UNIVERSITY OF CALIFORNIA SAN DIEGO

Bridging Language Models and Structured Knowledge: Extraction, Representation, and Reasoning

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Zilong Wang

Committee in charge:

Professor Jingbo Shang, Chair
Professor Taylor Berg-Kirkpatrick
Professor Zhiting Hu
Professor Julian McAuley

2025

The Dissertation of Zilong Wang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2025

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support, guidance, and encouragement of many individuals, to whom I am deeply grateful.

First and foremost, I extend my sincerest gratitude to my advisor, Professor Jingbo Shang, for his unwavering support, insightful guidance, and countless thought-provoking discussions. His mentorship has been instrumental in shaping not only this research but also my growth as a scholar. The collaborative and dynamic atmosphere in the lab provided a fertile ground for exploration, allowing us to delve into cutting-edge techniques with curiosity and boldness, making this journey both enriching and enjoyable.

I am also deeply thankful to my thesis committee members, Professor Taylor Berg-Kirkpatrick, Professor Julian McAuley, and Professor Zhiting Hu, for their invaluable support. Their willingness to accommodate my presentations and their thoughtful feedback have played a crucial role in refining and strengthening this dissertation.

I am grateful for the opportunities I had to intern at Adobe, Google, and Amazon, where I gained invaluable industry experience. My sincere appreciation goes to Vlad Morariu and Jiuxiang Gu at Adobe, Chen-Yu Lee, Zifeng Wang, Yichao Zhou, Wei Wei, and Sandeep Tata at Google, and Jingfeng Yang and Sheikh Sarwar at Amazon. Their mentorship and guidance provided me with deeper insights into real-world challenges and broadened my understanding of industry research.

My Ph.D. journey was greatly enriched by the incredible collaborations with Zihan Wang, Dheeraj Mekala, Letian Peng, Yuwei Zhang, Feng Yao, and all my other co-authors and labmates. The stimulating discussions, exchange of ideas, and shared passion for research made this journey more rewarding. I am also immensely grateful to my dear friends who have supported me and helped me adapt to life in America, making my days more colorful beyond the demands of research. Special thanks to Xiaoshuai Zhang, Minghua Liu, Xiyuan Zhang, Xinyue Wei, Yuheng Zhi, Zexue He, Zhankui He, Yunan Zhang for their unwavering friendship.

Finally, I owe my deepest gratitude to my family. Embarking on this Ph.D. journey

during an unprecedented global pandemic, coupled with strict international travel restrictions, was incredibly challenging. Their unconditional love and support carried me through the toughest moments. A special thanks to my lovely girlfriend, Li Zhong, who has been my best collaborator, my most trusted confidant, and my greatest source of joy. I cannot overstate my gratitude to her for always being by my side. This journey has been one of growth, learning, and resilience, and I am profoundly grateful to everyone who has been part of it.

Chapter 2 incorporates material from the publication "VRDU: A benchmark for visually-rich document understanding" by Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata, which was published in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. The author of this dissertation was the principal investigator and the lead author of this paper.

Chapter 3 incorporates material from the publication "Towards few-shot entity recognition in document images: A label-aware sequence-to-sequence framework" by Zilong Wang and Jingbo Shang, which was published in Findings of the Association for Computational Linguistics: ACL 2022. The author of this dissertation was the principal investigator and the lead author of this paper.

Chapter 4 incorporates material from the publication "Towards zero-shot relation extraction in web mining: A multimodal approach with relative xml path" by Zilong Wang and Jingbo Shang, which was published in The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. The author of this dissertation was the principal investigator and the lead author of this paper.

Chapter 5 incorporates material from the publication "Chain-of-table: Evolving tables in the reasoning chain for table understanding" by Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister, which was published in The Twelfth International Conference on Learning Representations, 2024. The author of this dissertation was the principal investigator and the lead author of this paper.

VITA

2016-2020	Bachelor of Science, Peking University
2023	M.S. in Computer Science, University of California San Diego
2025	Ph.D. in Computer Science, University of California San Diego

PUBLICATIONS

1. Zilong Wang, Zhaohong Wan, and Xiaojun Wan. Transmodality: An end2end fusion method with transformer for multimodal sentiment analysis. In *Proceedings of the web conference 2020*, pages 2514–2520, 2020

2. Jie Huang, Zilong Wang, Kevin Chang, Wen-Mei Hwu, and Jinjun Xiong. Exploring semantic capacity of terms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8509–8518, 2020

3. Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 898–908, 2020

4. Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. Layoutreader: Pre-training of text and layout for reading order detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4735–4744, 2021

5. Zilong Wang, Jiuxiang Gu, Chris Tensmeyer, Nikolaos Barmpalios, Ani Nenkova, Tong Sun, Jingbo Shang, and Vlad Morariu. Mgdoc: Pre-training with multi-granular hierarchy for document image understanding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3984–3993, 2022

6. Zihan Wang, Kewen Zhao, Zilong Wang, and Jingbo Shang. Formulating few-shot fine-tuning towards language model pre-training: A pilot study on named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3186–3199, 2022

7. Zilong Wang and Jingbo Shang. Towards few-shot entity recognition in document images: A label-aware sequence-to-sequence framework. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4174–4186, 2022

8. Zilong Wang and Jingbo Shang. Towards zero-shot relation extraction in web mining: A multimodal approach with relative xml path. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023

9. Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. Vrdu: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5184–5193, 2023

10. Alex Nguyen, Zilong Wang, Jingbo Shang, and Dheeraj Mekala. Docmaster: A unified platform for annotation, training, & inference in document question-answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 128–136, 2024

11. Li Zhong, Zilong Wang, and Jingbo Shang. Debug like a human: A large language model debugger via verifying runtime execution step by step. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024

12. Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations*, 2024

13. Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, Chen-Yu Lee, and Nan Hua. LMDX: Language model-based document information extraction and localization. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024

14. Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang. Answer is all you need: Instruction-following text embedding via answering the question. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024

15. Prashant Krishnan, Zilong Wang, Yangkun Wang, and Jingbo Shang. Towards few-shot entity recognition in document images: A graph neural network approach robust to image manipulation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16514–16526, 2024

16. Li Zhong and Zilong Wang. Can llm replace stack overflow? a study on robustness and reliability of large language model code generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21841–21849, 2024

17. Si-An Chen, Lesly Miculicich, Julian Martin Eisenschlos, Zifeng Wang, Zilong Wang, Yanfei Chen, Yasuhisa Fujii, Hsuan-Tien Lin, Chen-Yu Lee, and Tomas Pfister. Tablerag: Million-token table understanding with language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024

ABSTRACT OF THE DISSERTATION

Bridging Language Models and Structured Knowledge:
Extraction, Representation, and Reasoning

by

Zilong Wang

Doctor of Philosophy in Computer Science

University of California San Diego, 2025

Professor Jingbo Shang, Chair

Structured knowledge—diversely embedded in document images, web pages, and tabular data—presents distinct challenges for language models. Unlike free-form text, structured data encodes meaning through spatial arrangements, hierarchical structures, and relational dependencies, requiring models to extract, interpret, and reason beyond linguistic signals. This dissertation advances the integration of structured knowledge with language models, introducing novel methodologies for document understanding, web mining, and table-based reasoning.

We first introduce VRDU, a benchmark for Visually-Rich Document Understanding, designed to evaluate how models extract structured information from business documents with

complex layouts and hierarchical entities. By identifying key challenges in template generalization and few-shot adaptation, VRDU provides a more realistic assessment of multimodal language models.

Next, we present LASER, a label-aware sequence-to-sequence framework for few-shot entity recognition in document images. By embedding label semantics and spatial relationships directly into the decoding process, LASER enables models to recognize entities with minimal supervision, outperforming traditional sequence-labeling approaches in low-resource scenarios.

For web mining, we propose ReXMiner, a zero-shot relation extraction framework that captures structural dependencies within semi-structured web pages. By encoding relative XML paths in the Document Object Model (DOM) tree, ReXMiner improves the generalization of relation extraction across diverse and unseen web templates, demonstrating that structural signals enhance information retrieval from the web.

Finally, we introduce CHAIN-OF-TABLE, a framework for table-based reasoning that evolves tabular data iteratively. Unlike previous approaches that treat tables as static inputs, CHAIN-OF-TABLE dynamically applies structured transformations, enabling models to reason step-by-step over tabular data. This approach achieves state-of-the-art performance across multiple benchmarks in table-based question answering and fact verification.

Together, these contributions redefine how language models interact with structured knowledge, bridging the gap between unstructured text processing and structured data reasoning. By integrating multimodal signals, relational structures, and iterative reasoning mechanisms, this dissertation lays the foundation for more robust and generalizable models in structural knowledge understanding.

Chapter 1

Introduction

1.1 Background

Language models (LMs) have made significant strides in natural language processing (NLP), demonstrating impressive capabilities in text generation, comprehension, and reasoning [18, 8, 50, 105, 99, 86, 61]. However, much of this progress has been centered around free-form text, where meaning is derived from sequential linguistic patterns. In contrast, **structured knowledge**, embedded in **semi-structured data** such as document images, web pages, and tabular data, presents distinct challenges [121, 120, 112, 43, 122, 37, 107]. Unlike unstructured text, structured knowledge relies on **spatial layouts, hierarchical relationships, and relational structures**, requiring models to process information beyond simple text sequences [76, 114, 128, 106, 11, 87].

Understanding and reasoning over structured knowledge is critical for many real-world applications [9, 103, 40, 127, 39, 111]. In domains such as finance, healthcare, business intelligence, program development, and scientific research, a significant portion of valuable information is stored in structured formats rather than in free-flowing text. Extracting meaningful insights from these data sources requires not only language understanding but also the ability to interpret **structural dependencies, layout information, and relational constraints**. Despite advances in multimodal and semi-structured data processing, language models often struggle to generalize effectively across diverse structured formats [113, 109, 115].

1.2 Motivation

Many real-world applications require accurate and efficient extraction, reasoning, and representation of structured data. For instance, organizations rely on automated systems to process invoices, contracts, and regulatory documents; search engines and information retrieval systems need to extract relationships from web pages; and financial analysts depend on structured tables for decision-making. However, existing language models, designed primarily for sequential text, face limitations when applied to structured knowledge [113, 109, 115, 87, 114].

The need for **robust structured data processing** arises from the following key challenges:

- **Template Variability**: Structured documents and web pages follow diverse templates and formats, making it difficult for models to generalize across unseen structures.
- **Hierarchical and Relational Dependencies**: Extracting meaningful insights requires understanding nested and hierarchical entities, as well as relationships between different components of structured data.
- **Few-Shot and Zero-Shot Adaptation**: Real-world applications often lack abundant labeled data, necessitating methods that can generalize effectively with minimal supervision.
- **Iterative Reasoning**: Many structured knowledge tasks require more than a one-time effort; they demand step-by-step reasoning, particularly when working with tabular data, where intermediate results play a crucial role in complex analysis.

To address these challenges, this dissertation investigates the integration of structured knowledge with language models through novel methodologies that enhance **extraction, representation, and reasoning**.

1.3 Challenges in Bridging Language Models and Structured Knowledge

Despite the recent surge in multimodal and structured data research, several fundamental gaps remain:

1. **Limitations of Existing Benchmarks:** Many current datasets for document and table understanding fail to capture the complexities of real-world structured data.
2. **Inadequate Representation of Structure:** Traditional LMs primarily encode textual features, overlooking spatial, hierarchical, and relational aspects crucial for structured knowledge extraction.
3. **Scalability and Generalization:** Language models trained on specific structured formats often struggle to generalize to unseen documents, web structures, or table layouts.
4. **Lack of Iterative Processing Capabilities:** Most models treat structured data as static inputs, lacking the ability to iteratively refine representations through reasoning.

Addressing these gaps requires a paradigm shift in how language models process structured knowledge, moving beyond sequence-based processing to models that incorporate multimodal, hierarchical, and relational reasoning.

1.4 Contributions

This dissertation presents four key contributions to advancing the integration of structured knowledge with language models:

- **Visually-Rich Document Understanding (VRDU) [115]**: A benchmark designed to evaluate language models on extracting structured information from complex document layouts. VRDU introduces challenges related to **template generalization, hierarchical**

entity extraction, and few-shot learning, providing a realistic evaluation framework for multimodal document understanding.

- **Label-Aware Sequence-to-Sequence Framework (LASER) [108]**: A novel few-shot entity recognition model for document images that embeds **label semantics and spatial relationships** into a generative labeling scheme. Unlike traditional sequence labeling approaches, LASER enables more efficient generalization with minimal supervision, improving entity recognition under low-resource conditions.

- **ReXMiner: Zero-Shot Relation Extraction in Web Mining [109]**: A multimodal approach for structured information extraction from web pages, leveraging **relative XML paths in the Document Object Model (DOM) tree**. By encoding structural relationships, ReXMiner improves relation extraction and key-value pair detection in semi-structured web data, enhancing the adaptability of language models to unseen templates.

- **CHAIN-OF-TABLE: Iterative Table-Based Reasoning [114]**: A framework that enables **dynamic table evolution during reasoning**, addressing the limitations of treating tables as static inputs. CHAIN-OF-TABLE applies **structured transformations step-by-step**, allowing models to iteratively refine tabular representations and improve accuracy in **table-based question answering and fact verification**.

Together, these contributions advance the integration of **structured knowledge into language models**, improving their ability to **extract, represent, and reason** over semi-structured data.

This dissertation aims to bridge the gap between **language models and structured knowledge**, laying the foundation for more robust and generalizable models in **document understanding, web mining, and table reasoning**.

Chapter 2

VRDU: A Benchmark for Visually-rich Document Understanding

2.1 Introduction

Visually-rich documents, such as forms, receipts, invoices, are ubiquitous in various business workflows. Distinct from plain text documents, visually-rich documents have layout information that is critical to the understanding of documents. Given the potential to automate business workflows across procurement, banking, insurance, retail lending, healthcare, etc., understanding these documents, and in particular extracting structured objects from them has recently received a lot of attention from both industry and academia [56, 126, 89, 3, 25, 5].

While tasks such as classification [33] and Visual-QA [71] have been posed to study the understanding of such documents, in this paper, we focus on the task of extracting structured information. Optical character recognition engines (OCR) are typically used to extract the textual content and the bounding boxes of each of the words from the documents. Existing models rely on language models with multi-modal features to solve the task, where features from textual contents, images, and structural templates are jointly encoded through self-supervised training [120, 121, 3, 25, 52, 89]. Although recent models achieved impressive results [29, 81, 44, 95], we argue that existing benchmarks do not reflect the challenges encountered in practice, such as having to generalize to unseen templates, complex target schema, hierarchical entities, and small training sets.

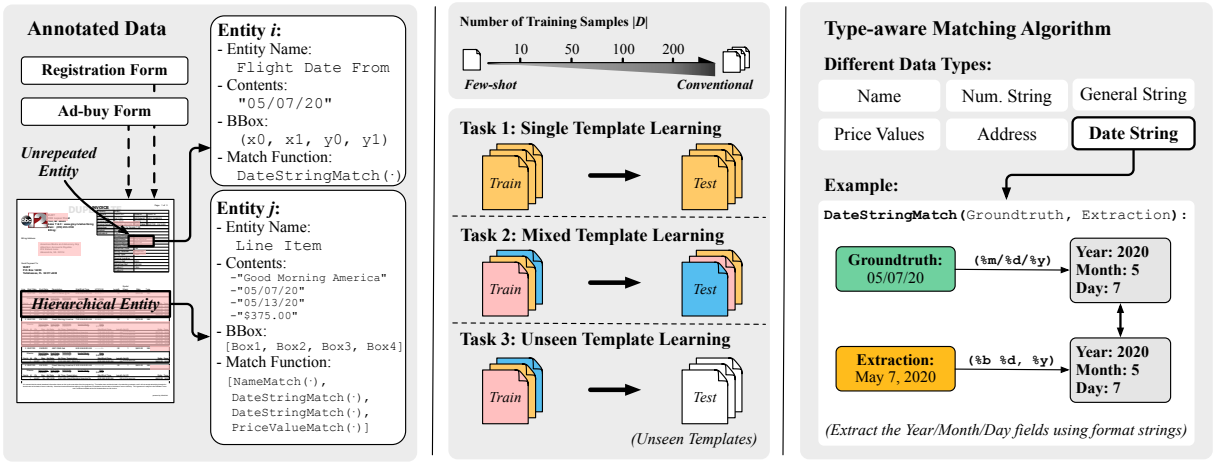


Figure 2.1. Overview of the VRDU benchmark: (a) high-quality annotation of rich labeling schema; (b) tasks of different difficulty levels and different number of training samples; (c) type-aware matching algorithm for entities of different data types.

We identify five desiderata (Section 2.3) for benchmarks on this topic based on our observations of drawbacks of existing datasets. First, most existing benchmarks suffer from the fact that they lack richness in labeling schema [29, 44, 95]. Entities are roughly considered as simple text strings while practical document types have a variety of types like numerical IDs, dates, addresses, currency amounts, etc. Further, real-world docs frequently have hierarchical and repeated fields like componentized addresses and line-items in invoices. Second, some benchmarks contain documents with limited layout complexity. Pages that are mostly organized in long paragraphs and sentences are more similar to plain text documents [95] and are not helpful evaluating our understanding of visually-rich documents. Third, the documents in some benchmarks may share the same template [44]. This makes it trivial for the models to deal with these document by simply memorizing the structure even if the single template is complex. Next, existing datasets use different OCR engines [81, 44]. The large variety of OCR engines make it hard to tell whether the improvements come from the advanced models or more accurate OCR results. Finally, some benchmarks only provide the textual contents for each entity without further annotating the specific tokens in the document that are involved in the entities [44, 95, 97], which means the models cannot be supervised with the token-level annotation. While this seems minor,

it is very difficult to re-construct the token-level annotation only with textual contents of entities since the same text (e.g. "0.0") may appear multiple times in the document but only one of them may correspond to the target entity. It is necessary to involve human annotators to fix the issue by relabeling the documents with precise token spans. Also note that most existing approaches on this topic are based on sequence labeling models [121, 120, 43, 126, 3, 25, 43, 102] that require token-level annotations to work.

Based on these observations, we propose a new benchmark, **VRDU**, for **V**isually-**R**ich **D**ocument **U**nderstanding task. VRDU is designed to reflect the challenges encountered in practice and eliminate the unnecessary factors affecting the research. We hope that this benchmark helps bridge the gap between academic research and practical scenarios to facilitate future study on this topic. As shown in Figure 2.1, we collected political ad-buy forms from the Federal Communications Commission (FCC)[1] and registration forms from the Foreign Agents Registration Act (FARA)[2], and constructed two datasets. We describe the annotated data, and the labeling protocol in Section 2.4.

Based on the two datasets, we then design three tasks of increasing difficulty. The tasks are designed to be similar to real applications. In Task 1 Single Template Learning, documents in the train and test sets are drawn from a single template. In Task 2 Mixed Template Learning, we increase the diversity of templates, but train and test sets for each document type are drawn from the same set of templates. In Task 3 Unseen Template Learning, the train and test sets are drawn from disjoint sets of templates to measure how well a model generalizes to unseen templates. Within each task, we compare the model performance with different number of training samples to understand the data efficiency for each approach. Finally, we evaluate the model performance with a type-aware match algorithm, where we use different matching functions for each entity according to its data type instead of simply using string matching when comparing the prediction results with the groundtruth. For example, when comparing numerical entities, we may want "4"

[1] https://publicfiles.fcc.gov
[2] https://www.justice.gov

and “4.0” to be considered equivalent, while for address fields, “4, Main St.” and “40 Main St.” ought not to be considered equivalent.

We report the performance of commonly-used baseline models, LayoutLM [121], LayoutLMv2 [120], LayoutLMv3 [43], and FormNet [52] in each task. Our work is *not* meant to be a comparison of these model architectures. Through our experiments, we highlight three areas of opportunity for all these models. First, while the models are great at extracting from new instances of documents with a layout that matches one seen during training (Task 1 Single Template Learning and Task 2 Mixed Template Learning), they do worse on new layouts (Task 3 Unseen Template Learning). Second, few-shot performance continues to be hard with substantial room for improvement. Third, extracting hierarchical or repeated entities is really challenging, and all models perform worse on this compared to simple fields.

We summarize our contribution as follows.

- We identify desiderata for benchmarks in the visually-rich document understanding task, arguing that the current datasets do not meet these requirements.
- We propose VRDU, a new comprehensive benchmark for visually-rich document understanding. We open-source the dataset with high-quality OCR results and annotations. We also define three tasks corresponding to different application scenarios, and open-source an evaluation toolkit with a type-aware matching algorithm. The toolkit and dataset can be found at https://github.com/google-research/google-research/tree/master/vrdu.
- VRDU satisfies all of our proposed desiderata and reflects practical challenges in extracting structured data from visually rich documents. It bridges the gap between academic research and practical scenarios to facilitate future study on this topic.
- Through experiments on multiple commonly-used baseline models, we show that there is substantial room for progress on the tasks in VRDU with regard to template transfer learning, few-shot settings, and hierarchical entity extractions.

Table 2.1. The statistics of VRDU and other existing benchmarks. * denotes the number of hierarchical entities in the dataset, where VRDU-Ad-buy Form involves 1 hierarchical entity and the hierarchical entity has 5 entities as components.

Dataset	Source	Doc #	Entity #	Desiderata				
				Rich Schema	Layout-rich Documents	Diverse Templates	High-quality OCR	Token-level Annotation
FUNSD	Lawsuits Forms	199	3	✗	✓	✓	✓	✓
CORD	Grocery Receipts	1000	30	✓	✓	✗	✗	✓
SROIE	Grocery Receipts	973	4	✗	✓	✗	✗	✗
Kleister-NDA	NDA Forms	540	4	✗	✗	✗	✓	✗
Kleister-Charity	Financial Reports	2778	8	✗	✓	✓	✓	✗
DeepForm	FCC	1100	5	✗	✓	✓	✓	✗
VRDU-Registration Form	**FARA**	**1915**	**6**	✓	✓	✓	✓	✓
VRDU-Ad-buy Form	**FCC**	**641**	**9+1(5)***	✓	✓	✓	✓	✓

2.2 Related Work

Several benchmarks are available to evaluate the performance of models in visually-rich document understanding. The properties of these benchmarks and the comparison with our proposed benchmark are shown in Table 2.1.

FUNSD [29] is a dataset widely used in the form understanding task, which contains 199 fully annotated forms with three different entity types, *Header*, *Question*, and *Answer*. This simple schema is too limited to reflect the rich schemas we encounter in practical scenarios. CORD [81] is a receipt dataset where the document images are photos of grocery receipts. While it does have a rich schema with different types including hierarchical and repeated fields, there is fairly limited template diversity. Furthermore, image artifacts (tilt, lighting, distortion) result in OCR errors. In our work, the focus is *not* on challenging OCR scenarios, but rather on benchmarks that help us understand how well models are able to extract information after high-quality OCR. SROIE [44] is another receipt dataset. A few key fields are labeled, such as *Company Name*, *Address*, and *Total Price* – a fairly simple target schema. Further, the receipts in the dataset use the same template, failing to satisfy the requirement for diverse templates. Kleister-NDA [95] collects non-disclosure agreements and labels important fields but the documents are full of plain text paragraphs and chapters and show few layout elements.

Kleister-Charity [95] and DeepForm [97, 6] collect charity financial reports and political ad-buy documents respectively. Compared with the datasets above, DeepForm and Kleister-Charity involve layout-rich documents of various templates. However, both of them fail to provide token-level annotation. Further, both datasets have a target schema with multiple types, but lacking hierarchical and repeated fields. As we describe in Section 2.3.5, token-level annotations are critical to properly training and evaluating sequence labeling models. Upon investigation, we found that the source documents for DeepForm do contain many more fields including hierarchical and repeated fields. We based one of the two dataset in VRDU, the ad-buy forms on the same source and designed the labeling task to include bounding boxes and token-level annotations.

This paper proposes VRDU, composed of two datasets of Registration Form and Ad-buy Form, both of which have rich schema, layout-rich documents, diverse templates, high-quality OCR outputs, and token-level annotations. The Ad-buy Forms provide hierarchical entity annotations, introducing a practical structural extraction task that has not been explored in any of the existing benchmarks.

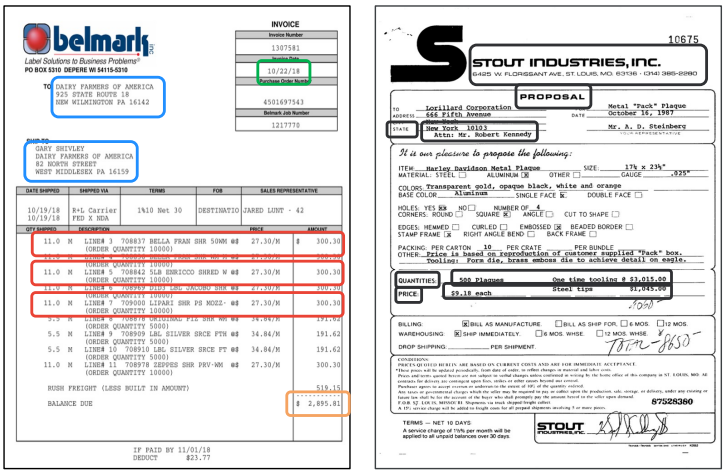
2.3 Benchmark Desiderata

We identify five key desiderata for a benchmark that reflects practical challenges in extracting structured data from visually rich documents. A benchmark on the visually-rich document understanding topic should involve rich schema, layout-rich documents, diverse template, high-quality OCR results, and token-level annotation.

2.3.1 Rich Schema

The structured data we need to extract from in practice reflect a rich diversity of schemas. Entities extracted have various types such as numerical IDs, names, addresses, dates, currency amounts, etc. They can be required, optional, or repeated for a given document. In several cases, we also see hierarchical entities. For example, a US address field contains address lines, city,

state, and zip code. A hierarchical entity is composed of all these components. Considering the heterogeneity of schema we encounter in practical settings, we believe a useful benchmark should reflect a rich schema. Contrast this with a dataset (see Figure 2.2) where the entities to be extracted are all treated as simple text strings named *header*, *question*, and *answer*.

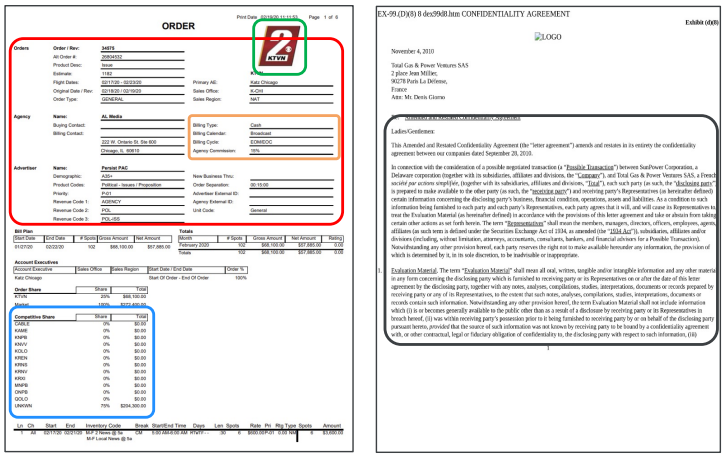


(a) (b)

Figure 2.2. Examples of labeling schema: (a) specifies data types for each text fields, such as *date strings*, *address lines*, *price values*, and *hierarchical entities* (denoted with , , , , respectively); (b) treats all text fields as *simple text strings* ignoring the specific data type (denoted with).

