Table Header Recognition Based on Large Language Models

Ilia I. Okhotin
Matrosov Institute for System Dynamics and Control Theory of SB RAS
Irkutsk, Russia
ISP RAS Research Center for Trusted Artificial Intelligence
Moscow, Russia
ilia.ohotin@yandex.ru

Nikita O. Dorodnykh
Matrosov Institute for System Dynamics and Control Theory of SB RAS
Irkutsk, Russia
ISP RAS Research Center for Trusted Artificial Intelligence
Moscow, Russia
nikidorny@icc.ru

ABSTRACT

Automatic table header recognition remains a challenging task due to the diversity of table layouts, including multilevel headers, non-standard formatting, and merged cells. In this paper, for the first time, we propose a methodology to evaluate the performance of large language models on this task. Our study covers eight different models and six strategies for prompt engineering with zero-shot and few-shot settings, on a prepared dataset of 237 tables. The results show that model size critically affects accuracy: large models (405b parameters) achieve the F1 score 80-85%, while small ones (7b parameters) show the F1 score 6-30%. Complicating prompts with step-by-step instructions, search criteria, and examples that improves the results only for large and medium models, while for small ones it leads to degradation due to context overload. The greatest errors occur when processing tables with hierarchical headers and merged cells, where even large models lose accuracy. The practical significance of this paper lies in identifying optimal configurations of prompts for different types of models. In particular, short instructions are effective for large models, and step-by-step instructions with search criteria are effective for medium models. This research opens up new possibilities for creating universal tools for automatic analysis of table headers.

VLDB Workshop Reference Format:
Ilia I. Okhotin and Nikita O. Dorodnykh. Table Header Recognition Based on Large Language Models. VLDB 2025 Workshop: Tabular Data Analysis (TaDA).

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/YRL-AIDA/Table_Header_Recognition.

1 INTRODUCTION

Tables are an important means for presenting structured data in scientific publications, financial reports, web documents, and other fields, but their automatic processing is hindered by the diversity of structures, including multilevel headers and merged cells [1, 2, 4, 19]. Modern approaches to table structure recognition, including neural network-based methods [8, 14, 25], as well as pre-trained language models based on the Transformer architecture [3, 9, 10, 15, 22, 26, 28], show progress in this field, but face limitations when working with hierarchical headers and noisy data. The emergence of large language models (LLMs) has opened up new possibilities for processing tabular data due to their ability to more deeply analyze the context and semantics of data [5, 20, 23]. However, their effectiveness depends on prompt engineering strategies, which remain understudied for the task of table header recognition.

In this paper, we address the task of table header recognition using LLMs and prompt engineering. Our contributions include:

(1) For the first time, an experimental evaluation of the performance of LLMs in the context of solving the problem of table header recognition is obtained.
(2) The influence of different prompt types, LLM sizes, and structural complexity of tables on the accuracy of header recognition is studied.

2 RELATED WORK

In the field of automatic table understanding [2, 16, 19], the main problems of table processing are considered such as **table detection** (searching and identifying tables in the original information source), **table structure recognition** (defining rows, columns, and cells, as well as headers) and **semantic table interpretation** (annotating table elements with concepts from the knowledge graph).

There are two main directions to the development of methods to solve the problems of table structure recognition:

- **Rule-based methods**: These methods are based on the rules of analysis and interpretation of tables [6, 17, 24]. Such solutions typically do not cover the full diversity of table layouts, formatting, and content. They are limited to conventional layouts, atomic cells, and flat headers, ignoring cases where these assumptions do not hold.
- **Data-driven methods**: Such solutions can use both traditional machine learning-based methods such as the support vector machine (SVM) and random forest [7], or cluster analysis [18], and deep learning-based methods [8, 14, 25] including as pre-trained language models such as TSR-Former [15], TableFormer [26], TableVLM [3], VAST [9], TATR [22], GTE [28], and TSR-DSAW [10].

