ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10448427

OPENTE: OPEN-STRUCTURE TABLE EXTRACTION FROM TEXT

Haoyu Dong, Mengkang Hu, Qinyu Xu, Haochen Wang, Yue Hu

Institute of Information Engineering, Chinese Academy of Sciences;
School of Cyber Security, University of Chinese Academy of Sciences;
The University of Hong Kong; Tsinghua University; Peking University

ABSTRACT

This paper presents an Open-Structure Table Extraction (OpenTE) task, which aims to extract a table with intrinsic semantic, calculational, and hierarchical structure from unstructured text. We devise a novel Identification-Extraction-Grounding (IEG) framework for language models (LMs) comprising three chaining steps: (1) identifying semantic and calculational relationships among columns, (2) extracting structured data from unstructured text, and (3) aligning extracted data with the source text and the table structure with a separate discrete grounding model. Experiment results suggest that OpenTE presents a significant challenge for state-of-the-art LMs and demonstrate that the IEG framework achieves superior performance on both datasets, with over 9% F1 improvements in the few-shot setting for GPT-3.5&4 and other large language models (LLMs) and over 4.9% F1 enhancements in the fine-tuning setting for open-source BART. We'll release the dataset to facilitate future research.

1. INTRODUCTION

Tables, commonly used for data presentation and management, are particularly valuable in open-domain information extraction due to their versatile and inherent structures for recording information. The versatility is manifested in two ways, as illustrated in Figure 1: (1) column names in tables can be flexibly specified to support open-ended entity types, allowing for inclusivity beyond a limited predefined entity set, and (2) semantic, calculational, and hierarchical relationships among columns display great adaptability, seamlessly aligning with the diverse demands of information extraction.

However, existing table extraction datasets, such as RotoWire, WikiTableText, E2E, and WikiBio, proposed by [1], exhibit clear limitations. RotoWire is single-domain, focusing exclusively on basketball teams and players, and is designed only for extracting simple relational tables; meanwhile, E2E, WikiTableText, and WikiBio primarily concentrate on extracting key-value pairs, neglecting tables with more generic structures. In this work, we propose Open-Structure Table Extraction (OpenTE) with the first benchmark dataset comprising fine-grained labels for various semantic, calculational, and hierarchical relationships among columns. Given a table's title and top header, the objective is to populate the table using information extracted from unstructured text, as shown in Figure

Example 1 from Wikipedia: Governors of the Territory of Oklahoma

#	Governor	Party	Took office year	Left Office year	Duration (year)
18	Henry Bellmon	Republican			
19	Dewey Follett Bartlett		1967	1971	4 (calc)
	Mary Fallin		2011	2019	8 (calc)

Dewey Follett Bartlett, served as the 19th Governor of Oklahoma from 1967 to 1971, following his same party Republican predecessor, Henry Bellmon.
Mary Fallin (born 1954) is the governor of Oklahoma from 2011 to 2019.

Example 2 from analysis report: Number with epilepsy and prevalence per 1,000, 2010 to 2012

Sex type	population (2011/2012)		Household population (2010/2011)		Total	
	Number '000	Prevalence per 1,000	Number '000	Prevalence per 1,000	Number '000	Prevalence per 1,000
Total	10.6	40.4	128.6		139.2 (calc)	
Male	5.2	57.1	59.6		64.8 (calc)	
Female	5.4	31.6	69.0		74.4 (calc)	

Based on data for the 2010-to-2012 period, about 10,600 Canadians in long-term care facilities and 128,600 Canadians in private households had epilepsy.
Among people in long-term care facilities, the overall prevalence of epilepsy was 40.4 per 1,000. The figure rose from 31.6 per 1,000 among women (5,400 female Canadians) to 57.1 per 1,000 among men (5,200 male Canadians).
In private households, about 69,000 female Canadians and 59,600 male Canadians had epilepsy.

Fig. 1. Examples of open-structure table extraction from text. We use "calc" to highlight cells that need calculation.

1. We collect text-to-table data from open-domain Wikipedia pages and statistical reports, leveraging existing table-to-text datasets [2, 3] which preserve the hierarchy information of tables. Human effort that devoted to addressing data quality issues result in a high-quality dataset. Importantly, fine-grained semantic and calculational column relationships are labeled in details to enable meticulous model training and experiment analysis.

