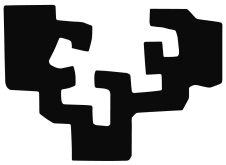
eman ta zabal zazu



Universidad del País Vasco / Euskal Herriko Unibertsitatea

Improving Fidelity and Table Representation in Table Understanding and Table-to-Text Generation

Iñigo Alonso González

Zuzendaria
Eneko Agirre

2025

EUSKAL HERRIKO UNIBERTSITATEA
Hizkuntzaren Azterketa eta Prozesamendua doktoretza-programa

Improving Fidelity and Table Representation in Table Understanding and Table-to-Text Generation

Iñigo Alonso Gonzalezek Eneko Agirreren zuzendaritzapean eginiko tesi txostena, Euskal Herriko Unibertsitatean Doktore titulua eskuratzeko aurkeztua.

Donostia, 2025ko Urtarrila.

Maybe the real thesis was
the friends we made
along the way.

Acknowledgements

Eskerrik asko...

Ixa taldeari eta ixakideei, ikertzaile bikainez betetako talde batean onartzeagatik eta ikerketa talde bat baina askoz gehiago izateagatik.

Eneko Agirreri irakatsi didan guztiagatik eta behar izan dudanean hor egoteagatik.

To the EdinburghNLP research group, especially Mirella Lapata, for teaching me so much and welcoming me as one of their own.

To my fellow workaholics at Ixa: Ander, Anar, and Olia. Lagun ikaragarriak izateagaitik, beti laguntzeko prest egoteagatik, маңдай терді ағызып, бас көтермей жұмыс істеген кездерді өте көңілді еткендерің үшін, и вдохнули жизнь в моё время в Доностии.

To all the people, soeurs, tesómnak, fratelli e sorelle, Schwestern, 兄弟姐妹, ziomków, and 仲間 in the Informatics Forum, for their motivation and companionship.

A aita, ama y mis hermanos, por ser mi referente, estar siempre ahí y nunca rendirse.

To Ale, Asier, and Jon, for all the laughs, advice, and adventures that led to this work.

Nire kuadrillako lagun guztioi. Eman ditudan buelta guzti hauetan hor egon zaretelako.

A Miguel Ángel Veganzones, por su apoyo y experiencia en Sherpa, y especialmente por haberme guiado durante el inicio de esta tesis.

And finally, to Jokin Rueda and all the teachers who never gave up on me.

Abstract

The field of Natural Language Processing (NLP) has advanced considerably, yet applying its techniques to structured data, like tables, introduces unique challenges. These challenges stem from the structured nature of tables and the need for accurate interpretation of their data. Among these challenges, a critical one in Table Understanding (TU) is the ability to represent all table information in a complete and efficient manner while ensuring, particularly in natural language generation tasks like table-to-text, that the generated texts remain faithful to the source data.

The goal of this thesis is to contribute to the field of TU by developing techniques that enhance fidelity in table-to-text generation and improve table representation to better capture information within tabular data. To this end, this thesis explores the use of structured semantics to guide table-to-text generation models in producing descriptions that faithfully represent table data. We dissect the critical components that play a key role in achieving this, including the grammar used to represent these semantics and the conditioning signals required to build them. We propose the use of automatically generated logical forms and analyze the impact of content selection in enhancing the system's accuracy. We demonstrate that using automatically generated logical forms significantly improves faithfulness and factual accuracy in table-to-text generation, achieving a 67% increase in fidelity over baseline models.

In addition, we propose a new method for effectively encapsulating information across a wider range of table formats. Specifically, we introduce the use of Visual Language Models (VLMs) to capture information from tables represented as images, highlighting their advantages over traditional text-based representations. We also address inherent challenges in this approach by proposing a new image-based structure learning curriculum to capture the structural dynamics of tabular data and reduce structure-related fidelity errors. Our proposed image-based table-

to-text generation model, PixT3, achieves state-of-the-art results, outperforming other baseline models in both automatic metrics and human evaluations of faithfulness. PixT3's strong performance on an out-of-domain dataset further demonstrates its adaptability to previously unseen tables.

Finally, we extend our image-based approach to additional TU tasks, such as Table Question Answering, Table Structure Recognition, Table Fact Verification, and Table Numerical Reasoning by creating a multimodal, instruction-based dataset that includes original table visualizations. We analyze state-of-the-art TU pre-training objectives to construct a dataset designed to instill foundational, generalizable knowledge of table interpretation into vision-based models. To this end, we introduce the largest multimodal, instruction-based TU dataset with original table visualizations from Wikipedia to date, comprising 2.5 million examples and 1.1 million unique table images across 11 different tasks. This dataset addresses a significant limitation of current multimodal TU datasets, which rely on lossy textual table representations, by incorporating original table visualizations instead.

This thesis contributes to the field of Table Understanding by introducing advancements that address the need for more reliable, scalable, and visually-aware methods for table-to-text generation. This work also proposes new research lines to further advance in this field. Our findings were published in a Journal Citation Reports (JCR) Q1-ranked journal and the main conference of ACL 2024.

Contents

List of Tables

List of Figures

1. CHAPTER

Introduction

This thesis is situated at the intersection of Natural Language Generation (NLG) and Table Understanding (TU), specifically focusing on the generation of coherent and accurate textual descriptions from tables, commonly referred to as table-to-text generation. In the area of Natural Language Processing (NLP), recent advancements have expanded the potential for machines to interpret and produce human language with increased fluency and accuracy. Structured data representation, however, requires additional layers of processing to accurately interpret and convey the information encoded within tables, an inherently organized and often complex form of data. This challenge is crucial as tables are widely used in domains such as finance, healthcare, and scientific research.

Table-to-text generation presents unique challenges beyond those encountered in general NLG tasks, due to the structured nature of the data and the need to accurately interpret its information. This work contributes three key advancements addressing these challenges, each designed to enhance the faithfulness, scalability, and applicability of Table Understanding and table-to-text systems. The first contribution focuses on improving fidelity in textual generation by using logical forms (LF) as intermediary representations. The second introduces a novel approach to representing tables as visual entities, taking advantage of recent advancements in vision language models (VLMs). Finally, the third contribution extends these techniques to a broader array of Table Understanding tasks, opening the path to bring the benefits of this multimodal approach to a wider range of tabular applications.

This thesis work was conducted within the Ixa group at the HiTZ research

center, University of the Basque Country. The Ixa group has a long-standing history of impactful NLP research, particularly in developing language tools for the Basque language, along with a strong record of contributions to NLP research more broadly. This research was also carried out in collaboration with the EdinburghNLP group at the University of Edinburgh. This collaboration has enabled the development of novel approaches presented in this thesis.

1.1 Motivation

The recent advancements in NLP, driven by large language models, have enabled models to perform a range of complex linguistic tasks with remarkable fluency. However, generating faithful and contextually accurate text from structured data sources like tables remains challenging. Being faithful to the original data is critical in table-to-text generation, where the goal is to create a reliable textual description from a table's contents. Despite advances, current systems often produce hallucinations, that is, content that appears plausible but is factually incorrect or irrelevant (Koehn and Knowles, 2017; Maynez et al., 2020; Bender et al., 2021).

The first part of this thesis addresses this issue by building on previous work in logical forms (LF), which have been shown to improve accuracy when used as an intermediary representation between table data and text (Chen et al., 2020d). This method, however, has traditionally required manual creation of logical forms, which is impractical for large datasets and real-world use cases. To overcome this limitation, we introduce a two-step model, Table-to-Logic-to-Text ($T\mathit{l}T$), that generates LFs automatically. By automating LF generation, this contribution enables large-scale applications of LF-based fidelity improvements, offering a scalable solution to the fidelity challenge in table-to-text generation.

Our initial research into table-to-text generation also revealed the limitations of traditional, text-only approaches in representing complex table structures. Tables in real-world applications often deviate from simple two-dimensional grids, incorporating visual formatting and layout features that convey information in ways that linearization, i.e. converting their textual content into a sequential text format, cannot effectively capture. Recent developments in VLMs, such as Pix2Struct (Lee et al., 2023), have demonstrated promising capabilities in tasks that contain visually represented text. In the second part of this thesis, we explore a multimodal approach to table-to-text generation by treating tables as visual entities. This approach, implemented in the PixT3 model, bypasses the need for textual linearization, allowing us to efficiently capture the structural richness of

tables as they appear visually, which in turn improves the model's ability to handle larger and more complex data structures.

Following these advancements, we further extend the visual perspective of table-to-text generation to broader Table Understanding tasks such as Table Question Answering, Table Numerical Reasoning, or Table Fact Checking. Traditional pre-training objectives for Table Understanding, such as next-token prediction and masked language modeling, are not ideally suited to capture the semantic relationships within tables, where the context may not naturally be correlated with their neighboring cells. Additionally, current efforts in multimodal TU rely on image renderings of text-based representations, which lose much of the visual and stylistic information, making them lossy representations compared to the original visualization. In this final contribution, we introduce a new multimodal TU dataset that preserves the visual integrity of tables, tracing each example back to its original source to capture a lossless view of the table. This dataset enables the application of multimodal TU approaches while retaining the benefits of the PixT3 model, that is, enhancing space efficiency and supporting richer contextual understanding.

The motivation for this research is thus rooted in the need for more reliable, scalable, and visually-aware methods for table-to-text generation and Table Understanding.