2.3.2 Layout-rich Documents

The documents should have complex layout elements. Challenges in practical settings come from the fact that documents may contain tables, key-value pairs, switch between single-column and double-column layout, have varying font-sizes for different sections, include pictures with captions, and even footnotes. Contrast this with datasets where most documents are organized in sentences, paragraphs, and chapters with section headers. Figure 2.3 shows an example of a document with rich layout and contrasts it with a more traditional document that is the focus of classic NLP literature on long inputs.



(a) (b)

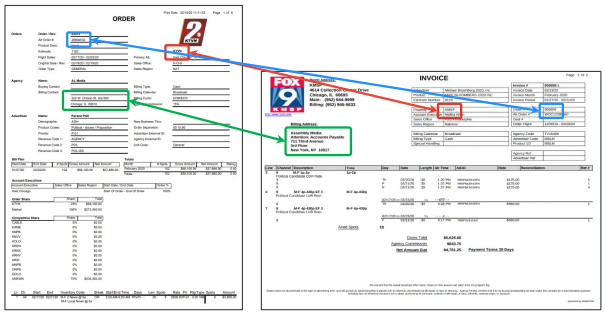
Figure 2.3. Examples of layout elements in the documents: (a) involves rich layout elements, such as *images*, *tables*, *key-value pairs*, and *multi-columns* (denoted with , , , , respectively); (b) largely contains natural language like *paragraphs*, *sentences*, *chapters* (denoted with).

2.3.3 Diverse Templates

A benchmark collection should involve different structural layouts or templates as shown in Figure 2.4. It is trivial to extract from a particular template by memorizing the structure. However, in practice one needs to be able to generalize to new templates. Consider, for instance, an invoice parser. If a company starts working with a new vendor (and enterprises routinely work with new vendors every year), a model that memorized the set of templates corresponding to existing vendors is likely to break since the new vendor may send invoices with a different template. In order to reflect this real-world requirement, a useful benchmark for extraction from visually-rich documents should have diverse templates and test a model's ability to generalize to unseen templates.

2.3.4 High-quality OCR Results

Documents should have high-quality OCR results. Our aim with this benchmark is to focus on the VRDU task itself and we want to exclude the variability brought on by the choice of OCR engine. Existing benchmarks use different OCR engines, which makes the evaluation results



(a)

(b)

Figure 2.4. Examples of document templates: (a) two examples of the same document type with different templates (entities denoted with , , for *address*, *contract ID*, and *TV station name*, respectively); (b) example of different documents that share the same template.

inconsistent and the comparison unfair. It is confusing whether the performance improvements come from the more advanced model design or are simply because of more accurate OCR results. Therefore, a benchmark should use the same high-quality engine ensuring the quality of OCR is satisfactory and the choice of OCR engine is not a factor influencing the results when comparing the performance.

2.3.5 Token-level Annotation

A good benchmark ought to provide the token spans in the document that correspond to each entity in the target schema rather than simply provide text strings and leave the task of mapping the values to the corresponding token ranges open. Existing approaches solve the extraction task using sequence labeling models and tend to build their models through extending BERT-like language models with multi-modal features [121, 120, 126, 3, 25, 43, 102]. They use the hidden states from the language models to classify tokens into the BIO tags [70, 93], i.e.,

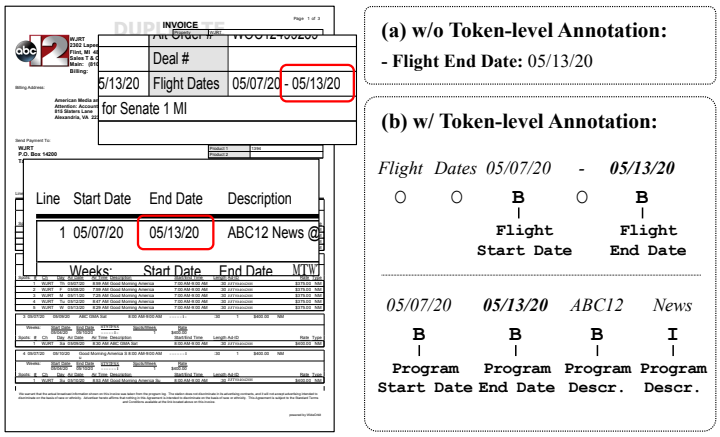


Figure 2.5. Examples of token-level annotation in visually-rich documents: (a) the dataset without token-level annotation where only the textual contents of entities are provided, and it is non-trivial to tell which "05/13/20" in the page is the value of *flight end date*; (b) the dataset with token-level annotation where all tokens are labeled with BIO tags.

Begin, Inside, Outside of an entity, and then extract entities accordingly. Thus token spans are required to construct training and evaluation sets. It is non-trivial to re-construct the token-level annotation only with the entity text. The possible ways are either labor-intensive or prone to errors. A intuitive approach is to find the phrases in the documents with the same textual contents with the entities, but these phrases are not necessarily to be the actual entity, as shown in Figure 2.5. [98] points out simply doing such value matching may result in worse F-1 scores in the performance. For instance, a dataset annotates the total amount field in a grocery store receipt as "10", but "10" may also appear in the receipt as the number of purchased items. Human annotators are needed to annotate the documents again to create the accurate token-level annotation. Therefore, token-level annotation is necessary to properly train and evaluate current baseline models and future works.

2.4 VRDU Benchmark

Based on the desiderata outlined in Section 2.3, we introduce VRDU, a new public benchmark for visually-rich document understanding. This benchmark includes two datasets: Ad-buy Forms and Registration Forms. These documents contain structured data with rich

schema including hierarchical repeated fields, have complex layouts that clearly distinguish them from long text documents, have a mix of different templates, and have high-quality OCR results. We provide token-level annotations for the ground truth ensuring there is no ambiguity when mapping the annotations to the input text. In the remainder of this section, we describe: (1) the process used for collecting and annotating the datasets, (2) the three extraction tasks we designed along with the prescribed train/validation/test splits, and (3) the design and implementation of the type-aware matching algorithm used to compare the extracted entities with the ground-truth

2.4.1 Data Collection

Visually-rich documents are common in various business workflows. However, there are still a large proportion of documents that fail to meet our proposed desiderata. To make things worse, documents with sensitive information can only be used as in-house datasets due to privacy issues, so they are unsuitable for public academic research. To find visually-rich documents that satisfy our desiderata and are available to the public, we crawl political ad-buy forms from the same resource as the DeepForm dataset, the Federal Communications Commission, and construct a new dataset, the Ad-buy Forms. DeepForm includes documents of high quality but fails to provide token-level annotation with rich schema so we collect the documents from the same source and annotate them from scratch. We also crawl documents from the Foreign Agents Registration Act and construct a separate dataset, the Registration Form. We use the state-of-the-art commercial OCR engines to recognize the raw data in the documents[3].

Ad-buy Forms

The Ad-buy Forms consist of 641 documents about political advertisements. Each document is an invoice or receipt signed between a TV station and a campaign group. The documents use tables, multi-columns, and key-value pairs to record the advertisement information, such as the product name, the flight dates, and the total price. They also have a large table showing more

[3]https://cloud.google.com/vision/docs/ocr

Table 2.2. The labeling schema of VRDU.

Registration Form	
Unrepeated Entity	*file_date, foreign_principle_name, registrant_name, registration_ID, signer_name, signer_title,*

Ad-buy Form	
Unrepeated Entity	*advertiser, agency, contract_ID, property, gross_amount product, TV_address, flight_from_date, flight_to_date*
Repeated Entity	*description, start_date, end_date, sub_price*
Hierarchical Entity	*line_item* (composed of *description, sub_price, start_date, end_date*)

details of the advertisements including the specific release date and time.

Registration Forms

The Registration Forms consist of 1915 documents about foreign agents registering with the US government. Each document records essential information about foreign agents involved in activities that require public disclosure. Contents include the name of the registrant, the address of related bureaus, the purpose of activities, and other details. We include three forms in the dataset, so the documents have three different templates, *Amendment*, *Short Form*, and *Dissemination Report*. All these forms are on the same topic so we label them using the same schema.

2.4.2 Human Annotation

After we collect visually-rich documents for the two datasets, we hire human annotators to annotate entities in the documents using a rich labeling schema. We describe the labeling schema, the labeling team, and the label protocol as follows.

Labeling Schema

The documents in our proposed benchmark present structured data with fairly rich schema, where entities can be repeated, unrepeated, or hierarchical, and the data types can be numerical strings, price values, etc. After examining a subset of the documents, we decide the target schema with 6 unrepeated entity names for Registration Forms, and 9 unrepeated entity names and 1 hierarchical repeated entity name for Ad-buy Forms. The entity names and their numbers are shown in Table 2.2 and Table 2.3.

- The unrepeated entities are the entities that only have one unique value in each document. Sometimes they may be present multiple times on a document, but with each instance having the exact same value. For example, a document may have several fields showing the contract ID but all these fields have the same content.

- The repeated entities are the entities that belong to the same type but have different values. For example, the names of purchased items are common repeated entities in grocery receipts. People may buy several items so there will be multiple values for the entity type, *purchased_item_name*.

- The hierarchical entities are the entities containing several repeated entities as components. For example, in Ad-buy Form, we design the *line_item* as a hierarchical entity, which corresponds to each TV program. Each *line_item* contains *description*, *start/end_date*, and *sub_price* of TV programs and all of these are repeated entities. In practice, we group the repeated entities that belong to a specific TV program as a *line_item*.

Labeling Team

We hired a labeling team of 30 annotators and 3 experts. All annotators and experts are experienced in labeling English documents and all of our data are in English. In our labeling task, the documents were first labeled by the annotators and then checked by the experts to guarantee the labeling quality. We acquired stats from our team of annotators on how long the classic

annotation takes for various document types. We found it averaged 6-8 min for an annotator to label a single-page document with fewer than 20 fields while it averaged 10-30 min for an annotator to label a multi-page document with 25 fields. So we picked a conservative value (6 min) as the estimated time of labeling one document in this paper.

Table 2.3. The statistics of entity numbers in VRDU. The *italic* entity names are hierarchical entities, which includes several repeat entities as components.

Registration Form			
Entity	**Number**	**Entity**	**Number**
Registration ID	1903	Foreign Principal	1132
Registrant Name	1902	Signer Name	1467
File Date	1873	Signer Title	549

Ad-buy Form			
Entity	**Number**	**Entity**	**Number**
Property	595	Flight From Date	540
TV Address	535	Flight To Date	538
Advertiser	635	Gross Amount	629
Product	607	Agency	283
Contract ID	624	*Line Item*	*9163*

Labeling Protocol

During the annotation, a pool of experienced annotators were provided with the previously annotated documents as reference and the labeling instruction as guidance. They drew bounding boxes to highlight the entities and labeled each entity into different categories. The system would collect the OCR results of the token span in the bounding box to construct token-level annotation, including the coordinates of the bounding box, the textual contents of entity, and the index in the sequence. If unrepeated entities occurred multiple times, they were instructed to identify all instances and the model only needs to extract one of them in the evaluation. When labeling the hierarchical entities, the annotators labeled the component entities as well as drew a larger bounding box that grouped the components together into a hierarchical entity. The system would

use the entities in the larger box to compose hierarchical entities in our dataset. After the first pass of annotation, a pool of experts were assigned to review the results labeled by the first pool. We took the final corrected results from the expert pool and used them in our experiments. This is the dataset we published.

Common Labeling Errors

To better understand the labeling protocol, we further study the annotators' common error types.

- Confusion of similar entities: In Ad-buy Form, the annotators are sometimes confused between the start/end dates of the flight and other time periods in the documents (e.g. the invoice period).
- Incomplete multi-line entities: In the Ad-buy Form dataset, the annotators sometimes ignore the last line of the address field since the address field usually contains multiple lines.

To cope with these errors, we give annotators previous annotated documents as reference and ask another expert annotator to double check all the annotation results. We believe our labeling protocol can well prevent the annotation mistakes and produce a high-quality benchmark.

2.4.3 Task Settings

We design three tasks with increasing difficulty:

Task 1: Single Template Learning (STL)

This is the simplest scenario where the training, testing, and validation sets only contain a single template. This simple task is designed to evaluate a model's ability to deal with a fixed template. Naturally, we expect very high F1 scores for this task.

Task 2: Mixed Template Learning (MTL)

This task is similar to the task that most related papers use: the training, testing, and validation sets all contain documents belonging to the same set of templates. We randomly sample documents from the datasets and construct the splits to make sure the distribution of the each template is not changed during the sampling.

Task 3: Unseen Template Learning (UTL)

This is the hardest setting, where we evaluate if the model is able to generalize to unseen templates. For example, in the Registration Forms dataset, we train the model with two of the three templates and test the model with the remaining one. The documents in the training, testing, and validation sets are drawn from disjoint sets of templates. To our knowledge, previous benchmarks and datasets do not explicitly provide such a task designed to evaluate the model's ability to generalize to templates not seen during training.

Dataset Splits

In each of the task mentioned above, we include 300 documents in the testing set. We build 4 different training sets with 10, 50, 100, 200 samples respectively. The objective is to evaluate models on their data efficiency. The prescribed dataset splits are published along with the datasets to enable and apples-to-apples comparison between different models using this benchmark.

2.4.4 Evaluation Toolkit

To evaluate extraction performance, we propose a type-aware fuzzy matching algorithm for each of the entities in our benchmark and report both the macro and micro F1 score for the dataset.

It is common practice to compare the extracted entity with the ground-truth using strict string matching [118]. However, such a simple approach may lead to unreasonable results in

Algorithm 1: Entity Grouping

Data: T is a set of entity names to be hierarchical, E is an entity list.
Result: N is a collection of hierarchical entities.

Function `Group(`T, E`):`
  $E' = \{e \in E | e.type \in T\}$ ▷ E' includes all component entities.
  $N = \phi$ ▷ N is to record all hierarchical entities.
  $M = \phi$ ▷ M is to memorize entity names.
  $i = 1, j = 1$
  while $i \leq j \leq E'.length$ **do**
    if $E'[j].type \notin M$ **then**
      $M = M \cup \{E[j].type\}$
      $j = j + 1$
    end
    else if $E'[j].type \in M$ **then**
      $N = N \cup \{E'[i : j-1]\}$
      $i = j$
      $M = \phi$ ▷ M is reset to refresh memory.
    end
  end
return N

many scenarios. For example, "$ 40,000" does not match with "40,000" because of the missing dollar sign when extracting the total price from a receipt, and "July 1, 2022" does not match with "07/01/2022". Dates may be present in different formats in different parts of the document, and a model should not be arbitrarily penalized for picking the wrong instance. We implement different matching functions for each entity name based on the data type. In the examples mentioned before, we will convert all price values into a numeric type before comparison. Similarly, date strings are parsed, and a standard date-equality function is used to determine equality.

2.4.5 Post-processing for Evaluation Toolkit

We include repeated, unrepeated, or hierarchical entity names in our proposed VRDU benchmark. Our benchmark requires the model to predict a unique value for unrepeated entity names and group component entities into a hierarchical entity. However, such constraints are usually ignored by existing models. For example, the *series ID* is an unrepeated entity and each document should only have one unique value for it, so the model is expected to extract a single

Table 2.4. Experiment results of Single Template Learning, Mixed Template Learning, Unseen Template Learning on Registration Form and Ad-buy Form.

$\|\mathscr{D}\|$	Model	Registration Form						Ad-buy Form			
		Task 1 (Single Template)		Task 2 (Mixed Template)		Task 3 (Unseen Template)		Task 2 (Mixed Template)		Task 3 (Unseen Template)	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
10	**LayoutLM**	65.91	53.64	36.41	28.98	25.54	18.37	20.20	48.13	19.92	47.73
	LayoutLMv2	80.05	68.89	69.44	63.79	54.21	45.38	25.36	58.13	25.17	57.84
	LayoutLMv3	72.51	61.13	60.72	53.37	21.17	15.15	10.16	21.97	10.01	21.89
	FormNet	74.22	62.95	63.61	56.53	50.53	40.24	20.47	55.15	20.28	54.80
50	**LayoutLM**	86.21	74.76	80.15	76.46	55.86	46.43	39.76	79.77	38.42	79.21
	LayoutLMv2	88.68	77.51	84.13	82.04	61.36	52.42	42.23	83.89	41.59	84.14
	LayoutLMv3	87.24	75.86	81.36	77.48	47.85	38.59	39.49	79.22	38.43	79.05
	FormNet	89.38	78.04	85.38	82.41	68.29	57.17	40.68	83.82	39.52	83.49
100	**LayoutLM**	88.70	78.79	86.02	84.04	63.68	53.43	42.38	83.41	41.46	82.27
	LayoutLMv2	90.45	80.03	88.36	86.38	65.96	57.39	44.97	86.38	44.35	85.62
	LayoutLMv3	89.23	78.91	87.32	85.06	57.69	47.84	42.63	82.66	41.54	81.51
	FormNet	90.91	80.82	88.13	85.82	72.58	62.23	40.38	84.24	39.88	83.57
200	**LayoutLM**	90.47	81.77	87.94	86.41	70.47	59.46	44.66	85.85	44.18	84.55
	LayoutLMv2	91.41	83.12	89.19	87.65	72.03	62.14	46.54	87.61	46.31	86.87
	LayoutLMv3	90.89	81.72	89.77	88.54	62.58	50.74	45.16	85.67	44.43	84.16
	FormNet	92.12	82.99	90.51	89.05	77.29	67.82	43.23	86.08	42.87	85.05

string with the highest confidence instead of providing a number of candidates for the users to choose from. When there is no confidence score provided by the model, we simply keep the first extracted entity as the answer for the unrepeated entity names.

The hierarchical entity is a new kind of entity name proposed by our benchmark. Since existing works only focus on the extraction of individual entities, we propose a heuristic method to group the related individual entities into hierarchical ones and evaluate the result accordingly. The method is shown in Algorithm 1. Specifically, the repeated entities are first extracted from the document by the extraction model. Then, we list all these entities according to their index in the reading order extracted by the OCR engine. Then, we run our algorithm to split the list into several spans and each span corresponds to a hierarchical entity. The split point is decided by the occurrence of entity types. Briefly, when an entity type appears the second time, we split the list and build a hierarchical entity with the span. For example, supposing we have 3 entity types, A, B, and C, the extracted list, [A, B, C, B, C], would be divided into [A, B, C] and [B, C] where the split point is the second B in the list.

2.5 Experiments

We conduct experiments on VRDU and evaluate baseline models on the three proposed tasks. We report the micro-F1 and the macro-F1 scores across the training sizes proposed. Our primary goal with these experiments is to demonstrate that several challenges remain open in this space. In fact, while performance on other datasets discussed in Section 2.2 might indicate that this is a solved problem, our results show all models fare worse on VRDU highlighting substantial room for improvements. However, comprehensive comparison between existing models is an explicit *non-goal* for this paper.

2.5.1 Baselines

We evaluate three models on the datasets, LayoutLM [121], LayoutLMv2 [120], LayoutLMv3 [43], and FormNet [52].

- LayoutLM: LayoutLM is a layout-aware pre-trained language model which encodes the absolute coordinates of bounding boxes in the embedding layers of BERT [18] to inform the model of the structural information. Although the visual features from ResNet [34] are appended to the hidden states of LayoutLM to solve the task by the authors, we ignore them since they are not incorporated in the pre-training stage and only serve as add-on features to enhance performance. Thus, LayoutLM is a multi-modal language model with text and layout features.

- LayoutLMv2: LayoutLMv2 further improves the layout embedding in LayoutLM by considering the relative distance between different bounding boxes and proposes the two-stream multi-modal Transformer encoder to learn the correlation between the image and the text. The visual features are properly integrated in the Transformer framework, so LayoutLMv2 is a multi-modal language model with text, layout, and visual features.

- LayoutLMv3: LayoutLMv3 improves the modeling with image features. Cross-modality

pre-training tasks are also incorporated to enhance the performance.

- FormNet: FormNet first uses the attention mechanism to model the 2D spatial relationship between words and further goes beyond simply sequence labeling approach by leveraging the graphs constructed by the layout elements in the documents to aggregate semantically meaningful information from neighboring tokens.

Although we acknowledge there are many other approaches to solving structured extractions from such documents [3, 5, 25, 89, 102, 126, 53, 43], we only consider these three commonly-used ones to highlight the challenges common to all three models and inspire possible directions for future study. As we said previously, a comprehensive comparison is outside the scope of this paper.

2.5.2 Experiment Results

We report the micro-F1 score and macro-F1 score of the three tasks, Single Template Learning (STL), Mixed Template Learning (MTL), and Unseen Template Learning (UTL), under different number of training samples in Table 2.4. Since Ad-buy Form dataset contains a variety of templates and there are only a limited number of documents for each template, we skip the STL task for it. We denote the number of training samples as $|\mathscr{D}|$. Under each setting, we build three training sets of the same size using different random seeds, and the reported numbers are the average result of each model on the three training sets.

First, comparing the results on VRDU and on other benchmarks in Table 2.4, it is clear that there is ample room for improvement. Even when $|\mathscr{D}| = 200$, the highest micro-F1 score is around 90% on Registration Form and around 45% on Ad-buy Form. In contrast, FormNet achieves 97.21% micro-F1 score and LayoutLMv2 achieves 96.01% micro-F1 score on CORD [120, 52]. LayoutLMv2 achieves 97.81% micro-F1 score on SROIE [120]. One might think that results on CORD and SROIE indicate that this is a solved problem. As results on VRDU show, a dataset that reflects challenges in practical settings shows that there is much room

for improvement. The performance of FormNet on FUNSD is 84.69% micro-F1 score, and that of LayoutLmv2 is 84.20% micro-F1 score [120, 52]. Although there is still room to improve, the simplistic labeling schema used in FUNSD makes the results less representative of practical tasks.

We also observe consistent improvement as training data size increases. Even for the simplest task, STL (on Registration Forms), the micro-F1 score of FormNet when $|\mathscr{D}| = 10$ is lower than that when $|\mathscr{D}| = 50$ by 15.16 points. This 15+ point gap remains across all tasks for both datasets between the $|\mathscr{D}| = 10$ and $|\mathscr{D}| = 50$ settings. This holds true for all three models, underscoring that few-shot performance is difficult for all models, even for the simple STL setting getting to micro-F1 scores of just 74.22%.

We then compare the performance of different tasks, STL, MTL, and UTL. The tasks are designed to study the template generalization of each model. From the results, we can see all models performs well in STL and MTL and achieve micro-F1 and macro-F1 scores higher than 80% in both datasets with 200 training samples. We attribute the performance to the fact that there are no unseen layout structures involved when generalizing to the testing set in STL and MTL. However, there is a noticeable gap between the performance of MTL and UTL. At 200 training documents, micro-F1 for UTL is 13–17 percentage points worse than the micro-F1 for MTL across the three models. The performance of UTL on Ad-buy Form is worse than MTL by about 3 points. Recall that the test set in UTL contains documents with templates (layouts) not seen in the training set. We believe techniques that allow models to generalize to new layouts even with modest training sets are of practical importance.

Studying the performance in Ad-buy Form, we see the macro-F1 scores are much higher than the micro-F1 scores. The micro-F1 score weighs every instance of an entity equally, while the macro-F1 scores average the F1 score for each entity. The huge difference between these scores for Ad-buy Form is because of the presence of hierarchical repeated entities with a very low F1 score.

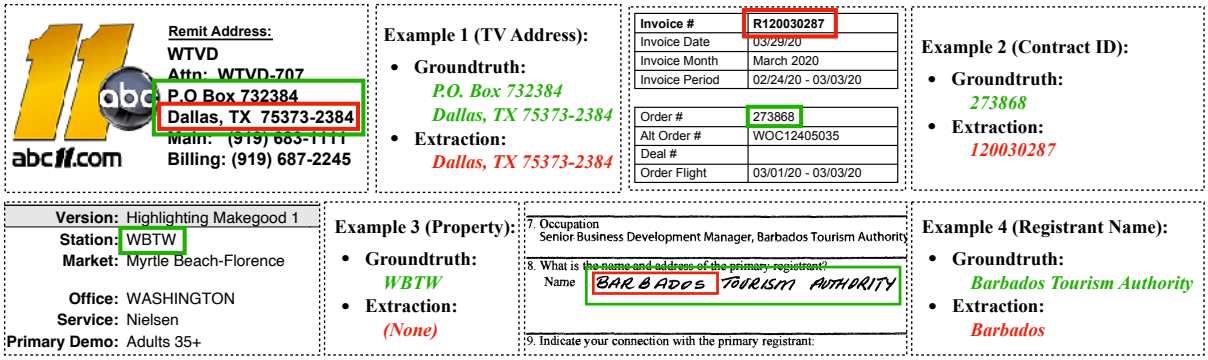


Figure 2.6. Loss cases found in the experiments: Example 1, 2, 3 are from Ad-buy Form, and Example 4 are from Registration Form. In each case, green indicates the ground-truth, and red indicates the extraction from the model.

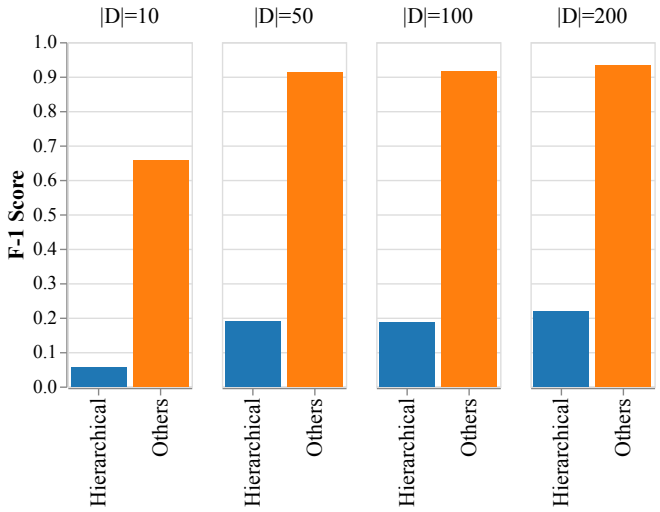


Figure 2.7. Comparison of FormNet on hierarchical and other entities in Mixed Template Learning, where $|\mathscr{D}|$ denotes the number of training samples.

2.5.3 Performance on Hierarchical Entities

We next study the performance of hierarchical entities in Ad-buy Form dataset. Consider the performance of FormNet on MTL. The performance of extracting hierarchical entities vs. other entities is plotted in Figure 2.7. As we can see, there is a huge gap of 60 – 70 points across different sizes of training sets when comparing the micro-F1 score of hierarchical entities and other entities. In contrast to unrepeated entities, the hierarchical entity requires the model not only to correctly extract the corresponding entities, but also to group the components together.

Currently, a heuristic method is used as a simple baseline to deal with the hierarchical entity since no existing models take the hierarchical entity type into consideration. We describe the method in detail in Section 2.4.5. However, such a heuristic results in very low F1 scores for the entity. It is still an open question for future research how to properly extract the hierarchical entities from visually-rich documents.

