THoRR: Complex Table Retrieval and Refinement for RAG

Kihun Kim*, Mintae Kim, Hokyung Lee, Seongik Park, Youngsub Han and Byoung-Ki Jeon

LG UPLUS, 71, Magokjungang 8-ro, Gangseo-gu, Seoul, Republic of Korea

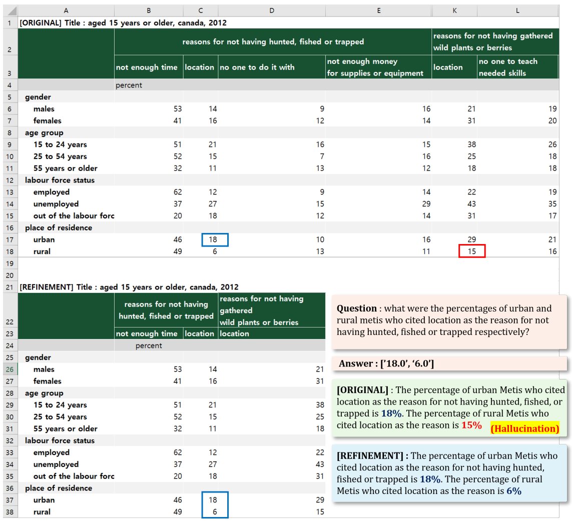
Abstract

Recent advancements in the contextual understanding and generation capabilities of Large Language Models (LLMs) have sparked increasing interest in the application of Retrieval-Augmented Generation (RAG) in specific domains and industry documents. Retrieving and understanding tables within these documents is crucial for generating correct answers in RAG systems. This study focuses on documents containing large and complex tables, such as statistical and industry reports and these presents two major challenges: 1) processing the large tables and 2) understanding complex tables. Previous studies faced challenges as they considered elements of tabular data such as cells, headers, and titles. In contrast, we designed the Table Header for Retrieval and Refinement (THoRR) method to address the aforementioned issues. THoRR performs two tasks: table retrieval and table refinement. In the table retrieval phase, we propose a table header representation approach that uses headers and titles, without considering cells. In the refinement phase, the model selects relevant table headers from the retrieved tables and processes them into refined tables containing the necessary information to answer the questions. This approach aids in understanding complex tables without chunking, by reorganizing information. Our models outperform existing approaches such as DTR and DPR-table. Moreover, we experimentally demonstrate that our refinement model can reduce hallucinations. To the best of our knowledge, our table refinement approach for RAG system is the first of its kind in the field.

Keywords

table retrieval, complex table, retrieval-augmented generation (RAG), table refinement, table representation

1. Introduction



[ORIGINAL] Title : aged 15 years or older, canada, 2012

	reasons for not having hunted, fished or trapped				reasons for not having gathered wild plants or berries	
	not enough time	location	no one to do it with	not enough money for supplies or equipment	location	no one to teach needed skills
	percent					
gender						
males	53	14	9	16	21	19
females	41	16	12	14	31	20
age group						
15 to 24 years	51	21	16	15	38	26
25 to 54 years	52	15	7	16	25	18
55 years or older	32	11	13	12	18	18
labour force status						
employed	62	12	9	14	22	19
unemployed	37	27	15	29	43	35
out of the labour forc	20	18	12	14	31	17
place of residence						
urban	46	18	10	16	29	21
rural	49	6	13	11	15	16

[REFINEMENT] Title : aged 15 years or older, canada, 2012

	reasons for not having hunted, fished or trapped		reasons for not having gathered wild plants or berries
	not enough time	location	location
	percent		
gender			
males	53	14	21
females	41	16	31
age group			
15 to 24 years	51	21	38
25 to 54 years	52	15	25
55 years or older	32	11	18
labour force status			
employed	62	12	22
unemployed	37	27	43
out of the labour forc	20	18	31
place of residence			
urban	46	18	29
rural	49	6	15

Figure 1: Example of tableQA with gpt-3.5-turbo. Comparing the result of the original complex table (top) and the refined table (bottom).