1.2 Goals and Research Lines

The main goal of this thesis is to contribute to the field of Table Understanding by developing techniques that enhance fidelity in table-to-text generation, as well as improving table representation to better capture information in tabular data. To achieve this, we establish the following research lines:

[RL1] Improving Faithfulness in Table-to-Text Generation. Explore the use of structured semantics to guide table-to-text generation models in producing descriptions that are faithful to the table data. In this line, we dissect the different components that play a key role in achieving this, such as the grammar used to represent these semantics and the conditioning signals required to build them. We propose the use of automatically generated logical forms to achieve this goal and analyze the impact of content selection in a system like this.

[**RL2**] **Improving Information Representation in Tabular Data.** Explore innovative ways of efficiently encapsulating all information represented across a broader range of table formats. In this line, we propose the use of VLMs to capture information from tables represented as images, highlighting their advantages over traditional textual representations. We also address inherent challenges in this approach, such as capturing the structural dynamics of tabular data to avoid fidelity errors. This requires the development and adoption of novel approaches to understanding table structure in visual tables.

[**RL3**] **Expanding Our Findings to a Broad Range of Table Understanding Tasks.** Establish a foundation for extending our findings in table-to-text to other Table Understanding tasks. In this line, we analyze the state-of-the-art pre-training objectives in TU to build a dataset that can instill generalizable foundational knowledge of Table Understanding into vision-based table language models. This work addresses the shortcomings of current approaches to multimodal Table Understanding, such as the reliance on lossy textual representations of tables.

1.3 List of Scientific Contributions

In this section, we present the scientific contributions resulting from the research conducted in this PhD thesis, along with another first-authored work unrelated to the thesis topic, developed for a separate research project within the group over the course of this PhD.

1.3.1 Research Contributions within this Thesis

[**A.1**] Alonso and Agirre (ESWA 2024) presented in Chapter **3**

Alonso I., and Agirre E. (2024). Automatic Logical Forms improve fidelity in table-to-text generation. In *Expert Systems with Applications (Volume 238, Part D, 15 March 2024, 121869)*.

Abstract: Table-to-text systems generate natural language statements from structured data like tables. While end-to-end techniques suffer from low factual correctness (fidelity), a previous study reported fidelity gains when using manually

produced graphs that represent the content and semantics of the target text called Logical Forms (LF). Given the use of manual LFs, it was not clear whether automatic LFs would be as effective, and whether the improvement came from the implicit content selection in the LFs. We present TlT, a system which, given a table and a set of pre-selected table values, first produces LFs and then the textual statement. We show for the first time that automatic LFs improve the quality of generated texts, with a 67% relative increase in fidelity over a comparable system not using LFs. Our experiments allow to quantify the remaining challenges for high factual correctness, with automatic selection of content coming first, followed by better Logic-to-Text generation and, to a lesser extent, improved Table-to-Logic parsing.[1]

[A.2] Alonso et al. (ACL 2024) presented in Chapter **4**

Alonso I., Agirre E., and Lapata M. (2024). PixT3: Pixel-based Table-To-Text Generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Abstract: Table-to-text generation involves generating appropriate textual descriptions given structured tabular data. It has attracted increasing attention in recent years thanks to the popularity of neural network models and the availability of large-scale datasets. A common feature across existing methods is their treatment of the input as a string, i.e., by employing linearization techniques that do not always preserve information in the table, are verbose, and lack space efficiency. We propose to rethink data-to-text generation as a visual recognition task, removing the need for rendering the input in a string format. We present PixT3, a multimodal table-to-text model that overcomes the challenges of linearization and input size limitations encountered by existing models. PixT3 is trained with a new self-supervised learning objective to reinforce table structure awareness and is applicable to open-ended *and* controlled generation settings. Experiments on the ToTTo (Parikh et al., 2020) and Logic2Text (Chen et al., 2020e) benchmarks show that PixT3 is competitive and, in some settings, superior to generators that operate solely on text.[2]

[1] TlT code, models, and data are available at https://github.com/alonsoapp/tlt

[2] PixT3 code, models, and data are available at https://github.com/alonsoapp/PixT3.

1.3.2 Research Contributions beyond this Thesis

[**A.3**] Alonso et al. (AIM 2024)

Alonso I., Oronoz M., and Agerri R. (2024). MedExpQA: Multilingual Benchmarking of Large Language Models for Medical Question Answering. In *Artificial Intelligence in Medicine (Volume 155, September 2024, 102938)*.

Abstract: Large Language Models (LLMs) have the potential of facilitating the development of Artificial Intelligence technology to assist medical experts for interactive decision support. This potential has been illustrated by the state-of-the-art performance obtained by LLMs in Medical Question Answering, with striking results such as passing marks in licensing medical exams. However, while impressive, the required quality bar for medical applications remains far from being achieved. Currently, LLMs remain challenged by outdated knowledge and by their tendency to generate hallucinated content. Furthermore, most benchmarks to assess medical knowledge lack reference gold explanations which means that it is not possible to evaluate the reasoning of LLMs predictions. Finally, the situation is particularly grim if we consider benchmarking LLMs for languages other than English which remains, as far as we know, a totally neglected topic. In order to address these shortcomings, in this paper we present MedExpQA, the first multilingual benchmark based on medical exams to evaluate LLMs in Medical Question Answering. To the best of our knowledge, MedExpQA includes for the first time reference gold explanations, written by medical doctors, of the correct and incorrect options in the exams. Comprehensive multilingual experimentation using both the gold reference explanations and Retrieval Augmented Generation (RAG) approaches show that performance of LLMs, with best results around 75 accuracy for English, still has large room for improvement, especially for languages other than English, for which accuracy drops 10 points. Therefore, despite using state-of-the-art RAG methods, our results also demonstrate the difficulty of obtaining and integrating readily available medical knowledge that may positively impact results on downstream evaluations for Medical Question Answering. Data, code, and fine-tuned models will be made publicly available[3]

[3] https://huggingface.co/datasets/HiTZ/MedExpQA

2. CHAPTER

Background

This thesis presents a computational approach to Table Understanding (TU) from a natural language perspective. Tables are structured arrangements of information or data, typically organized in rows and columns, though sometimes with more complex structures. They are widely used in communication, research, and data analysis, and can usually be found in print media, handwritten notes, computer software, and other forms of communication. Information conveyed through tables, referred to as tabular data, can consist exclusively of, or a combination of, textual, numerical, or other visually representable data.

There are notable similarities between tabular data and linear natural language. The main and most obvious one is when information appears in textual format. However, regardless of the type of information within a table, tables also contain implicit semantics in the form of contextual relationships between cells that may not be explicitly stated. Tables also come frequently alongside natural language captions to provide additional information, and many tasks involving tables are either supported by or result in natural language utterances.

Therefore, it is a natural consequence that the field of computational Table Understanding is tightly related to Natural Language Processing (NLP) and has traditionally benefited from the developments in the latter field.

2.1 Natural Language Processing

Natural Language Processing is the interdisciplinary field at the intersection of computer science, artificial intelligence, and linguistics, aimed at enabling machines to process, understand, and generate human language. Broadly, NLP tasks range from machine translation, sentiment analysis, and named entity recognition, to more complex activities such as question answering, summarization, and dialogue systems.

This chapter provides an overview of the key advancements in NLP that have contributed to the field of Table Understanding. It is worth noting that both fields have experienced unprecedented growth and attention during the course of this thesis, which has shaped the research goals and approaches throughout its development.

2.1.1 Early Approaches to NLP

Early NLP approaches relied on rule-based systems that used hand-crafted linguistic rules to process language (Weizenbaum, 1966; Winograd, 1972), which were effective in limited contexts but struggled with the ambiguity of natural language. More notably, other approaches like probabilistic n-gram models (Shannon, 1951), followed a data-driven approach to predict the likelihood of a sequence of words based in the n previous words, establishing the defining characteristic of language models (LM). This approach became the cornerstone of consequent works that applied statistical methods to improve tasks like machine translation (Brown et al., 1990). However, these models still had limitations in capturing long-range dependencies.

Neural Networks

The adoption of neural networks and its application on language modeling (Bengio et al., 2000), marked a significant paradigm shift in this field. Recurrent Neural Networks (RNNs) (Mikolov et al., 2010) emerged as an effective architecture for processing sequences by maintaining a hidden state across time steps. However, RNNs suffered from the vanishing gradient problem (Hochreiter et al., 2001), making it challenging to model long-range dependencies in sequences. To address this, Long Short-Term Memory (LSTM) networks Hochreiter and Schmidhuber (1997) introduced a gating mechanism to regulate the flow of information, enabling the retention of relevant data over longer time spans. Gated Recurrent

Units (GRU) (Cho et al., 2014a) provided a simpler alternative with fewer parameters, offering comparable performance to LSTMs. Both Sutskever et al. (2014)'s and Cho et al. (2014b)'s works also introduced the RNN encoder-decoder architecture that became an effective and standard approach for both neural machine translation and sequence-to-sequence prediction in general.

The Attention Mechanism

The introduction of the attention mechanism (Bahdanau et al., 2015) significantly improved sequence modeling, allowing models to focus on relevant parts of the input without relying on strict sequential order. Attention paved the way for more sophisticated architectures, addressing RNNs' limitations and providing better performance in machine translation, text summarization, and related tasks. The attention mechanism laid the foundation for subsequent architectures like the Transformer, enabling a more effective approach to understanding and generating human language.