2.5.4 Case Study

We select four loss cases in the experiments of FormNet and visualize the errors in Figure 2.6. We hope this spurs ideas for future improvements.

Incomplete Extraction

Example 1 and 4 suffer from the incomplete extraction, i.e., the model can correctly locate the ground-truth entity but fails to include all the necessary information. In Example 1, the *TV_address* field is hidden in complex context, which makes it hard to recognize the P.O. Box as part of the address. In Example 4, the error of *Registrant_name* is because of the handwritten characters in different sizes and fonts. The models cannot group the characters together to extract the right entity.

Misleading Key Words

The errors in Example 2 and 3 result from misleading key words. Specifically, in Example 2, the model is confused by the similar key word, "Invoice #", and extract the Invoice ID instead of the Order ID, although there are cases in the training set where the key word for *contract_ID* field is "Order #". In Example 3, the model fails to extract any entity as *Property* since the document is in a new template where "Station" is used as the key word for *Property* field. To solve the rare case in Example 3, it is useful to take into consideration that "WBTW" is common in the training set as *Property* field.

2.6 Conclusions and Future Study

In this paper, we identify five benchmark desiderata to measure progress on solving structured extractions from visually-rich documents in real application. We argue that existing benchmarks fall short on these and propose a new comprehensive benchmark, VRDU, including the dataset with high-quality OCR results and annotations, the tasks corresponding to different application scenarios, and the evaluation toolkit using the type-aware matching algorithm. Based on the novel task settings and extensive experiments, we highlight three areas of opportunity in the visually-rich document understanding task, including the generalization to new templates, the extraction under few-shot scenarios, and the extraction of complex hierarchical-repeated fields. We make the two datasets, all train/validation/test splits, and the evaluation toolkit publicly available. We hope this facilitates progress in this area. In future study, we would further evaluate the existing models using our benchmark to understand how models perform when incorporating multi-modal features into the language models and explore whether there are any potential directions of new frameworks solving the task. We will also focus on the three areas of opportunity discovered in this paper, and explore approaches that can solve the visually-rich document understanding task in the scenarios with unknown templates, limited number of training samples, and hierarchical entities.

2.7 Acknowledgments

Chapter 2, in full, is a reprint of the material as it appears in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata [115]. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Towards Few-shot Entity Recognition in Document Images: A Label-aware Sequence-to-Sequence Framework

3.1 Introduction

Entity recognition lies in the foundation of document image understandings, which aims at extracting word spans that perform certain roles from the document images, such as *header*, *question*. Distinct from the text-only named entity recognition task, the document images, such as forms, tables, receipts, and multi-columns, provide a perfect scenario to apply multi-modal techniques into practice where the rich layout formats in such document images serve as the new, complementary signals for entity recognition performance in addition to the existing textual data.

Recent methods [121, 37, 25] follow the traditional sequence labeling framework to extract the word spans using the standard `IOBES` tagging schemes [70, 93] in named entity recognition tasks. Entity types are treated as class IDs and the semantics of the label surface names are ignored. These methods also largely extend the label space by including combinations of the boundary identifiers (`B`, `I`, `E`, `S`) and entity types. For instance, when there are 3 target entity types, the extended label space would have 13 (i.e., $4 \times 3 + 1$) dimensions. As a result, they fail to learn from the data efficiently and require extensive datasets and high-quality annotations to create the connection between entities and their entity types. Meanwhile, document images

typically include various formats and have a high diversity of entities within each page. It is expensive or almost impossible to enumerate all required entity types and obtain enough annotated data for them. Moreover, ethical concerns would arise when it comes to the receipts or consent forms, which makes it even harder to collect enough data.

Due to the inefficiency of traditional methods and the data limitation in real application scenarios, it is necessary to resort to few-shot learning for entity recognition in document images. We aim at exploiting the potential of a limited number of training pages and try to generalize our model on the much larger number of new pages for testing. In our method, we go beyond the sequence labeling framework and reformulate the entity recognition as a sequence-to-sequence task. Specifically, we propose a new generative labeling scheme for entity recognition — the label surface name is explicitly generated right after each entity as a part of the target sequence. In this way, different entity types are no longer independent dimensions in the label space and models can leverage the semantic connect between the entities and entity types.

To this end, we propose a label-aware sequence-to-sequence framework for entity recognition, LASER. Our implementation is based on a pre-trained language model LayoutReader [112], which is a layout-aware pre-trained sequence-to-sequence model.

As shown in Figure 3.1, LASER extends the architecture of LayoutReader for our proposed generative labeling scheme to better solve the few-shot entity recognition task for document images. Specifically, after generating certain word spans, the model can choose to generate either the following words in the source sequence or label surface names. The entity labels are explicitly inserted in the generated sequence so that the probability of the entity types conditioned on the entity, $P(\text{type}|\text{entity})$, can be maximized not only by the signals from the training data but also by the knowledge from the pre-training of the language models. We also embed the label surface names into the spatial embedding space, so the generation of labels is also aware of the correlation between labels and the regions in the page.

Benefit from the novel generative labeling scheme and the semantics of labels, LASER is able to effectively recognize entities in document images with only a limited number of

training samples. In contrast, the sequence labeling models use less efficient tagging scheme, thus requiring more data and failing in the few-shot settings.

We validate LASER using two benchmarks, FUNSD [29] and CORD-Lv1 [81]. Both datasets are from real scenarios and fully-annotated with textual contents and bounding boxes. We compare our model with strong baselines and study the label-entity semantic and spatial correlations. We summarize our contribution as follows.

- We reformulate the entity recognition task and propose a new generative labeling scheme that embeds the label surface names into the target sequence to explicitly inform the model of the label semantics.
- We propose a novel label-aware sequence-to-sequence framework LASER to better handle few-shot entity recognition tasks for document images than the traditional sequence labeling framework using both label semantics and layout format learning.
- Extensive experiments on two benchmark datasets demonstrate the effectiveness of LASER under few-shot settings.

Reproducibility. We will release the code and datasets on Github[1].

3.2 Problem Formulation

The few-shot entity recognition in the document images is to take the text and layout inputs from a limited number of training samples to predict the boundary of each entity and classify the entity into categories. Given a document image page $\mathscr{P}$, the words within the page are annotated with their textual contents w and the bounding boxes $B = (x_0, y_0, x_1, y_1)$ (top-left and bottom-right corners) by human annotators or the OCR engines, and all the words and bounding boxes are listed in a sequence serving as the inputs from textual and layout modalities. In this way, the entities are spans of these words referring to precise concepts, which makes it

[1] github.com/zlwang-cs/LASER-release

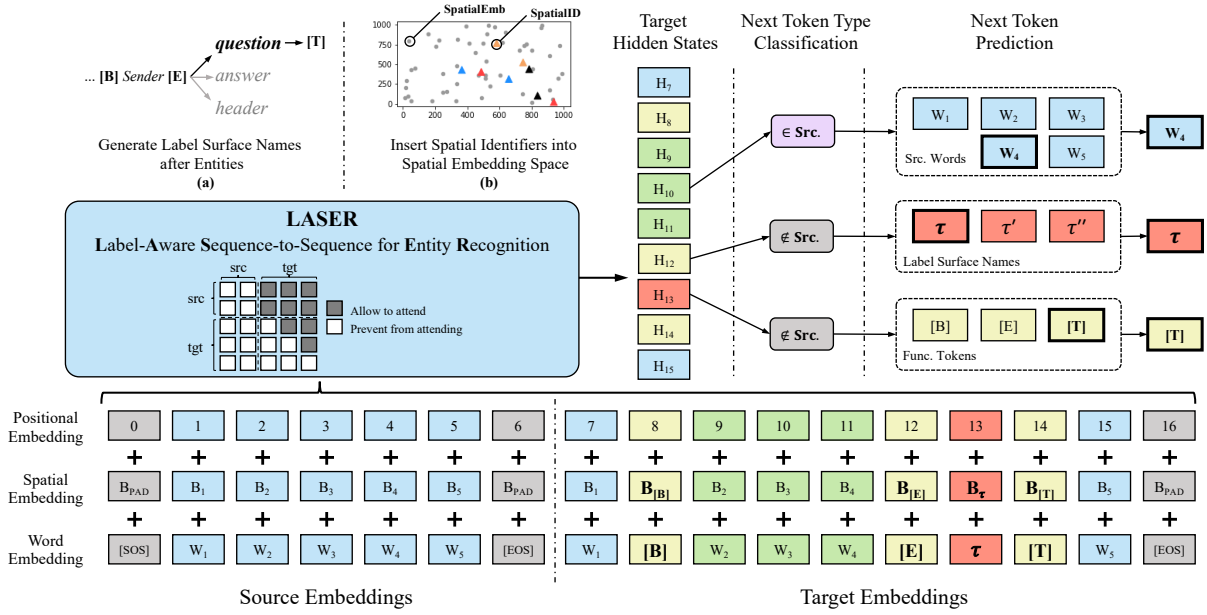


Figure 3.1. The Framework of LASER: `[B]`, `[E]`, `[T]` denote the boundaries; τ, τ', τ'' are the label surface names; (a) is the process of generative labeling scheme; (b) shows the alignment of the spatial identifiers and embeddings.

possible to conduct entity recognition using sequence labeling or generative labeling scheme. We randomly select a small subset of training samples and evaluate the performance under the k-shot training, where k denotes the number of the training samples.

3.3 Our Generative Labeling Scheme

We propose our labeling scheme of entity recognition in the generative manner which generates the entity boundaries and the label surface names explicitly. Specifically, given an entity $e = [w_i, w_{i+1}..., w_j]$, we use the `[B]` and `[E]` to denote the boundary of the entity and append the label surface name afterwards. Overall, the generative formulation is to generate:

$$w_{i-1}, \mathtt{[B]}, w_i, ..., w_j, \mathtt{[E]}, \tau_1, ..., \tau_k, \mathtt{[T]}, w_{j+1}$$

where `[B]` and `[E]` denote the start and end of the entity; $\tau_1 ... \tau_k$ are the words in the label surface name; `[T]` denotes the end of the label surface name. For example, "*Sender*" and

"*Charles Duggan*" are a pair of *question* and *answer* from a document image. According to the generative labeling scheme, the corresponding generated sequence is that: `[B]` *Sender* `[E]` *question* `[T]` `[B]` *Charles Duggan* `[E]` *answer* `[T]`.

3.4 Our LASER Framework

In this section, we introduce our label-aware sequence-to-sequence framework for entity recognition in document images. First, we introduce our method in a bird's eye view. Then we dive into the details of each part including the multi-modal prefix language model, the label-aware generation.

3.4.1 Overview

Our proposed LASER is a label-aware sequence-to-sequence model for entity recognition in document images. The framework is shown in Figure 3.1. The model follows the prefix language model paradigm [91, 20, 4] and is built upon the pre-trained language model, LayoutReader [112]. With extensive knowledge learned in pre-training stage, the model leverages the semantic meaning of label surface names during generation.

Since the functional tokens (e.g. `[B]`, `[E]`) and the label surface names are foreign words in the given page, their layout features are nonexistent. We use trainable vectors as special layout identifiers for these extra tokens and these vectors are well aligned into the spatial embedding space. In this way, the spatial correspondence between layout formats and labels can be learned.

To reinforce the model to distinguish the functional tokens (e.g. `[B]`, `[E]`) and ordinary words, an extra binary classification module is added, and the probability is used in the next token prediction.

Equipped with all the components, our proposed model is able to conduct entity recognition efficiently and effectively under the few-shot setting.

3.4.2 Multi-modal Prefix LM

LASER is built on the layout-aware prefix language model, LayoutReader [112]. Prefix language model refers to a multi-layered Transformer where the source sequence and target sequence are packed together and a "partially-triangle" mask is used to control the attention between tokens in the two sequences. In LASER, the source sequence has full self-attention and the target sequence only attends to the previous tokens so the conditional generative probability is learned.

Input Embedding

The input embedding layer of LASER includes the word embedding, spatial embedding, and positional embedding. We normalize and round the bounding box coordinates to integers ranging from 0 to 1000, and embed them as trainable vectors as spatial embeddings [121, 120, 122, 112]. So the input embeddings of the ordinary words are as follows:

$$e_{w_i} = \text{WordEmb}(w_i) + \text{SpatialEmb}(B_i) + \text{PosEmb}(i)$$

where WordEmb, SpatialEmb, PosEmb are the word embedding, the spatial embedding, and the positional embedding lookup tables, respectively; i is the index of the word in the packed sequence.

The functional tokens and label surface names are new tokens in the given page. We cannot extract the layout features from the bounding boxes of them because their bounding boxes are nonexistent. Instead of the actual bounding boxes, we design unique embedding vectors for each new tokens as their layout identifiers. These identifiers can perform in the same way as real bounding boxes during training to embed the functional tokens and label surface names into the spatial embedding space. The input embedding replaces the spatial embedding with the spatial

identifiers:

$$e_\lambda = \text{WordEmb}(\lambda) + \text{SpatialID}(\lambda) + \text{PosEmb}(i)$$

where SpatialID is the spatial identifier lookup table; i is the index of the word in the packed sequence; $\lambda \in \{\texttt{[B]}, \texttt{[E]}, \texttt{[T]}, \tau_1, ..., \tau_t\}$.

Within the input embedding layer, the pre-trained model learns the semantic and layout formats from word embeddings or spatial features. The spatial embeddings are already pre-trained and further fine-tuned in the downstream tasks, and the spatial identifiers are new to the model and completely trained in the downstream tasks.

Attention Mask

As mentioned, LASER depends on a "partially-triangle" mask to realize sequence-to-sequence training within one encoder. To be more specific, the "partially-triangle" attention mask has two parts, the source part and the target part. In the source part, the tokens can attend to each other, which enables the model to be aware of the entire sequence. In the target part, to predict the next token in a sequence-to-sequence way, we design the "triangle" mask which prevents the tokens from attending to the tokens after them. Therefore, the generative probability conditioned on the previous tokens can be computed.

Output Hidden States

To learn the conditional generative probability of the next token, we take the output hidden states corresponding to the target sequence which is denoted as $\mathbf{h}_{n+1}, \mathbf{h}_{n+2}, ..., \mathbf{h}_{n+m}$, where $n+1$ is the beginning of the target sequence in the packed sequence. According to the "partially-triangle" attention mask, $\mathbf{h}_{n+k}$ is produced with the attention to the source tokens and the previous target tokens, i.e., the input embeddings whose index ranges from 1 to $n+k$. Therefore, $\mathbf{h}_{n+k}$ is used to predict the $(k+1)$-th token in the target sequence.

3.4.3 Label-aware Generation

In the sequence-to-sequence setting, LASER estimates the probability of next token conditioned on the previous context, i.e. $P(x_k|x_{<k})$ and $x_k \in \mathscr{C}$, where $\mathscr{C} = \{w_1...w_n\} \cup \{\tau_1...\tau_t\} \cup \{\texttt{[B]}, \texttt{[E]}, \texttt{[T]}\}$ is the set of all candidate words. Following LayoutReader, we restrain the candidates within the source words instead of the whole dictionary, and we go beyond it and extend the candidate set to include the functional tokens and label surface names. Moreover, to distinguish whether the next word belongs to the source or not, we design an extra binary classification module.

Specifically, we take the hidden states h_k to predict whether the next token is from the source or not. We denote the probability $P(x_{k+1} \in \text{src}) = p_{k+1}$. Then we use p_{k+1} to weight the next token prediction. The probability that the next token is the i-th word in the source is computed as follows:

$$P(x_{k+1} = w_i|x_{\leq k}) = \frac{p_{k+1} \exp\left(\mathbf{e}_{w_i}^T \mathbf{h}_k + b_k\right)}{\sum_j \exp\left(\mathbf{e}_{w_j}^T \mathbf{h}_k + b_k\right)}$$

where w_i is the i-th word in the source; $\mathbf{e}_{w_i}$ is the input embedding of w_i; $\mathbf{b}_k$ is the bias.

Similarly, the probability that the next token is one of the functional tokens or label surface names is computed as follows:

$$P(x_{k+1} = \lambda|x_{\leq k}) = \frac{(1 - p_{k+1}) \exp\left(\mathbf{e}_{\lambda}^T \mathbf{h}_k + b'_k\right)}{\sum_{\lambda'} \exp\left(\mathbf{e}_{\lambda'}^T \mathbf{h}_k + b'_k\right)}$$

where λ is a functional token or label surface name, i.e. $\lambda \in \{\texttt{[B]}, \texttt{[E]}, \texttt{[T]}, \tau_1, ..., \tau_t\}$; $1 - p_{k+1}$ is the probability that $(k+1)$-th token is a functional token or label surface name; $\mathbf{b}'_k$ is the bias.

Label Semantics Learning

With the log likelihood loss of generative language modeling, the model maximize the dot production between the hidden states $\mathbf{h}$ and the input embeddings e. The semantic correlation

is learned considering that the input embeddings of the labels surface names are encoded in the word embeddings.

Spatial Identifier Learning

From the layout format perspective, the input embedding of the label surface names also includes the spatial identifiers. When predicting the next token, the log likelihood also strengthens the relation between the spatial identifiers and the layout context. In this way, LASER inserts the spatial identifiers into the hyperspace of the spatial embeddings. In other words, LASER predicts where a certain label is more likely to be. Similar to the joint probability of language modeling, LASER maximizes the joint probability of a mixture of spatial identifiers and spatial embeddings: $P(...,B_{k-1},B_k,\tau,B_{k+1},...)$ where B_k is the bounding boxes of the words in the page and the τ is the label to predict. Further visualization is conducted in Section 3.5.7.

3.4.4 Sequential Decoding

After training, LASER follows the prefix language modeling paradigm and generates the target sequence sequentially. We input the source sequence into the model and take the last hidden states to predict the first token in the target. Then we append the result to the end of input and repeatedly run the generation. We cache the states of the model and achieve generation in linear time.

3.5 Experiments

In this section, we conduct experiments and ablation study on FUNSD [29] and CORD-Lv1 [81] under few-shot settings. We replace the original label surface names with other tokens to study the importance of semantic meaning. We also plot the heatmaps of the similarity between the spatial identifiers and the spatial embeddings to interpret the spatial correspondence. Case studies are also conducted.

3.5.1 Experimental Setups

All the experiments are under few-shot settings using 1, 2, 3, 4, 5, 6, 7 shots. We use 6 different random seeds to select the few-shot training samples and the data augmentation is conducted to solve the data sparsity. We train all the models using the same data and compute the average performance and the standard deviation. We only report the result of 1, 3, 5, 7 shots for space limitation. To evaluate our model, we first convert our results into `IOBES` tagging style and compute the word-level precision, recall, and F-1 score using the APIs from [74] so that all comparisons with sequence labeling methods are under the same metrics. We believe such experiment settings guarantee the results are representative.

3.5.2 Datasets

Our experiments are conducted on two real-world data collections: FUNSD and CORD-Lv1. Both datasets provide rich annotations for the document image understandings includes the words and the word-level bounding boxes. The details and statistics of these two datasets are as follows.

- **FUNSD:** FUNSD consists of 199 fully-annotated, noisy-scanned forms with various appearance and format which makes the form understanding task more challenging. The word spans in this datasets are annotated with three different labels: `header`, `question` and `answer`, and the rest words are annotated as `other`. We use the original label names.

- **CORD-Lv1:** CORD consists of about 1000 receipts with annotations of bounding boxes and textual contents. The entities have multi-level labels. We select the first level and denote the dataset as CORD-Lv1. The first level includes `menu`, `void-menu`, `subtotal` and `total`. We simplify `subtotal` as `sub` and `void-menu` as `void`.

Table 3.1. Dataset Statistics.

Dataset	# Train Pages	# Test Pages	# Entities / Page
FUNSD	149	50	42.86
CORD-Lv1	800	100	13.82

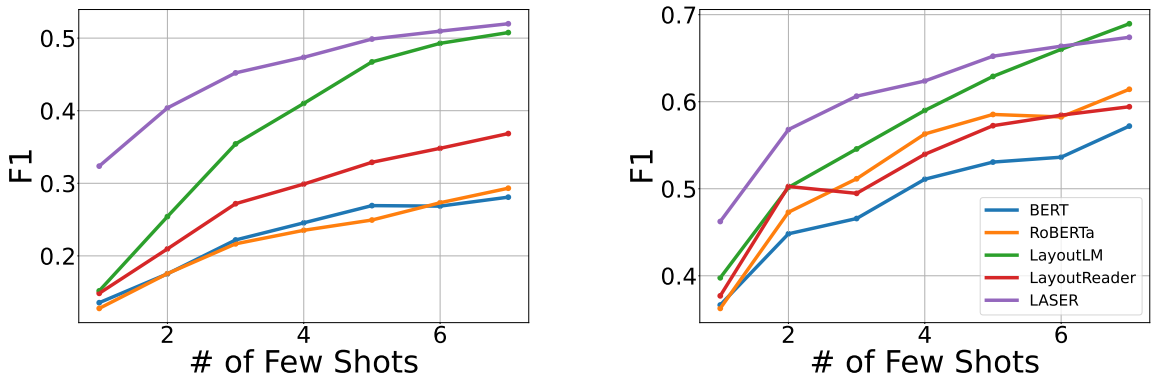
Table 3.2. Evaluation Results with Different Sizes of Few-shot Training Samples: **Bold** denotes the best model; Underline denotes the second-best model.

$\|\mathscr{P}\|$	Model	FUNSD			CORD-Lv1		
		Precision	Recall	F-1	Precision	Recall	F-1
1	BERT	9.62±2.24	24.14±3.46	13.55±2.09	30.64±2.80	45.60±3.45	36.64±3.10
	RoBERTa	9.29±1.57	22.06±5.64	12.76±1.91	30.66±4.25	44.39±6.72	36.25±5.18
	LayoutLM	11.39±1.12	24.73±7.38	15.18±2.17	33.27±7.32	**49.49±10.26**	39.77±8.47
	LayoutReader	11.32±0.62	22.53±4.80	14.84±1.25	32.17±4.64	45.61±6.54	37.70±5.31
	LASER	**30.40±4.89**	**35.20±7.20**	**32.36±5.14**	**47.63±3.90**	45.52±5.84	**46.24±3.01**
3	BERT	16.42±4.30	34.74±5.36	22.19±5.05	39.62±3.99	56.65±4.03	46.58±3.94
	RoBERTa	16.71±3.63	31.28±3.55	21.66±3.84	44.51±4.69	60.18±4.69	51.15±4.70
	LayoutLM	28.67±6.56	**47.22±8.31**	35.42±7.00	47.68±7.49	**63.93±7.04**	54.57±7.46
	LayoutReader	22.37±2.03	35.19±4.97	27.19±2.56	43.85±4.72	56.90±2.47	49.47±3.95
	LASER	**43.66±1.97**	47.08±5.72	**45.21±3.74**	**61.16±3.11**	60.33±5.65	**60.63±4.00**
5	BERT	20.57±2.59	39.25±1.10	26.93±2.46	45.73±4.31	63.29±3.68	53.06±4.14
	RoBERTa	19.47±2.32	35.04±1.89	24.94±1.93	52.21±4.55	66.63±5.52	58.54±4.92
	LayoutLM	39.24±4.33	**58.20±2.45**	46.72±3.12	56.13±7.39	**71.66±6.13**	62.91±7.04
	LayoutReader	27.52±3.44	41.17±4.01	32.89±3.28	51.97±8.42	63.82±7.87	57.24±8.32
	LASER	**47.25±1.93**	52.85±1.22	**49.87±1.29**	**65.62±3.79**	64.90±5.78	**65.23±4.70**
7	BERT	21.44±2.07	40.87±3.79	28.09±2.48	50.13±4.35	66.67±3.67	57.20±4.07
	RoBERTa	23.68±3.06	38.74±3.54	29.32±3.08	55.14±4.49	69.35±4.16	61.43±4.42
	LayoutLM	43.23±5.27	**61.73±5.97**	50.76±5.30	62.87±3.98	**76.38±2.72**	**68.96±3.49**
	LayoutReader	31.22±3.14	45.08±3.83	36.85±3.26	54.43±5.89	65.48±5.34	59.42±5.68
	LASER	**50.62±3.26**	53.63±2.89	**51.98±2.00**	**68.02±3.16**	66.87±4.82	67.40±3.76

3.5.3 Compared Methods

We evaluate LASER against several strong sequence labeling methods as follows.

- **BERT** [18] is a text-only auto-encoding pre-trained language model using the large-scale mask language modeling. We fine-tune the pre-trained BERT-base model with the few-shot training samples on each datasets.
- **RoBERTa** [65] extends the capacity of BERT and achieves better performance in multiple natural language understanding tasks. We also conduct the fine-tuning with few-shot



(a) FUNSD **(b)** CORD-Lv1

Figure 3.2. F-1 Curves with Different Sizes of Few-shot Training Samples.

training samples.

- **LayoutLM** [121] is a multi-modal language model which includes the layout and text information. It is built upon BERT and adds the extra spatial embeddings into the BERT embedding layer. Following LayoutLM, LayoutLMv2 [120] leverages extra computer vision features and improves the performance, which are strong signals but absent in our settings. For a fair comparison, we do not include LayoutLMv2 in our comparative experiments.

- **LayoutReader** [112] is a layout-aware sequence-to-sequence model for reading order detection. We append a linear layer upon the hidden states to conduct sequence labeling.

These compared methods are in their base version and follow the `IOBES` tagging scheme.

3.5.4 Implementation Details

We build LASER on the base of LayoutReader. We use the Transformers [118] and the s2s-ft toolkits from the repository of [20]. We use one NVIDIA A6000 to finetune with batch size of 8. We optimize the model with AdamW optimizer and the learning rate is 5×10^{-5}.

3.5.5 Experimental Results

From Table 3.2 and Figure 3.2, the results show that, under few-shot settings, our proposed model, LASER, achieves the SOTA overall performance compared with sequence labeling models. We conclude that the gain of performance comes mostly from the generative labeling scheme since LASER largely outperforms LayoutReader although both of them share the same backbone.

Specifically, compared with the second-best baseline, LASER improves the F-1 scores by 8.59% on FUNSD and by 3.32% on CORD-Lv1 on average across the different shots and LASER (IRLVT) also surpasses the baselines under most settings.

Moreover, the improvement on precision is remarkable. LASER improves the precision by 12.35% on FUNSD and by 10.62% on CORD-Lv1 on average across the different shots. Especially, under 1-shot setting, it surpasses the best sequence labeling model on FUNSD by 19.01% on precision, 10.47% on recall and 17.18% on F-1 score.

We can also observe a drop in the improvement with the increasing number of training samples. We conclude that, with enough training samples, the sequence labeling learns the meaning of each label and the semantics of each label surface names no longer provides extra useful information.

Based on these comparison, we safely come to the conclusion that our proposed generative labeling scheme is superior to the traditional sequence labeling scheme in few shot settings.

3.5.6 Ablation Study

In the ablation study, we aim at study the role of the label surface names. We introduce an ablation version, LASER (IRLVT), by replacing the label surface names with irrelevant tokens. We also design more different sets of words as substitutes denoted Sub1 and Sub2. The detailed substitutes are introduced in Table 3.3.