Recent advancements in the contextual understanding and generative capabilities of Large Language Models (LLMs) have heightened interest in Retrieval-Augmented Generation (RAG)[1, 2] for specific domains such as open domain or industry-specific documents. Industry or finance domain documents often contain large and complex tables. The understanding of which is critical for a RAG system to produce accurate responses. However, this task presents several challenges. Our research seeks solutions to two primary challenges.

The first challenge involves processing large and complex tables. Previous studies, such as DTR[3], and DPR-table[4], were designed with relatively simple open-domain tables in mind, such as those found in the nq-table[5] dataset, thus de-emphasizing the processing of large tables. Similar to the processing of text documents, previous methods involved dividing data tables into fixed-length segments (chunking), or even cutoff parts that exceeded a maximum input length. The chunking method complicates data retrieval by not only increasing the number of retrieval targets but also making it challenging to compare values across segmented tables. Moreover, disregarding overflow sections risks losing table information, diminishing the probability of obtaining a sufficient table representation. These problems can ultimately affect table retrieval performance.

The second challenge is the difficulty in understanding tables due to their complex structure. Complex tables typically feature hierarchical headers and numerous values, presenting a challenge for generator to consider vast amounts of information. Insufficient table comprehension can lead to incorrect answers (hallucinations). Figure 1 demonstrates an example where GPT-3.5-turbo[6] is used to perform tableQA on a hierarchical table. It showcases how the original table leads to incorrect responses, whereas the refined table, as processed by our proposed model, yields the correct answers.

In this paper, we propose **T**able **H**eader F**or** **R**etrieval and **R**efinement (**THoRR**) to solve this problem. These method is grounded in a heuristic assumption that, when finding and understanding tables, headers are more critical than values. THoRR has two models, a retriever and a refinement model, performed sequentially. Each is different from the previous one. THoRR: Retriever uses a table header representation. It performs table retrieval using only the header without considering the cells of the table. THoRR: Refinement performs relevant table header detection in the retrieved table to select table headers that are relevant to the question, and refines them into a simple table that contains only the necessary information, reducing the amount of information the generator needs to consider.

We compare THoRR with DTR[3] and DPR-table[4] and show that it has better retrieval performance in fine-tuning and zero-shot experiments on the HiTab[7] and AIT-QA[8]

IR-RAG'24: Information Retrieval's Role in RAG Systems (IR-RAG), July 18, 2024, IR-RAG, Washington D.C.

*Corresponding author.
kimkihun@lguplus.co.kr (K. Kim); iammt@lguplus.co.kr (M. Kim); hogay88@lguplus.co.kr (H. Lee); spark32@lguplus.co.kr (S. Park); yshan042@lguplus.co.kr (Y. Han); bkjeon@lguplus.co.kr (B. Jeon)
0009-0005-9453-7443 (K. Kim)

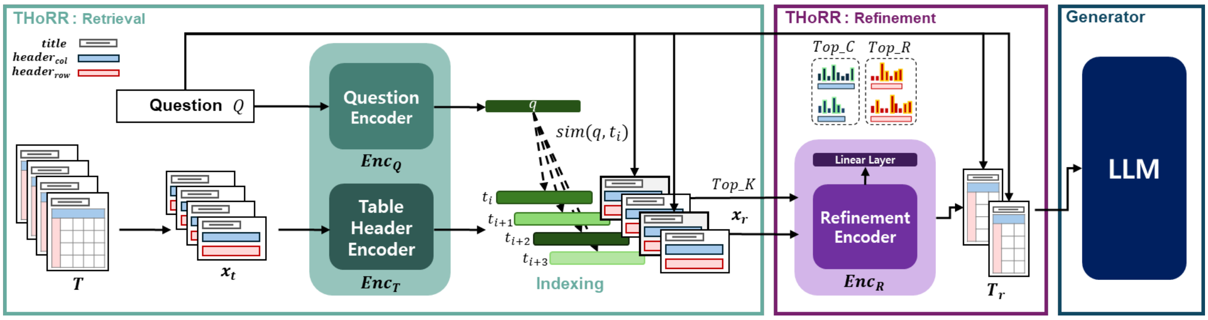


Figure 2: Architecture of the RAG system with LLM for table data, featuring our proposed THoRR method.

datasets while reducing the information (number of cells) needed to input the generator. Furthermore, our proposed methodology enables an efficient reduction in the number of tokens required for table inputs in the generator.