Recently, approaches based on LLMs have emerged (e.g., Table-GPT [13] or TableLlama [27]), which are increasingly being used to solve various tabular tasks using prompt engineering and contextual learning. The representation of a prompt for a table can play

Proceedings of the VLDB Endowment. ISSN 2150-8097.

an important role in the ability of models to process tabular data. In particular, the ability of LLMs to understand the structure of tables with different layouts using the prompt engineering technique is investigated in [20, 23]. In addition, a performance evaluation is provided for individual table presentation formats and the impact of noise in the data. However, such approaches are aimed only at structural tasks that relate to the identification of rows, columns, and cells by specified indexes, omitting the processing of headers. Other successful examples are approaches [11, 12] that focus on the ability of LLMs to understand the semantics of tabular data, making one of the key contributions to research on column annotation and the relationships between them using prompt engineering. However, such approaches are aimed at mapping table columns (cell values with data) to semantic types (classes and properties) from the target knowledge graph (e.g., DBpedia or Wikidata) and are not able to determine metadata (headers) inside the table itself.

3 METHODOLOGY

This study evaluates the performance of LLMs in the task of table header recognition. Specifically, we explore the following aspects: 1) *How does prompt complexity and detail affect the model's ability to accurately identify headings?* 2) *How does the number of LLM parameters affect the accuracy of table header recognition?* 3) *How does the structural complexity of tables (e.g., the presence of merged cells, multilevel headers, or non-standard header placement) also affect recognition accuracy?*

The general scheme of our methodology is presented in Figure 1.

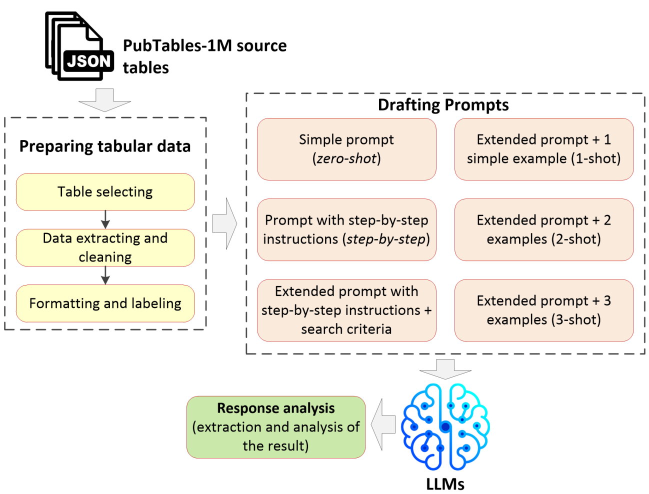

Figure 1: The general scheme of the proposed methodology, that supports controlled ablation of: (1) LLM scale effects, (2) structural table complexity (hierarchical headers/merged cells), and (3) prompting strategies. Our methodology establish a reproducible benchmark for table header recognition.

3.1 Models

We selected eight models with varying numbers of parameters to study the ability of LLMs to recognize table headers. The models are divided into three categories: **small** (<70 billion parameters), **medium** (>70 and <100 billion parameters), and **large** (>100 billion parameters), which determines the impact of model size on performance. Table 1 lists the LLMs with their brief characteristics.

Table 1: LLMs used in the study. We categorized 8 state-of-the-art LLMs by parameter scale (Small: 7–27B; Medium: 70–72B; Large: 405B), establishing a foundational framework for model capability analysis. The selection encompasses cutting-edge open-source (e.g., Llama 3, Gemma) and distilled models (DeepSeek-R1), enabling systematic investigation of scaling ways in tabular data understanding.

Model name	# Parameters
Small	
Mistral-7B-Instruct-v0.3	7 billion
Llama-3.1-8B-Instruct	8 billion
Mistral-Small-24B-Instruct-2501	24 billion
Gemma-2-27B-it	27 billion
Medium	
Llama-3.3-70B-Instruct-Turbo	70 billion
DeepSeek-R1-Distill-Llama-70B	70 billion
Qwen2-72B-Instruct	72 billion
Large	
Llama-3.1-405B-Instruct	405 billion

3.2 Dataset Pre-processing

Our study employs the large-scale PubTables-1M corpus that contains nearly one million tables extracted from scientific articles available in the PubMed Open Access archive [21]. This corpus provides rich annotations for table discovery, table structure recognition, and functional analysis tasks, including information about cell layout, content, and roles (e.g., headers, cell values with data).