2.1.2 Transformers and Encoder-Decoder Architectures

Transformer Architecture

The introduction of the Transformer model by Vaswani et al. (2017) was a turning point in NLP. The Transformer replaced recurrent architectures with fully attention-based mechanisms, significantly improving parallelization during training. The original transformer architecture uses an encoder-decoder structure, where both components rely on multi-head self-attention and feed-forward layers, with the decoder also incorporating cross-attention to integrate encoder outputs. This model architecture became the backbone of state-of-the-art systems in NLP, dramatically improving the performance of tasks such as machine translation, text generation, and summarization.

Encoder-Only and Decoder-Only Architectures

While the original Transformer model included both encoder and decoder components, subsequent variations of the Transformer architecture tailored the design for specific tasks. For instance, BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al. (2019a), is an encoder-only architecture designed for tasks that require deep contextual understanding of the entire

sequence, such as sentence classification, named entity recognition, and question-answering. BERT uses a masked language model (MLM) pre-training objective, allowing it to capture bidirectional context, as well as predicting whether two given sentences come one after the other. This set the standard for using encoder-only models in language understanding tasks (Liu et al., 2018; Yang et al., 2019; He et al., 2020).

On the other hand, GPT (Generative Pretrained Transformer), introduced by Brown et al. (2020), and the later LLaMA (Touvron et al., 2023) are Transformer-based decoder-only architectures optimized for text generation tasks. Unlike BERT, these architectures use a left-to-right auto-regressive language model that predicts the next word in a sequence based on the previous context, making it highly effective for tasks such as text generation and completion.

Transfer Learning: Pre-training and Fine-Tuning

One of the most significant advancements in modern NLP has been the adoption of transfer learning. Instead of training models from scratch for each task, models are first pre-trained on large-scale corpora using self-supervised objectives, such as predicting masked tokens (BERT) or the next word (GPT). After pre-training, the models are fine-tuned on specific tasks with significantly less task-specific data. This paradigm, led to substantial improvements across a wide range of NLP tasks, including text classification, named entity recognition, and machine translation.

More recently, as models grow in size and complexity, techniques like Low-Rank Adaptation (LoRA) (Hu et al., 2022), introduce an efficient model adaptation method that reduces the number of trainable parameters by factorizing the weight matrices in transformers into lower-dimensional matrices during fine-tuning. This significantly lowers the computational burden while maintaining high performance on downstream tasks.

2.2 Table Understanding

Table Understanding is the subfield of Natural Language Processing focused on interpreting, processing, and generating information from tabular data. While it shares some similarities with other NLP tasks, TU needs to understand the structure, relationships, and semantics embedded within table formats. The field has gained traction due to the growing number of applications requiring table data processing, from information retrieval and data analytics to summarization and ques-

tion answering. Early works such as RoboCup (Chen and Mooney, 2008), WeatherGov (Liang et al., 2009), and Rotowire (Wiseman et al., 2017a) demonstrated the potential of NLP techniques to handle structured data like tables, paving the way for advances in table-centric tasks and applications.

2.2.1 Table Taxonomy

Tables can vary significantly in format, structure, and complexity. While there is no standardized taxonomy, tables are commonly categorized into three main types based on their structural characteristics:

- **Key-Value Pair Tables**: These are simple collections of key-value pairs, resembling a single row in a 2-dimensional table. An example is a Wikipedia infobox, where each entry consists of a key (attribute) and its corresponding value. The simplicity of this format limits the need for structural understanding beyond key-value matching. Table (a) in Figure 2.1 belongs to this category.

- **Regular Tables**: In this format, tables have a predefined set of columns, with one value per column, forming a matrix-like structure. These tables typically represent relationships where each cell provides a value for a property defined by the column header and corresponding to an entity in the current row. This is the most common table format in tabular datasets due its simplicity and wide use, and can be found in a wide range of sources from documents to spreadsheets. Relational databases also involve strictly two dimensional tables but these are often approached through semantic parsing techniques due to the large amount of rows they typically contain. Table (b) in Figure 2.1 belongs to this category.

- **Irregular Tables**: Also known as hierarchical tables, these tables feature complex two dimensional structures with cells spanning multiple rows or columns. Additional information can be conveyed through style formatting, such as cell background or text formatting. Beyond the typical data found in the previous two categories, these tables can also include other types of visually representable data, such as images. Irregular tables often appear in documents, web pages, and other visually oriented formats, requiring a visual representation for proper consumption. Table (c) in Figure 2.1 belongs to this category.

Willie Park, Jr.	
Name	Willie Park, Jr.
Born	4 February 1864
Died	22 May 1925
Nationality	Scotland
Status	Professional
Masters T.	NYF
PGA	DNP
U.S. Open	CUT: 1919
The Open	Won: 1887, 1889
Hall of Fame	2013

(a) Key-Value Pair Table

Place	Player	Country	Score
1	Willie Park, Jr.	Scotland	151
2	Harry Vardon	Jersey	154
T3	Thomas Renouf	Jersey	156
T3	J.H. Taylor	England	156
T5	Harold Hilton	England	157
T5	David Kinnell	Scotland	157
T7	James Kinnell	Scotland	158
T7	Freddie Tait	Scotland	158
9	Sandy Herd	Scotland	159
10	David Herd	Scotland	160

(b) Table with a Regular Structure

Place	Player	Countr	Score
1	Willie Park, Jr.		151
T2	David Kinnell		157
	James Kinnell		157
Total Scotland			**465**
T3	J.H. Taylor		158
	Harold Hilton		158
Total England			**316**
4	Harry Vardon		159
5	Thomas Renouf		161
Total Jersey			**320**
World Golf Hall of Fame 1974 (member page)			

(c) Table with an Irregular Structure

2.1 Figure – Comparison between the three table structure categories in this thesis.

2.2.2 Task Typology

There are numerous established tasks in the area of Table Understanding. While some tasks share overlapping methodologies and allow for knowledge transfer across them, others present distinct challenges that remain active areas of research. Below, we describe the most actively researched tasks in TU, discussing relevant works, datasets, and evaluation methods.

Complexity Levels in Table Understanding

Before diving into the characteristics of each specific task, it is important to highlight that each task can be approached at different levels of complexity. While many of these complexity factors are specific to individual tasks, some are common to all tasks, including multi-table scenarios, reasoning requirements, hybrid data sources, diverse table structures, output complexity, and table length.

- **Multiple Tables**: While many tasks focus on a single table, increasing the number of tables to process is another factor of complexity that can be applied to many tasks. For example, MultiTabQA (Pal et al., 2023) tackled Table Question Answering across multiple tables, while Zhang et al. (2024b) explored multi-table settings for table-to-text generation.

- **Reasoning Required**: The need to perform reasoning based on table data can also be considered a common complexity factor. This can range from logical reasoning, as explored by Chen et al. (2020e), to mathematical rea-

soning as seen in datasets like FinQA (Chen et al., 2021) and TabMWP (Lu et al., 2023a).

- **Hybrid Source**: The requirement of processing tables alongside other modalities like text, can also increase the complexity of many tasks. For example, HybridQA (Chen et al., 2020c) combines table data with textual passages to answer questions.
- **Table Structure**: Tasks can be carried out on regular or irregular tables, the latter requiring a more challenging understanding of the table's structure. Works like ToTTo (Parikh et al., 2020) and HiTab (Cheng et al., 2022) address this challenge in table-to-text and TableQA tasks, respectively.
- **Output Complexity**: The length and intricacy of the expected output also affect task difficulty. For example, Rotowire (Wiseman et al., 2017a) involves generating long-form text descriptions averaging 337 tokens.
- **Table Length**: The length, in both the number of rows and the number of columns, also poses a considerable increase in the challenge of a task. Model context length limitations have historically kept researchers away from tackling the challenges posed by long tables. However, works such as MATE (Eisenschlos et al., 2021) have addressed the problem of encoding long tables based on the Transformer context limitations of their time.

Regardless of these modifiers, the complexity of tasks also varies among them. When following a training curriculum, low-complexity tasks typically come first, serving as an initial exposure for the model to the domain of tabular data. Higher-complexity tasks are used later to fine-tune the model for more complex downstream objectives and, sometimes, with the aim of improving its ability to generalize to unseen tasks (Li et al., 2023a; Hu et al., 2024; Zheng et al., 2024).

The first stages of the training curriculum typically include tasks centered on fundamental Table Understanding principles, such as semantic comprehension, structural awareness, and relational understanding of the table and its entities. More advance stages, on the other hand, include tasks that not only require basic understanding of table mechanics but also involve performing additional operations, such as table question answering, table-to-text generation, numerical reasoning, or fact-checking.

However, the distinction between these two categories is not always clear-cut, as some low complexity tasks can also serve as final downstream tasks. For example, a TU model designed specifically to parse the table into a different format

may require using structure recognition objectives as its primary objective. On the other hand, more complex tasks can also function as intermediate steps aimed at imparting higher-level knowledge to the model, without necessarily being the final task. For instance, training a model to perform mathematical reasoning may serve as a means to enhance its overall reasoning capabilities.