LLMs have demonstrated remarkable performance in tasks involving textual understanding and tabular reasoning. Yet, when it comes to the unique challenges of table extraction, their capabilities remain uncharted. We are the first to employ state-of-the-art LLMs for the OpenTE task, utilizing carefully crafted prompts. Then we introduce a novel chaining Identification-Extraction-Grounding (IEG) framework tailored for LMs on OpenTE, which comprises three pivotal steps: (1) To cater to the challenge of diverse table structures, we design a dedicated module that identifies semantic and calculational relationships among columns. (2) We harness the advancement of state-of-the-art LMs to extract pertinent information from unstructured text. (3) To avoid common flaws made by LMs, such as incorrect row/column correspondences or excessive cells generated with hallucination, we introduce a separate grounding model. This model ensures the extracted

data aligns with both the source text and the table structure. Our framework has been validated on GPT 3.5&4, showcasing the superiority of LLMs over prior methods [1]. Remarkably, the IEG framework delivers over 9% F1 improvements in the few-shot setting for GPT-3.5&4, and more than 4.9% F1 enhancements in the fine-tuning setting for BART on both datasets.

2. PRELIMINARIES

Task Formulation This task aims to extract information from unstructured text and populate a target table, given the table's title and top header. The purpose of providing the title and header as inputs is to convey the semantic, calculational, and hierarchical table structure, catering to specific and diverse user demands regarding table organization. It also substantially reduces ambiguity during evaluation related to multiple eligible structures (e.g., flat or hierarchical, two columns of "Year" and "Month" or one column of "Date"), data formats (e.g., expressed in thousands or millions), etc.

Evaluation We utilize Exact Match (EM), BERTScore (BERT), and Chrf to evaluate cell-level precision, recall, and F1 for cells to be extracted in the table, following [1]. However, constructing triples by assuming the top row and left column distinctly index a cell [1] is only applicable to simple relational or matrix tables and not suitable for tables with other structures. Instead, we compute cell-by-cell EM, BERT, and Chrf scores for all pairs of the predicted row and the ground-truth row, calculate the average of cell scores for all pairs of rows, and employ the Hungarian algorithm [4] to determine the best row-match strategy in polynomial time. Then we use the best row-match strategy to evaluate precision, recall, and F1 for each table and compute the average over all tables.

Formulation of column relationship Fully understanding flexible column relationships is both a challenge and a prerequisite for table extraction. We formulate column relationships in two aspects. (1) Semantic column relationship: Following [5], we identify the parent of every column header to construct a tree-structured ontology for each table, as illustrated in Figure 2. (2) calculational relationship: We define a computed column as a column calculated from others, which can be represented using a formula, as demonstrated in Figure 2.

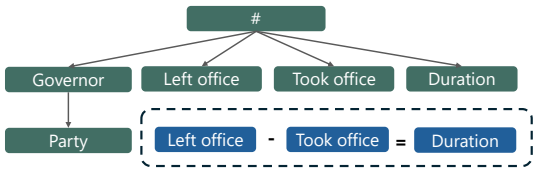

Fig. 2. Examples of a ontology tree and a formula to indicate column relationships.

Fine-grained labeled (All)	Wikipedia	Statistical reports
# Table	800	457
# Mentioned columns	4,480	1,830
# Mentioned rows	4,503	1,324
# Mentioned cells	18,315	2,260
# Sentences	5,621	1,553
Hierarchical header	35.2%	100.0%
Complex ontology tree	51.4%	0.0%
Computed cells	1.8%	7.5%

Table 1. Dataset statistics of OpenTE.

3. DATASET CONSTRUCTION

We construct the OpenTE dataset from two main sources: the ToTTo dataset [2] of Wikipedia articles and the HiTab dataset [3] of statistical reports. Both datasets consist of numerous tables with associated text for the task of text generation over tables, and highlighted (mentioned) cells are labeled to facilitate controllable generation. We filter tables with no fewer than four sentences and four mentioned cells. Then we sample 800 tables from Wikipedia and 457 tables from statistical reports. We reverse the dataset for text-to-table following [1]. However, there are data quality issues: (1) it includes many excessive and missing mentioned cells, and (2) numerous cell values are too precise to be extracted from mentions in text (e.g., "63,241" in cell but "63,000" in text). We employ five students from top universities for correcting quality issues. Annotators should be careful on deleting unmentioned cells in the table, adding missing cells to the table, and revising inconsistent cells in the table. We labeled 800 Wikipedia tables in ToTTo, 600 for training and 200 for testing. The labeled subset comprises a total of 5,621 sentences and 18,315 referred cells. The average table size is 5.48 mentioned columns by 5.63 mentioned rows. 35.2% of tables have hierarchical top headers, and 51.4% of tables have ontology trees with more than two layers. On the other hand, the HiTab dataset includes 457 tables, 305 for training and 152 for testing. The dataset encompasses 1,553 sentences in total. The computed cells in HiTab account for 7.6% of the dataset. By combining these two datasets, we create a diverse and rich dataset that covers various domains and table structures.