To implement the ablation study, we simply replace the word embedding of label surface

Table 3.3. Ablation Study of Different Label Surface Names in LASER. IRLVT uses the irrelevant tokens as labels; ORIG uses the original label surface names; Sub1 and Sub2 use some reasonable alternative label surface names. as substitutes. **Bold** denotes the best model; Underline denotes the second-best model.

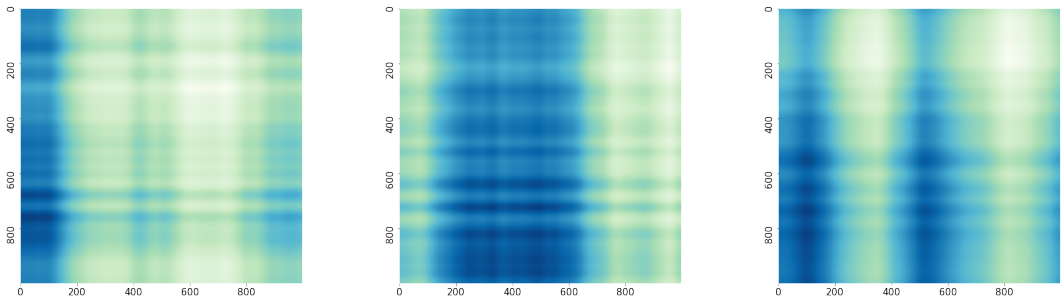
$\|\mathscr{P}\|$	FUNSD				CORD-Lv1			
	Label Surface Names	**Precision**	**Recall**	**F-1**	**Label Surface Names**	**Precision**	**Recall**	**F-1**
1	IRLVT [*x*, *y*, *z*]	30.64±5.89	33.45±9.14	31.62±6.61	IRLVT [*w*, *x*, *y*, *z*]	**48.57±4.93**	44.12±6.36	45.84±3.57
	ORIG [*header*, *question*, *answer*]	30.40±4.89	35.20±7.20	32.36±5.14	ORIG [*menu*, *void*, *sub*, *total*]	47.63±3.90	45.52±5.84	46.24±3.01
	Sub1 [*title*, *key*, *value*]	**31.78±4.75**	34.21±7.44	32.66±5.10	Sub1 [*info*, *etc*, *small*, *number*]	48.12±4.15	**48.47±6.60**	**48.04±4.06**
	Sub2 [*page*, *topic*, *value*]	30.90±5.20	**35.97±8.57**	**33.03±6.31**	Sub2 [*page*, *non*, *part*, *price*]	45.59±5.68	44.09±7.87	44.38±5.39
3	IRLVT [*x*, *y*, *z*]	43.51±1.46	47.92±5.93	45.44±3.36	IRLVT [*w*, *x*, *y*, *z*]	61.50±2.52	59.17±4.11	60.27±2.99
	ORIG [*header*, *question*, *answer*]	43.66±1.97	47.08±5.72	45.21±3.74	ORIG [*menu*, *void*, *sub*, *total*]	61.16±3.11	**60.33±5.65**	60.63±4.00
	Sub1 [*title*, *key*, *value*]	43.87±1.33	47.11±6.07	45.26±3.44	Sub1 [*info*, *etc*, *small*, *number*]	61.54±2.76	58.79±6.76	60.00±4.57
	Sub2 [*page*, *topic*, *value*]	**43.88±1.34**	**48.01±6.86**	**45.65±3.93**	Sub2 [*page*, *non*, *part*, *price*]	**61.85±2.16**	60.29±2.85	**61.03±2.10**
5	IRLVT [*x*, *y*, *z*]	46.94±1.87	52.96±2.03	49.74±1.63	IRLVT [*w*, *x*, *y*, *z*]	63.67±3.82	61.10±5.21	62.33±4.48
	ORIG [*header*, *question*, *answer*]	47.25±1.93	52.85±1.22	49.87±1.29	ORIG [*menu*, *void*, *sub*, *total*]	**65.62±3.79**	**64.90±5.78**	**65.23±4.70**
	Sub1 [*title*, *key*, *value*]	47.43±2.29	52.19±2.09	49.68±1.98	Sub1 [*info*, *etc*, *small*, *number*]	65.05±5.59	63.64±7.16	64.31±6.34
	Sub2 [*page*, *topic*, *value*]	**47.46±2.18**	**53.50±1.01**	**50.26±1.16**	Sub2 [*page*, *non*, *part*, *price*]	65.57±3.04	64.71±3.97	65.12±3.38
7	IRLVT [*x*, *y*, *z*]	50.30±2.26	**54.14±3.48**	52.08±2.26	IRLVT [*w*, *x*, *y*, *z*]	66.08±3.26	64.73±5.08	65.32±3.74
	ORIG [*header*, *question*, *answer*]	**50.62±3.26**	53.63±2.89	51.98±2.00	ORIG [*menu*, *void*, *sub*, *total*]	**68.02±3.16**	**66.87±4.82**	**67.40±3.76**
	Sub1 [*title*, *key*, *value*]	50.22±3.20	53.79±3.13	51.88±2.56	Sub1 [*info*, *etc*, *small*, *number*]	67.61±4.19	66.64±5.72	67.08±4.72
	Sub2 [*page*, *topic*, *value*]	50.43±2.88	54.03±2.71	**52.10±2.09**	Sub2 [*page*, *non*, *part*, *price*]	66.64±3.97	63.59±7.00	65.02±5.47

names. For example, in LASER (Sub1) on FUNSD, we use the wording embedding of *title* instead of the original *header*.

From Table 3.3, we compare the performance of all the ablation models. We observe that LASER performs differently with distinct label semantics. In most cases, the human-designed labels can provide stronger semantic correlation with the entities than the irrelevant labels so they can further improve the performance. However, there are also drops due to improper labels. Overall, we conclude that the semantic meanings of the label surface names are useful to bridge the gap between the labels and entities.

3.5.7 Spatial Correspondence Interpretation

In this section, we study the ability of LASER to capture the spatial correspondence between certain areas and the labels. The experiment is based on the results of LASER on FUNSD with 7 shots. As mentioned in Section 3.4.2, we design unique spatial identifiers for the label surface names. The identifiers are in the same form as the spatial embeddings and LASER inserts the identifiers into the original spatial embedding space during sequence-to-



(a) Header **(b)** Question **(c)** Answer

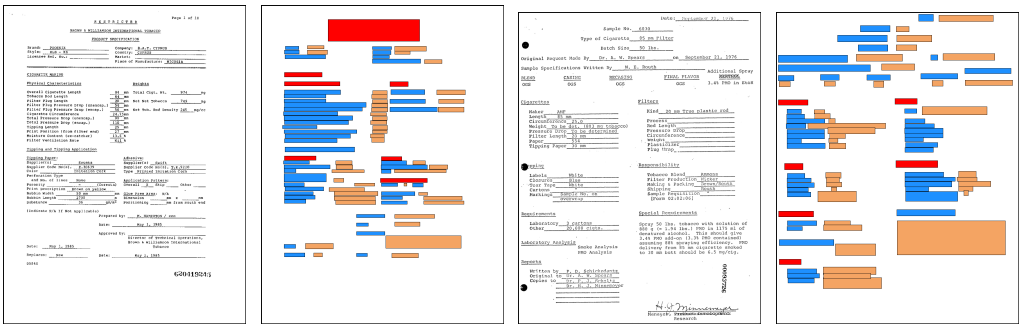
Figure 3.3. Spatial correspondence visualization on FUNSD for different entity types.

sequence training. Ideally, the model can learn where a certain label is more likely to appear. To visualize such patterns, we compute the cosine similarity matrix M of identifiers and the spatial embeddings as $M_{ij} = \cos\left(\text{SpatialID}(\tau), \text{SpatialEmb}((\text{i}, \text{j}))\right)$ where (i, j) is the normalized coordinate pair; $\tau \in \{\tau_1, ..., \tau_t\}$. Then we plot the heatmap of the similarity matrix, where the highlight areas mean the higher similarities.

From Figure 3.3, we observe that the label `header` is more likely to be in the middle column of the page and may appear in the bottom part as well when there are multiple paragraphs. Intuitively, the label `question` and `answer` should appear in pairs and it is observed in Figure 3.3 that their heatmaps are almost complementary to each other. Several examples from FUNSD are selected to demonstrate the visualization results in 3.4. Comparing the examples and the visualization results, we conclude that the spatial identifiers of labels capture the formats of pages and LASER leverages these features to better extract the entities under few shot settings.

3.5.8 Case Study

We visualize cases from the 5-shot setting. From Figure 3.5, we observe LASER can extract the entities correctly, and the errors of LayoutLM comes from the failure to extract the entities or wrong entity type predictions. Since the sequence labeling groups the words into spans through `IOBES` tagging, which creates great uncertainty. Meanwhile, LASER also



(a) Original Image **(b)** Labeled Entities **(c)** Original Image **(d)** Labeled Entities

Figure 3.4. Layout Format Examples from FUNSD: , , denotes `question`, `answer`, `header`.

learns `questions` and `answers` appear in pairs (see Figure 3.5b). It also properly predicts a numerical string as `menu` even if numbers are likely to be `total` (see Figure 3.5e).

Table 3.4. Text-only Dataset Statistics

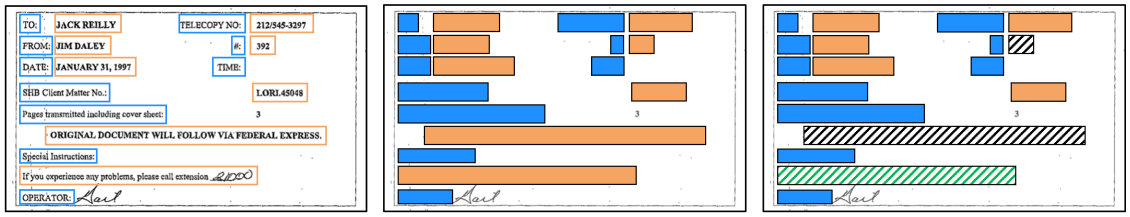
Dataset	# Train	# Test	# Entity Type
OntoNotes	60.0k	8.3k	18
Mit Movie	7.8k	2.0k	12

Table 3.5. Results of 10-way-5-shot Experiments

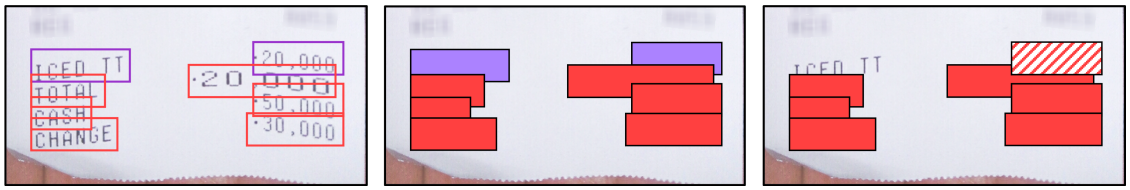
Model	OntoNotes	MIT Movie
	F-1	F-1
BERT	60.79±0.97	47.88±0.97
RoBERTa [41]	57.70	51.30
UniLM	60.82±1.26	51.09±1.40
LASER	**61.11±1.08**	**51.88±1.27**

3.5.9 Text-only Entity Recognition

LASER is designed for the entity recognition task in document images where both text and layout can be leveraged to acquire essential information. However, the generative labeling scheme is not constrained in the scenario of document images. We briefly explore the potential of the generative labeling scheme in text-only scenario. We initialize LASER with a text-only



(a) Test Image and Expected Labels **(b)** LASER Results **(c)** LayoutLM Results



(d) Test Image and Expected Labels **(e)** LASER Results **(f)** LayoutLM Results

Figure 3.5. Case Studies. (a), (b), (c) from FUNSD; (d), (e), (f) from CORD-Lv1; , , , , denote `menu`, `question`, `answer`, `other`; , , denote `menu`, `total`; , /// denote the right, wrong predictions.

language model, UniLM [20], based on the experiments in [112], and apply it onto text-only entity recognition task. Following [41], we conduct 10-way-5-shot experiments on two datasets, OntoNotes [117] and MIT Movie [60], which cover general domains and review domains, respectively. The dataset statistics are shown in Table 3.4 and the results are as shown in Table 3.5. We observe that our method can also surpass the sequence labeling methods in these two datasets, showing the great potential of the generative labeling scheme in the entity recognition tasks.

3.6 Related Work

Layout-aware LMs

Since the post-OCR processing has great application prospects, existing works propose to adapt the language pre-training to the layout formats learning. LayoutLM [121] is the pioneer in this area, which successfully uses the coordinates to represent the layout information in the embedding layer of BERT [18]. Following LayoutLM, the upgraded version, LayoutLMv2 [120], is further proposed to leverage the visual features and benefits from the alignment between

words and the regions in the page. LAMBERT [25] and BROS [37] continue studying the layout representation which uses the sinusoidal function or apply the relative positional biases from T5 [91]. LayoutReader [112] aims to predict the reading order of words from the OCR results. ReadingBank [112] is proposed to facilitate the pre-training of reading order detection, which annotates the reading order of millions of pages.

Generalized Seq2Seq

Sequence-to-sequence architecture is basic in natural language processing and is originally designed for machine translation. With the rise of large pre-trained models, sequence-to-sequence models are increasingly used with new problem formulation. Existing works exploit the potential latent knowledge and stronger representation ability of sequence-to-sequence modeling. GENRE [17] creatively reformulates the entity retrieval task into the sequence-to-sequence settings. It inferences the lined entities using the generation of BART. Recent works on prompt learning also leverage the pre-trained sequence-to-sequence language models to conduct few shot learning [62, 90, 30].

3.7 Conclusions and Future Work

In this paper, we present LASER, a label-aware sequence-to-sequence framework for entity recognition in document images under few-shot settings. It benefits from the generative labeling scheme which reformulates the entity recognition task into the sequence-to-sequence setting. The label surface names are embedded into the generated sequence. Compared with the sequence labeling methods, LASER leverages the rich semantics of the label surface names and overcome the limitation of training data. Moreover, we design spatial identifiers for each label and well insert them into the spatial embedding hyperspace. In this way, LASER can inference the entity labels from the layout formats perspective and empirical experiments demonstrate our method can learn the layout formats though limited number of training samples.

For further research, we will investigate the selection of label surface names and how to

better leverage the semantics from the pre-trained sequence-to-sequence models. We also notice that such labeling scheme can cope with unknown categories. We will focus on the generalization of our method.

3.8 Acknowledgments

Chapter 3, in full, is a reprint of the material as it appears in Findings of the Association for Computational Linguistics: ACL 2022. Zilong Wang and Jingbo Shang [105]. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Towards Zero-shot Relation Extraction in Web Mining: A Multimodal Approach with Relative XML Path

4.1 Introduction

The internet is a vast repository of semi-structured web pages that are characterized by the use of HTML/XML markup language. Compared to plain text in traditional natural language understanding tasks, these web pages possess additional multimodal features such as the semi-structured visual and layout elements from the HTML/XML source code. These features can be effectively generalized across different websites and provide a richer understanding of the web pages [66, 67, 68].

The dynamic nature of the modern internet poses significant challenges for web mining models due to its rapid pace of updates. It is infeasible to annotate emerging web pages and train targeted models for them. Modern web mining models are expected to perform zero-shot information extraction tasks with little prior knowledge of emerging subjects or templates [68, 10]. In this context, the multimodal features extracted from the HTML/XML source code as well as the textual contents are crucial for dealing with zero-shot information extraction task on the countless emerging web pages.

Previous approaches to the problem of zero-shot web mining have primarily focused on

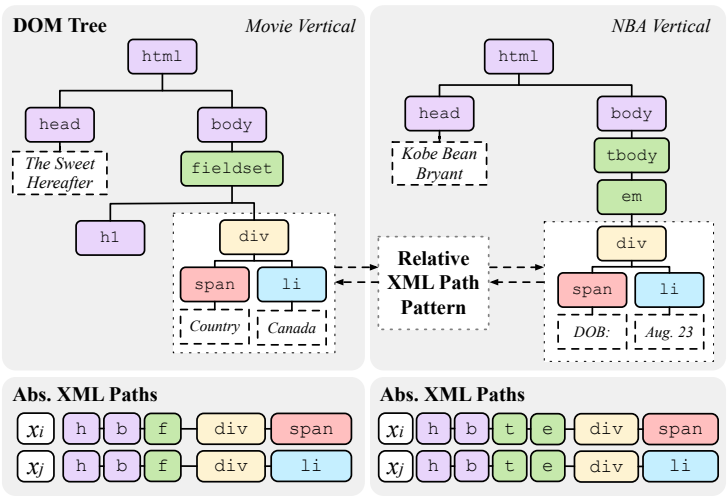


Figure 4.1. The structural information from semi-structured web pages. Based on the DOM Tree from the HTML source code, the absolute and relative XML Paths are extracted. We believe the web page structure is well modeled by the XML Paths to predict the attribute of text nodes and the relative XML Paths provide extra signals to predict the relation between text nodes.

creating rich representations through large-scale multimodal pre-training, utilizing XML Paths of text nodes[1] [58, 131, 55]. As shown in Figure 4.1, XML Paths are sequences of tags (e.g., `div`, `span`, `li`) indicating the location of the text node in the DOM Tree[2] of the page. These pre-training approaches extend vanilla language models by embedding the absolute XML Paths but fail to take into account the relative local relationship expressed by the relative XML Paths. The related nodes tend to be close to each other in the DOM tree, which results in a long common prefix in their XML Paths, as shown in Figure 4.1. Such local relation is more common than the absolute XML Path patterns. Therefore, it is easy to transfer the relative XML Paths to new web pages, and the relative XML Paths serve as a more efficient and meaningful signal in predicting the relation between text nodes.

Additionally, existing web mining approaches tend to treat each web page separately and focus on memorizing their various templates, ignoring the fact that the relevance across different web pages of the same website is also meaningful to identify the related text nodes [131, 55, 68]. Intuitively, a text node is more likely to be a key word if it appears frequently in a collection

[1] https://en.wikipedia.org/wiki/XPath
[2] https://en.wikipedia.org/wiki/Document_Object_Model

of web pages and its surrounding words are not fixed. For example, in web pages about NBA players, the statistics about the height, age are common text fields in the player introduction, so the text nodes, such as "Height:" and "Age:" should appear more frequently than other text nodes and the surrounding text contents should be different.

In light of the aforementioned challenges in web mining, we propose a web mining model with Relative XML Path, ReXMiner, for tackling the zero-shot relation extraction task from semi-structured web pages. Our approach aims to learn the local relationship *within* each web page by exploiting the potential of the DOM Tree. Specifically, we extract the shortest path between text nodes in the DOM Tree as the relative XML Path, which removes the common prefix in the XML Paths. Inspired by the relative position embedding in T5 [91], we then embed the relative XML Paths as attention bias terms in the multi-layered Transformer. Additionally, we incorporate the popularity of each text node by counting the number of times it occurs *across* different web pages, and embed the occurrence logarithmically in the embedding layer. Furthermore, we address the data sparsity issues in the relation extraction task by adopting contrastive learning during training which is widely used in related works [96, 36, 54]. We randomly generate negative cases and restrict their ratio to the positive ones, allowing the model to properly discriminate related node pairs from others.

By learning from the relationships between text nodes *within* and *across* pages, ReXMiner is able to effectively transfer knowledge learned from existing web pages to new ones. We validate our approach on web pages from three different verticals from the SWDE dataset [31], including Movie, University, and NBA. The relation labels are annotated by [67]. We summarize our contribution as follows.

- We propose a novel multimodal framework, ReXMiner, that effectively exploit the relative local relationship *within* each web page and incorporate the popularity of text nodes *across* different web pages in the relation extraction task.
- We represent the relative local relation and the popularity of text nodes in the language

models through relative XML Paths in the DOM Tree and the occurrence number of text nodes across different web pages.

- Extensive experiments on three different verticals from SWDE dataset demonstrate the effectiveness of ReXMiner in the zero-shot relation extraction task in web mining.

Reproducibility. The code will be released on Github.[3].

4.2 Related Work

Information Extraction in Web Mining

How to efficiently and automatically gathering essential information from the internet is always a hot topic in the academia of natural language processing and data mining due to the enormous scale and vast knowledge within the internet. The open information extraction task in web mining is originally proposed by [22] and further developed by following works, including [23, 7, 72] which rely on the syntactic constraints or heuristic approaches to identify relation patterns, and [16, 66, 67, 119, 57] which introduce neural networks to solve the task under supervision or distant supervision settings. Our proposed method follows the task formulation of the zero-shot relation extraction in web mining proposed by ZeroShotCeres [68] where the models are required to transfer relation knowledge from the existing verticals to the unseen ones. ZeroShotCeres adopts the graph neural network to understand the textual contents and model the layout structure. It finally produces rich multimodal representation for each text node and conduct binary classification to extract related pairs.

Layout-aware Multimodal Transformers

The pre-trained language models, such as BERT [18], XLNet [123], GPT [8], T5 [91], are revolutionary in the academia of natural language processing. It achieves state-of-the-art performance in text-only tasks. To further deal with multimodal scenarios, various features are extracted and incorporated into the Transformer framework. Recent study has shown that it

[3] github.com/zlwang-cs/ReXMiner-release

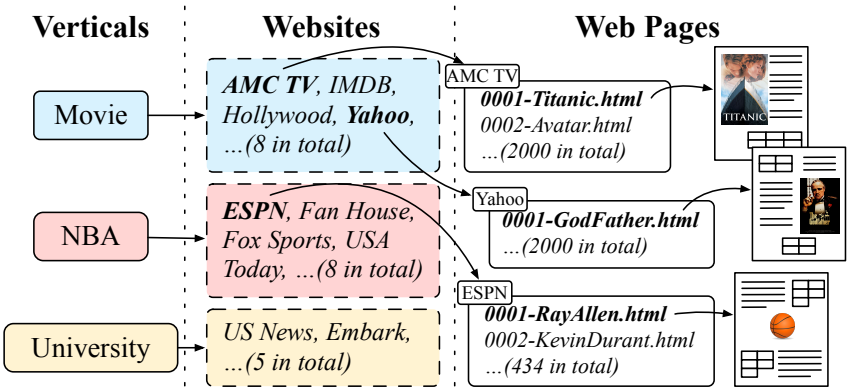


Figure 4.2. The web pages in the SWDE dataset. There are three verticals, Movie, NBA, University. Each vertical includes several websites. Each website includes hundreds web pages.

is beneficial to incorporate multimodal features, such as bounding box coordinates and image features,into pre-trained language models to enhance overall performance in understanding visually-rich documents [121, 120, 43, 27, 107]. Similarly, web pages are rendered with HTML/XML markup language and also represent layout-rich structures. Multimodal features from the DOM Tree or rendered web page images are incorporated in the pre-trained language models to solve the tasks in the semi-structured web pages [58, 131, 55, 103].

4.3 Problem Formulation

The zero-shot relation extraction in web mining is to learn knowledge of related pairs in the existing web pages and transfer the knowledge to the unseen ones [68]. The unseen web pages should be orthogonal to the existing ones with regard to vertical, topic, and template. The zero-shot setting requires the web mining models to extract relevant pairs based on both the textual content and the DOM Tree structure of web pages. Specifically, each web page is denoted as a sequence of text nodes, $P = [x_1, x_2, ..., x_n]$, where n is the number of nodes in the page. Each node involves textual contents and the XML Path extracted from the DOM Tree, $x_i = (w_i, xpath_i)$. The goal of the zero-shot relation extraction task is to train a model using related pairs, $(x_i \rightarrow x_j)$, from a set of web pages, and subsequently extract related pairs from unseen ones. For example, as shown in Figure 4.2, one of our tasks is to train models with web

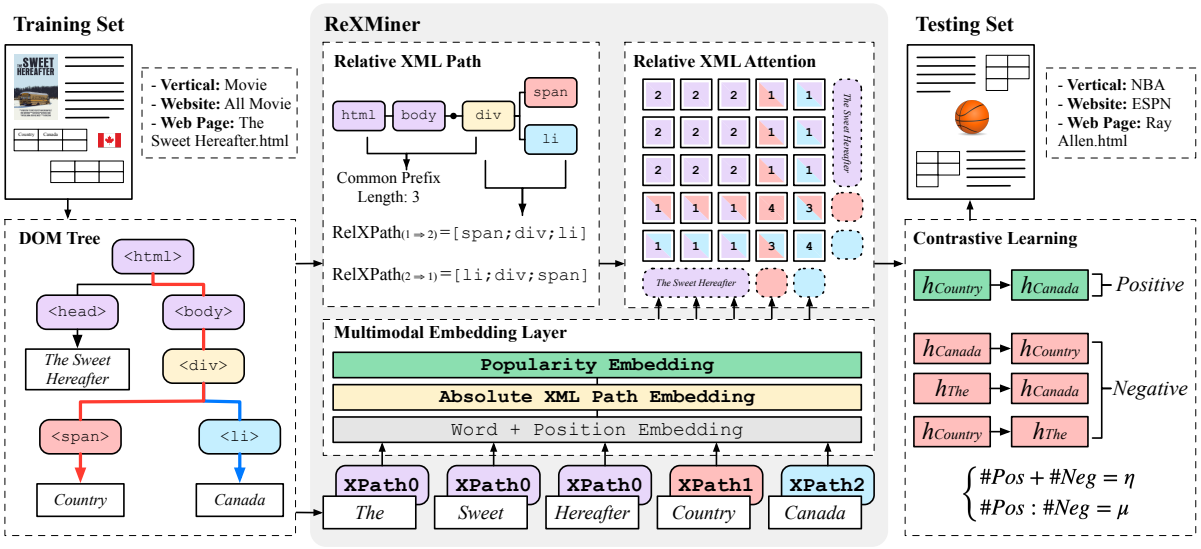


Figure 4.3. The framework of ReXMiner. We extract the DOM Tree of each web page from the HTML source code and further extract the absolute and relative XML Paths. We embed the popularity of text nodes and absolute XML Paths in the embedding layer and embed the relative XML Paths in the attention layers. We reduce the binary classification loss of the relation pairs sampled by negative sampling. In this figure, we train ReXMiner using web pages from the Movie vertical and test it on unseen web pages from the NBA vertical.

pages from Movie and NBA verticals and test the models with web pages from the University vertical.

4.4 Methodology

We extend the text-only language models with multimodal features and propose a novel framework, ReXMiner, for zero-shot relation extraction task in web mining. Figure 4.3 shows the components in our framework. We adopt the absolute XML Path embedding in MarkupLM [55], and further extend it with popularity embedding and relative XML Path attention. To cope with the sparsity issue in the relation extraction task, we adopt the contrastive learning strategy where we conduct negative sampling to control the ratio between positive cases and negative cases.