Table Understanding Tasks

Table Understanding includes a variety of tasks aimed at extracting, interpreting, and reasoning over information encoded in tables. These tasks serve as a foundation for numerous applications, from answering natural language queries based on tables to generating coherent textual descriptions. In this section, we describe key tasks in TU, their methodologies, datasets, and evaluation metrics, highlighting recent advancements and their implications for the field.

Table Question Answering This task is one of the most prominent tasks in Table Understanding and involves answering questions based on the content of a table. The complexity of this task ranges from simple fact lookup to complex reasoning tasks requiring multi-step operations.

Early approaches to Table Question Answering (TableQA) focused on simple table lookups where the answer is a direct extraction from a table cell. For instance, works like TaPas (Herzig et al., 2020) use BERT-based models fine-tuned for TableQA tasks to extract answers from tables directly. TaPas demonstrated the feasibility of using transformer models for table-based tasks by combining structured data with pre-trained language models.

Further approaches perform reasoning over table cells to answer questions requiring aggregation, comparison, or arithmetic operations. Notable works in this direction include TAT-QA (Zhu et al., 2021), which tackles numerical reasoning by combining symbolic operations with deep learning approaches. Similarly, FinQA (Chen et al., 2021) extends TableQA to financial documents, where complex multi-step reasoning is essential. ReasTAP (Zhao et al., 2022) further enhances the reasoning capabilities of question answering models through explicit reasoning chains.

Some approaches involve answering questions based on multiple tables or combining information from tables and other sources such as text. For example, MultiTabQA (Pal et al., 2023) addresses the challenge of question answering across multiple tables by introducing techniques that identify relevant tables and perform joint reasoning over the extracted information.

Finally, other subcategories involve leveraging both tabular and textual data sources to improve accuracy. HybridQA (Chen et al., 2020c) and TableLlama (Zhang et al., 2024a) are examples where models utilize table data combined with additional textual context, requiring models to navigate and integrate information from different modalities.

Notable TableQA datasets include:

- WikiTableQuestions (Pasupat and Liang, 2015): One of the earliest and most prominent datasets for TableQA, featuring questions derived from Wikipedia tables. It serves as a benchmark for evaluating the ability of models to interpret various table structures.
- SQA (Iyer et al., 2017): A dataset of sequences of questions over semi-structured tables, simulating follow-up queries where the context evolves across multiple turns.
- TAT-QA (Zhu et al., 2021): Focuses on numerical reasoning tasks over financial tables. It is designed to test models on more challenging numerical reasoning tasks, including multi-step arithmetic operations.
- HybridQA (Chen et al., 2020c): Combines tables with linked passages to perform question answering over two information sources. The dataset requires models to make use of both structured tabular data and unstructured textual information to answer questions.

Evaluation in TableQA typically uses metrics such as Exact Match (EM) accuracy, which measures whether the model's output exactly matches the reference answer, and F1 score, which accounts for partial matches (more information about these metrics shown in Section 2.4). For models that predict specific table cells or regions as the answer (instead of generating free-form text), the evaluation usually focus on how accurately the model selects the correct cells. Other NLG metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) may also be applied to assess answer quality, especially in cases involving free-form or descriptive responses.

Table Semantic Parsing Table semantic parsing involves translating a natural language query into a structured representation (e.g., SQL query, logical form, or graph) based on the content of a table. This task requires models to understand both the semantics of the query and the table structure to produce accurate representations for database querying or logical reasoning.

Several approaches have been developed for table semantic parsing, like TAPEX (Liu et al., 2022), which follows a pre-training approach in which a model learns to execute SQL-like operations over tables. This work demonstrates that training models to act as a SQL execution engine can enhance their understanding of table semantics. Also, the WikiTableQuestions (Pasupat and Liang, 2015) TableQA dataset was originally aimed at learning semantic parsers that can map natural language questions to logical forms, which can then be executed on tables. Although it is not directly mapped to SQL, it focuses on table-based semantic parsing tasks. Other notable dataset for this task is WikiSQL (Zhong et al., 2017). This is a large-scale dataset containing natural language questions paired with SQL queries over Wikipedia tables. It serves as a primary benchmark for evaluating semantic parsing over tabular data.

Table semantic parsing is evaluated based on the accuracy of generating the correct structured representation (e.g., SQL query). This may involve exact match accuracy against reference representations or measuring execution accuracy, which checks whether the generated representation produces the correct output when executed against the table.

Table Entailment Table entailment, or table fact verification, involves classifying whether a statement is supported or refuted based on the content of a table. This task requires not only table comprehension but also reasoning capabilities to match and validate facts against tabular data.

Notable works and datasets include: TabFact (Chen et al., 2020b), a collection of Wikipedia tables and statements labeled as supported or refuted. It requires models to perform multi-step reasoning and recognize entailment relationships. Eisenschlos et al. (2020) also improve table entailment by adapting TAPAS with data augmentation and table pruning. InfoTabs (Gupta et al., 2020), a dataset that extends the entailment task to sentence-level comparisons with 1D tabular data extracted from Wikipedia Info-boxes. InfoTabs provides a richer set of entailment scenarios, including those requiring relational and numerical reasoning. Finally, Feverous (Aly et al., 2021) is a large-scale fact verification dataset based on Wikipedia, incorporating not only textual claims but also tabular evidence. This dataset challenges models to integrate information from both text and tables.

Table entailment approaches often leverage large language models fine-tuned on entailment tasks, incorporating table-specific representations to understand structured data. Evaluation is performed using accuracy, precision, recall, and F1 scores to measure a model's performance in classifying statements. Human

evaluation may also be conducted for complex entailment cases to assess model reliability in reasoning about table content.

Table Numerical Reasoning Table numerical reasoning involves solving mathematical problems using table data, such as performing arithmetic operations, comparisons, and aggregations. This task is particularly challenging for language models, which often struggle with numerical reasoning.

Notable works and datasets include, TAT-QA (Zhu et al., 2021) and FinQA (Chen et al., 2021), two TableQA oriented benchmarks focused on mathematical reasoning over financial tables, including numerical operations such as summation, subtraction, and percentage calculations. TabMWP (Lu et al., 2023a), a dataset for solving math word problems using table data and challenge models to understand tabular contexts and perform multi-step numerical computations.

Notably, in the Chain-of-Table (Wang et al., 2024) work, the authors introduce a model that, given a query and a table, chooses and programatically performs a series of transformations to simplify the table. These transformations can include actions like calculating the average of a column or summing the values in a row. The goal is to gradually reduce the complexity of the table so that the question can be answered directly, after which the model provides the solution.

Approaches for table numerical reasoning often incorporate symbolic reasoning components to handle arithmetic operations, in addition to language model fine-tuning. Evaluation is typically based on numerical accuracy, and models may also be evaluated on their ability to generate reasoning steps, providing transparency into their decision-making process.

Table-to-Text Generation Table-to-text generation refers to the task of generating descriptive text based on table content. Like the other tasks, there are different variations that raise the complexity from simple table summarizations. One of the pioneering works in this subfield, Rotowire (Wiseman et al., 2017a) proposes generating long-form sport reports from game statistic tables. The length of the desired summaries combined with the complexity of the tables makes this a challenging dataset.

The type of table also plays a key role in defining table-to-text challenges. For instance, in WikiBio, Lebret et al. (2016a) propose using Key-Value Pair Tables to generate biographies of historical figures. Other works like Logic2Text (Chen et al., 2020e) or ReTAG (Ghosal et al., 2023), reduce the length of the target text to focus on generating faithful descriptions that contain facts that are a result of

reasoning over the table data. Meanwhile, approaches like QFMTS (Zhang et al., 2024b) extend summarization to scenarios involving multiple tables.

The task of table-to-text generation presents a challenge, as, in its original definition, it is underspecified. Without a content selection signal to guide the generation, the range of possible verbalizations is so broad that it becomes difficult to determine whether one generated text is preferable to another, as long as both remain faithful to the source data. To address this issue, several works have proposed different settings in which a content selection signal is included in the input to narrow down the possible verbalizations. For instance, ToTTo (Parikh et al., 2020) provides a set of highlighted cells alongside the table, indicating the table content on which the target text is based. Another approach, QTSumm (Zhao et al., 2023), introduces the concept of Query-Focused Summarization (QFS), where a query is added alongside the table to guide the generation towards the desired description. The key distinction between this method and Table Question Answering is that while TableQA requires a direct answer to a question, in QFS, the answer is expected to be formulated as a longer, more descriptive sentence.

Evaluation in table-to-text generation includes traditional text generation metrics like BLEU, ROUGE, and METEOR, which measure the similarity between generated and reference texts. Additionally, metrics specific to table-to-text generation, such as Content Selection (CS) and PARENT (Dhingra et al., 2019), assess the accuracy and relevance of content generated based on the table. Human evaluation of faithfulness, that is, whether the generated text accurately reflects the source table, is also common (Puduppully et al., 2019; An et al., 2022; Zhao et al., 2023).

Table Structure Recognition Table Structure Recognition (TSR) involves identifying and parsing the structure of tables, often presented in visual format in images or documents. It can be considered a multimodal task that lies at the intersection of Table Understanding and Document Understanding, as it requires models to recognize both the visual and structural components of tables. The table knowledge required to solve this task is usually leveraged during the first step of a curriculum training to instill solid foundational knowledge of table structure. Notable works like TURL (Deng et al., 2020) and TUTA (Wang et al., 2021) extend beyond structural recognition to incorporate semantic understanding as well.