2. Method

In this section, we present the **T**able **H**eader F**or** **R**etrieval and **R**efinement (**THoRR**) method, designed to retrieve and refine tables within the RAG system [2]. THoRR is divided into two phases, retrieval and refinement, as shown in Figure 2. These two phases are separately trained and serve distinct purposes. The retrieval phase is utilized for embedding and indexing tables. Subsequently, as a question is input, it retrieves the pre-indexed tables. In the refinement phase, the retrieved tables are processed to extract the necessary information, refining them into smaller tables.

The goal of this method is to obtain the Top_K refined Tables T_r relevant to the given M target Tables T when a question Q is provided. We denote the components of T as $title$, $header_{row}$, and $header_{col}$, representing the row headers, column headers, and title, respectively. The comparative experiments between THoRR and the existing table retrieval baseline are explained in Section 3.1

2.1. Table Retriever

Given M target tables T, Our THoRR:retrieval model aims to retrieve the Top_K candidate tables containing information relevant to the question Q. In this paper, we follow the structure of DPR[9] for comparison with DPR-table[4]. we use two different encoders (the table header encoder (Enc_T) and the question encoder (Enc_Q), both utilizing the base model of [10]. Enc_T maps target M tables to table header represents t and builds an index t that will be used for retrieval. The input x_t to Enc_T is defined in equation 1. When given a question Q, obtain a question representation q using the Enc_Q, and then select the Top_K closest candidate tables for indexed t from it. The similarity between t and q is defined by using the dot product, as in [9] (equation 2), and the encoder uses the base model of [10].

$$x_t = \{[CLS]\ title\ [SEP]\ header_{col}\ [SEP]\ header_{row}\ [SEP]\} \tag{1}$$

$$Sim(q,t) = Enc_Q(Q)^\top \cdot Enc_T(x_t) \tag{2}$$

In this process, a difference aspect of our retriever compared to previous research lies in the table header representation t. Our method, which utilizes the table's header and title without considering every cell, is relatively free from the input limitations of the encoder. The chunking method and our comparative experiments are explained in Section 3.2

The objective of the training is to minimize the distance between questions q and positive table t_i^+ while maximizing the distance between queries and the number of n negative tables t_i^- in a given training dataset $D = \{(q_i, t_i^+, t_{i,1}^-, t_{i,2}^-, ..., t_{i,n}^-)\}_{i=1}^{M}$. The loss function, optimized as Negative Log Likelihood (NLL) :

$$\begin{aligned} &L_{retriever}(q_i, t_i^+, t_{i,1}^-, t_{i,2}^-, ..., t_{i,n}^-) \\ &= -log \frac{e^{sim(q_i,t_i^+)}}{\sum_{j=1}^{n} e^{sim(q_i,t_i^+)} + e^{sim(q_i,t_{i,j}^-)}} \end{aligned} \tag{3}$$

2.2. Table Refinement Model

This paper introduces a new task called Table Refinement, defined as simplifying a table while preserving specific information. Accordingly, our THoRR:refinement model aims to obtain refined tables, denoted as t_r, for the Top_K candidate tables from the retrieval phase. The input x_r is defined by equation 4, where the $header \in [header_{row}, header_{col}]$. Similar to equation 5, x_r is input to the refinement encoder Enc_R to obtain hidden states. And then, the linear layer takes in these hidden states and outputs the relevant header score, denoted as h. Using h, we obtain the Top_C relevant column headers indices (I_c) and Top_R relevant row header indices (I_r) as specified in Equation 6. Subsequently, we refine candidate tables using selected row and column indices to obtain t_r.