4.4.1 Absolute XML Path Embedding

We follow the idea in MarkupLM and embed the absolute XML Paths in the embedding layer. We introduce it in this section for self-contained purpose. The XML Path is a sequence of tags from HTML/XML markup language (e.g., `div`, `span`, `li`). Both of the tag names and the order of tags are important to the final representation. Therefore, in the embedding layer, we first embed each tag as a embedding vector, and all these tag embeddings are concatenated. To be more specific, we pad or truncate the XPath to a tag sequence of fixed length, $[t_1, ..., t_n]$, and embed the tags as $\text{Emb}(t_1), ..., \text{Emb}(t_n)$ where t_i is the i-th tag and $\text{Emb}(t_i) \in \mathbb{R}^s$ is its embedding. We further concatenate the vectors as $\text{Emb}(t_1) \circ ... \circ \text{Emb}(t_n) \in \mathbb{R}^{n \cdot s}$ to explicitly encode the ordering information, where $\circ$ is the operation of vector concatenation. To fit in with the hyperspace of other embedding layers $\in \mathbb{R}^d$, a linear layer is used to convert the concatenation into the right dimension.

$$\begin{aligned}&\text{AbsXPathEmb}(xpath_i)\\ =&\text{Proj}(\text{Emb}(t_1) \circ ... \circ \text{Emb}(t_n)) \in \mathbb{R}^d\end{aligned}$$

where $\text{Proj}(\cdot)$ is a linear layer with parameters $W \in \mathbb{R}^{ns \times d}$ and $b \in \mathbb{R}^d$.

4.4.2 Popularity Embedding

We propose Popularity Embedding to incorporate the occurrence of the text nodes into the pre-trained framework. Web pages from the same website use similar templates. The popularity of a certain text node across different web pages of the same website is meaningful in the relation extraction task in the web mining. Intuitively, a text node is more likely to be a key word if it appears frequently and its neighboring words are not fixed.

In details, given a text node $(w, xpath)$ and N web pages $P_1, ..., P_N$ from the same website, we iterate through all the text nodes in each web page and compare their textual contents with w, regardless of their XML Paths. We count the web pages that involves nodes with the same

text and define the number of these web pages as the *popularity* of w. Thus, higher popularity of a text node means that the same textual contents appears more frequently in the group of web pages.

$$\sigma(w,P) = \begin{cases} 1, & \text{if } \exists xpath', \text{ s.t.}(w,xpath') \in P \\ 0, & \text{otherwise} \end{cases}$$

$$pop(w) = \sum_{i=1}^{N} \sigma(w,P_i)$$

where $pop(w)$ is the popularity of w. Then we normalize it logarithmically and convert the value into indices ranging from 0 to τ. Each index corresponds to an embedding vector.

$$\text{PopEmb}(w) = \text{Emb}\left(\left\lfloor \tau \cdot \frac{\log pop(w)}{\log N} \right\rfloor\right) \in \mathbb{R}^d$$

where $\text{Emb}(\cdot)$ is the embedding function; τ is the total number of popularity embeddings; d is the dimension of embedding layers.

Formally, along with the absolute XML Path embedding, the embedding of the i-th text node, $(w_i, xpath_i)$, is as follows.

$$\begin{aligned} e_i = &\text{PopEmb}(w_i) + \text{AbsXPathEmb}(xpath_i) \\ &+ \text{WordEmb}(w_i) + \text{PosEmb}(i) \end{aligned}$$

4.4.3 Self-Attention with Relative XML Paths

The local relation within each web page is essential to the zero-shot relation extraction since the related nodes are more likely to be close in the DOM Tree. As shown in Figure 4.4, they present a long common prefix in their XML Paths, and the rest parts of their XML Paths compose the relative XML Paths between them. The relative XML Paths can be seen as the shortest path between text nodes in the DOM Tree. Therefore, the relative XML Paths are useful

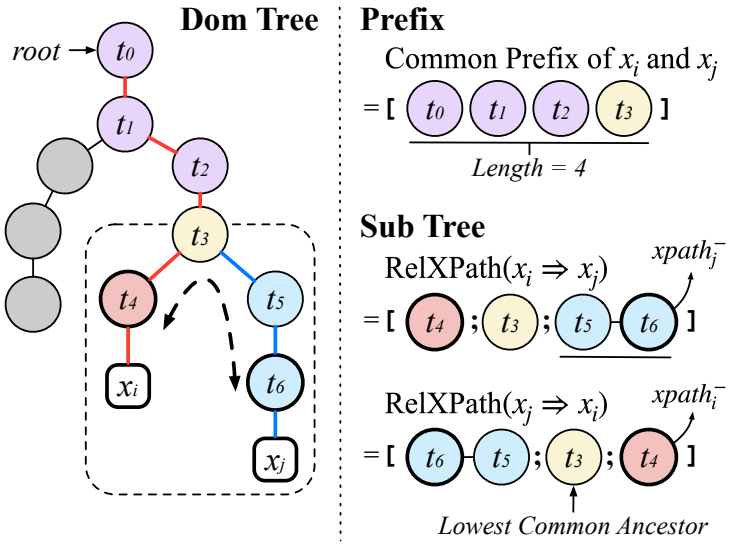


Figure 4.4. The relative XML Path illustration. In Prefix, we focus on the length of the common prefix of the pair of nodes showing their depth in the DOM Tree, and embed it in the first α attention layers. In Sub Tree, we focus on the shortest path between the pair of nodes, and embed it in the following β attention layers.

signals and could be well transferred into unseen web pages. Enlightened by [123, 91, 83, 84], we model the relative XML Paths as bias terms and incorporate them into the multi-layer self-attention of Transformer. Specifically, we embed the common prefix length in the first α layers of self-attention and embed the relative XML Paths tags in the next β layers of self-attention, where $(\alpha+\beta)$ equals to the total number of layers. In the case of Figure 4.4, we embed the common prefix length 4 as well as the relative XML Paths $[t_4, t_3, t_5, t_6]$.

Extracting Relative XML Paths

Given a pair of text nodes, x_i and x_j, we first extract the common prefix of their XML Paths which shows the path from the root to the lowest common ancestor of these two nodes in the DOM Tree (e.g. $[t_0, t_1, t_2, t_3]$ in Figure 4.4). We denote the prefix length as d_{ij}. The rest parts in the XML Paths shows the path from the lowest common ancestor to the text node. We denote them as $xpath_i^-$ and $xpath_j^-$ which are the XML Paths without the common prefix (e.g. $[t_5, t_6]$ and $[t_4,]$ in Figure 4.4). They also compose the shortest path between these nodes in the DOM

Tree:

$$\text{RelXPath}(x_i \Rightarrow x_j) = [\text{rev}(xpath_i^-); t; xpath_j^-]$$
$$\text{RelXPath}(x_j \Rightarrow x_i) = [\text{rev}(xpath_j^-); t; xpath_i^-]$$

where $\text{rev}(\cdot)$ is to reverse the tag sequence; t is the lowest common ancestor of x_i and x_j (e.g. t_3 in Figure 4.4). In the case of Figure 4.4, $\text{rev}(xpath_j^-)$ equals $[t_6, t_5]$, the lowest common ancestor is t_3, and $xpath_i^-$ equals $[t_4,]$, so $\text{RelXPath}(x_j \Rightarrow x_i)$ equals $[t_6, t_5, t_3, t_4]$.

Adding Bias Terms

In the first α layers of the self-attention, we embed the common prefix length d_{ij} as bias terms. The attention weight between x_i and x_j is computed as

$$A_{ij}^{\alpha} = \frac{1}{\sqrt{d}}(W^Q e_i)^\top (W^K e_j) + \mathbf{b}^{\text{pre}}(d_{ij})$$

where the common prefix length d_{ij} is a bounded integer and each integer is mapped to a specific bias term by $\mathbf{b}^{\text{pre}}(\cdot)$.

In the next β layers of the self-attention, we embed the relative XML Paths as bias terms. Following the absolute XML Path embedding (introduced in Section 4.4.1), we project the embedding of tags in $\text{RelXPath}(x_i \Rightarrow x_j)$ into bias terms. Specifically, we split the relative XML Path at the lowest common ancestor tag and embed each part separately. When embedding $\text{RelXPath}(x_i \Rightarrow x_j)$, the two sub-sequences of tags are $[\text{rev}(xpath_i^-); t]$ and $[t; xpath_j^-]$.

In the equation, t_m is the lowest common ancestor (e.g. t_3 in Figure 4.4); $[t_1, ..., t_m]$ is the path from x_i to the lowest common ancestor (e.g. $[t_4, t_3]$ in Figure 4.4); $[t_m, ..., t_n]$ is the path from the lowest common ancestor to x_j (e.g. $[t_3, t_5, t_6]$ in Figure 4.4). The bias term is as follows,

$$\mathbf{b}^{\text{xpath}}(x_i, x_j) = \mathbf{b}(\text{Emb}(t_1) \circ ... \circ \text{Emb}(t_m)) + \mathbf{b}'(\text{Emb}'(t_m) \circ ... \circ \text{Emb}'(t_n)) \in \mathbb{R}$$

where $\circ$ is the operation of vector concatenation; Emb is the embedding function; $\mathbf{b}$ is a linear layer projecting embedding to $\mathbb{R}$. We also use two sets of modules to differentiate the two sub-sequences of tags, $(\mathbf{b}, \text{Emb})$ and $(\mathbf{b}', \text{Emb}')$. Thus, the attention weight between x_i and x_j is computed as

$$A_{ij}^{\beta} = \frac{1}{\sqrt{d}}(W^Q e_i)^\top (W^K e_j) + \mathbf{b}^{\text{xpath}}(x_i, x_j)$$

4.4.4 Contrastive Learning

We observe the sparsity issues in the relation extraction task, where only a small proportion of nodes are annotated as related pairs so the negative cases are much more than the positive ones. To tackle this issue, we adopt the contrastive learning and conduct negative sampling to control the ratio between the positive cases and negative ones.

Negative Sampling

The number of positive cases and negative cases in the sampling should follow,

$$\#\text{Pos} + \#\text{Neg} = \eta \ ; \quad \#\text{Pos} : \#\text{Neg} = \mu$$

where we denote the number of related pairs in the groundtruth as #Pos and the number of negative samples as #Neg; η and μ are two hyper-parameters.

Loss Function

To distinguish the positive samples from the negative ones, we train our model with cross-entropy loss. First, we define the probability of a related pair, $(x_i \rightarrow x_j)$ using the Biaffine attention [77] and the sigmoid function σ.

$$\text{Biaffine}(u, v) = u^\top M v + W(u \circ v) + b$$
$$\mathscr{P}(x_i \rightarrow x_j) = \sigma(\text{Biaffine}(h_i, h_j))$$

Table 4.1. The statistics of the SWDE datset.

Vertical	# Websites	# Web Pages	# Pairs per Web Page
Movie	8	16000	34.80
NBA	8	3551	11.94
University	5	8090	28.44

where h_i and h_j are the hidden states from ReXMiner corresponding to x_i and x_j; M, W, b are trainable parameters; $\circ$ is the vector concatenation. During training, we reduce the cross entropy of training samples against the labels.

$$\mathscr{L} = \sum_{(x_i, x_j)} \text{CrossEntropy}(\mathscr{P}(x_i \to x_j), L(x_i, x_j))$$

where $L(x_i, x_j)$ is the label of (x_i, x_j), either positive or negative, indicating whether these two nodes are related or not.

4.5 Experiments

We conduct experiments and ablation study of zero-shot relation extraction on the websites of different verticals from the SWDE dataset following the problem settings proposed in [68].

4.5.1 Datasets

Our experiments are conducted on the SWDE dataset [31]. As shown in Figure 4.2, the SWDE dataset includes websites of three different verticals, Movie, NBA, and University, and each vertical includes websites of the corresponding topic. For example, http://imdb.com and http://rottentomatoes.com are collected in the Movie vertical, and http://espn.go.com and http://nba.com are collected in the NBA vertical. Then the SWDE dataset collects web pages in each website and extracts their HTML source code for web mining tasks. Based on the original SWDE dataset, [67, 68] further annotates the related pairs in the web pages, and propose the

zero-shot relation extraction task in web mining. The statistics of the SWDE dataset is shown in Table 4.1, where we report the total number of websites in each vertical, the total number of web pages in each vertical, and the average number of annotated pairs in each web page.

Table 4.2. The experiment results of ReXMiner and baseline models. † The results of Colon Baseline and ZeroshotCeres (ZSCeres) are from [68]. ‡ We introduce the contrastive learning module of ReXMiner to the MarkupLM framework to solve the relation extraction task.

Model	Unseen Vertical									Average
	Movie			NBA			University			
	Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**F1**
Colon†	47	19	27	51	33	40	46	31	37	35
ZSCeres-FFNN†	42	38	40	44	46	45	50	45	48	44
ZSCeres-GNN†	43	42	42	48	49	48	49	45	47	46
MarkupLM‡	**48.93**	40.56	44.35	44.45	**71.35**	54.78	58.50	**62.37**	60.37	53.17
Ours	45.36	**49.36**	**47.28**	**65.86**	64.94	**65.40**	**68.43**	60.97	**64.48**	**59.05**

4.5.2 Experiment Setups

The zero-shot relation extraction task requires that the unseen web pages in the testing set and the existing web pages in the training set are of different verticals. Therefore, we follow the problem settings, and design three tasks based on the SWDE dataset, where we train our model on web pages from two of the three verticals and test our model on the third one. We denote the three tasks as,

- *Movie+NBA⇒Univ*: Train models with the Movie and NBA verticals, and test them on the University vertical;
- *NBA+Univ⇒Movie*: Train models with the NBA and University verticals, and test them on the Movie vertical;
- *Univ+Movie⇒NBA*: Train models with the University and Movie verticals, and test them on the NBA vertical.

We report the precision, recall, and F-1 score.

4.6 Implementation Details

We use the open-source Transformers framework from Huggingface [118] and build ReXMiner on the base of MarkupLM [55]. We initialize ReXMiner with the pre-trained weights of MarkupLM, initialize the extra modules with Xavier Initialization [26], and further finetune ReXMiner on the relation extraction tasks. We do not incorporate further pre-training on extra corpus. We use one NVIDIA A6000 to train the model with batch size of 16. We optimize the model with AdamW optimizer [69], and the learning rate is 2×10^{-5}. We set the number of popularity embeddings (τ) as 20, the number of attention layers with the common prefix length (α) as 12, the number of attention layers with the relative XML Path (β) as 3, the total number of samples (η) as 100, and the ratio between the positive and negative samples (μ) as $\frac{1}{5}$.

4.6.1 Compared Methods

We evaluate ReXMiner against several baselines.

Colon Baseline

The Colon Baseline is a heuristic method proposed in [68]. It identifies all text nodes ending with a colon (":") as the relation strings and extracts the closest text node to the right or below as the object. The Colon Baseline needs no training data, so it satisfies the requirement of the zero-shot relation extraction.

ZeroshotCeres

ZeroshotCeres [68] is a graph neural network-based approach that learns the rich representation for text nodes and predicts the relationships between them. It first extracts the visual features of text nodes from the coordinates and font sizes, and the textual features by inputting the text into a pre-trained BERT model [18]. Then the features are fed into a graph attention network (GAT) [101], where the graph is built based on the location of text nodes in the rendered web page to capture the layout relationships. The relation between text nodes is predicted as a binary classification on their feature concatenation.

MarkupLM

MarkupLM [55] is a pre-trained transformer framework that jointly models text and HTML/XML markup language in web pages. It embeds absolute XML Paths in the embedding layer of the BERT framework and proposes new pre-training tasks to learn the correlation between text and markup language. These tasks include matching the title with the web page, predicting the location of text nodes in the DOM Tree, and predicting the masked word in the input sequence. We use MarkupLM as a backbone model and append it with the contrastive learning module of ReXMiner to solve relation extraction task.

Table 4.3. The results of ablation study, where we compare ReXMiner with two ablation variants, ReXMiner w/o RelXPath and ReXMiner w/o RelXPath + PopEmb. PopEmb denotes the popularity embedding, and RelXPath denotes the relative XPath bias terms.

Model	Unseen Vertical									Average
	Movie			NBA			University			
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	F1
ReXMiner	45.36	49.36	47.28	65.86	64.94	65.40	68.43	60.97	64.48	59.05
- *w/o RelXPath*	45.60	45.68	45.64	47.13	73.54	57.44	54.82	74.63	63.21	55.43
- *w/o RelXPath + PopEmb*	48.93	40.56	44.35	44.45	71.35	54.78	58.50	62.37	60.37	53.17

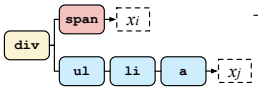
4.6.2 Experimental Results

We report the performance of ReXMiner in Table 4.2 and compare it with baseline models. From the result, we can see that our proposed model, ReXMiner, achieves the state-of-the-art performance in the zero-shot relation extraction task in all three verticals of the SWDE dataset. Specifically, ReXMiner surpasses the second-best model, MarkupLM, by 5.88 in the average F-1 score. In each task, we can observe a remarkable improvement of 2.93, 10.62 and 4.11 in F-1 score when the Movie, NBA, or University verticals are considered as the unseen vertical, respectively.

ZeroshotCeres is the previous state-of-art model proposed to solve the zero-shot relation extraction which leverages the graph neural network to model the structural information. We copy its performance from [68]. In the comparison with MarkupLM and ReXMiner, we observe

Table 4.4. The extraction results of the ablation models on `Quiz Show.html` in *NBA+Univ⇒Movie*. The green pairs denote the new true positive predictions compared with the previous ablation model, and the red pairs denote the missing true positive predictions compared with the previous ablation model.

NBA+Univ ⇒ Movie (Prediction result on *Quiz Show.html*)

Relative XML Path Pattern	Model	Extracted Pairs	
		Ture Positive	***False Positive***
RelXPath($x_i \Rightarrow x_j$) = [span; div; ul li a] 	**ReXMiner** ***(w/ RelXPath + PopEmb)***	(Color type, Technicolor prints); (Moods, Food for Thought); (Set In, 1958); (Genres, Drama); (Sound by, Dolby); ***(Produced by, Buena Vista);*** ~~***(From book, Remembering America);***~~ ***(Keywords, Advertising);*** ***(Types, Docudrama)***	(Director, Americana); (Types, Drama); (MPAA Rating, USA); (Keywords, Scandal)
	ReXMiner ***(w/o RelXPath, w/ PopEmb)***	(Color type, Technicolor prints); (Moods, Food for Thought); ***(Genres, Drama); (Sound by, Dolby); (From book, Remembering America);***	(Director, Americana); (Genres, Dolby); (Moods, Technicolor prints); (Types, Drama); (Tones, Technicolor prints)
	ReXMiner ***(w/o RelXPath + PopEmb)***	(Color type, Technicolor prints); (Moods, Food for Thought)	(Director, Drama); (Flags, Americana)

that directly modeling the XML Path information using Transformer framework achieves better performance, where MarkupLM and ReXMiner surpass ZeroshotCeres by 7.17 and 13.05 in average F-1 score. The multimodal attention mechanism with absolute XML Path embedding from MarkupLM enhance the performance in each task, and ReXMiner achieves the state-of-the-art overall performance after incorporating the relative XML Paths and the popularity of text nodes.

Though the performance of ReXMiner varies in different verticals, we can safely come to the conclusion that our proposed model, ReXMiner, is superior to the baselines in solving zero-shot relation extraction task in web mining. Further analysis is conducted in Ablation Study and Case Study to study the multimodal features.

4.6.3 Ablation Study

In the ablation study, we aim at studying the role of multimodal features proposed in ReXMiner, including the Relative XML Path Attention and the Popularity Embedding. We introduce three ablation versions of ReXMiner by removing certain features in Table 4.3. From Table 4.3, we compare the performance of all ablation models. We find that using the

Popularity Embedding enhances F-1 score by 2.84 and 2.66 in *Movie+NBA⇒Univ* task and *Univ+Movie⇒NBA* task, respectively. After incorporating the Relative XML Path Attention, the F-1 score are further improved in all three tasks. Thus, the ablation model with all multimodal features achieve the highest F-1 score. We conclude that the Relative XML Path contributes to the high precision while the popularity embedding enhances recall leading to the best performance in F-1 score.

Table 4.5. The experiment results of ReXMiner and baseline models. † The results of ZeroshotCeres (ZSCeres) are from [68]. ‡ We introduce the contrastive learning module of ReXMiner to the MarkupLM framework to solve the relation extraction task.

Model	Movie			NBA			University		
	Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
ZSCeres-FFNN†	37	50	45	35	49	41	47	59	52
ZSCeres-GNN†	49	51	50	47	39	42	50	49	50
MarkupLM‡	**55.98**	71.30	62.72	46.49	**73.66**	57.00	67.09	**70.56**	68.78
Ours	55.15	**79.50**	**65.12**	**68.06**	62.56	**65.19**	**73.08**	68.73	**70.84**

4.7 Zero-shot Relation Extraction on Unseen Websites

In this paper, we propose ReXMiner to solve the zero-shot relation extraction task where web pages in the training set and the testing set are from different verticals. Here, we report the results of the additional experiments for the zero-shot relation extraction on unseen websites. To be more specific, in these additional experiments, the web pages in the training set and the testing set are from the same vertical but different websites. For each vertical in the SWDE dataset, we select a subset of websites as the testing set and train the model with the rest websites. We select “rottentomatoes" and “yahoo" from Movie Vertical, “yahoo” from NBA Vertical, and “ecampustours" and “usnews" from University Vertical. We report the results in Table 4.5.

4.7.1 Case Study

In Table 4.4, we show the extraction results of the ablation models on a web page, `Quiz Show.html`, in *NBA+Univ⇒Movie*. We select one relative XML Path pattern as an example, `[span;div;ul,li,a]`, and list the corresponding extracted pairs into two groups, true positive extractions and false positive extractions. From the results, we can see that ReXMiner with all proposed features shows the best performance, which is also demonstrated in the ablation study. Specifically, by incorporating the Popularity Embedding, ReXMiner (w/o RelXPath, w/ PopEmb) depends on the frequency when predicting the related pairs so it intends to extract more text node pairs and contributes to a higher recall. After adding the Relative XML Path Attention, the extracted pairs are further filtered by the relative XML Path patterns in ReXMiner (w/ RelXPath + PopEmb) so it can extract similar number of true positive pairs and largely reduce the number of false positive cases, but it leads to the missing extraction of *(From book, Remembering America)*.

4.8 Conclusion and Future Work

In this paper, we present ReXMiner, a web mining model to solve the zero-shot relation extraction task from semi-structured web pages. It benefits from the proposed features, the relative XML Paths extracted from the DOM Tree and the popularity of text nodes among web pages from the same website. Specifically, based on MarkupLM, and we further incorporate the relative XML Paths into the attention layers of Transformer framework as bias terms and embed the popularity of text nodes in the embedding layer. To solve the relation extraction task, we append the backbone model with the contrastive learning module and use the negative sampling to solve the sparsity issue of the annotation. In this way, ReXMiner can transfer the knowledge learned from the existing web pages to the unseen ones and extract the related pairs from the unseen web pages. Experiments demonstrate that our method can achieve the state-of-the-art performance compared with the strong baselines.

For future work, we plan to explore the new problem settings with limited supervision, such as few-shot learning and distant supervision, and further study the topological structure information in the DOM Tree to explore more meaningful signals in understanding the semi-structured web pages in web mining tasks.

4.9 Limitations

We build ReXMiner based on MarkupLM and incorporate new features, including the relative XML Paths, the popularity of text nodes, and the contrastive learning. After initializing our model with the pre-trained weights of MarkupLM, the additional modules are finetuned on the datasets of downstream tasks without large-scale pre-training, due to the limited computing resource. We believe more promising results can be achieved if it is possible to pre-train our proposed framework enabling all parameters to be well converged.

4.10 Acknowledgments

Chapter 4, in full, is a reprint of the material as it appears in The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. Zilong Wang and Jingbo Shang [109]. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding

5.1 Introduction

Tables are a popular data format and widely used in daily life [9]. Understanding tabular data with language models can benefit various downstream tasks, such as table-based fact verification [14], and table-based question answering [47]. Distinct from pure text, tables deliver rich information through the interaction between rows and columns in the tabular structure, which enhances the data capacity but also increases the difficulty for language models to understand them. Thus, reasoning over the tabular data is an important direction in natural language processing and attracts increasing attention from both academia and industry.

In recent years, several approaches have been suggested to tackle the problem of table understanding by *training* language models. One common direction is to add specialized embedding layers or attention mechanisms into language models and pre-train the models by recovering table cells or segments [35, 104, 28, 1]. In this way, the pre-trained models are aware of the tabular structure. Another direction is to synthesize SQL query-response pairs and pre-train an encoder-decoder model as a neural SQL executor [21, 63, 46].

Recently, large language models (LLMs) achieve outstanding performance across diverse tasks solely by *prompting*, thanks to the massive scale of pre-training [8, 50]. As series of works on prompting techniques have further improved the reliability of LLMs by designing reasoning

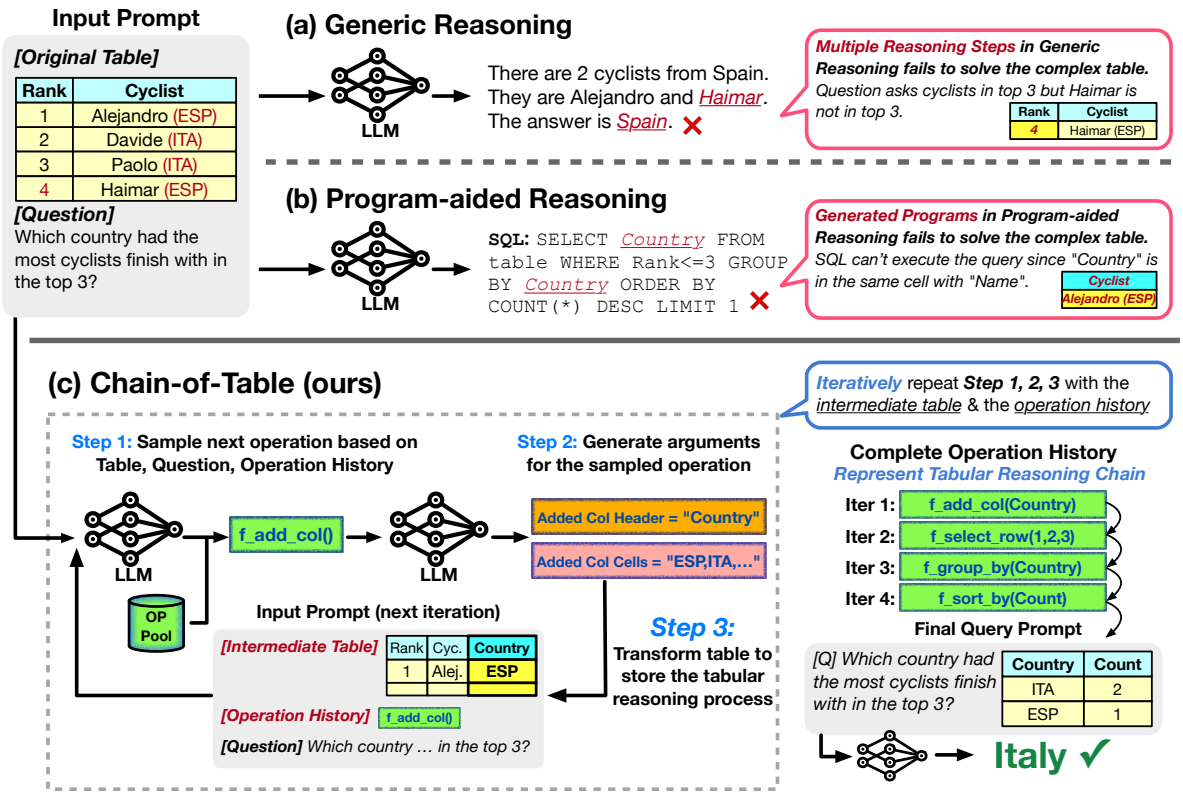


Figure 5.1. Illustration of the comparison between (a) generic reasoning, (b) program-aided reasoning, and (c) the proposed CHAIN-OF-TABLE. Given a complex table where a cyclist's nationality and name are in the same cell, (a) is unable to provide the correct answer through multi-step reasoning due to the complexity; (b) generates and executes programs (e.g. SQL queries) to deliver the answer, but it also falls short in accurately parsing the name and nationality in the table. In contrast, (c) CHAIN-OF-TABLE iteratively samples a chain of operations that effectively transform the complex table into a version specifically tailored to the question. With the assistance of CHAIN-OF-TABLE, the LLM can arrive at the correct answer.

chains, such as Chain-of-Thought [116], Least-to-Most [130], Program-of-Thought [13] and Tree-of-Thought [124]. Different works have also explored the possibility of using LLMs to solve table-based problems [12, 15, 125]. However, these approaches [38] often represent reasoning steps in free-form text or code, which are not ideally suited for addressing scenarios involving complex tables, as shown in Figure 5.1(a) and Figure 5.1(b).