Notable TSR datasets include:

- ICDAR 2013 (Gobel et al., 2013) and the follow up ICDAR 2019 cTDaR (Gao et al., 2019) Datasets include a collection of scanned document images

with a wide range of table types and layouts. It is widely used for table detection and recognition tasks.

- PubTabNet (Zhong et al., 2020) is a large-scale dataset containing tabular data in scientific publications. It includes tables extracted from PubMed Central and provides annotations in HTML format that represent the table structure.

- TableBank (Li et al., 2020) is a large-scale dataset containing over 417,000 tables from LaTeX documents and Microsoft Word documents, providing a rich resource for table structure recognition tasks.

Typical metrics for TSR include row/column prediction accuracy and Tree-Edit-Distance-based Similarity when parsing tables into structured formats like HTML or LaTeX. TSR evaluation may also involve assessing the model's ability to recognize complex structural elements such as merged cells or multi-level headers.

Table Representation and Processing

In early works, tables were often represented as triplets or graphs. However, these approaches presented challenges: triplet-based representations were overly simplistic and unable to capture complex table structures, while graph-based representations tended to be overly complicated. Although Graph Neural Networks (GNNs) have been a promising approach for encoding table structures, they have not consistently achieved sufficient results to justify the complexity of the system. Some notable efforts in using GNNs for table encoding include works by Zhang et al. (2020) and Liu et al. (2021), which attempted to leverage GNNs to capture relational information in tables but faced difficulties in scaling and performance.

With the rise of Transformer-based models, many approaches began to represent tables as flat textual linearizations (An et al., 2022; Wang et al., 2022a). This method introduced a trade-off: either all information was represented in a verbose manner, resulting in inefficiencies, or some information was lost, especially in the case of complex, structured tables.

Recent approaches, including the works developed in this thesis, have taken advantage of the latest advancements in Vision Language Models (VLM) to encode tables as images, effectively capturing all information without significant loss in a more efficient manner (Zheng et al., 2024; Deng et al., 2024). This shift

allows for a better representation of the visual and structural aspects of tables while maintaining the fidelity of the original data.

2.3 Vision Language Models

When recurrent neural networks (RNNs) began gaining popularity in natural language processing, convolutional neural networks (CNNs) were experiencing a similar rise in the field of computer vision. The emergence of deep learning saw deep CNNs becoming widely used for image recognition, particularly after Krizhevsky et al. (2012a) demonstrated their superior performance over previous methods on ImageNet classification (Krizhevsky et al., 2012b), and Simonyan and Zisserman (2015) highlighted the importance of their depth and capacity. Early challenges with vanishing gradients were mitigated by introducing skip connections between convolutional layers (He et al., 2016; Xie et al., 2017; Szegedy et al., 2017), establishing CNNs as the default choice for visual representation encoding until recent times.

The transformer architecture is not limited to text processing. Dosovitskiy et al. (2020) adapted an encoder-only transformer for image recognition by segmenting images into non-overlapping patches, converting them into one-dimensional embeddings suitable for the encoder, and adding a classifier head. CNN-based and transformer-based models now show comparable performance in image recognition, with both approaches remaining in use today.

However, the integration of transformers into vision tasks did not stop at image recognition. Multimodal models, which combine information in multiple modalities auch as language and vision, have gained traction by leveraging transformers' ability to handle different types of inputs. LLaVA (Liu et al., 2023b), for example, extends the use of transformers to understand visually-situated language by combining vision transformers (ViTs) with large language models. LLaVA processes visual inputs through a vision transformer that encodes image patches, while the language model generates a unified representation to understand and produce text grounded in visual context.

2.3.1 Visually Situated Language

Traditional research on language and vision has mainly focused on tasks where images and text are treated as separate channels. However, encoding visually represented tables involves a blend of visual and textual elements that require holistic

understanding. Traditional approaches to visually-situated language understanding often relied on task-specific engineering and external tools like OCR, limiting their adaptability and general applicability, although recent efforts are moving towards end-to-end models that reduce these dependencies.

In their work *Language Modeling with Pixels*, Rust et al. (2023) explore a novel approach to visually situated language understanding by treating pixels directly as the input for language modeling, aiming to integrate visual and textual information in a unified framework, without relying on external tools like OCR. Soon after, models such as Dessurt (Davis et al., 2022), Donut (Kim et al., 2022), and Pix2Struct (Rust et al., 2023) demonstrated that end-to-end holistic approaches achieve competitive performance against OCR-based models in document understanding. Following these approaches, recent works like MatCha (Liu et al., 2023a) and UniChart (Masry et al., 2023) have extended these capabilities to include chart understanding and numerical reasoning, allowing these approaches to succeed in areas where OCR-based and unimodal systems have struggled.

2.4 Evaluation

Many NLP and Table Understanding (TU) tasks involve generating coherent, human-readable text. Tasks such as summarization, question answering, and table-to-text generation require models to produce coherent and informative text based on contextual data. These tasks fall under the domain of Natural Language Generation (NLG). The intrinsic ambiguity, variability, and rich semantic context of human languages make evaluating the quality of generated text in such open-ended tasks challenging.

NLG evaluation methods are divided into three categories (Celikyilmaz et al., 2020):

- **Human-Centric Evaluation**: This remains the gold standard for evaluating most NLG tasks and involves human judges assessing the quality of generated texts. In these methods, human evaluators rate and compare generated texts against human-written references or their own knowledge of the language. Some tasks require evaluators to be domain experts, while others only need proficiency in the language of the generated text. Although this form of evaluation is considered the most reliable, it is also more expensive and time-consuming than other methods. Therefore, it is typically used for final evaluations rather than during the development phase.

- **Automatic Metrics**: These methods compare generated texts against gold references, usually relying on metrics based on n-gram overlap, string distance, or lexical diversity. This evaluation approach is widely used due to its high comparability, applicability, and ease of implementation. However, these metrics often fail to replicate human judgment and cannot fully assess many desired characteristics of generated text (Reiter and Belz, 2009; Krahmer and Theune, 2010; Reiter, 2018a; Wiseman et al., 2017a). Despite their limitations, automatic metrics are especially useful during the development phase and as auxiliary measures in the final evaluation alongside human assessment.

- **Machine-Learned Metrics**: These metrics aim to model human judgment, combining the comparability and applicability of automatic metrics with the effectiveness of human evaluation. Machine-learned models compare generated texts against each other or against human-written references, offering a more detailed and reliable assessment of generated text quality (Lu et al., 2023b; Vu et al., 2022).

2.4.1 Human-Based Evaluation Methods

Regardless of the task, machine generated natural language is typically meant to be addressed to human consumption. For this reason, despite recent advancements in machine-learned metrics, human evaluation remains a crucial aspect to consider when assessing NLG systems.

However, human evaluation techniques come with their own set of trade-offs. Evaluating with human judges can be expensive and time-consuming, especially when domain expertise is required. Crowdsourcing platforms such as Amazon Mechanical Turk[1] and Prolific[2] can help mitigate these issues, but they also introduce new challenges such as maintaining quality control (Ipeirotis et al., 2010; Mitra et al., 2015), and increased evaluation costs. Additionally, inconsistency in human evaluation processes makes it difficult for researchers to reproduce experiments and compare results across different systems. Although many publications report human evaluation results (Hashimoto et al., 2019), they often lack crucial details about the evaluation process (Van Der Lee et al., 2019).

An important aspect of human evaluation is the agreement between evaluators. Human assessments always exhibit a certain degree of subjectivity, leading to dis-

[1] https://www.mturk.com
[2] https://www.prolific.com

crepancies between evaluator responses. This rate of mismatch can be measured using inter-evaluator agreement metrics that assess the level of agreement between different evaluators when evaluating the same generated text. High inter-evaluator agreement indicates that the evaluation results are consistent and not heavily influenced by individual biases or subjectivity. Some commonly used inter-evaluator agreement metrics include Cohen's Kappa (Cohen, 1960), which measures the degree of agreement between two raters beyond chance, and Fleiss' Kappa (Fleiss, 1971), an extension of Cohen's Kappa for more than two raters.

2.4.2 Automatic Evaluation Metrics

The use of automatic metrics for evaluating NLG systems involves methods that compare generated texts against reference texts. This is a common evaluation approach due to its ease of implementation and rapid assessment. Automatic metrics can be grouped into five categories (Celikyilmaz et al., 2020): n-gram overlap metrics, distance-based metrics, diversity metrics, content overlap metrics, and grammatical feature-based metrics. This section will focus on the n-gram overlap metrics most relevant to this thesis.

F-Score

The F-Score, is the harmonic mean of precision and recall. Although it is typically used to evaluate classification tasks, this metric is also used in NLG to compare generated and reference n-grams. *Precision*, also called specificity, is the number of overlapping n-grams (tp) divided by the total number of n-grams in the generated text (tp + fp). *Recall*, also called sensitivity, is the number of overlapping n-grams (tp) divided by the total number of n-grams in the reference text (tp + fn).