4. METHOD

To mitigate three major challenges, namely understanding complex column relationships, extracting information from unstructured text, and ensuring consistency between text and the extracted table, we introduce an Identification-Extraction-Grounding (IEG) pipeline to unlock the power of LMs.

4.1. Identification

Table titles and headers are typically concise. A vague understanding of the table structure may lead to ambiguities in the generation process, such as inserting a text snippet into an

incorrect cell or overlooking computed values. (1) This necessitates that LMs fully comprehend entities and their relationships based on succinct column names (e.g., "#" and "Governor"). By formulating an ontology tree to represent semantic and calculational column relationships, we employ parentheses during decoding to depict a tree structure for LMs. (2) Subsequently, LMs are instructed to construct formulas (using column names as operators) to represent underlying calculation and aggregation relationships among columns. The input for the identification module consists of the table's title and header. The instruction and output are as follows:

Instruction:
Given the table title:
Governors of the Territory of Oklahoma
Given column headers of the table:
| Governor | Party | Took office | Left office | Duration
1. Generate the ontology tree of columns;
2. Generate the calculational relationships among columns:

The expected generated answer:
1. # (Governor (Party), Took office, Left office, Duration)
2. Left office - Took office = Duration

4.2. Extraction

The extraction process presents challenges in extracting and inferring information from text (e.g., inferring "18" based on "Oklahoma followed Henry as the 19th governor"). This information must be placed in the correct position within a table and organized in a consistent format, such as number format (e.g., "10.6" given the header "number '000" indicating the unit) and string format (e.g., "1971" given the header keyword "year"). The input includes the table's title and header, the extracted ontology tree and formulas by the Identification module, and the text to be extracted. The output is the polulated table. The instruction and output are as follows:

Instruction:
Given the table title, header, ontology tree, and formula.
Generate a table with the header from sentences as follows:
Dewey Follett Bartlett, served as the 19th Governor of ...

The expected generated answer:
| Governor | Party | Took office year | ...
18 | Henry Bellmon | Republican | - | ...
...

4.3. Grounding

LMs often exhibit flaws such as generating cells not mentioned in the text (hallucination) and placing cells in incorrect positions. To address these issues, we propose a grounding step that requires LMs to reverse-engineer grounded sentences for each extracted cell, validating if each cell has been accurately extracted and positioned. Only those cells confirmed to be correct are preserved in the table, eliminating unmentioned and wrongly-positioned cells. The input for the grounding module includes the extracted table and all text sentences. The output consists of the descriptive sentences for each cell and if the cell is correctly extracted and positioned. The instruction and output is as follows:

Instruction:
Given the extracted table (with title, ontology tree, and formula) and source sentences.
Generate the grounded sentences for each extracted cell and validate if each cell is correctly extracted and put in a correct position in the following format:
Cell|Grounded sentences|If correctly extracted and positioned

The expected generated answer:
Dewey Follett Bartlett|Dewey Follett Bartlett, served as...|Yes
Henry Bellmon|...following his same party Republican predecessor, Henry Bellmon|Yes
...

Finally, we use a script with executors to explicitly validate the calculation based on the column relationships recorded by formulas, resolving incorrect and missing computed cells.

5. EXPERIMENTS

In this section, we examine the performance of BART-Large [6] and GPT 3.5&4 [7, 8]. We also evaluate a text-to-table baseline introduced by [1] by augmenting BART-Large with table constraint and table relation embeddings (denoted as BART_A).