$$x_r = \{[CLS]\ Q\ [SEP]\ header\ [SEP]\} \tag{4}$$

$$h = Enc_R(x_r) \tag{5}$$

$$\begin{aligned} I_{row} &= argmax(h_{row}, Top\_R) \\ I_{col} &= argmax(h_{col}, Top\_C) \end{aligned} \tag{6}$$

The learning objective aims to identify the index of the question and relevant header, with the goal of increasing the score of the answer's header index h_i. The loss function is as described in Equation (5), We optimized Cross Entropy Loss. Where, N represents the number of tokens in input x_r and y is the gold relevant header index.

$$L_{refinement}(h,y) = -log \frac{exp(h_y)}{\sum_{n=1}^{N} = exp(h_n)} \tag{7}$$

Table 1
Comparison of retrieval accuracy performance of our THoRR method and the baselines. Fine-tuning denotes training with the HiTab dataset and Zero-shot denotes evaluation of AIT-QA using fine-tuned model.

Model	Refinement		HiTab Fine-tuning					AIT-QA Zero-shot				
	Top_C	*Top_R*	HIT@1	HIT@5	HIT@10	HIT@20	HIT@50	HIT@1	HIT@5	HIT@10	HIT@20	HIT@50
DTR[3]	-	-	19.00	40.97	51.96	64.27	77.53	8.74	20.39	28.16	41.75	71.07
DPR-table[4]	-	-	40.40	69.51	77.15	84.03	90.66	19.61	41.75	55.15	71.26	89.51
THoRR	5	-	45.39	74.75	82.83	87.31	91.60	22.52	47.38	62.91	74.95	92.82
(Ours)	5	10	43.50	71.84	79.55	84.03	88.07	21.75	44.27	59.03	69.51	84.85
	7	-	**45.77**	**75.51**	**83.59**	**88.07**	**92.49**	**23.50**	**48.54**	**64.47**	**76.89**	**94.76**
	7	10	43.88	72.60	80.30	84.79	88.95	22.72	45.44	60.58	71.46	86.80

3. EXPERIMENTS

Dataset We conduct experiments on two complex table benchmark datasets. HiTab[7] is a Table QA dataset with a hierarchical structure. This dataset consists of questions that require complex numerical calculations, including tables from Wikipedia and statistical reports. It contains a total of 10,672 question-answer pairs, with 7,417 for training, 1,671 for validation, and 1,584 for testing. There are a total of 3,597 tables in this dataset. We use this dataset for fine-tuning. AIT-QA[8] is a Table QA dataset specific to the Airline industry, composed of tables extracted from the U.S. public SEC filings. It includes specialized vocabulary terms for a specific domain and also has a hierarchical structure like HiTab[7]. It consists of 515 questions and answers, with a total of 116 tables. In this paper, this dataset is used to evaluate the zero-shot performance of the fine-tuning model.

Baseline In order to demonstrate the performance of our method, we compare it with baseline methods. DTR[3] is a table encoder that uses a table-specific structure. DPR-table[4], on the other hand, processes tables linearly, similar to understanding text passages. Both of these baselines have been trained on the nq-dataset[5] and their pretrained models are publicly available. We fine-tune these pre-train models as backbones and compare them with our model.