The F-Score is defined as the harmonic mean of the model's precision and recall. It is possible to adjust the F-Score to give more importance to precision over recall, or vice versa. Common adjusted F-Scores are the F0.5-Score and the F2-Score, along with the standard F1-Score. The F1-Score is defined as:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \tag{2.1}$$

In Table Understanding text generation tasks, such as table-to-text, the F-Score provides an approximation of the quality of the generated sequence produced by a model.

BLEU

BLEU (Bilingual Evaluation Understudy) is one of the most widely used metrics in NLG evaluation. Originally developed to evaluate machine translation (Papineni et al., 2002), this metric calculates the precision of n-grams between the generated text and one or more reference texts, where a BLEU score of 1.0 indicates a perfect match and a score of 0.0 indicates no match. This comparison disregards word order and considers only the occurrence of words in the reference text, meaning a candidate text is not rewarded for generating an excess of relevant words beyond what is necessary. Mathematically, the BLEU score is defined as:

$$p_n = \frac{\sum_s \min(\text{count}(s, \hat{y}), \text{count}(s, y))}{\sum_s \text{count}(s, \hat{y})} \tag{2.2}$$

where $\hat{y}$ is the candidate sequence, y is the reference sequence, s is an n-gram sequence of $\hat{y}$, and $\text{count}(s, \hat{y})$ is the number of times s appears in $\hat{y}$. The BLEU score is then calculated as:

$$\text{BLEU} = \text{BP} \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.3}$$

where N is the total number of n-gram precision scores used (usually $N = 4$), w_n is the weight for each precision score, often set to $1/N$, and BP is the *brevity penalty* to penalize sequences that are too short.

$$\text{BP} = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} \tag{2.4}$$

where c and r are the candidate and reference sequence lengths, respectively.

BLEU also supports the calculation of individual and cumulative n-gram scores. It can use a fixed n-gram or the weighted mean of multiple n-gram scores. The weights can be assigned equally or set differently, giving more importance to certain n-gram scores. In BLEU-4, for example, each of the 1-gram, 2-gram, 3-gram, and 4-gram scores is weighted at 0.25.

BLEU was originally proposed for evaluating machine translation tasks, and it has been reported to correlate well with human judgment for this purpose (Zhang et al., 2004). However, it has been shown that BLEU does not perform as well on tasks outside of machine translation (Reiter, 2018b) lacking awareness of semantic meaning and global coherence (Caccia et al., 2018). Despite these limitations, BLEU is still commonly used for other generation tasks (Rebuffel et al., 2020;

Iso et al., 2020; Gehrmann et al., 2018), as it remains a relatively good proxy for assessing the quality of generated text compared to references.

ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a recall-based metric used primarily for summarization tasks (Lin, 2004). The most common variant, ROUGE-N, calculates the recall of n-grams between the generated text and reference texts:

$$\text{ROUGUE-N} = \frac{\sum_r \sum_n match(gram_{n,r})}{\sum_r \sum_s count(gram_n)} \tag{2.5}$$

where $\sum_n$ sums over all n-grams of length n (e.g., if $n = 2$, the formula measures the number of times a matching bigram is found in the machine-generated and the reference text). If there are more than one reference summaries, the outer summation ($\sum_n$) repeats the process over all reference summaries.

ROUGE places a strong emphasis on recall, which makes it particularly suitable for evaluating summarization quality, ensuring that key content from the reference is included in the generated summary. However, like BLEU, ROUGE has limitations in capturing semantic meaning and often fails to reward well-phrased but slightly different summaries.

Content Selection, Relation Generation and Content Ordering

Content Selection (CS), Relation Generation (RG), and Content Ordering (CO) are a collection of three evaluation metrics introduced by Wiseman et al. (2017a) to evaluate these three specific characteristics of table-to-text generation. Unlike BLEU, which rewards fluency without ensuring that important information is conveyed coherently, these metrics were designed to determine whether a summary accurately represents the desired information.

These metrics are extractive, meaning they involve extracting data from the generated text and comparing it to the reference input data. In order to achieve this they require an information extraction model to obtain data from the generated text. This extracted data comes in the form of entity-value-type triples. These triples are then contrasted against the input table to calculate the following set of values:

- Content Selection (CS): **precision**, **recall** and **F1** score of unique triples extracted from the generated text that are also extracted from the gold ref-

erence text. This measures how well the generated text matches the gold document in terms of selecting which records to generate. This metric targets the *"what to say"* aspect of evaluation.

- Relation Generation (RG): **precision** and **amount of unique triples** extracted from the generated text that also appear in the input table. This metric measures the factual correctness of the generated text targeting both, the *"what to say" and "how to say it"* aspects of evaluation.
- Content Ordering (CO): **normalized Damerau Levenshtein Distance** (Brill and Moore, 2000) between the sequences of records extracted from the gold reference text and the generated text. It measures the order in which the model presents the records it chooses to discuss. This metric targets the *"how to say it"* aspect of evaluation.

PARENT

PARENT (Precision and Recall of Entailed N-grams from the Table) is another metric introduced by Dhingra et al. (2019) to address the limitations of traditional evaluation metrics like BLEU and ROUGE in table-to-text generation tasks. Unlike these metrics, which only measure overlap between generated text and reference, PARENT takes into account both the reference text and the source table. It ignores reference content not present in the table and rewards information in the generated text that is correctly sourced from the table, even if absent in the reference. Specifically, PARENT calculates precision and recall over n-grams that are supported by the table, ensuring that the generated text is evaluated based on both its semantic coverage of reference content and its alignment with the factual information in the source.

2.4.3 Machine-Learned Metrics

Machine-learned metrics represent an evolution in the evaluation of NLG systems, offering a balance between the efficiency of automatic metrics and the quality of human evaluation. Unlike n-gram-based automatic metrics, which focus on lexical overlap, machine-learned metrics can leverage pre-trained models to capture semantic similarity and contextual alignment between generated and reference texts. Additionally, some models can follow a set of guidelines to evaluate text even without a reference text. This allows them to provide a more nuanced assessment that better aligns with human judgment with the benefits of automated

metrics. In this section, we explore key machine-learned metrics used in this thesis.

BERTScore

BERTScore (Zhang et al., 2019) leverages the pre-trained BERT model to compute similarity scores between generated and reference texts. Instead of relying solely on n-gram overlap, BERTScore uses contextual embeddings to compare the semantic similarity between each token in the generated text and the reference text. The metric is computed by aligning each token in the generated text with the most similar token in the reference, thereby providing a precision, recall, and F1 score based on semantic overlap. This makes BERTScore more effective in capturing nuanced differences in meaning, compared to traditional metrics like BLEU. However, its reliance on BERT means that it inherits the biases and limitations of the underlying pre-trained model, which can affect its reliability in some contexts.

BARTScore

BARTScore (Yuan et al., 2021) builds on the BART model, a transformer-based sequence-to-sequence model, to evaluate generated texts by framing the evaluation as a text generation problem. Specifically, BARTScore estimates the likelihood of the reference text given the generated text and vice versa, essentially measuring how well the generated output could reproduce the reference content. This approach allows BARTScore to evaluate fluency, coherence, and semantic correctness in a more integrated manner. BARTScore is particularly effective for tasks like summarisation and machine translation, where it is essential to evaluate both fidelity to the original content and the quality of the generated output. Nevertheless, like other machine-learned metrics, BARTScore's performance is dependent on the quality of the pre-trained model it uses.

2.4.4 Large Language Models as Evaluators

Recently, large language models (LLMs) like LLaMA (Touvron et al., 2023) and commercial products such as GPT-4 (OpenAI, 2023) or Claude (Anthropic, 2023) have been explored as evaluation metrics. These models are capable of understanding complex semantic relationships and generating human-like text, making

them suitable for evaluating NLG outputs. By prompting LLMs to assess the quality of generated text or directly compare it with reference texts, these metrics can achieve a level of evaluation that closely matches human judgment. This approach has the advantage of being adaptable to a wide range of tasks, as LLMs can be fine-tuned or prompted to focus on specific evaluation criteria such as relevance, coherence, or fluency. However, using LLMs as evaluation metrics is computationally expensive, and their responses may vary depending on the prompt, requiring researchers to measure their alignment with human judgment and leading to concerns about consistency and reproducibility.

3. CHAPTER

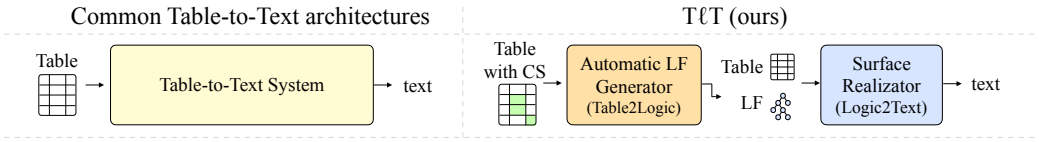
Improving faithfulness in Table-to-Text Generation

3.1 Motivation and Contributions

Table-to-text generation involves producing coherent and accurate textual descriptions from information contained in tables. This process has many applications, such as generating weather forecasts from meteorological data (Goldberg et al., 1994), summarizing sports events (Wiseman et al., 2017b), and creating descriptions from biographical information (Lebret et al., 2016b). A key challenge in these scenarios is ensuring that the generated text remains faithful[1] to the source data. This challenge is particularly pronounced because the task is inherently underspecified, meaning that multiple valid verbalisations can be derived from the same data. As a result, improving fidelity in this task is crucial, especially since many current systems continue to struggle with producing outputs free from hallucinations, i.e., where the generated content inaccurately reflects the input data (Koehn and Knowles, 2017; Maynez et al., 2020; Bender et al., 2021).