We prepared a dataset including 237 tables from the PubTables-1M corpus with various table structures, in particular:

- 122 **simple tables** contain headers in the first row or column.
- 95 **medium tables** contain header cells within the first three rows (there may be merged cells and a hierarchical structure).
- 20 **complex tables** contain non-standard multilevel headers (there may be merged cells and a hierarchical structure), where header cells are located both in rows (they can be located below the third row) and in columns.

The selected tables were not intentionally noised. The key statistics for our dataset are provided in Table 2. According to [20], when solving various structured tabular tasks, LLMs better understand the DataFrame format (a JSON-like format used in the pandas library), which has shown the highest efficiency in experiments. Thus, we used this format for the subsequent processing and transferring of tabular data to our models.

Table 2: Statistics of the prepared dataset including 237 tables from the large-scale PubTables-1M corpus. The tables in this dataset are divided into three types: simple, medium and complex.

Statistics	Value
Total number of tables	237
Number of simple tables	122
Number of medium tables	95
Number of complex tables	20
Number of columns	1293
Average number of columns per table	5,46
Number of rows	3457
Average number of rows per table	14,59
Total number of cells	20353
Number of header cells	1546
Number of data cells	18807
Number of merged cells	602

3.3 Prompting Strategies

We used general guidelines[1] for writing prompts, but made these prompts specific to our task, since they include search criteria and instructions for header detection. In particular, prompt with step-by-step instructions provides a structured approach, guiding the model to a systematic analysis of the table, which can improve recognition accuracy. The prompt with search criteria takes the instructions further by providing the model with clear criteria to identify headers, which is especially useful for complex tables. We have also included zero-shot and few-shot settings in our prompts. An example of a prompt using the described format is found in Figure 2.

Thus, we designed various strategies for LLMs prompting in Table 3.

Table 3: Main characteristics of prompting strategies. We systematized 6 prompting configurations from zero-shot to few-shot (1–3 table examples), augmented with algorithmic step-by-step instructions (*+int*) and header search criteria (*+crit*). "N/A" denotes not table example in prompt.

#	Prompt type	Example type
1	*zero-shot*	N/A
2	*zero-shot+int*	N/A
3	*zero-shot+int+crit*	N/A
4	*1-shot+int+crit*	header in the first row
5	*2-shot+int+crit*	headings on top and left
6	*3-shot+int+crit*	hierarchical headers

[1] https://github.blog/ai-and-ml/generative-ai/prompt-engineering-guide-generative-ai-llms/

You are a specialist in analyzing complex tables.
Your task is to identify **all headers** in the given table: **{table}**.

Please follow the detailed algorithm below:

1. Visually examine the table to understand its structure.
2. Identify potential headers based on summarizing text and formatting.
3. Analyze the positioning of headers (top, left, right).
4. Generate a JSON response with the coordinates of the identified headers.

Header identification criteria:

- Semantic analysis to detect keywords indicating categories.
- Positional analysis to determine the typical placement of headers.
- Structural pattern analysis (e.g., repeated header rows or columns).
- Contextual analysis of relationships between cells.
- Metadata analysis (such as font size, alignment).
- Hierarchical analysis to detect multi-level headers.

Example of a table with recognized headers:
{table example}

Figure 2: An example of prompt pattern with step-by-step instructions, header identification criteria, and table instance (1-shot).

4 RESULTS AND DISCUSSION

We developed a special environment for conducting experiments in Python. We used the LangChain library[2] and the Together.ai API[3]. All requests to LLMs were made through this API with a generation temperature of 0.0 to ensure the stability and reproducibility of the results. The prepared tabular dataset, all types of prompts, and the experiment results are published in an open repository on GitHub[4].

Standard metrics were used to evaluate header recognition quality: *precision*, *recall*, and *F1 score*. The distribution of F1 scores across models is shown in Figure 3.

Sequential processing of 11376 queries (237 tables × 6 prompt types × 8 LLMs) took about 49.3K seconds (13.7 hours). Switching to parallel processing in five threads reduced this time to 9.8K seconds (2.7 hours), which accelerated query processing almost five times.