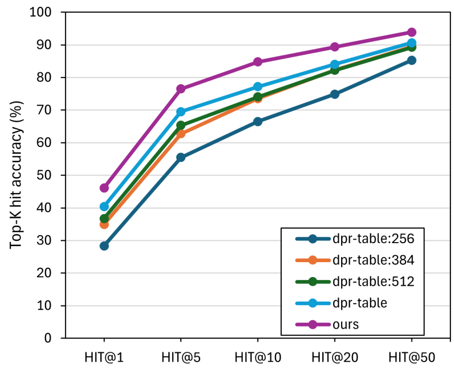
3.1. Main Result : THoRR

The experiments in this paper evaluate the proposed models, THoRR, in a two-phase process as shown in Figure 1 (THoRR:retrieval and THoRR:refinement). The performance of the models is evaluated using the 'Hits accuracy' as the main evaluation metric. This metric measures the ratio of correct answers included in the Top_K selected tables by the models. Where, Top_K takes values 1, 5, 10, 20, 50 to evaluate the accuracy of the models.

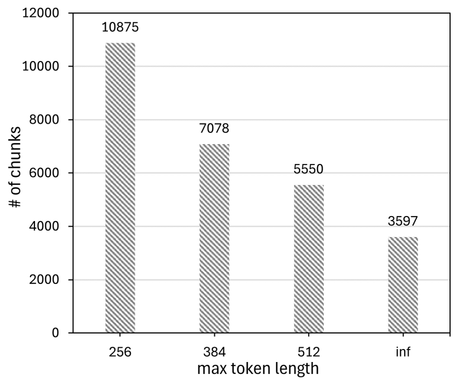
Fine-tuning To compare fine-tuning experiments on the complex table dataset, we train THoRR and baselines using the HiTab [7] training set. Table 1 presents the performance of the THoRR method compared to baseline models. The experimental results indicate that the proposed models outperformed baselines in most cases. When $Top\_C = 7$ and $Top\_K <= 10$, the proposed models exhibit an accuracy improvement of more than 5% compared to the baseline's best accuracy. The superior performance at a small Top_K indicates the importance in the RAG system, as it indicates effective utilization of a limited number of reference pieces of information, which is common when the Top_K is less than 10.

Zero-shot The zero-shot experiment intend to observe how the model performs on complex table data from a new domain. In this process, a fine-tuned model using the HiTab [7] dataset is used to make predictions on the AIT-QA[8] dataset without any additional training, and the results are evaluated. Through this experiment, we aim to demonstrate that the proposed models can handle complex table retrieval in previous unseen domains. Table 1 presents the results of this experiment, showing superior performance compared to the baselines and indicating well THoRR works on complex tables in different domains.

3.2. Retrieval Result



(a)



(b)

Figure 3: (a) Retriever accuracy with DPR-table's chunking method vs THoRR:retrival's table header representation method. (b) Comprison between the number of chunks by the max token length.

We compare our proposed table header representation method and chunking method in terms of retrieval accuracy. Figure 3(a) illustrates the performance of [4] with the chunking method and our method. (”inf” refers to the use of the original table without chunking.) As shown in Figure 3(b), we observe a decrease in retrieval accuracy as the lower max token length, indicating that the number of retrieval targets affects the performance significantly in retrieval tasks. Our approach demonstrates superior performance compared to methods that consider all values. This highlights the effectiveness of our method, which relies solely on table headers for table representation, especially in retrieving large and complex tables. Moreover, our method demonstrates superior performance compared to existing approaches that consider all values, thereby experimentally validating our heuristic assumption that headers are crucial elements in table retrieval.

3.3. Refinement Result

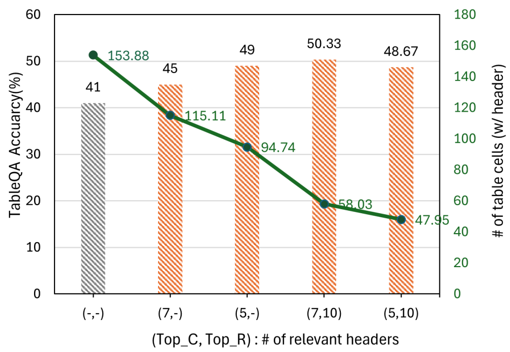


Figure 4: Comparison of human evaluation performance on TableQA and the number of refined table cells.