In this first work of the thesis, we address the issue of improving fidelity in table-to-text generation by building on the advancements of Chen et al. (2020d), which incorporate logical forms (LFs) as an intermediary step between table data and text generation. Their work demonstrated that incorporating LFs in the table-to-text generation process can significantly improve faithfulness, raising factual correctness from 20% to over 80%. However, their approach required LFs to be manually produced, making their benefits costly and impractical for large-scale,

[1] We use the terms faithfulness, factual correctness, and fidelity interchangeably.



3.1 Figure – Our proposed system to improve fidelity, TlT, (right) alongside a typical table-to-text architecture (left).

real-world applications. This also raised questions about whether the benefits of LFs could be transferred to automatically generated ones. This gap motivates the key contribution of our work:

TlT (short from Table-to-Logic-to-Text), a two-step model that produces descriptions by, first, automatically generating LFs from the table (Table-to-Logic parsing), and then using those LFs alongside the table to produce the text (Logic-to-Text generation). Our model (see Figure 3.1) enables practical usage of LFs while preserving their fidelity benefits. Our research demonstrates that automatically generated LFs can significantly enhance the accuracy of table-to-text systems. Empirical results confirm the advantages of using automatic LFs, showing improvements in both content selection and fidelity when compared to systems that do not utilize LFs.

Additionally, this work provides a detailed analysis of the model's performance, identifying content selection as the most critical factor influencing fidelity, followed by logical form generation and, to a lesser extent, the parsing process. These findings highlight the potential for further advancements in automatic content selection and LF generation techniques, paving the way for more reliable and scalable table-to-text generation systems.

Our findings were published in the Expert Systems with Applications scientific journal (Alonso and Agirre, 2024). All code, models and derived data are also publicly available [2].

3.2 Methodology

In this section, we outline the methodological framework developed to enhance fidelity in table-to-text generation. We begin by formally defining our problem, and then introducing the concept of logical forms, which serve as structured representations of table semantics, and detail their grammar and execution. Following

[2] https://github.com/alonsoapp/tlt

this, we present our two-step Table-to-Logic-to-Text (TlT) model, which consists of generating LFs from table data and subsequently producing textual descriptions guided by these forms. This approach integrates semantic parsing techniques and pre-trained language models to generate logical forms, enabling more accurate and factually consistent text generation from tables.

3.2.1 Problem Formulation

The task of table-to-text generation aims to take a structured table $\boldsymbol{t}$ as input and output a natural language description $\boldsymbol{y} = [y_1, \ldots, y_k]$ where k is the length of the description. Table $\boldsymbol{t}$ is typically reformatted as a sequence of textual records $\boldsymbol{t} = [t_{1,1}, t_{1,2}, \ldots, t_{i,j}, \ldots, t_{m,n}]$ where m and n respectively denote the number of rows and columns of $\boldsymbol{t}$.

3.2.2 Logical Forms

The LFs used in this work are tree-structured logical representations that capture the semantics of a statement related to a table, similar to Abstract Meaning Representation graphs (AMR) (Banarescu et al., 2012). These LFs are built following the grammar rules established by Chen et al. (2020d). Each LF can be executed against a table yielding a result based on the set operations it represents. Since these graphs represent factual statements, the root node is always a boolean operation that returns "True" upon successful execution if the statement is supported by the table. Figure 3.2 provides an example of a table, its caption, and the corresponding logical form.

Logical Form Grammar

Logical forms follow a grammar containing several non-terminal elements (nodes in the graph, some of which can be found in Fig. 3.2), which include:

Stat: Represents boolean comparative statements such as "greater than", "less than", "equals" (denoted as *eq* in the figure), "not equals", "most equals", or "all equals". This forms the root of the LF graph.

C: Refers to a specific column in the input table (e.g., *attendance* and *result* in the figure).

V: Represents specific values, which may either be explicitly stated in the table (e.g., *w* in the figure) or arbitrary values used in comparisons or filters (e.g., *52500* in the figure).

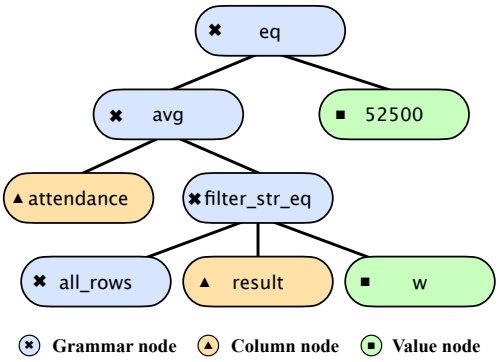
Caption:
1979 philadelphia eagles season

Table:

opponent	result	attendance
new york giants	w 23-17	67000
atlanta falcons	l 14-10	39700
new orleans saints	w 26-14	54000
new york giants	w 17-13	27500
pittsburgh steelers	w 17-14	61500

Statement: In the 1979 Philadelphia Eagles season there was an average attendance of 52500 in all winning games.

LF: eq { avg { filter_str_eq { all_rows ; result ; w } ; attendance } ; 52500 } = True



Content Selection values: 52500, w

3.2 Figure – Example of a table with its caption, a logical form (in linearized and graph forms), its corresponding content selection values and the target statement. Note that *w* in the table stands for *win*. More details in the text.

View: Represents a set of rows selected based on a filter applied across all rows. These filters define conditions on the values within a specific column (e.g., *greater*). In the figure, *all_rows* retrieves all rows, while *filter_str_eq* filters rows containing the substring *"w"* in the *result* column.

N: Performs operations that return numeric values from a specified view and column, such as sums, averages (denoted as *avg* in the figure), and minimum or maximum values, as well as counts.

Row: Selects a single row based on maximum or minimum values within a column.

Obj: Extracts values from columns within rows (either views or specific rows). The most common operation is the *hop* function, which extracts a value from a specific row. For example, *str_hop_first* extracts a string from the first row in a given *View*.

I: Represents values used in ordinal operations within *N* and *Row* rules. For instance, *I* would be set to 2 when selecting the "second highest".

We refer to B.3 for full details. Note that *Stat*, *View*, *N*, *Row*, and *Obj* serve as internal nodes, forming the structure of the LF (shown in blue in the figure), while *C* (columns), *V* (values), and *I* (indices) are always leaf nodes.

Original Grammar Issues

We identified several ambiguities in the original grammar definition that hindered the creation of a consistent framework for generating LFs and ensuring their unambiguous execution.

String ambiguity The first ambiguity affects functions that handle strings. In the LF execution engine proposed by Chen et al. (2020d), these functions are split into two categories: one handles numeric and date-like strings, while the other strictly processes other string values. To resolve this, we explicitly represented these functions as two distinct groups within the grammar: one for numerical and date-like values, and another for non-numeric strings, denoted by the suffix "_str."

Hop operation ambiguity The second ambiguity concerns the *hop* function. When applied to a *Row*, this function extracts the value of one of its columns. Although the grammar specifies that *hop* should only be applied to *Row* objects, in 25% of dataset examples, it is applied to *View* objects, which can represent multiple rows. To address this, we introduced a new function, *hop_first*, designed specifically for these cases.

The updated grammar, which resolves these ambiguities, is presented in B.3. Additionally, we automatically converted all LFs in the dataset to align with the unambiguous grammar. We published the conversion script alongside the code, models and derived data.

Content Selection

To evaluate the effects of content selection independently from the full LFs, we extracted the content of the *Value* nodes from LFs to assess model performance with and without content selection. These extracted values include both those explicitly present in table cells and other values from the LF that are inferred, such as results of arithmetic operations. These values serve as supplementary input for the systems utilizing content selection (CS). We categorize these values as follows:

- **TAB**: Values that appear in table cells either completely or as sub-strings. For example, in Figure 3.2, "w" is a substring of several cells. 72.2% *Value* nodes are of this type.

- **INF**: Values that are inferred but not explicitly present in the table, such as results of arithmetic operations. For instance, *52500* in Figure 3.2 corresponds to the average attendance. 20.8% of *Value* nodes are of this type.
- **AUX**: Auxiliary values that are neither present in the table nor inferred, but used in operations like comparisons (e.g., "All scores are greater than 20"). Only 7.1% are of this type.

In principle, a separate model could be trained to select and generate all necessary content selection values for any table-to-text model. The steps would be as follows: 1) Select values from table cells (TAB); 2) Infer values through operations like averaging or counting (INF); 3) Generate values for use in comparisons (AUX). To differentiate the impact of content selection from LF generation, in this work we focused on using manually derived content selection values from gold reference LFs in the dataset, feeding these into the models. Experiments in Section 3.3.3 show that content selection is critical, and without it, current models fail. The task of developing fully automated content selection mechanisms remains an open area for future research.