On the other hand, inference on tables typically involves a series of intermediate reasoning steps and each of them aligns with specific tabular operations. We propose CHAIN-OF-TABLE, where we conduct step-by-step reasoning as step-by-step tabular operations to form a *chain* of

tables. The tables in the chain are the transformed tables by the tabular operations, representing the intermediate reasoning results. This procedure resembles the *thought* of reasoning in Chain-of-Thought [116]. Specifically, we define a set of table operations, such as adding columns, selecting rows, grouping, and more, which are commonly-used in SQL and DataFrame development [88, 94, 48]. We then prompt LLMs to conduct step-by-step reasoning. In each step, the LLM dynamically generates an operation as the next step along with its required arguments, and then we execute the operation on the table programmatically. This operation can either enrich the table by adding detailed intermediate results or condense it by removing irrelevant information. Intuitively, visualizing the intermediate results is essential for reaching correct predictions. We feed the transformed table back for the next step. This iterative process continues until an ending state is achieved. We argue that the tables obtained during the reasoning steps are better structured representations of the intermediate thoughts than free-form text. Finally, the CHAIN-OF-TABLE reasoning results in tables from which it is easier for LLMs to derive a final answer to the question.

We validate CHAIN-OF-TABLE with three tabular benchmarks to evaluate table-based reasoning: WikiTQ [82], TabFact [14], and FeTaQA [75]. We conduct our experiments using PaLM 2 [2] and GPT-3.5 [8, 79] to demonstrate that our proposed method CHAIN-OF-TABLE is able to generalize to various LLM options. We summarize our contribution as follows:

- We extend the concept of Chain-of-Thought to the tabular setting, where we transform the input table to store intermediate results. This multi-step tabular reasoning approach with table evolution leads to more accurate table understanding.
- Extensive experiments on table-based fact verification and question answering show that CHAIN-OF-TABLE archives state-of-the-art performance in WikiTQ, TabFact, and FeTaQA datasets.

5.2 Related Work

Fine-tuning Language Model for Table Understanding

Tables are effective in organizing, storing, and analyzing information. Efforts have been made to fine-tune language models (LMs) to tackle table understanding tasks. Following the successful mask language modeling (MLM) proposed in BERT [18], TaPas [35] adopts this approach and asks the model to reconstruct certain cells in the table during pre-training. Pasta [28] and TUTA [104] further propose to mask the entire columns or segments in the table. On the other hand, TAPEX [63] pre-trains an encoder-decoder model with a large synthetic SQL dataset so that it can perform as a SQL executor to better understand the tabular structure. [21] and [46] also leverage synthesized SQL with additional consideration of the alignment between SQL and natural language questions by pre-training the model with both natural and synthetic data.

Prompting Language Model for Table Understanding

LLMs can learn from a few samples as prompts through in-context learning. This strategy is widely used to give models additional instructions to better solve downstream tasks. Chain-of-Thought (CoT) [116] proposes to generate reasoning steps before answering instead of directly generating an end-to-end answer. Following CoT, Least-to-Most [130] and DecomP [49] propose to break down the question into subproblems in the reasoning chain. During reasoning, the latter steps are aware of the previous ones. Such iterative chains with task decomposition further improve the results on complex problems by leveraging the intermediate results from solving subproblems. [132] enhances CoT through a table-filling procedure, with a primary focus on text-based tasks where the input and output are in textual format. However, the line of works following CoT is not specifically designed for tabular data. As reported in [12], large language models with these generic reasoning methods can achieve decent results, but there are still gaps between these methods and those specialized for table scenarios [15, 125]. We propose CHAIN-OF-TABLE to fill the gap by directly incorporating intermediate tables from

tabular operations as a proxy of intermediate thoughts.

To better solve table-based tasks with LLMs, researchers go beyond general text and resort to using external tools. [13, 24] propose solving reasoning tasks by generating Python programs, which are then executed using the Python interpreter. This approach greatly improves the performance of arithmetic reasoning. In the scenario of table understanding, Text-to-SQL with LLMs [92] is a straightforward application of this idea. To further push the limits of programs, Binder [15] generates SQL or Python programs and extends their capabilities by calling LLMs as APIs in the programs. LEVER [78] also proposes solving the table-based tasks with programs but with the additional step of verifying the generated programs with their execution results. However, the assistant programs in these program-aided methods still fall short in solving difficult cases that involve complex tables. These limitations are primarily due to the constraints of the *single-pass* generation process, where the LLMs lack the capability to modify the table in response to a specific question, requiring them to perform reasoning over a static table. Our method, on the contrary, is a *multi-step* reasoning framework that conducts tabular reasoning step by step. It transforms the tables tailored to the given question.

To the best of our knowledge, Dater [125] is the only model that modifies the tabular context while solving table-based tasks. However, the table decomposition in Dater is motivated by the idea that tables could be too large for LLMs to conduct reasoning. It is, therefore, more similar to an LLM-aided data pre-processing than to a part of the reasoning chain since the tabular operations are limited to column and row selections, and fixed for all tables and questions. In contrast, our CHAIN-OF-TABLE generalizes a larger set of generic table operations and *dynamically* generates reasoning chains in an adaptive way based on the inputs, leveraging the planning ability [100, 32] of LLMs.

5.3 CHAIN-OF-TABLE Reasoning

Problem Formulation.

In table-based reasoning, each entry can be represented as a triplet (T, Q, A), where T stands for the table, Q represents a question or statement related to the table, and A is the expected answer. Particularly, in the table-based question answering task, Q and A are the question and expected answer in natural language form; in the table-based fact verification task, Q is a statement about the table contents and $A \in \{\texttt{True}, \texttt{False}\}$ is a Boolean value that indicates the statement's correctness. The objective is to predict the answer A given the question Q and the table T. To facilitate table-based reasoning within the same paradigm employed for generic reasoning, we convert all data values, including tables, into textual representations (see Appendix A.4 for the tabular format encoding method).

5.3.1 Overview

CHAIN-OF-TABLE enables LLMs to *dynamically plan* a chain of operations over a table T in response to a given question Q. It utilizes atomic tool-based operations to construct the table chain. These operations include adding columns, selecting rows or columns, grouping, and sorting, which are common in SQL and DataFrame development (see Appendix A.1 for more details).

Previously, Dater [125] employs a dedicated yet fixed procedure for decomposing tables and questions, which limits its compatibility with new operations. Also, Binder [15], while potentially compatible with new operations, is restricted to those that work with code interpreters such as SQL or Python. In contrast, our framework is extendable and can incorporate operations from a wide range of tools thanks to the flexible in-context learning capability to sample and execute effective operations.

As illustrated in Algorithm 2, at each iteration, we prompt the LLM to sample one of the pre-defined atomic operations denoted as `f` using the corresponding question Q, the latest

table state T, and the operation chain `chain` (Line 4). Then, we query the LLM to generate the required arguments `args` for `f` (Line 5) and execute it to transform the table T (Line 6). We keep track of the operation `f` performed on the table in the operation chain `chain` (Line 7). The process finishes when the ending tag `[E]` is generated (Line 8). Finally, we feed the latest table into the LLM to predict the answer (Line 9). This series of operations serves as the reasoning steps leading LLMs to understand the input table and better generate the final answer.

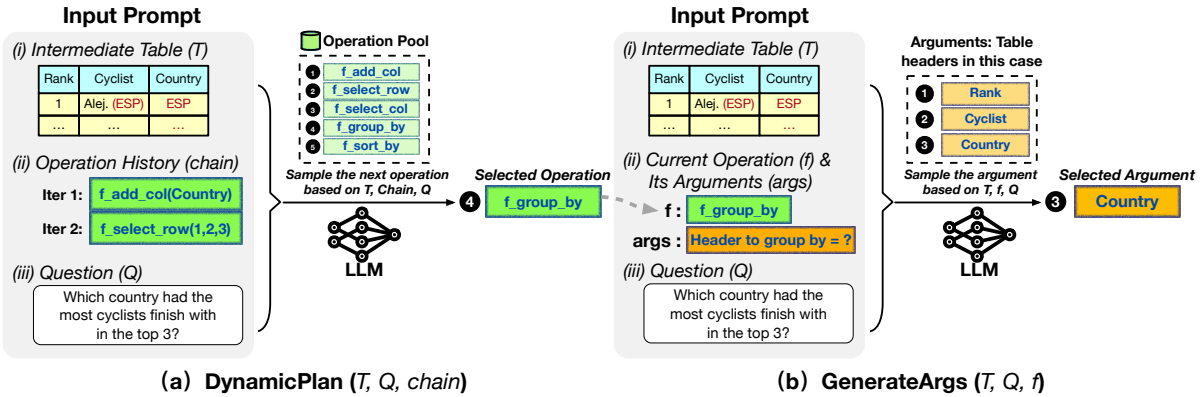
Algorithm 2: CHAIN-OF-TABLE Prompting

Data: (T, Q) is a table-question pair.
Result: $\hat{A}$ is the predicted answer to the question.

```
Function Chain-of-Table(T, Q):
    chain ← [([B],ϕ),]          ▷ Initialize the operation chain chain with [B] and ϕ, where [B] is
                                ▷ the beginning tag, and ϕ means it requires no arguments
    repeat
        f ← DynamicPlan(T,Q,chain)   ▷ Generate next operation f based on the table, the question, and
                                     ▷ the current operation chain
        args ← GenerateArgs(T,Q,f)   ▷ Generate the arguments args for the next operation
        T ← f(T,args)                ▷ Perform the next operation on the table to obtain updated T
        chain ← chain.append((f,args))   ▷ Keep track of the operations in the operation chain chain
    until f = [E]                ▷ Iteratively update the table until the ending tag [E] is generated
    Â ← Query(T,Q)               ▷ Query the LLM with the resulting table to get the final answer Â
return Â
```

5.3.2 Dynamic Planning

CHAIN-OF-TABLE instructs the LLM to dynamically plan the next operation by in-context learning. As shown in Figure 5.2(a), `DynamicPlan` involves three components: the most recent intermediate table T (Figure 5.2(a)(i)), the history of the previous operations chain `chain` (Figure 5.2(a)(ii)), and the question Q (Figure 5.2(a)(iii)). We guide the LLM to select the subsequent operation `f` from the operation pool given (T, `chain`, Q). The LLM is then able to dynamically plan the next operation and build a tabular reasoning chain step by step. See Appendix A.5.1 for detailed prompts.



(a) DynamicPlan (*T, Q, chain*) **(b) GenerateArgs (*T, Q, f*)**

Figure 5.2. Illustration of the main components `DynamicPlan`$(T, Q,$ `chain`$)$ and `GenerateArgs`$(T, Q,$ `f`$)$ in the proposed CHAIN-OF-TABLE, where T is a intermediate table; Q is the question; `chain` is a list of operations already performed on the table; `f` is the operation selected by `DynamicPlan`. **Left:** `DynamicPlan` samples the next operation from the operation pool, according to (T, `chain`, Q). **Right:** `GenerateArgs` takes the selected operation `f` as input and generates its arguments based on (T, `f`, Q). The operations, along with their arguments, act as a proxy of the tabular reasoning process to effectively tackle table understanding tasks.

5.3.3 Argument Generation

The next step, `GenerateArgs`, involves generating arguments for the selected table operation `f` sampled by `DynamicPlan`, as depicted in Figure 5.2. `GenerateArgs` involves three key components: the most recent intermediate table T (Figure 5.2(b)(i)), the selected operation `f` along with its arguments `args` (Figure 5.2(b)(ii)), and the question (Figure 5.2(b)(iii)). We employ simple regular expressions to account for varying number of arguments required by different operations (see Appendix A.5.2 for more details). Finally, we apply programming languages to execute the operation and create the corresponding intermediate tables.

5.3.4 Final Query

We transform the table through dynamic planning (Section 5.3.2) and argument generation (Section 5.3.3). During this process, we create a chain of operations that acts as a proxy for the tabular reasoning steps. These operations generate intermediate tables that store and present the results of each step to the LLM. Consequently, the output table from this chain of operations

contains comprehensive information about the intermediate phases of tabular reasoning. We then employ this output table in formulating the final query. As illustrated in Figure 5.1 (bottom right), we input both the output table and the question into the LLM, which provides the final answer to the question (see Line 9 in Algorithm 2).

5.4 Experiments

We evaluate the proposed CHAIN-OF-TABLE on three public table understanding benchmarks: WikiTQ [82], FeTaQA [75], and TabFact [14]. WikiTQ and FeTaQA are datasets focused on table-based question answering. They require complex tabular reasoning over the provided table to answer questions. WikiTQ typically requires short text span answers, whereas FeTaQA demands longer, free-form responses. TabFact, on the other hand, is a table-based binary fact verification benchmark. The task is to ascertain the truthfulness of a given statement based on the table. For WikiTQ evaluation, we use the official denotation accuracy [82], and for TabFact, we employ the binary classification accuracy. Given the nature of FeTaQA, which involves comparing predictions with longer target texts, we utilize BLEU [80], ROUGE-1, ROUGE-2, and ROUGE-L [59] for assessment. In our experiments, we use PaLM 2-S[1], GPT 3.5 (turbo-16k-0613)[2] as the backbone LLMs. We incorporate few-shot demo samples from the training set into the prompts to perform in-context learning. Examples of these prompts can be found in Appendix A.5. Details regarding the LLM inference parameters and the number of demonstration samples used are provided in Appendix A.3.

5.4.1 Baselines

The baseline methods are categorized into two groups: (a) generic reasoning, which includes End-to-End QA, Few-Shot QA, Chain-of-Thought [116]; and (b) program-aided reasoning, which includes Text-to-SQL [92], Binder [15], Dater [125]). Detailed descriptions of

[1] https://cloud.google.com/vertex-ai/docs/generative-ai/learn/generative-ai-studio
[2] http://openai.com/api/

Table 5.1. Table understanding results on WikiTQ and TabFact with PaLM 2 and GPT 3.5. (underline denotes the second-best performance; **bold** denotes the best performance; the improvement is measured against the second-best performing method.)

Prompting	PaLM 2		GPT 3.5	
	TabFact	**WikiTQ**	**TabFact**	**WikiTQ**
Generic Reasoning				
End-to-End QA	77.92	60.59	70.45	51.84
Few-Shot QA	78.06	60.33	71.54	52.56
Chain-of-Thought [116]	79.05	60.43	65.37	53.48
Program-aided Reasoning				
Text-to-SQL [92]	68.37	52.42	64.71	52.90
Binder [15]	76.98	54.88	79.17	56.74
Dater [125]	84.63	61.48	78.01	52.81
CHAIN-OF-TABLE (ours)	**86.61** (+1.98)	**67.31** (+5.83)	**80.20** (+1.03)	**59.94** (+3.20)

these baseline methods are provided below.

Generic Reasoning

End-to-End QA guides the LLM to directly produce the answer when provided with a table and a question as input prompts. Few-Shot QA operates similarly, but it includes few-shot examples of (Table, Question, Answer) triplets in the prompt, as detailed in [8]. We select these examples from the training set, and the model also outputs the answer directly. Chain-of-Thought [116] prompts the LLM to articulate its reasoning process in text format before delivering the question. See Appendix A.6 for the prompts of baselines.

Program-aided Reasoning

Text-to-SQL [92] utilizes in-context samples to guide LLMs in generating SQL queries for answering questions. This approach follows the concepts introduced by [13, 24]. Binder [15] integrates a language model API with programming languages such as SQL or Python. This integration prompts the LLM to produce executable programs that perform table reasoning tasks on the given table and question. Dater [125] employs few-shot samples for efficient deconstruction of table contexts and questions, enhancing end-to-end table reasoning with

Table 5.2. Distribution of the number of samples v.s. the required length of operation chain in CHAIN-OF-TABLE with PaLM 2 on WikiTQ and TabFact datasets. We observe that the majority of samples need 2 to 4 operations to generate the final output.

Dataset	Length of operation chain				
	1	2	3	4	5
WikiTQ	95	1308	1481	1084	341
TabFact	4	547	732	517	223

decomposed sub-tables and sub-questions.

5.4.2 Results

We compare CHAIN-OF-TABLE with generic reasoning methods and program-aided reasoning methods on three datasets: WikiTQ, TabFact, and FeTaQA. The results on WikiTQ and TabFact are presented in Table 5.1. We have additional results on FeTaQA in Appendix A.2. We follow the previous works and report the performance using the official evaluation pipeline[3].

Table 5.1 shows that CHAIN-OF-TABLE significantly outperforms all generic reasoning methods and program-aided reasoning methods on TabFact and WikiTQ across PaLM 2 and GPT 3.5. This is attributed to the dynamically sampled operations and the informative intermediate tables in CHAIN-OF-TABLE. CHAIN-OF-TABLE iteratively generates operations that act as proxies for tabular reasoning steps. These operations produce and present tailored intermediate tables to the LLM, conveying essential intermediate thoughts (see the example in Figure 5.4). With the support of CHAIN-OF-TABLE, the LLM can reliably reach the correct answer.

From the results, we observe a performance decrease on WikiTQ due to the complexity of tabular structure when vanilla Chain-of-Thought is introduced to End-to-End QA using PaLM 2. In contrast, our proposed CHAIN-OF-TABLE consistently enhances End-to-End QA performance by 8.69% on TabFact and 6.72% on WikiTQ with PaLM 2.

[3]Dater [125] with OpenAI Codex LLM achieves 65.9% and 85.6% accuracy on WikiTQ and TabFact, respectively. It also achieves 27.96 in BLEU, 0.62 in ROUGE-1, 0.40 in ROUGE-2, and 0.52 in ROUGE-L on FeTaQA. However, because Codex is no longer publicly available, we do not compare CHAIN-OF-TABLE with Dater with Codex.

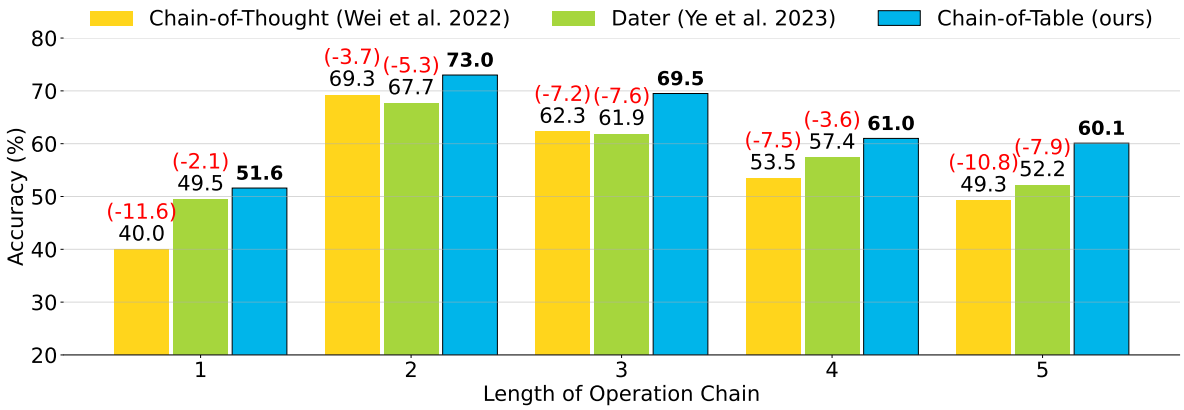


Figure 5.3. Performance of Chain-of-Thought, Dater, and the proposed CHAIN-OF-TABLE on WikiTQ for questions that require an operation chain of varying lengths. Our proposed atomic operations allow our proposed method CHAIN-OF-TABLE to dynamically transform the input table through multiple reasoning iterations. This significantly improves performance over generic and program-aided reasoning counterparts.

5.4.3 Performance Analysis under Different Operation Chain Lengths

In CHAIN-OF-TABLE, the selection of each operation is dynamically determined based on the difficulty and complexity of the questions and their corresponding tables. Therefore, we conduct a detailed study on the performance under different numbers of operations by categorizing the test samples according to their operation lengths. We report the distribution of the number of samples v.s. the required length of operation chain in Table 5.2. This analysis focuses on samples that require operations in the reasoning process. We use the results with PaLM 2 as an example. Our observations reveal that the majority of samples require 2 to 4 operations to generate the final output.

For each chain length, we further compare CHAIN-OF-TABLE with Chain-of-Thought and Dater, as representative generic and program-aided reasoning methods, respectively. We illustrate this using results from PaLM 2 on WikiTQ. We plot the accuracy of all methods using bar charts in Figure 5.3, highlighting the gap between the compared methods and our method. Notably, CHAIN-OF-TABLE consistently surpasses both baseline methods across all operation chain lengths, with a significant margin up to 11.6% compared with Chain-of-Thought, and up

Table 5.3. Performance of Binder, Dater, and the proposed CHAIN-OF-TABLE on small (<2000 tokens), medium (2000 to 4000 tokens), large (>4000 tokens) tables from WikiTQ. We observe that the performance decreases with larger input tables while CHAIN-OF-TABLE diminishes gracefully, achieving significant improvements over competing methods. (underline denotes the second-best performance; **bold** denotes the best performance; the improvement is measured against the second-best performing method.)

Prompting	Table Size		
	Small (<2k)	**Medium** (2k∼4k)	**Large** (>4k)
Binder [15]	56.54	26.13	6.41
Dater [125]	62.50	42.34	34.62
CHAIN-OF-TABLE (ours)	**68.13** (+5.63)	**52.25** (+9.91)	**44.87** (+10.25)

to 7.9% compared with Dater.

Generally, the performance of these methods decreases as the number of tabular operations required in the tabular reasoning chain increases due to higher difficulty and complexity of questions and tables. Nevertheless, our proposed CHAIN-OF-TABLE declines gracefully compared to other baseline methods. For example, CHAIN-OF-TABLE exhibits only a minimal decrease in performance when the number of operations increases from four to five.

5.4.4 Performance Analysis under Different Table Sizes

Large tables present significant challenges to LLMs since LLMs often struggle to interpret and integrate contexts in long input prompts [61, 125]. To assess the performance on tables of various sizes, we categorize the input tables from WikiTQ into 3 groups based on token count: small (<2000 tokens), medium (2000 to 4000 tokens) and large (>4000 tokens). We then compare CHAIN-OF-TABLE with Dater [125] and Binder [15], the two latest and strongest baselines, as representative methods. Detailed results are presented in Table 5.3.

As anticipated, the performance decreases with larger input tables, as models are required to process and reason through longer contexts. Nevertheless, the performance of the proposed CHAIN-OF-TABLE diminishes gracefully, achieving a significant 10+% improvement over the second best competing method when dealing with large tables. This demonstrates the efficacy of the reasoning chain in handling long tabular inputs.

Table 5.4. Number of samples generated for a single question in Binder, Dater, and the proposed CHAIN-OF-TABLE on the WikiTQ dataset. Notably, CHAIN-OF-TABLE generates the fewest samples among the baselines – 50% less than Binder and 75% less than Dater. For a detailed description of the steps involved in Binder and Dater, please refer to the corresponding papers.

Prompting	**Total # of generated samples**	**# of generated samples in each steps**
Binder [15]	50	Generate Neural-SQL: 50
Dater [125]	100	Decompose Table: 40; Generate Cloze: 20; Generate SQL: 20; Query: 20
CHAIN-OF-TABLE (ours)	$\leq$**25**	DynamicPlan: $\leq$5; GenerateArgs: $\leq$19; Query: 1

5.4.5 Efficiency Analysis of CHAIN-OF-TABLE

We analyze the efficiency of CHAIN-OF-TABLE by evaluating the number of required generated samples. We compare CHAIN-OF-TABLE with Binder [15] and Dater [125], the two latest and most competitive baseline method. The analysis results on WikiTQ are presented in Table 5.4. Binder generates Neural-SQL queries, requiring 50 samples for self-consistent results. Dater involves multiple delicate yet fixed steps, such as decomposing the tables and generating cloze queries for the questions. In each step, Dater also employs self-consistency to improve accuracy of the LLM outputs, leading to a high number of required generated samples. For a detailed description of these frameworks, please refer to the corresponding papers, [125] and [15].

Unlike these previous methods, our proposed CHAIN-OF-TABLE employs a greedy search strategy in its tabular reasoning process, instead of relying on self-consistency sampling for boosting performance. This approach results in a reduced query count for our method, despite CHAIN-OF-TABLE adopting an iterative reasoning process. To be more specific, we observe that the number of queries needed by CHAIN-OF-TABLE is the lowest among the most recent baselines – 50% less than Binder and 75% less than Dater. We attribute the query efficiency of our method to the proposed dynamic operation execution through the tabular reasoning. The model is able to find an effective reasoning process that reaches the final output quicker and

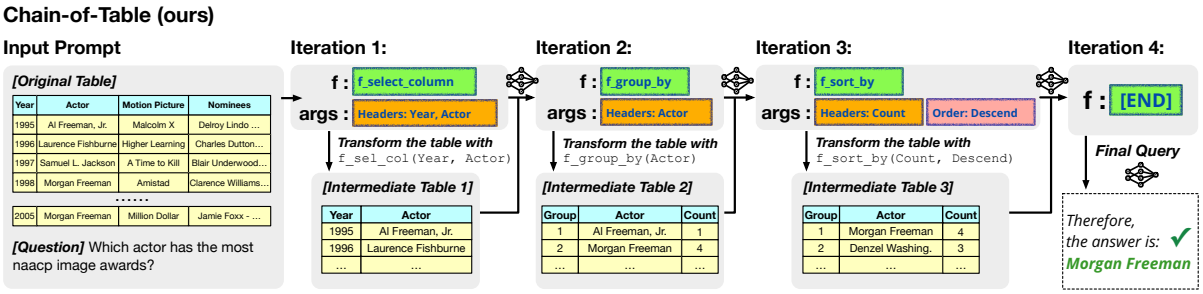


Figure 5.4. Illustration of the tabular reasoning process in CHAIN-OF-TABLE. This iterative process involves dynamically planning an operation chain and accurately storing intermediate results in the transformed tables. These intermediate tables serve as tabular thought process that can guide the LLM to land to the correct answer more reliably.

more reliably.