5.1. Implementation details

Table headers are encoded using "|" to separate cells in a row, and multiple rows are used if hierarchical headers exist. Textual information is provided as a list of sentences. We fine-tune all parameters in BART-Large (denoted as BART-FT). We experiment with two settings: (1) Few-shot setting: GPT uses the same five-shot examples randomly sampled from the labeled training set in prompts. Few-shot examples are randomly sampled three times, and the average is used as the final result. (2) Fine-tuning setting: For Wikipedia tables, we use 600 fine-grained labeled training samples. Since the unlabeled 4,432 tables in Wikipedia may have data quality issues and lack fine-grained column relationship labeling, we do not use them in this paper, although they could provide noisy and partial/distant supervision. For tables from statistical reports, we use the 305 labeled tables for fine-tuning. Fine-tuning takes 10 epochs, and the IEG modules trained sequentially using the results of previous modules.

5.2. Experiment Result and Analysis

Table 2 presents the experiment results of LMs on table extraction in Wikipedia and statistical reports. GPT-4-E, one of

Cell-level F1	Wikipedia			Statistics Report		
%	EM	Chrf	BERT	EM	Chrf	BERT
BART_A-FT-E	41.2	44.1	47.9	43.7	44.7	45.6
BART-FT-E	38.4	42.0	47.2	41.1	42.5	44.8
BART-FT-EG	44.2	47.8	53.5	46.7	48.0	49.5
BART-FT-IEG	46.3	49.6	55.2	49.1	50.5	51.4
GPT-3.5-E	50.4	53.7	55.2	47.3	48.0	48.9
GPT-3.5-EG	57.2	59.8	62.1	55.4	56.2	57.0
GPT-3.5-IEG	60.2	63.3	65.8	56.9	58.0	58.7
GPT-4-E	58.5	61.3	63.1	54.8	55.5	56.2
GPT-4-EG	66.2	68.9	69.1	62.3	63.2	64.0
GPT-4-IEG	68.0	70.6	71.1	63.8	64.5	65.4

Table 2. Results of OpenTE evaluation on Wikipedia and statistical reports. Suffixes FT, I, E, and G mean fine-tuning and the Identification, Extraction, and Grounding modules.

EM	Hierarchy		Ontology tree		calculation	
Cell-level F1 %	Flat	Deep	Flat	Deep	No	Yes
BART_A-FT-E	42.4	39.0	45.0	37.6	46.5	8.9
BART-FT-E	39.6	36.2	42.5	34.5	43.8	7.1
BART-FT-EG	45.0	42.8	48.5	40.1	49.8	8.9
BART-FT-IEG	47.3	44.5	48.9	43.8	51.2	23.2
GPT-3.5-E	53.2	45.3	51.6	49.3	49.4	21.4
GPT-3.5-EG	59.5	53.0	58.7	55.7	57.9	25.0
GPT-3.5-IEG	62.0	56.8	60.8	59.6	58.3	39.2
GPT-4-E	60.8	54.2	59.9	57.2	57.4	23.2
GPT-4-EG	67.6	63.6	68.1	64.4	65.2	26.8
GPT-4-IEG	69.4	65.4	68.3	67.7	63.4	44.6

Table 3. Results of table extraction by hierarchy, semantic, and calculation types. Due to the dataset distribution, the results of hierarchy and ontology tree are based on Wikipedia tables and the results of calculation are based on statistical report tables.

the most powerful LLMs, significantly outperforms the previous state-of-the-art model (BART_A-FT-E) by a large margin (17.3% for Wikipedia and 11.1% for statistical reports in F1-EM). Moreover, the Identification and Grounding modules yield over 9% F1 gains in the few-shot setting for GPT-4 on both Wikipedia and statistical reports, as well as substantial F1-EM gains in the fine-tuning setting for BART (7.9% and 8.0%).

Table 3 displays experiment results by table structures and calculation types. It reveals that the Identification and Grounding modules produce larger accuracy gains for challenging cases containing hierarchical top headers, deep ontology trees with more than 2 layers, and cells computed by others. Especially for computed cells, the Identification and Grounding modules provide more than 12% F1 additional gains for all methods, even 21.4% for GPT-4.

5.3. Case Study

We analyze the failed cases of GPT-4-IEG and categorize the errors into the following types: (1) missing cells, especially for those requiring inference, such as "18" in the first case in Figure 1; (2) incorrect cells, including erroneous unit conversions, e.g., extracting "128.6" from "128,600" as shown in the second case in Figure 1; (3) incorrect positions, particularly for snippets in vague contexts or column names with ambiguity; (4) correct semantics but inconsistent formats. Future work might benefit from more meticulous metrics, considering that even though "2009-10" and "2009 to 2010" convey the same semantic meaning, BERTScore identifies them as distinct.

