canim. 2020a. Ad hoc table retrieval using intrinsic and extrinsic similarities. In *Proceedings of The Web Conference 2020*, pages 2479–2485.

Roee Shraga, Haggai Roitman, Guy Feigenblat, and Mustafa Canim. 2020b. Ad hoc table retrieval using intrinsic and extrinsic similarities. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2479–2485. ACM / IW3C2.

Roee Shraga, Haggai Roitman, Guy Feigenblat, and Mustafa Cannim. 2020c. Web table retrieval using multimodal deep learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1399–1408, New York, NY, USA. Association for Computing Machinery.

Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. Table cell search for question answering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 771–782.

Yibo Sun, Zhao Yan, Duyu Tang, Nan Duan, and Bing Qin. 2019. Content-based table retrieval for web queries. *Neurocomputing*, 349:183–189.

Ellen M. Voorhees and Donna K. Harman. 2005. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press.

Yalin Wang and Jianying Hu. 2002. A machine learning based approach for table detection on the web. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, page 242–250, New York, NY, USA. Association for Computing Machinery.

Xiaojun Xu, Chang Liu, and Dawn Song. 2018. SQLNet: Generating structured queries from natural language without reinforcement learning.

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.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Shuo Zhang and K. Balog. 2018a. Ad hoc table retrieval using semantic similarity. *Proceedings of the 2018 World Wide Web Conference*.

Shuo Zhang and Krisztian Balog. 2018b. Ad hoc table retrieval using semantic similarity. In *Proceedings of the 2018 World Wide Web Conference*, pages 1553–1562.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.