5.4.6 Case Study

In Figure 5.4, we illustrate the tabular reasoning process by CHAIN-OF-TABLE. The question is based on a complex table and requires multiple reasoning steps to 1) identify the relevant columns, 2) conduct aggregation, and 3) reorder the aggregated intermediate information. Our proposed CHAIN-OF-TABLE involves dynamically planning an operation chain and accurately storing intermediate results in the transformed tables. These intermediate tables serve as tabular thought process that can guide the LLM to land to the correct answer more reliably.

5.5 Conclusion

Our proposed CHAIN-OF-TABLE enhances the reasoning capability of LLMs by leveraging the tabular structure to express intermediate thoughts for table-based reasoning. It instructs LLMs to dynamically plan an operation chain according to the input table and its associated question. This evolving table design sheds new light on the understanding of prompting LLMs for table understanding.

5.6 Reproducibility Statement

We include the prompt examples of `DynamicPlan(`T`,`Q`,chain)` in Appendix A.5.1, the demo examples of `GenerateArgs(`T`,`Q`,f)` in Appendix A.5.2, the prompt examples of `Query(`T`,`Q`)` in Appendix A.5.3. We run the generic reasoning methods (End-to-End QA, FewShot QA, Chain-of-Thought) using the prompts reported in Appendix A.6. We run Text-to-SQL and Binder using the official open-sourced code and prompts in https://github.com/HKUNLP/Binder. We run Dater using the official open-sourced code and prompts in https://github.com/AlibabaResearch/DAMO-ConvAI. We revise the code to use publicly available GPT 3.5 and PaLM 2 (Section 5.4) as the LLM backbone instead of the OpenAI Codex due to its inaccessibility.

5.7 Acknowledgments

Chapter 5, in full, is a reprint of the material as it appears in The Twelfth International Conference on Learning Representations, 2024. Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister [114]. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Conclusion and Future Directions

6.1 Summary

In this dissertation, we have explored the intersection of **language models and structured knowledge**, addressing the challenges associated with processing **semi-structured data** such as document images, web pages, and tabular data. Traditional language models, primarily designed for free-form text, struggle with the spatial, hierarchical, and relational complexities inherent in structured data. To bridge this gap, we introduced four key contributions that enhance language models' capabilities in **extraction, representation, and reasoning** over structured knowledge.

First, we introduced **VRDU**, a benchmark for visually-rich document understanding, highlighting the challenges that multimodal models face when extracting structured information from complex document layouts. This benchmark provides a robust evaluation framework for testing real-world generalization in document extraction tasks.

Second, we proposed **LASER**, a label-aware sequence-to-sequence framework for few-shot entity recognition in document images. By leveraging label semantics and spatial structure, LASER demonstrated improved generalization with minimal supervision, outperforming traditional sequence-labeling approaches in low-resource settings.

Third, we developed **ReXMiner**, a multimodal approach for zero-shot relation extraction in web mining. By encoding structural relationships through relative XML paths in the Document Object Model (DOM) tree, ReXMiner significantly improved the ability to extract structured

knowledge from semi-structured web pages, even under unseen templates.

Finally, we introduced **CHAIN-OF-TABLE**, a novel framework for iterative table-based reasoning. Unlike previous approaches that treat tables as static inputs, CHAIN-OF-TABLE dynamically evolves tabular data by applying structured transformations, enabling step-by-step reasoning and leading to state-of-the-art performance on multiple table-based question-answering and fact-verification benchmarks.

Together, these contributions advance the field of **structured knowledge integration in language models**, offering new methodologies and benchmarks that push the boundaries of document understanding, web mining, and table reasoning.

6.2 Future Directions

While this dissertation presents significant advancements in bridging structured knowledge and language models, several open challenges remain, offering opportunities for future research:

- **Generalization to Unseen Structures:** Despite improvements in template adaptation, models still struggle to generalize across highly diverse document layouts, web structures, and table formats. Future work could explore self-supervised or meta-learning techniques to enhance adaptability.
- **Multimodal Integration Beyond Text:** While current models leverage textual, spatial, and structural features, incorporating richer modalities such as images, charts, and graphs could further improve structured knowledge understanding.
- **Few-Shot and Zero-Shot Adaptation:** Developing more efficient few-shot and zero-shot learning techniques could enhance models' ability to perform robust structured knowledge extraction with minimal labeled data.
- **Interpretable Structured Reasoning:** As language models become increasingly powerful,

improving their interpretability in structured reasoning tasks—especially in domains requiring transparency, such as finance and healthcare—remains an important research direction.

- **Efficient and Scalable Architectures:** Processing structured data at scale remains computationally expensive. Future research could focus on more lightweight and scalable architectures that maintain high accuracy while reducing computational overhead.

- **Bridging Structured and Unstructured Knowledge:** While this dissertation focuses on structured data, real-world applications often require models to combine structured and unstructured sources. Developing hybrid models that can reason jointly over structured knowledge and free-form text could unlock new possibilities in knowledge-intensive applications.

6.3 Final Remarks

The integration of structured knowledge into language models is a rapidly evolving field with profound implications for information extraction, reasoning, and decision-making. By addressing key challenges in structured data processing, this dissertation contributes to bridging the gap between language models and real-world structured knowledge. We hope that the methodologies and insights presented here will inspire further research and innovation in this exciting area.

Appendix A

Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding

A.1 Atomic Operations in CHAIN-OF-TABLE

A.1.1 Introduction

In this study, we adopt a set of five table operations, which are commonly-used in SQL and DataFrame development, as an example. We note that our framework can trivially accommodate additional operations, which we leave for future work.

- `f_add_column()` adds a new column to the table to store intermediate reasoning or computational results.
- `f_select_row()` selects a subset of rows that are relevant to the question. Tables may contain irrelevant information for the given question [125]. This operation helps locate the necessary context.
- `f_select_column()` selects a subset of columns. A column usually corresponds to an attribute in the table. This operation allows the model to locate the necessary attributes to answer the question.
- `f_group_by()` groups the rows by the contents of a specific column and provides the count of each enumeration value in that column. Many table-based questions or statements

involve counting, but LLMs are not proficient at this task [45].

- `f_sort_by()` sorts the rows based on the contents of a specific column. When dealing with questions or statements involving comparison or extremes, LLMs can utilize this operation to rearrange the rows. The relationship can be readily inferred from the order of the sorted rows.

A.1.2 Ablation Study

To demonstrate the effectiveness of our proposed atomic operations, we perform an ablation study by creating five leave-one-out variants of our method, each of which removes one of the pre-defined operations from the pre-defined operation pool. For example, *w/o* `f_add_column()` means `f_add_column()` is removed from the operation pool. As a result, the LLM is only able to plan from the remaining four operations (`f_select_column`, `f_select_row`, `f_group_by`, and `f_sort_by`) to construct operation chains. We report the results of the ablation study in Table A.1.

Table A.1. Ablation study of the atomic operations used in CHAIN-OF-TABLE with PaLM 2 on WikiTQ and TabFact datasets. We observe that row selection and group-by operations have the biggest impact on the final table understanding performance.

Prompting	TabFact	WikiTQ
	Accuracy	Accuracy
CHAIN-OF-TABLE	**86.61**	**67.31**
w/o `f_add_column()`	85.23 (-1.38)	65.88 (-1.43)
w/o `f_select_column()`	82.61 (-4.00)	65.68 (-1.63)
w/o `f_select_row()`	82.21 (-4.40)	65.06 (-2.25)
w/o `f_group_by()`	84.78 (-1.83)	61.88 (-5.43)
w/o `f_sort_by()`	86.21 (-0.40)	65.85 (-1.46)

As shown in Table A.1, all five operations contribute to the final state-of-the-art performance of CHAIN-OF-TABLE, as removing any operation results in a decrease in performance. In particular, we observe that `f_select_row()` and `f_select_column()` contribute the most on TabFact, while `f_group_by()` contributes the most on WikiTQ. This suggests

that different tasks require different operations to help the LLM determine the correct answer. Therefore, leveraging the LLM to design custom operation chains through dynamic planning naturally fits different tasks, resulting in superior performance of our method.

A.2 Experiments of CHAIN-OF-TABLE on FeTaQA

Table A.2 shows that CHAIN-OF-TABLE also improves the performance of free-form question answering on FeTaQA across all metrics, whereas Dater [125] fails to improve the ROUGE scores compared with End-to-End QA. We also observe the marginal improvement of CHAIN-OF-TABLE compared with the baseline methods. We attribute this to the nature of the n-gram text similarity metrics of ROUGE-1/2/L [59]. As discussed in [73, 19], these metrics are known to be insensitive to capturing improvements when using in-context learning since the model is unable to learn the expected style of the long form text just from an instruction or a few examples. We sample several cases from FeTaQA as shown in Figure A.1 where the ROUGE metrics assign low scores; however, upon review, we observe that the generated answers were correct.

Table A.2. Table understanding results on the FeTaQA benchmark using PaLM 2 with the best results in bold and improvements over Dater [125] reported. (underline denotes the second-best performance; **bold** denotes the best performance; the improvement is measured against the second-best performing method.)

Prompting	FeTaQA			
	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
End-to-End QA	28.37	0.63	0.41	0.53
Dater [125]	29.47	0.63	0.41	0.53
CHAIN-OF-TABLE (ours)	**32.61** (+3.14)	**0.66** (+0.03)	**0.44** (+0.03)	**0.56** (+0.03)

Example from FeTaQA

Question: Who were the last two finishers in the 2000 Summer Olympics Mens 100 metre freestyle?
Answer: Russia's Denis Pimankov (49.36) and Australia's Chris Fydler (49.44) rounded out the finale.
Prediction: The last two finishers in the 2000 Summer Olympics Mens 100 metre freestyle were Chris Fydler and Denis Pimankov.
Results: ROUGE-1=0.33; ROUGE-2=0.12; ROUGE-L=0.11

Explanation: The generated response correctly answers the question but the sentence styles are different. From the metrics, we can see the ROUGE scores are below the average.

Figure A.1. Result example of CHAIN-OF-TABLE on FeTaQA using the ROUGE scores as metrics, where the ROUGE metrics assign very low scores but the generated answers were correct.

A.3 Inference Parameters and Number of Demo Samples of CHAIN-OF-TABLE

We report the parameters and demo sample numbers we used in CHAIN-OF-TABLE in Table A.3, A.4 and A.5. Overall, we annotate 29 samples and use them across different datasets. There are a large overlapping between the usage on different functions. For example, we use the same demo sample to introduce how to use `f_add_column` in the function `DynamicPlan` across different datasets. We guarantee that all demo samples are from the training set so they are unseen during testing. We argue that this further demonstrates our framework does not rely on a specific set of demos and can be well generalized to new datasets with the same prompts.

Table A.3. LLM parameters and number of demo samples in CHAIN-OF-TABLE on WikiTQ

Function	WikiTQ				
	temperature	top_p	decode_steps	n_samples	n_demos
DynamicPlan()	0.0	1.0	200	-	4
f_add_column()	0.0	1.0	200	-	6
f_select_row()	1.0	1.0	200	8	3
f_select_column()	1.0	1.0	200	8	8
f_group_by()	0.0	1.0	200	-	2
f_sort_by()	0.0	1.0	200	-	2
query()	0.0	1.0	200	-	1

Table A.4. LLM parameters and number of demo samples in CHAIN-OF-TABLE on TabFact

Function	TabFact				
	temperature	top_p	decode_steps	n_samples	n_demos
DynamicPlan()	0.0	1.0	200	-	4
f_add_column()	0.0	1.0	200	-	7
f_select_row()	0.5	1.0	200	8	4
f_select_column()	0.5	1.0	200	8	8
f_group_by()	0.0	1.0	200	-	2
f_sort_by()	0.0	1.0	200	-	2
query()	0.0	1.0	200	-	4

Table A.5. LLM parameters and number of demo samples in CHAIN-OF-TABLE on FeTaQA

Function	FeTaQA				
	temperature	top_p	decode_steps	n_samples	n_demos
DynamicPlan()	0.0	1.0	200	-	3
f_add_column()	0.0	1.0	200	-	6
f_select_row()	1.0	1.0	200	8	3
f_select_column()	1.0	1.0	200	8	8
f_group_by()	0.0	1.0	200	-	2
f_sort_by()	0.0	1.0	200	-	2
query()	0.0	1.0	200	-	8

A.4 Tabular Format Encoding Comparison

In alignment with prior studies [64, 63, 46] and the baseline methods [15, 125], we adopt PIPE encoding in CHAIN-OF-TABLE (as shown in Appendix A.5). This decouples the performance gains of the proposed tabular CoT with atomic operations from the influence of various table formatting choices.

To further understand the impact of different encoding methods on table understanding performance, we conduct additional experiments using 3 additional table representations: HTML, TSV, and Markdown. For these experiments, we use End-to-End QA on WikiTQ with PaLM 2 as a running example. The results are shown in Table A.6. These findings show that different tabular format encoding methods lead to different outcomes. Notably, the PIPE format adopted in our study yields the highest performance among the four encoding methods tested.

Table A.6. Tabular format encoding comparison on WikiTQ with PaLM 2

Prompting	Tabular Format Encoding			
	PIPE	**HTML**	**TSV**	**Markdown**
End-to-End QA	60.6	56.1	58.1	58.0

A.5 Prompts in CHAIN-OF-TABLE

A.5.1 `DynamicPlan`

We illustrate the prompting method used by `DynamicPlan(T,Q,chain)` in Figure A.2 where T is the latest intermediate table and Q is its corresponding question; `chain` is the list of operations performed on the table.

With `DynamicPlan`, the LLM can generate the rest of the operation chain for the current sample (Figure A.2(c)). We denote the generated operations as $\texttt{f}_{i+1}(\texttt{args}_{i+1}) \rightarrow ... \rightarrow$ `[E]` given that $\texttt{f}_i$ is the last operation of the input open-ended operation chain. Although a complete chain is generated, we only consider the first generated operation, $\texttt{f}_{i+1}$, and ignore the

rest of the generation including the arguments and remaining operations. $\texttt{f}_{i+1}$ is generated based on the latest intermediate table from the previous operations, while the generation of subsequent operations are not based on the most up-to-date intermediate table so there could be mistakes in the generated contents. Therefore, we believe $\texttt{f}_{i+1}$ is the most reliable generation among all operations in the generated chain. See Figure A.5 for more detailed prompts.

A.5.2 `GenerateArgs`

We illustrate the demonstration and prompts used by `GenerateArgs(T,Q,f)` in Figure A.3 where T is the latest intermediate table and Q is its corresponding question; `f` is the selected tabular operations. The detailed prompts for each operation and the regular expressions for extracting the generated arguments are as follows.

- `f_add_column`: See Figure A.6.
- `f_select_row`: See Figure A.8.
- `f_select_column`: See Figure A.7.
- `f_group_by`: See Figure A.9.
- `f_sort_by`: See Figure A.10.

A.5.3 `Query`

We illustrate the prompts used by `Query(T,Q)` in Figure A.4 where T is the resulting table from CHAIN-OF-TABLE and Q is the question. See Figure A.11 for more detailed prompts.

A.6 Implementation Details of Baseline Methods

We run Text-to-SQL and Binder using the official open-sourced code and prompts in https://github.com/HKUNLP/Binder. We run Dater using the official open-sourced code and prompts in https://github.com/AlibabaResearch/DAMO-ConvAI. We revise the code to use

publicly available GPT 3.5 and PaLM 2 (Section 5.4) as the LLM backbone instead of the OpenAI Codex due to its inaccessibility. We report the detailed prompts used in other baseline methods as follows.

- **End-to-End QA**: See Figure A.12.
- **Few-Shot QA**: See Figure A.13.
- **Chain-of-Thought**: The demonstration samples of Chain-of-Thought for WikiTQ and TabFact are from [12] (https://github.com/wenhuchen/TableCoT). See Figure A.14.

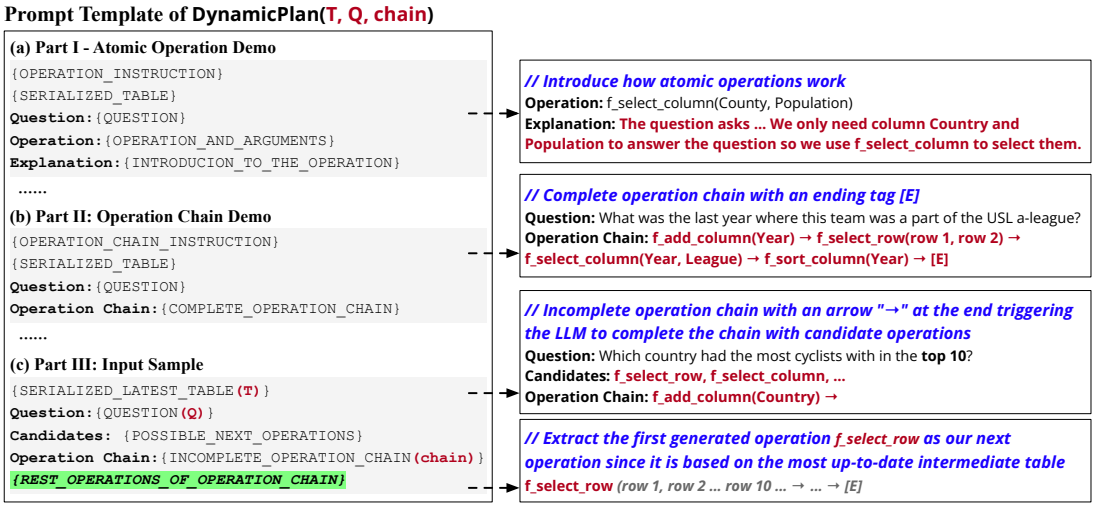


Figure A.2. Illustration of `DynamicPlan(`T`,`Q`,chain)`. **Left**: Overall prompt template and expected generation, including (a) demonstration of how atomic operations work, (b) demonstration of how to generate a complete operation chain to answer a given question, and (c) prompt for actual input table and its question, and its expected generation from the LLM (highlighted in green). **Right**: Examples and brief explanations of each part in the prompt and generation.

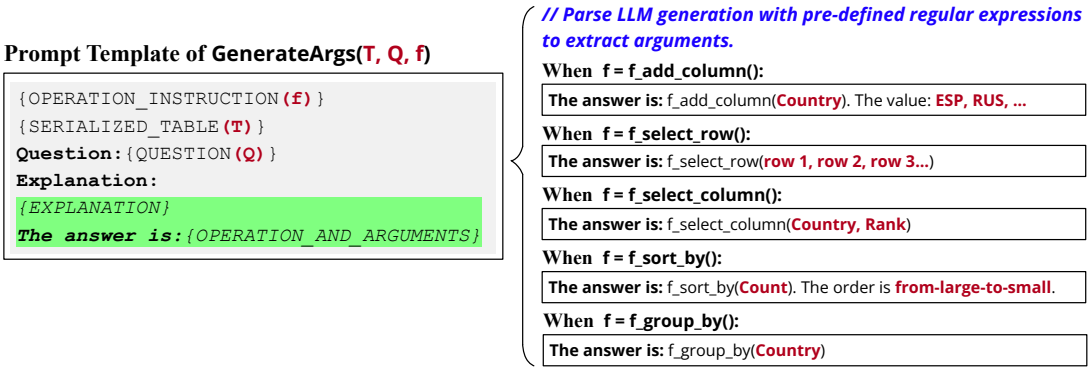


Figure A.3. Illustration of `GenerateArgs`$(T, Q, \texttt{f})$. After a specific operation `f` is sampled by the LLM as the next operation, we ask the LLM to generate the required arguments by calling `GenerateArgs`. Then we parse the generation results of the LLM according to the pre-defined templates to extract the arguments.

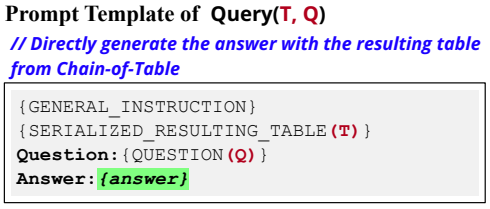


Figure A.4. Illustration of `Query`(T, Q). The resulting table from the operation chain serves as a proxy for the intermediate thoughts of reasoning, allowing us to directly generate the answer without providing the reasoning chain in textual format.

```
======================================== Prompt ========================================

If the table only needs a few rows to answer the question, we use f_select_row() to select
these rows for it. For example,
/*
col : Home team | Home Team Score | Away Team | Away Team Score | Venue | Crowd
row 1 : st kilda | 13.12 (90) | melbourne | 13.11 (89) | moorabbin oval | 18836
row 2 : south melbourne | 9.12 (66) | footscray | 11.13 (79) | lake oval | 9154
row 3 : richmond | 20.17 (137) | fitzroy | 13.22 (100) | mcg | 27651
*/
Question : Whose home team score is higher, richmond or st kilda?
Function: f_select_row(row 1, row 3)
Explanation: The question asks about the home team score of richmond and st kilda. We need
to know the the information of richmond and st kilda in row 1 and row 3. We select row 1
and row 3.

If the table only needs a few columns to answer the question, we use
f_select_column() to select these columns for it. For example,
......

If the question asks about items with the same value and the number of these items, we use
f_group_by() to group the items. For example,
......

If the question asks about the order of items in a column, we use f_sort_by() to sort
the items. For example,
......

Here are examples of using the operations to answer the question.
/*
col : Date | Division | League | Regular Season | Playoffs | Open Cup
row 1 : 2001/01/02 | 2 | USL A-League | 4th, Western | Quarterfinals | Did not qualify
row 2 : 2002/08/06 | 2 | USL A-League | 2nd, Pacific | 1st Round | Did not qualify
row 5 : 2005/03/24 | 2 | USL First Division | 5th | Quarterfinals | 4th Round
*/
Question: what was the last year where this team was a part of the usl a-league?
Function Chain: f_add_column(Year) -> f_select_row(row 1, row 2) ->
f_select_column(Year, League) -> f_sort_by(Year) -> <END>
......

/*
col : Rank | Cyclist | Team | Time | UCI ProTour; Points | Country
row 1 : 1 | Alejandro Valverde (ESP) | Caisse d'Epargne | 5h 29' 10" | 40 | ESP
row 2 : 2 | Alexandr Kolobnev (RUS) | Team CSC Saxo Bank | s.t. | 30 | RUS
row 3 : 3 | Davide Rebellin (ITA) | Gerolsteiner | s.t. | 25 | ITA
row 4 : 4 | Paolo Bettini (ITA) | Quick Step | s.t. | 20 | ITA
row 5 : 5 | Franco Pellizotti (ITA) | Liquigas | s.t. | 15 | ITA
row 6 : 6 | Denis Menchov (RUS) | Rabobank | s.t. | 11 | RUS
row 7 : 7 | Samuel Sánchez (ESP) | Euskaltel-Euskadi | s.t. | 7 | ESP
row 8 : 8 | Stéphane Goubert (FRA) | Ag2r-La Mondiale | + 2" | 5 | FRA
row 9 : 9 | Haimar Zubeldia (ESP) | Euskaltel-Euskadi | + 2" | 3 | ESP
row 10 : 10 | David Moncoutié (FRA) | Cofidis | + 2" | 1 | FRA
*/
Question: which country had the most cyclists finish within the top 10?
The next operation must be one of f_select_row() or f_select_column() or f_group_by()
or f_sort_by().
Function Chain: f_add_column(Country) ->

====================================== Completion ======================================

f_select_row(row 1, row 10) -> f_select_column(Country) -> f_group_by(Country) -> <END>
```

Figure A.5. `DynamicPlan(T,Q,chain)` Prompt used for WikiTQ

```
To answer the question, we can first use f_add_column() to add more columns to the table.

The added columns should have these data types:
1. Numerical: the numerical strings that can be used in sort, sum
2. Datetype: the strings that describe a date, such as year, month, day
3. String: other strings

/*
col : Week | When | Kickoff | Opponent | Results; Final score | Results; Team record
row 1 : 1 | Saturday, April 13 | 7:00 p.m. | at Rhein Fire | W 27-21 | 1-0
row 2 : 2 | Saturday, April 20 | 7:00 p.m. | London Monarchs | W 37-3 | 2-0
row 3 : 3 | Sunday, April 28 | 6:00 p.m. | at Barcelona Dragons | W 33-29 | 3-0
*/
Question: what is the date of the competition with highest attendance?
The existing columns are: "Week", "When", "Kickoff", "Opponent", "Results; Final score",
"Results; Team record", "Game site", "Attendance".
Explanation: the question asks about the date of the competition with highest score. Each
row is about one competition. We extract the value from column "Attendance" and create a
different column "Attendance number" for each row. The datatype is Numerical.
Therefore, the answer is: f_add_column(Attendance number). The value: 32092 | 34186 | 17503

/*
col : Rank | Lane | Player | Time
row 1 :  | 5 | Olga Tereshkova (KAZ) | 51.86
row 2 :  | 6 | Manjeet Kaur (IND) | 52.17
row 3 :  | 3 | Asami Tanno (JPN) | 53.04
*/
Question: tell me the number of athletes from japan.
The existing columns are: Rank, Lane, Player, Time.
Explanation: the question asks about the number of athletes from japan. Each row is about
one athlete. We need to know the country of each athlete. We extract the value from column
"Player" and create a different column "Country of athletes" for each row. The datatype
is String.
Therefore, the answer is: f_add_column(Country of athletes). The value: KAZ | IND | JPN
```

Figure A.6. Demos used for `GenerateArgs(T,Q,f_add_column)`. We use the regular expression: `f_add_column((.*)).The value:(.*)` to extract the arguments from the generated text.

```
Use f_select_column() to filter out useless columns in the table according to information
in the statement and the table.

/*
{
  "table_caption": "south wales derby",
  "columns": ["competition", "total matches", "cardiff win", "draw", "swansea win"],
  "table_column_priority": [
    ["competition", "league", "fa cup", "league cup"],
    ["total matches", "55", "2", "5"],
    ["cardiff win", "19", "0", "2"],
    ["draw", "16", "27", "0"],
    ["swansea win", "20", "2", "3"]
  ]
}
*/
statement : there are no cardiff wins that have a draw greater than 27.
similar words link to columns :
no cardiff wins -> cardiff win
a draw -> draw
column value link to columns :
27 -> draw
semantic sentence link to columns :
None
The answer is : f_select_column([cardiff win, draw])
```

Figure A.7. Demos used for `GenerateArgs(T,Q,f_select_column)`. We use the regular expression: `f_select_column([(.*)])` to extract the arguments from the generated text.

```
Using f_select_row() to select relevant rows in the given table that support or oppose the
statement.
Please use f_select_row([*]) to select all rows in the table.

/*
table caption : 1972 vfl season.
col : home team | home team score | away team | away team score | venue | crowd
row 1 : st kilda | 13.12 (90) | melbourne | 13.11 (89) | moorabbin oval | 18836
row 2 : south melbourne | 9.12 (66) | footscray | 11.13 (79) | lake oval | 9154
row 3 : richmond | 20.17 (137) | fitzroy | 13.22 (100) | mcg | 27651
row 4 : geelong | 17.10 (112) | collingwood | 17.9 (111) | kardinia park | 23108
row 5 : north melbourne | 8.12 (60) | carlton | 23.11 (149) | arden street oval | 11271
row 6 : hawthorn | 15.16 (106) | essendon | 12.15 (87) | vfl park | 36749
*/
statement : what is the away team with the highest score?
explain : the statement want to ask the away team of highest away team score. the highest
away team score is 23.11 (149). it is on the row 5.so we need row 5.
The answer is : f_select_row([row 5])
```