In this section, we experiment with our refinement model to reduce cell information in mitigating hallucinations. In Figure 4, the green line indicates a decreasing trend in the number of cells in tables when using our model. Furthermore, Figure 4 illustrates the human evaluation accuracy on the results obtained by input refined tables into Llama2[11] 7B-Chat. Where, ”(-,-)” denotes the original table. We randomly sample 300 questions from the HiTab test dataset for human evaluation. Llama2[11] 7B-Chat takes a gold table as input to generate responses. If the generated response contains exactly the answer and is correct, we mark it as correct. Otherwise, we consider it as a hallucination. Three master’s students in the field of AI evaluated the generated results. To ensure the reliability of the evaluations, one evaluator and two validators were assigned roles in the evaluation process. As a result, by setting $Top\_C = 7$ and $Top\_R = 10$, we demonstrate that our refinement model reduces the number of table cells from 153.88 to 58.03, resulting in a 62.2% decrease compared to the original table. Additionally, we observe a 9.33% improvement in the reduction of hallucinations. This validates the superiority of our refinement approach.

4. Related Works

Research on table encoders has been focused on pre-training tabular data with table-specific architectures[12, 13, 14, 15, 16, 17]. TAPAS[12] introduces a pre-training method using Masked-Language-Modeling for the cells of tabular data. TaBERT[13] introduces a pre-training model that jointly learns over 26 million natural language questions and tables. TURL[18] introducing a structure-aware Transformer encoder and Masked Entity Recovery (MER) objective for pre-training. StruG[14] proposes a semi-supervised learning framework for learning the connection between text and SQL. MATE[15] demonstrates the efficient restriction of Transformer attention flow on tabular data, enabling training with larger sequence lengths. Tableformer[16] learns from tables using attention biases, making it better at understanding tabular data. TABBIE[17] introduces a method to improve performance on table-based prediction tasks by pre-training only tabular data.

Research on table retrieval includes methodologies such as [19, 3, 4, 20, 21]. Table2vec[19] proposes a method for obtaining table embeddings by considering various table elements such as captions, headers, cells, and entities. DTR[9] introduces a table-specific model suitable for open-domain table question answering. DPR-table[4] linearizes tables to handle them similar to text passages, instead of using table-specific models. GTR[20] introduces a model that transforms tables into graphs, capturing both cell and layout structures. [21] introduces a method for enhancing the similarity between queries and tables for table retrieval, employing various semantic spaces and similarity measurement methods.

5. Conclusion

We propose the THoRR method, which uses the table headers to retrieve and help understand the complex and large tables. We use the table header representations in the retriever that can retrieve tables without chunking them. Additionally, we propose a novel methodology for refining tables by detecting the table headers that are relevant to the questions within the table. This approach aims to simplify the tables in which an excessive amount of information is present, particularly in complex tables. THoRR is capable of handling large and complex tables without dividing them into smaller chunks, reducing the information required for preventing hallucinations in LLM generator. Furthermore, the Table Refinement task is the first of its kind in this field, therefore, it is expected to contribute significantly to the future research in this field. Our future work involves exploring methods to detect the table headers. Additionally, we aim to prevent potential information loss in questions by selecting fewer relevant headers during the refinement phase.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, ArXiv abs/2312.10997 (2023). URL: https://api.semanticscholar.org/CorpusID:266359151.

[3] J. Herzig, T. Müller, S. Krichene, J. Eisenschlos, Open domain question answering over tables via dense re-

trieval, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 512–519. URL: https://aclanthology.org/2021.naacl-main.43. doi:10.18653/v1/2021.naacl-main.43.

[4] Z. Wang, Z. Jiang, E. Nyberg, G. Neubig, Table retrieval may not necessitate table-specific model design, in: W. Chen, X. Chen, Z. Chen, Z. Yao, M. Yasunaga, T. Yu, R. Zhang (Eds.), Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI), Association for Computational Linguistics, Seattle, USA, 2022, pp. 36–46. URL: https://aclanthology.org/2022.suki-1.5. doi:10.18653/v1/2022.suki-1.5.