Experiments demonstrated that model size significantly affects performance. The large model (405B) achieved the F1 score 82-84%, while the small models (7-8B) showed the F1 score 6-70%. The Llama-3.1-8B, Qwen2-72B and Mistral-7B models demonstrate the greatest instability when using complex prompts. Instead of the expected quality improvement through step-by-step instructions, search criteria or a few-shot settings, these models showed the opposite result.

Key observations from the model analysis:

- **The negative impact of complex prompts:** Complex instructions sometimes misled the model, while the overall F1 score decreased.

[2] https://www.langchain.com/
[3] https://www.together.ai/
[4] https://github.com/YRL-AIDA/Table_Header_Recognition

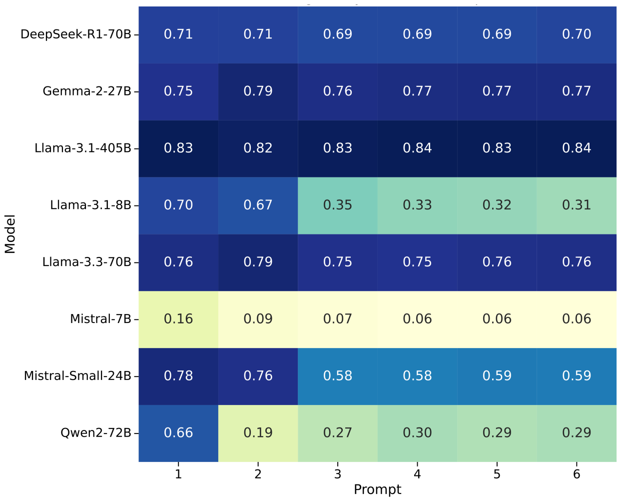

Figure 3: Heatmap of F1 scores across model-prompt configurations. Evaluation demonstrates the decisive impact of model scale on header recognition accuracy. Large models (405B) achieve peak performance (F1: 80-85%) even with minimal prompts. Medium models (70B) show significant gains (+15% F1) from criteria-augmented prompts, while small models (7B) exhibit severe degradation (F1: 6-30%) due to context overload. Thus, we highlight the inflection point (τ70B parameters) for reliable in-context learning in tabular domains.

- **Non-standard response format:** LLMs often returned not the coordinates of the header cells, but their contents, despite explicit instructions in the prompts to return the coordinates of the header. This is probably due to the architectural specificity of transformer-based generative models, which are optimized for predicting the next tokens rather than numeric coordinates; therefore, they extract and return the text header value rather than the position. In addition, LLMs are not specialized for this task and were trained using basic tables with a header in the first row; therefore, they tend to use this pattern. To compensate for such inconsistent responses, we implemented an additional step that provides text values are obtained by searching the source table and matching them with coordinates. This fix partially corrected this error, which was reflected in the final scores.
- **Header omission in complex tables:** LLMs suffered significant performance losses when the headers were not in the first row. Thus, almost all models showed a low recall for medium and complex tables, skipping true headings located outside the first row. Large and medium models handled such cases better, but still showed a noticeable decrease in the F1 score compared to simple tables.
- **False positive header identification:** LLMs exhibit a tendency to misclassify cells with text content as headers, especially in dense tables (such tables tend to contain the maximum amount of information in a minimal area without empty cells and may also lack obvious visual separators), resulting in reduced precision. An example of such a table is shown in Figure 4.
- **Coordinate identification errors:** Errors in determining the header boundaries are prevalent in tables containing merged cells or irregular layout, causing a decrease in the F1 score.

Table 2: Hematologic toxicity. Number of patients affected with each side effect are listed in the corresponding rows.

NCI CTC Grade	1	2	3	4
Leukopedia	7	1	0	1
Neutropenia	4	7	3	0
Anaemia	3	1	0	0

Figure 4: An illustrative example of a simple table where the Llama-3.3-70B model is wrong in automated header recognition. Ground-truth headers (highlighted in green) are compared with model-predicted headers (highlighted in red) for the Llama-3.3-70B model under a *zero-shot-int+crit* prompt. Despite the high overall efficiency of this model (F1: 75%), this dense tabular structure with a minimal visual separators and header-like numeric values in data cells lead to critical precision errors. Specifically, the model mistakenly attributes two data cells in the first column ("Leukopedia" and "Neutropenia") as a header while overlooking the true header in the first cell of the last column.