Figure A.8. Demos used for `GenerateArgs(T,Q,f_select_row)`. We use the regular expression: `f_select_row([(.*)])` to extract the arguments from the generated text.

```
To answer the question, we can first use f_group_by() to group the values in a column.

/*
col : Rank | Lane | Athlete | Time | Country
row 1 : 1 | 6 | Manjeet Kaur (IND) | 52.17 | IND
row 2 : 2 | 5 | Olga Tereshkova (KAZ) | 51.86 | KAZ
row 3 : 3 | 4 | Pinki Pramanik (IND) | 53.06 | IND
row 4 : 4 | 1 | Tang Xiaoyin (CHN) | 53.66 | CHN
row 5 : 5 | 8 | Marina Maslyonko (KAZ) | 53.99 | KAZ
*/
Question: tell me the number of athletes from japan.
The existing columns are: Rank, Lane, Athlete, Time, Country.
Explanation: The question asks about the number of athletes from India. Each row is about
an athlete. We can group column "Country" to group the athletes from the same country.
Therefore, the answer is: f_group_by(Country).
```

Figure A.9. Demos used for `GenerateArgs(T,Q,f_group_by)`. We use the regular expression: `f_group_by((.*))` to extract the arguments from the generated text.

```
To answer the question, we can first use f_sort_by() to sort the values in a column to get the
order of the items. The order can be "large to small" or "small to large".

The column to sort should have these data types:
1. Numerical: the numerical strings that can be used in sort
2. DateType: the strings that describe a date, such as year, month, day
3. String: other strings

/*
col : Position | Club | Played | Points | Wins | Draws | Losses | Goals for | Goals against
row 1 : 1 | Malaga CF | 42 | 79 | 22 | 13 | 7 | 72 | 47
row 10 : 10 | CP Merida | 42 | 59 | 15 | 14 | 13 | 48 | 41
row 3 : 3 | CD Numancia | 42 | 73 | 21 | 10 | 11 | 68 | 40
*/
Question: what club placed in the last position?
The existing columns are: Position, Club, Played, Points, Wins, Draws, Losses, Goals for,
Goals against
Explanation: the question asks about the club in the last position. Each row is about a
club. We need to know the order of position from last to front. There is a column for
position and the column name is Position. The datatype is Numerical.
Therefore, the answer is: f_sort_by(Position), the order is "large to small".
```

Figure A.10. Demos used for `GenerateArgs(T,Q,f_sort_by)`. We use the regular expression: `f_sort_by((.*)),the order is "(.*)".` to extract the arguments from the generated text.

```
======================================== Prompt ========================================

Here is the table to answer this question. Please understand the table and answer the
question:

/*
col : Rank | City | Passengers Number | Ranking | Airline
row 1 : 1 | United States, Los Angeles | 14749 | 2 | Alaska Airlines
row 2 : 2 | United States, Houston | 5465 | 8 | United Express
row 3 : 3 | Canada, Calgary | 3761 | 5 | Air Transat, WestJet
row 4 : 4 | Canada, Saskatoon | 2282 | 4 |
row 5 : 5 | Canada, Vancouver | 2103 | 2 | Air Transat
row 6 : 6 | United States, Phoenix | 1829 | 1 | US Airways
row 7 : 7 | Canada, Toronto | 1202 | 1 | Air Transat, CanJet
row 8 : 8 | Canada, Edmonton | 110 | 2 |
row 9 : 9 | United States, Oakland | 107 | 5 |
*/
Question: how many more passengers flew to los angeles than to saskatoon from manzanillo
airport in 2013?
The anwser is: 12467

Here is the table to answer this question. Please understand the table and answer the
question:

/*
col : Rank | Country
row 1 : 1 | ESP
row 2 : 2 | RUS
row 3 : 3 | ITA
row 4 : 4 | ITA
row 5 : 5 | ITA
row 6 : 6 | RUS
row 7 : 7 | ESP
row 8 : 8 | FRA
row 9 : 9 | ESP
row 10 : 10 | FRA
*/
Group the rows according to column "Country":
/*
Group ID | Country | Count
1 | ITA | 3
2 | ESP | 3
3 | RUS | 2
4 | FRA | 2
*/
Question: which country had the most cyclists in top 10?
The answer is:

====================================== Completion ======================================
Italy.
```

Figure A.11. Prompt Example used for `Query(T,Q)`

```
======================================== Prompt ========================================

Here is the table to answer this question. Answer the question.
/*
col : Name | League | FA Cup | League Cup | JP Trophy | Total
row 1 : Scot Bennett | 5 | 0 | 0 | 0 | 5
row 2 : Danny Coles | 3 | 0 | 0 | 0 | 3
row 3 : Liam Sercombe | 1 | 0 | 0 | 0 | 1
row 4 : Alan Gow | 4 | 0 | 0 | 0 | 4
row 5 : John O'Flynn | 11 | 0 | 1 | 0 | 12
row 6 : Guillem Bauza | 2 | 0 | 0 | 0 | 2
row 7 : Jimmy Keohane | 3 | 0 | 0 | 0 | 3
row 8 : Pat Baldwin | 1 | 0 | 0 | 0 | 1
row 9 : Jamie Cureton | 20 | 0 | 0 | 0 | 20
row 10 : Arron Davies | 3 | 0 | 0 | 0 | 3
row 11 : Jake Gosling | 1 | 0 | 0 | 0 | 1
row 12 : OWN GOALS | 0 | 0 | 0 | 0 | 0
row 13 : Total | 0 | 0 | 0 | 0 | 0
*/
Question: does pat or john have the highest total?
The answer is:


====================================== Completion ======================================

John.
```

Figure A.12. Prompt of End-to-end QA used for WikiTQ.

```
======================================== Prompt ========================================

Here is the table to answer this question. Answer the question.
/*
col : Rank | Cyclist | Team | Time | UCI ProTour; Points
row 1 : 1 | Alejandro Valverde (ESP) | Caisse d'Epargne | 5h 29' 10" | 40
row 2 : 2 | Alexandr Kolobnev (RUS) | Team CSC Saxo Bank | s.t. | 30
row 3 : 3 | Davide Rebellin (ITA) | Gerolsteiner | s.t. | 25
row 4 : 4 | Paolo Bettini (ITA) | Quick Step | s.t. | 20
row 5 : 5 | Franco Pellizotti (ITA) | Liquigas | s.t. | 15
row 6 : 6 | Denis Menchov (RUS) | Rabobank | s.t. | 11
row 7 : 7 | Samuel Sánchez (ESP) | Euskaltel-Euskadi | s.t. | 7
row 8 : 8 | Stéphane Goubert (FRA) | Ag2r-La Mondiale | + 2" | 5
row 9 : 9 | Haimar Zubeldia (ESP) | Euskaltel-Euskadi | + 2" | 3
row 10 : 10 | David Moncoutié (FRA) | Cofidis | + 2" | 1
*/
Question: which country had the most cyclists finish within the top 10?
The answer is: Italy.

Here is the table to answer this question. Please provide your explanation first, then
answer the question in a short phrase starting by 'therefore, the answer is:'
/*
col : Rank | Cyclist | Team | Time | UCI ProTour; Points
row 1 : 1 | Alejandro Valverde (ESP) | Caisse d'Epargne | 5h 29' 10" | 40
row 2 : 2 | Alexandr Kolobnev (RUS) | Team CSC Saxo Bank | s.t. | 30
row 3 : 3 | Davide Rebellin (ITA) | Gerolsteiner | s.t. | 25
row 4 : 4 | Paolo Bettini (ITA) | Quick Step | s.t. | 20
row 5 : 5 | Franco Pellizotti (ITA) | Liquigas | s.t. | 15
row 6 : 6 | Denis Menchov (RUS) | Rabobank | s.t. | 11
row 7 : 7 | Samuel Sánchez (ESP) | Euskaltel-Euskadi | s.t. | 7
row 8 : 8 | Stéphane Goubert (FRA) | Ag2r-La Mondiale | + 2" | 5
row 9 : 9 | Haimar Zubeldia (ESP) | Euskaltel-Euskadi | + 2" | 3
row 10 : 10 | David Moncoutié (FRA) | Cofidis | + 2" | 1
*/
Question: how many players got less than 10 points?
The answer is: 4.

Here is the table to answer this question. Answer the question.
/*
col : Name | League | FA Cup | League Cup | JP Trophy | Total
row 1 : Scot Bennett | 5 | 0 | 0 | 0 | 5
row 2 : Danny Coles | 3 | 0 | 0 | 0 | 3
row 3 : Liam Sercombe | 1 | 0 | 0 | 0 | 1
row 4 : Alan Gow | 4 | 0 | 0 | 0 | 4
row 5 : John O'Flynn | 11 | 0 | 1 | 0 | 12
row 6 : Guillem Bauza | 2 | 0 | 0 | 0 | 2
row 7 : Jimmy Keohane | 3 | 0 | 0 | 0 | 3
row 8 : Pat Baldwin | 1 | 0 | 0 | 0 | 1
row 9 : Jamie Cureton | 20 | 0 | 0 | 0 | 20
row 10 : Arron Davies | 3 | 0 | 0 | 0 | 3
row 11 : Jake Gosling | 1 | 0 | 0 | 0 | 1
row 12 : OWN GOALS | 0 | 0 | 0 | 0 | 0
row 13 : Total | 0 | 0 | 0 | 0 | 0
*/
Question: does pat or john have the highest total?
The answer is:


====================================== Completion ======================================

John.
```

Figure A.13. Prompt of Few-shot QA used for WikiTQ

```
======================================== Prompt ========================================
Here is the table to answer this question. Please provide your explanation first, then
answer the question in a short phrase starting by 'therefore, the answer is:'
/*
col : Rank | Cyclist | Team | Time | UCI ProTour; Points
row 1 : 1 | Alejandro Valverde (ESP) | Caisse d'Epargne | 5h 29' 10" | 40
row 2 : 2 | Alexandr Kolobnev (RUS) | Team CSC Saxo Bank | s.t. | 30
row 3 : 3 | Davide Rebellin (ITA) | Gerolsteiner | s.t. | 25
row 4 : 4 | Paolo Bettini (ITA) | Quick Step | s.t. | 20
row 5 : 5 | Franco Pellizotti (ITA) | Liquigas | s.t. | 15
row 6 : 6 | Denis Menchov (RUS) | Rabobank | s.t. | 11
row 7 : 7 | Samuel Sánchez (ESP) | Euskaltel-Euskadi | s.t. | 7
row 8 : 8 | Stéphane Goubert (FRA) | Ag2r-La Mondiale | + 2" | 5
row 9 : 9 | Haimar Zubeldia (ESP) | Euskaltel-Euskadi | + 2" | 3
row 10 : 10 | David Moncoutié (FRA) | Cofidis | + 2" | 1
*/
Question: which country had the most cyclists finish within the top 10?
Explanation: ITA occurs three times in the table, more than any others. Therefore, the
answer is: Italy.

Here is the table to answer this question. Please provide your explanation first, then
answer the question in a short phrase starting by 'therefore, the answer is:'
/*
col : Rank | Cyclist | Team | Time | UCI ProTour; Points
row 1 : 1 | Alejandro Valverde (ESP) | Caisse d'Epargne | 5h 29' 10" | 40
row 2 : 2 | Alexandr Kolobnev (RUS) | Team CSC Saxo Bank | s.t. | 30
row 3 : 3 | Davide Rebellin (ITA) | Gerolsteiner | s.t. | 25
row 4 : 4 | Paolo Bettini (ITA) | Quick Step | s.t. | 20
row 5 : 5 | Franco Pellizotti (ITA) | Liquigas | s.t. | 15
row 6 : 6 | Denis Menchov (RUS) | Rabobank | s.t. | 11
row 7 : 7 | Samuel Sánchez (ESP) | Euskaltel-Euskadi | s.t. | 7
row 8 : 8 | Stéphane Goubert (FRA) | Ag2r-La Mondiale | + 2" | 5
row 9 : 9 | Haimar Zubeldia (ESP) | Euskaltel-Euskadi | + 2" | 3
row 10 : 10 | David Moncoutié (FRA) | Cofidis | + 2" | 1
*/
Question: how many players got less than 10 points?
Explanation: Samuel Sánchez,  Stéphane Goubert, Haimar Zubeldia and David Moncoutié
received less than 10 points.  Therefore, the answer is: 4.

Here is the table to answer this question. Please provide your explanation first, then
answer the question in a short phrase starting by 'therefore, the answer is:'
/*
col : Name | League | FA Cup | League Cup | JP Trophy | Total
row 1 : Scot Bennett | 5 | 0 | 0 | 0 | 5
row 2 : Danny Coles | 3 | 0 | 0 | 0 | 3
row 3 : Liam Sercombe | 1 | 0 | 0 | 0 | 1
row 4 : Alan Gow | 4 | 0 | 0 | 0 | 4
row 5 : John O'Flynn | 11 | 0 | 1 | 0 | 12
row 6 : Guillem Bauza | 2 | 0 | 0 | 0 | 2
row 7 : Jimmy Keohane | 3 | 0 | 0 | 0 | 3
row 8 : Pat Baldwin | 1 | 0 | 0 | 0 | 1
row 9 : Jamie Cureton | 20 | 0 | 0 | 0 | 20
row 10 : Arron Davies | 3 | 0 | 0 | 0 | 3
row 11 : Jake Gosling | 1 | 0 | 0 | 0 | 1
row 12 : OWN GOALS | 0 | 0 | 0 | 0 | 0
row 13 : Total | 0 | 0 | 0 | 0 | 0
*/
Question: does pat or john have the highest total?
Explanation:
====================================== Completion ======================================
John O'Flynn has the highest total of 12 goals. Pat Baldwin has the lowest total of 1 goal.
Therefore, the answer is: John.
```

Figure A.14. Prompt of Chain-of-Thought used for WikiTQ

Bibliography

[1] Ewa Andrejczuk, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, and Yasemin Altun. Table-to-text generation and pre-training with TabT5. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6758–6766, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[3] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003, 2021.

[4] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR, 2020.

[5] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16548–16558, 2022.

[6] Łukasz Borchmann, Michał Pietruszka, Tomasz Stanislawek, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Graliński. Due: End-to-end document understanding benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[7] Mirko Bronzi, Valter Crescenzi, Paolo Merialdo, and Paolo Papotti. Extraction and integration of partially overlapping web sources. *Proceedings of the VLDB Endowment*, 6(10):805–816, 2013.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing*

systems, 33:1877–1901, 2020.

[9] Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: Exploring the power of tables on the web. *Proc. VLDB Endow.*, 1(1):538–549, aug 2008.

[10] Jiaoyan Chen, Yuxia Geng, Zhuo Chen, Ian Horrocks, Jeff Z. Pan, and Huajun Chen. Knowledge-aware zero-shot learning: Survey and perspective. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4366–4373. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.

[11] Si-An Chen, Lesly Miculicich, Julian Martin Eisenschlos, Zifeng Wang, Zilong Wang, Yanfei Chen, Yasuhisa Fujii, Hsuan-Tien Lin, Chen-Yu Lee, and Tomas Pfister. Tablerag: Million-token table understanding with language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[12] Wenhu Chen. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[13] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

[14] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*, 2019.

[15] Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. Binding language models in symbolic languages. In *International Conference on Learning Representations*, 2022.

[16] Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413, 2018.

[17] N De Cao, G Izacard, S Riedel, and F Petroni. Autoregressive entity retrieval. In *ICLR 2021-9th International Conference on Learning Representations*, volume 2021. ICLR, 2020.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the*

2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

[19] Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, 2019.

[20] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

[21] Julian Eisenschlos, Syrine Krichene, and Thomas Müller. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online, November 2020. Association for Computational Linguistics.

[22] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[23] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545, 2011.

[24] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR, 2023.

[25] Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Graliński. Lambert: Layout-aware language modeling for information extraction. In *International Conference on Document Analysis and Recognition*, pages 532–547. Springer, 2021.

[26] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[27] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. Unidoc: Unified pretraining framework for document understanding. *Advances in Neural Information Processing Systems*, 34:39–50, 2021.

[28] Zihui Gu, Ju Fan, Nan Tang, Preslav Nakov, Xiaoman Zhao, and Xiaoyong Du. PASTA: Table-operations aware fact verification via sentence-table cloze pre-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4971–4983, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[29] Jean-Philippe Thiran Guillaume Jaume, Hazim Kemal Ekenel. Funsd: A dataset for form understanding in noisy scanned documents. In *Accepted to ICDAR-OST*, 2019.

[30] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: Word-level Adversarial ReProgramming. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online, August 2021. Association for Computational Linguistics.

[31] Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: a unified solution for structured web data extraction. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 775–784, 2011.

[32] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

[33] Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995, 2015.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[36] William Hogan, Jiacheng Li, and Jingbo Shang. Fine-grained contrastive learning for relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1083–1095, 2022.

[37] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model focusing on text and layout for better key

information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10767–10775, 2022.

[38] Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 2023.

[39] Haochen Huang, Bingyu Shen, Li Zhong, and Yuanyuan Zhou. Protecting data integrity of web applications with database constraints inferred from application code. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 632–645, 2023.

[40] Haochen Huang, Chengcheng Xiang, Li Zhong, and Yuanyuan Zhou. {PYLIVE}:{On-the-Fly} code change for python-based online services. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 349–363, 2021.

[41] Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. Few-shot named entity recognition: An empirical baseline study. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[42] Jie Huang, Zilong Wang, Kevin Chang, Wen-Mei Hwu, and Jinjun Xiong. Exploring semantic capacity of terms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8509–8518, 2020.

[43] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022.

[44] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE, 2019.

[45] Shima Imani, Liang Du, and Harsh Shrivastava. MathPrompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42, Toronto, Canada, July 2023. Association for Computational Linguistics.

[46] Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States, July 2022. Association for Computational Linguistics.

[47] Nengzheng Jin, Joanna Siebert, Dongfang Li, and Qingcai Chen. A survey on table question answering: recent advances. In *China Conference on Knowledge Graph and Semantic Computing*, pages 174–186. Springer, 2022.

[48] George Katsogiannis-Meimarakis and Georgia Koutrika. A survey on deep learning approaches for text-to-sql. *The VLDB Journal*, pages 1–32, 2023.

[49] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *International Conference on Learning Representations*, 2022.

[50] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, 2022.

[51] Prashant Krishnan, Zilong Wang, Yangkun Wang, and Jingbo Shang. Towards few-shot entity recognition in document images: A graph neural network approach robust to image manipulation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16514–16526, 2024.

[52] Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. Formnet: Structural encoding beyond sequential modeling in form document information extraction. In *ACL*, 2022.

[53] Chen-Yu Lee, Chun-Liang Li, Chu Wang, Renshen Wang, Yasuhisa Fujii, Siyang Qin, Ashok Popat, and Tomas Pfister. Rope: reading order equivariant positional encoding for graph-based document information extraction. In *ACL*, 2021.

[54] Dongyang Li, Taolin Zhang, Nan Hu, Chengyu Wang, and Xiaofeng He. Hiclre: A hierarchical contrastive learning framework for distantly supervised relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2567–2578, 2022.

[55] Junlong Li, Yiheng Xu, Lei Cui, and Furu Wei. Markuplm: Pre-training of text and markup language for visually rich document understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

6078–6087, 2022.

[56] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. DocBank: A benchmark dataset for document layout analysis. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[57] Zimeng Li, Bo Shao, Linjun Shou, Ming Gong, Gen Li, and Daxin Jiang. Wiert: web information extraction via render tree. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13166–13173, 2023.

[58] Bill Yuchen Lin, Ying Sheng, Nguyen Vo, and Sandeep Tata. Freedom: A transferable neural architecture for structured information extraction on web documents. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1092–1102, 2020.

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

[60] Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and Jim Glass. Query understanding enhanced by hierarchical parsing structures. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–77. IEEE, 2013.

[61] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12, 2024.

[62] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), January 2023.

[63] Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. TAPEX: Table pre-training via learning a neural sql executor. In *International Conference on Learning Representations*, 2021.

[64] Qian Liu, Fan Zhou, Zhengbao Jiang, Longxu Dou, and Min Lin. From zero to hero: Examining the power of symbolic tasks in instruction tuning. *arXiv preprint arXiv:2304.07995*, 2023.

[65] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[66] Colin Lockard, Xin Luna Dong, Arash Einolghozati, and Prashant Shiralkar. Ceres: Distantly supervised relation extraction from the semi-structured web. *Proceedings of the VLDB Endowment*, 11(10), 2018.

[67] Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. Openceres: When open information extraction meets the semi-structured web. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3047–3056, 2019.

[68] Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. Zeroshotceres: Zero-shot relation extraction from semi-structured webpages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8105–8117, 2020.

[69] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[70] Lluis Marquez, Pere Comas, Jesús Giménez, and Neus Catala. Semantic role labeling as sequential tagging. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 193–196, 2005.

[71] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

[72] Mausam Mausam. Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, pages 4074–4077, 2016.

[73] Joshua Maynez, Priyanka Agrawal, and Sebastian Gehrmann. Benchmarking large language model capabilities for conditional generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9194–9213, 2023.

[74] Hiroki Nakayama. seqeval: A python framework for sequence labeling evaluation, 2018. Software available from https://github.com/chakki-works/seqeval.

[75] Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022.

[76] Alex Nguyen, Zilong Wang, Jingbo Shang, and Dheeraj Mekala. Docmaster: A uni-

fied platform for annotation, training, & inference in document question-answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 128–136, 2024.

[77] Dat Quoc Nguyen and Karin Verspoor. End-to-end neural relation extraction using deep biaffine attention. In *European conference on information retrieval*, pages 729–738. Springer, 2019.

[78] Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-tau Yih, Sida Wang, and Xi Victoria Lin. Lever: Learning to verify language-to-code generation with execution. In *International Conference on Machine Learning*, pages 26106–26128. PMLR, 2023.

[79] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

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

[81] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: A consolidated receipt dataset for post-ocr parsing. In *Document Intelligence Workshop at Neural Information Processing Systems*, 2019.

[82] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics.

[83] Han Peng, Ge Li, Wenhan Wang, Yunfei Zhao, and Zhi Jin. Integrating tree path in transformer for code representation. *Advances in Neural Information Processing Systems*, 34:9343–9354, 2021.

[84] Han Peng, Ge Li, Yunfei Zhao, and Zhi Jin. Rethinking positional encoding in tree transformer for code representation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3204–3214, 2022.

[85] Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang. Answer is all you need: Instruction-following text embedding via answering the question. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

[86] Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang,

and Jingbo Shang. Answer is all you need: Instruction-following text embedding via answering the question. *arXiv preprint arXiv:2402.09642*, 2024.

[87] Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, Chen-Yu Lee, and Nan Hua. LMDX: Language model-based document information extraction and localization. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[88] Richard Pönighaus. 'favourite'sql-statements—an empirical analysis of sql-usage in commercial applications. In *International Conference on Information Systems and Management of Data*, pages 75–91. Springer, 1995.

[89] Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer. In *International Conference on Document Analysis and Recognition*, pages 732–747. Springer, 2021.

[90] Raul Puri and Bryan Catanzaro. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*, 2019.

[91] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[92] Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. Evaluating the text-to-sql capabilities of large language models. *arXiv preprint arXiv:2204.00498*, 2022.

[93] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, 2009.

[94] Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. On the potential of lexico-logical alignments for semantic parsing to sql queries. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[95] Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. Kleister: key information extraction datasets involving long documents with complex layouts. In *International Conference on Document Analysis and Recognition*, pages 564–579. Springer, 2021.

[96] Peng Su, Yifan Peng, and K Vijay-Shanker. Improving bert model using contrastive learning for biomedical relation extraction. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 1–10, 2021.

[97] S Svetlichnaya. Deepform: Understand structured documents at scale, 2020.

[98] Sandeep Tata, Navneet Potti, James B. Wendt, Lauro Beltrão Costa, Marc Najork, and Beliz Gunel. Glean: Structured extractions from templatic documents. *Proc. VLDB Endow.*, 14(6):997–1005, apr 2021.

[99] Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*, 2023.

[100] Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[101] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[102] Jiapeng Wang, Lianwen Jin, and Kai Ding. LiLT: A simple yet effective language-independent layout transformer for structured document understanding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7747–7757, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[103] Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. Webformer: The web-page transformer for structure information extraction. In *Proceedings of the ACM Web Conference 2022*, pages 3124–3133, 2022.

[104] 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*, pages 1780–1790, 2021.

[105] Zihan Wang, Kewen Zhao, Zilong Wang, and Jingbo Shang. Formulating few-shot fine-tuning towards language model pre-training: A pilot study on named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3186–3199, 2022.

[106] Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. Officebench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024.

[107] Zilong Wang, Jiuxiang Gu, Chris Tensmeyer, Nikolaos Barmpalios, Ani Nenkova, Tong

Sun, Jingbo Shang, and Vlad Morariu. Mgdoc: Pre-training with multi-granular hierarchy for document image understanding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3984–3993, 2022.

[108] Zilong Wang and Jingbo Shang. Towards few-shot entity recognition in document images: A label-aware sequence-to-sequence framework. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4174–4186, 2022.

[109] Zilong Wang and Jingbo Shang. Towards zero-shot relation extraction in web mining: A multimodal approach with relative xml path. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[110] Zilong Wang, Zhaohong Wan, and Xiaojun Wan. Transmodality: An end2end fusion method with transformer for multimodal sentiment analysis. In *Proceedings of the web conference 2020*, pages 2514–2520, 2020.

[111] Zilong Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, et al. Speculative rag: Enhancing retrieval augmented generation through drafting. *arXiv preprint arXiv:2407.08223*, 2024.

[112] Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. Layoutreader: Pre-training of text and layout for reading order detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4735–4744, 2021.

[113] Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 898–908, 2020.

[114] Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations*, 2024.

[115] Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. Vrdu: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5184–5193, 2023.

[116] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[117] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23, 2013.

[118] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[119] Chenhao Xie, Wenhao Huang, Jiaqing Liang, Chengsong Huang, and Yanghua Xiao. Webke: Knowledge extraction from semi-structured web with pre-trained markup language model. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2211–2220, 2021.

[120] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, 2021.

[121] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

[122] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *arXiv preprint arXiv:2104.08836*, 2021.

[123] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[124] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[125] Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decomposing evidence and questions for

table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 174–184, New York, NY, USA, 2023. Association for Computing Machinery.

[126] Peng Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. Trie: end-to-end text reading and information extraction for document understanding. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1413–1422, 2020.

[127] Li Zhong. A survey of prevent and detect access control vulnerabilities. *arXiv preprint arXiv:2304.10600*, 2023.

[128] Li Zhong and Zilong Wang. Can llm replace stack overflow? a study on robustness and reliability of large language model code generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21841–21849, 2024.

[129] Li Zhong, Zilong Wang, and Jingbo Shang. Debug like a human: A large language model debugger via verifying runtime execution step by step. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[130] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*, 2022.

[131] Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, and Sandeep Tata. Simplified dom trees for transferable attribute extraction from the web. *arXiv preprint arXiv:2101.02415*, 2021.

[132] Jin Ziqi and Wei Lu. Tab-CoT: Zero-shot tabular chain of thought. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10259–10277, Toronto, Canada, July 2023. Association for Computational Linguistics.