6. REFERENCES

Information Extraction (IE) focuses on extracting structured data from unstructured text. This consists of tasks such as named entity recognition, relation extraction, and event extraction. Typically, the schema for table extraction is predefined, often using straightforward structures like key-value pairs and relational tuples. Existing datasets for table extraction, such as RotoWire, WikiTableText, E2E, and WikiBio, have limitations as discussed by [1]. RotoWire is confined to a single domain, specifically basketball teams and players, and only supports the extraction of basic relational tables. In contrast, E2E, WikiTableText, and WikiBio focus predominantly on extracting key-value pairs, thereby overlooking more complex table structures. Recent works exploring to model generally structured tables and text together [9] are summarized in this survey [10]. Open Information Extraction (OpenIE) concentrates on extracting information from texts without the need for explicitly defined schema, although the output often remains in the form of simple relational tuples [11, 12]. [13] introduced hierarchical information extraction. The pioneering effort in deriving tables from textual content is the "text-to-table" method, as presented by [1, 14, 15]. However, their proposed datasets and evaluation metrics only address relational or simple matrix tables. OpenTE stands out as the first benchmark for open-structured table extraction, catering to tables that have versatile hierarchical, calculational, and semantic interrelations among columns. Additionally, OpenTE introduced the first LLM-based table extraction pipeline, demonstrating significant effectiveness in this task.

7. CONCLUSION

OpenTE targets open-structured table extraction from text, featuring a well-labeled benchmark. This dataset presents unique challenges in terms of semantics, hierarchy, and calculation. Moreover, it provides human labeling of semantic and calculation relationships among columns. A chaining extraction pipeline is proposed, which includes a pre-extraction module for interpreting column relationships and a post-extraction module to validate the extracted table cell-by-cell. This approach substantially improves the accuracy of BART and GPT3.5&4 by large margins in both few-shot and fine-tuning settings.

8. REFERENCES

[1] Xueqing Wu, Jiacheng Zhang, and Hang Li, "Text-to-table: A new way of information extraction," *arXiv preprint arXiv:2109.02707*, 2021.

[2] Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das, "Totto: A controlled table-to-text generation dataset," *arXiv preprint arXiv:2004.14373*, 2020.

[3] Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang, "Hitab: A hierarchical table dataset for question answering and natural language generation," *arXiv preprint arXiv:2108.06712*, 2021.

[4] Derek Bruff, "The assignment problem and the hungarian method," *Notes for Math*, vol. 20, no. 29-47, pp. 5, 2005.

[5] Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al., "Dart: Open-domain structured data record to text generation," *arXiv preprint arXiv:2007.02871*, 2020.

[6] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, Eds., 2020.

[8] OpenAI, "GPT-4 technical report," *CoRR*, vol. abs/2303.08774, 2023.

[9] Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang, "Tuta: Tree-based transformers for generally structured table pre-training," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1780–1790.

[10] Haoyu Dong, Zhoujun Cheng, Xinyi He, Mengyu Zhou, Anda Zhou, Fan Zhou, Ao Liu, Shi Han, and Dongmei Zhang, "Table pretraining: A survey on model architectures, pretraining objectives, and downstream tasks," *arXiv preprint arXiv:2201.09745*, 2022.

[11] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld, "Open information extraction from the web," *Communications of the ACM*, vol. 51, no. 12, pp. 68–74, 2008.

[12] Michele Banko and Oren Etzioni, "The tradeoffs between open and traditional relation extraction," in *Proceedings of ACL-08: HLT*, 2008, pp. 28–36.

[13] Kai Zhang, Yuan Yao, Ruobing Xie, Xu Han, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun, "Open hierarchical relation extraction," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5682–5693.

[14] Tong Li, Zhihao Wang, Liangying Shao, Xuling Zheng, Xiaoli Wang, and Jinsong Su, "A sequence-to-sequence&set model for text-to-table generation," *arXiv preprint arXiv:2306.00137*, 2023.

[15] Michał Pietruszka, Michał Turski, Lukasz Borchmann, Tomasz Dwojak, Gabriela Pałka, Karolina Szyndler, Dawid Jurkiewicz, and Lukasz Garncarek, "Stable: Table generation framework for encoder-decoder models," *arXiv preprint arXiv:2206.04045*, 2022.