[5] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, S. Petrov, Natural questions: A benchmark for question answering research, Transactions of the Association for Computational Linguistics 7 (2019) 452–466. URL: https://aclanthology.org/Q19-1026. doi:10.1162/tacl_a_00276.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[7] Z. Cheng, H. Dong, Z. Wang, R. Jia, J. Guo, Y. Gao, S. Han, J.-G. Lou, D. Zhang, HiTab: A hierarchical table dataset for question answering and natural language generation, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 1094–1110. URL: https://aclanthology.org/2022.acl-long.78. doi:10.18653/v1/2022.acl-long.78.

[8] Y. Katsis, S. Chemmengath, V. Kumar, S. Bharadwaj, M. Canim, M. Glass, A. Gliozzo, F. Pan, J. Sen, K. Sankaranarayanan, S. Chakrabarti, AIT-QA: Question answering dataset over complex tables in the airline industry, in: A. Loukina, R. Gangadharaiah, B. Min (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 2022, pp. 305–314. URL: https://aclanthology.org/2022.naacl-industry.34. doi:10.18653/v1/2022.naacl-industry.34.

[9] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. URL: https://aclanthology.org/2020.emnlp-main.550. doi:10.18653/v1/2020.emnlp-main.550.

[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[11] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[12] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, J. Eisenschlos, TaPas: Weakly supervised table parsing via pre-training, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 4320–4333. URL: https://aclanthology.org/2020.acl-main.398. doi:10.18653/v1/2020.acl-main.398.

[13] P. Yin, G. Neubig, W.-t. Yih, S. Riedel, TaBERT: Pretraining for joint understanding of textual and tabular data, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8413–8426. URL: https://aclanthology.org/2020.acl-main.745. doi:10.18653/v1/2020.acl-main.745.

[14] X. Deng, A. H. Awadallah, C. Meek, O. Polozov, H. Sun, M. Richardson, Structure-grounded pretraining for text-to-SQL, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 1337–1350. URL: https://aclanthology.org/2021.naacl-main.105. doi:10.18653/v1/2021.naacl-main.105.

[15] J. Eisenschlos, M. Gor, T. Müller, W. Cohen, MATE: Multi-view attention for table transformer efficiency, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 7606–7619. URL: https://aclanthology.org/2021.emnlp-main.600. doi:10.18653/v1/2021.emnlp-main.600.

[16] J. Yang, A. Gupta, S. Upadhyay, L. He, R. Goel, S. Paul, TableFormer: Robust transformer modeling for table-text encoding, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 528–537. URL: https://aclanthology.org/2022.acl-long.40. doi:10.

18653/v1/2022.acl-long.40.

[17] H. Iida, D. Thai, V. Manjunatha, M. Iyyer, TABBIE: Pretrained representations of tabular data, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 3446–3456. URL: https://aclanthology.org/2021.naacl-main.270. doi:10.18653/v1/2021.naacl-main.270.

[18] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, Turl: table understanding through representation learning, Proc. VLDB Endow. 14 (2020) 307–319. URL: https://doi.org/10.14778/3430915.3430921. doi:10.14778/3430915.3430921.

[19] L. Zhang, S. Zhang, K. Balog, Table2vec: Neural word and entity embeddings for table population and retrieval, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1029–1032. URL: https://doi.org/10.1145/3331184.3331333. doi:10.1145/3331184.3331333.

[20] F. Wang, K. Sun, M. Chen, J. Pujara, P. Szekely, Retrieving complex tables with multi-granular graph representation learning, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1472–1482. URL: https://doi.org/10.1145/3404835.3462909. doi:10.1145/3404835.3462909.

[21] S. Zhang, K. Balog, Semantic tablenbsp;retrieval using keyword and table queries, ACM Trans. Web 15 (2021). URL: https://doi.org/10.1145/3441690. doi:10.1145/3441690.