5 CONCLUSION

In this paper, we investigated the capabilities of LLMs to recognize table headers with different structural layouts. The experiments were carried out on a set of 237 tables selected from the large-scale PubTables-1M corpus. Our study demonstrates the potential of LLMs in table header recognition, highlighting the importance of model size and prompting strategies. Future work will focus on expanding the dataset, including tables on low-resource languages, and more examples of tables with complex structures. We plan to conduct experiments on existing table header detection datasets from the related work to make the results directly comparable and to explore hybrid approaches that combine LLMs with computer vision methods[9].

ACKNOWLEDGMENTS

This work was supported by a grant, provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925-P4G0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated June 20, 2025 No. 139-15-2025-011.

REFERENCES

[1] G. Badaro, M. Saeed, and P. Papotti. 2023. Transformers for Tabular Data Representation: A Survey of Models and Applications. *Transactions of the Association for Computational Linguistics* 11 (2023), 227–249. https://doi.org/10.1162/tacl_a_00544

[2] S. Bonfitto, E. Casiraghi, and M. Mesiti. 2021. Table understanding approaches for extracting knowledge from heterogeneous tables. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11 (2021). https://doi.org/10.1002/widm.1407

[3] L. Chen, C. Huang, X. Zheng, J. Lin, and X. Huang. 2023. TableVLM: Multi-modal Pre-training for Table Structure Recognition. In *Proc. the 61st Annual Meeting of the Association for Computational Linguistics (ACL'23)*. 2437–2449. https://doi.org/10.18653/v1/2023.acl-long.137

[4] H. Dong, Z. Cheng, X. He, M. Zhou, A. Zhou, F. Zhou, A. Liu, S. Han, and D. Zhang. 2022. Table Pre-training: A Survey on Model Architectures, Pre-training Objectives, and Downstream Tasks. *Proc. the Thirty-First International Joint Conference on Artificial Intelligence* (2022), 5426–5435. https://doi.org/10.24963/ijcai.2022/761

[5] H. Dong and Z. Wang. 2024. Large Language Models for Tabular Data: Progresses and Future Directions. In *Proc. the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'24)*. 2997–3000. https://doi.org/10.1145/3626772.3661384

[6] D. W. Embley, M. S. Krishnamoorthy, G. Nagy, and S. Seth. 2016. Converting heterogeneous statistical tables on the web to searchable databases. *International Journal on Document Analysis and Recognition* 19, 2 (2016), 119–138. https://doi.org/10.1007/s10032-016-0259-1

[7] J. Fang, P. Mitra, Z. Tang, and C. L. Giles. 2012. Table header detection and classification. In *Proc. the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12)*. 599–605. https://doi.org/10.5555/2900728.2900814

[8] B. Fetahu, A. Anand, and M. Koutraki. 2019. TableNet: An Approach for Determining Fine-grained Relations for Wikipedia Tables. In *Proc. the World Wide Web Conference (WWW'19)*. 2736–2742. https://doi.org/10.1145/3308558.3313629

[9] Y. Huang, N. Lu, D. Chen, Y. Li, Z. Xie, S. Zhu, L. Gao, and W. Peng. 2023. Improving Table Structure Recognition with Visual-Alignment Sequential Coordinate Modeling. In *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11134–11143. https://doi.org/10.1109/CVPR52729.2023.01071

[10] A. Jain, S. Paliwal, M. Sharma, and L. Vig. 2021. TSR-DSAW: Table Structure Recognition via Deep Spatial Association of Words. In *Proc. the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2021)*. 257–262. https://doi.org/10.14428/esann/2021.es2021-109

[11] Keti Korini and Christian Bizer. 2023. Column Type Annotation using ChatGPT. In *Proceedings of the Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) — TaDA'23: Tabular Data Analysis Workshop*. 1–10.

[12] Keti Korini and Christian Bizer. 2024. Column Property Annotation Using Large Language Models. In *Proceedings of the Semantic Web: ESWC 2024 Satellite Events*. 61–70. https://doi.org/10.1007/978-3-031-78952-6_6

[13] Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle R. Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2024. Table-GPT: Table Fine-tuned GPT for Diverse Table Tasks. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–28. https://doi.org/10.1145/3654979

[14] X. H. Li, F. Yin, H. S. Dai, and C. L. Liu. 2022. Table Structure Recognition and Form Parsing by End-to-End Object Detection and Relation Parsing. *Pattern Recognition* 132, C (2022). https://doi.org/10.1016/j.patcog.2022.108946

[15] W. Lin, Z. Sun, C. Ma, M. Li, J. Wang, L. Sun, and Q. Huo. 2022. TSRFormer: Table Structure Recognition with Transformers. In *Proc. the 30th ACM International Conference on Multimedia (MM'22)*. 6473–6482. https://doi.org/10.1145/3503161.3548038

[16] Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. 2023. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics* 76 (2023), 100761. https://doi.org/10.1016/j.websem.2022.100761

[17] R. Rastan, H.-Y. Paik, and J. Shepherd. 2019. TEXUS: a unified framework for extracting and understanding tables in PDF documents. *Information Processing and Management: an International Journal* 56, 3 (2019), 895–918. https://doi.org/10.1016/j.ipm.2019.01.008

[18] J. C. Roldán, P. Jiménez, P. Szekely, and R. Corchuelo. 2021. TOMATE: A heuristic-based approach to extract data from HTML tables. *Information Sciences* 577 (2021), 49–68. https://doi.org/10.1016/j.ins.2021.04.087

[19] A. Shigarov. 2022. Table understanding: Problem overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 13 (2022). https://doi.org/10.1002/widm.1482

[20] A. Singha, J. Cambronero, S. Gulwani, V. Le, and C. Parnin. 2023. Tabular Representation, Noisy Operators, and Impacts on Table Structure Understanding Tasks in LLMs. In *Proc. Table Representation Learning Workshop at NeurIPS 2023*. 1–14. https://doi.org/10.48550/arXiv.2310.10358

[21] B. Smock, R. Pesala, and R. Abraham. 2022. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4634–4642. https://doi.org/10.1109/CVPR52688.2022.00459

[22] B. Smock, R. Pesala, and R. Abraham. 2023. Aligning Benchmark Datasets for Table Structure Recognition. In *Proc. 17th International Conference of Document Analysis and Recognition (ICDAR'2023)*. 371–386. https://doi.org/10.1007/978-3-031-41734-4_23

[23] Y. Sui, M. Zhou, M. Zhou, S. Han, and D. Zhang. 2024. Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study. In *Proc. the 17th ACM International Conference on Web Search and Data Mining (WSDM'24)*. 645–654. https://doi.org/10.1145/3616855.3635752

[24] X. Wu, H. Chen, C. Bu, S. Ji, Z. Zhang, and V. S. Sheng. 2023. HUSS: A heuristic method for understanding the semantic structure of spreadsheets. *Data Intelligence* 5, 3 (2023), 537–559. https://doi.org/10.1162/dint_a_00201

[25] W. Xue, B. Yu, W. Wang, D. Tao, and Q. Li. 2021. TGRNet: A Table Graph Reconstruction Network for Table Structure Recognition. In *Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 1295–1304. https://doi.org/10.1109/ICCV48922.2021.00133

[26] J. Yang, A. Gupta, S. Upadhyay, L. He, R. Goel, and S. Paul. 2022. TableFormer: Robust Transformer Modeling for Table-Text Encoding. In *Proc. the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22)*. 528–537. https://doi.org/10.18653/v1/2022.acl-long.40

[27] Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. TableLlama: Towards Open Large Generalist Models for Tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 6024–6044.

[28] X. Zheng, D. Burdick, L. Popa, X. Zhong, and N. X. R. Wang. 2021. Global Table Extractor (GTE): A Framework for Joint Table Identification and Cell Structure Recognition Using Visual Context. In *Proc. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 697–706. https://doi.org/10.1109/WACV48630.2021.00074