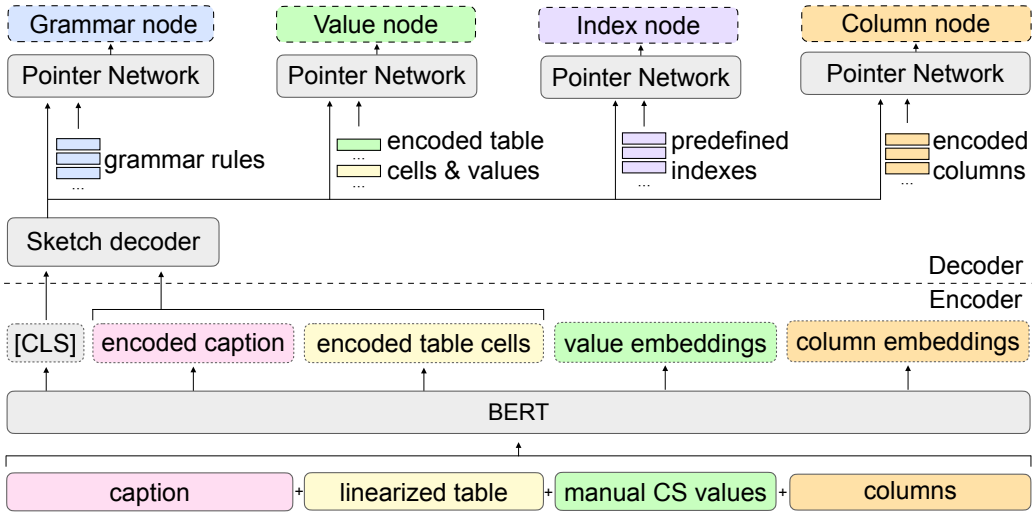
3.2.3 Generating Text via Logical Forms

Our Table-to-Logic-to-Text ($T|T$) system consists of two primary modules in a pipeline. Given a table, its caption, and optionally selected content, the **Table2Logic** module generates a logical form (LF). Using the same table information and the generated LF, the **Logic2Text** module then produces the textual statement.

Table2Logic Model

We frame the Table2Logic model as a semantic parser, adapting the IRNet grammar-based decoder by Guo et al. (2019) to generate LFs. Specifically, we base our implementation on Valuenet by Brunner and Stockinger (2021), a modern revision of IRNet. Both models are Natural Language to SQL semantic parsers that generate grammatically correct SQL queries from natural language descriptions. We adapted the system to produce LFs instead of SQL. The architecture of Table2Logic is shown in Figure 3.3.

During an execution of Table2Logic, we first use a pre-trained BERT encoder (Devlin et al., 2019b) to process the concatenated input, which includes the table's caption, the table content in linearized form, the column names, and, in some



3.3 Figure – The architecture of the Table2Logic system, which consists of two primary components: the BERT encoder and the LSTM-based grammar decoder. The input to the system includes the table's caption, column names, and linearized table content. Additionally, in some configurations, content selection values are incorporated, which are extracted from the gold reference logical forms. The BERT encoder processes these inputs, generating embeddings that are fed into the LSTM decoder. The decoder, guided by four pointer networks, generates the logical form in a two-step process: first, by producing a sketch LF containing only grammar-related nodes, and then by filling in placeholders for values, columns, and indices during a second iteration. The architecture allows constrained decoding to ensure that the generated LF adheres to the predefined grammar structure, ultimately yielding an executable logical form that represents the table data. The False Candidate Rejection (FCR) policy is used during inference to ensure that only logically correct LFs are selected for final output.

configurations, a set of content selection values extracted from the associated gold reference LF. More details on content selection values are provided in Section 3.2.2.

The output embeddings from the *CLS* token, caption tokens, and linearized table values are then passed into an LSTM decoder Hochreiter and Schmidhuber (1997). At each decoding step, the LSTM's attention vector is used by four different pointer networks (Vinyals et al., 2015), each specializing in generating one node type: *grammar*, *Value*, *Column*, and *Index*. We apply a constrained decoding strategy, selecting the appropriate pointer network based on the next node

type required by the LF grammar. The pointer networks use the attention vector alongside a set of embeddings. For *Value* and *Column* nodes, these embeddings consist of content selection values and column encodings produced by BERT. For *Index* and *grammar* nodes, a separate set of predefined embeddings is used.

Following Guo et al. (2019), Table2Logic performs two decoding iterations. The first iteration generates a "sketch" LF using the grammar pointer network, producing only grammar-related nodes (shown in blue in Fig. 3.2). The second iteration fills in the placeholders for *Value*, *Column*, and *Index* nodes using the corresponding pointer networks.

We train the model using a teacher-forcing strategy. During the first iteration, the loss is computed by accumulating the cross-entropy loss for each grammar node generated, given the previous gold reference nodes. This sketch is then used to compute the cross-entropy loss for generating *Value*, *Column*, and *Index* nodes. The model weights are updated based on the sum of these losses.

During inference, beam search is used to generate a set of candidate LFs. We also introduce a False Candidate Rejection (FCR) policy to filter out LFs in the beam that result in a "False" statement, which would lead to factually incorrect text. As explained in 3.2.2, the root node of each LF is a boolean rule that returns "True" upon successful execution if the statement is supported by the table. We exploit this property to discard LFs that, despite being grammatically correct, convey false information, i.e., return "False" upon execution. Only the LF that both executes to "True" and has the highest beam probability is selected. Section 3.3.3 details experiments using FCR.

Logic2Text Module

For text generation, we use the top-performing model from Chen et al. (2020d), which is a GPT-2 large (Radford et al., 2019) fine-tuned to generate text from tables and human generated manual LFs. This model, referred to as Logic2Text, takes as input the table caption, table headers, linearized table content, and logical form. The model generates a sentence strongly conditioned by the semantics of the LF. With Logic2Text, we produce natural language statements based on the automatically generated LFs from the Table2Logic module.

3.3 Experiments

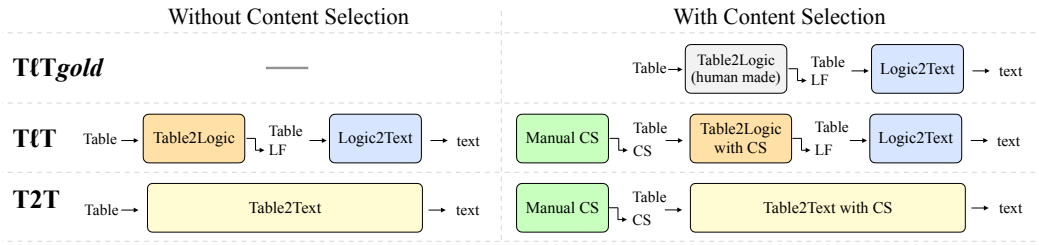
In this section, we present the experiments and evaluation results for our (TlT) system. Our experiments are designed to evaluate the faithfulness of our model, comparing the effectiveness of automatically generated LFs versus manually produced ones. We evaluate multiple configurations of our system across several dimensions, including the impact of content selection, the performance of different model configurations, and the benefits of rejecting false candidates during inference. We use both automatic metrics and human evaluations. We also conduct an extensive qualitiative analysis to identify key areas for future improvement. We begin by introducing the dataset used, followed by the details of the models and their configurations, and conclude with the results of our evaluation.

3.3.1 Dataset

The dataset used in this study was introduced by Chen et al. (2020d) and consists of 4,992 open-domain tables, extracted from the LogicNLG dataset (Chen et al., 2020a). Each table is associated with an average of two human-written statements that describe factual information present in the table. Annotators followed a predefined questionnaire to explain the logical structure behind these statements, which allowed Chen et al. (2020d) to generate Logical Forms for each statement. The final dataset contains 10,753 examples, divided into 8,566 training instances, 1,092 development instances, and 1,095 test instances, all featuring high-quality human-annotated LFs, their corresponding table data, and human-generated statements. These manually generated LFs are referred to as "gold LFs", in contrast to the automatic LFs produced by our system. As noted earlier, the task of generating text from tables is underspecified, meaning there are multiple factually correct statements (and LFs) that could be derived from the same data, even though they may not be explicitly included in the dataset.

3.3.2 Model Configurations

The model configurations used in our experiments are shown in Figure 3.4. All models process all table-related input, which includes the table caption, linearized table content, and column headers. The top row features the upper-bound model $TlTgold$, which takes as input both the table and the manually created gold reference LF. The middle row shows our TlT system, which is composed of two modules: Table2Logic and Logic2Text. Both TlT and $TlTgold$ share the same



3.4 Figure – Model configurations used in the main experiments.

Logic2Text module, but while $TlTgold$ uses manually generated LFs, TlT relies on LFs generated automatically. We evaluate TlT in two configurations: with content selection (TlT) and without content selection ($TlTnoCS$). The hyperparameters for Logic2Text were set according to the defaults from Chen et al. (2020d).

The baseline models (T2T, short for Table2Text) are shown in the bottom row. These models generate text directly from table information, either with or without content selection. For consistency, the baseline models use the same GPT-2 architecture as Logic2Text, but without LFs (during neither training nor inferece). T2T_{noCS} receives only the linearized table as input, while T2T incorporates the same list of manual content selection values used by TlT.

3.3.3 Content Selection Ablation Study

To understand the role of content selection and the impact of filtering out LFs that evaluate to *False* (False Candidate Rejection, FCR), we conducted an ablation study using the development set. Accuracy was evaluated based on strict match with the gold LFs. Both sketch accuracy (where placeholders are used for non-grammar nodes) and full accuracy were measured. While multiple LFs could be valid for a single table, accuracy remains a useful proxy for comparing model performance. The results shown are taken from the best model checkpoints after 50 training epochs, based on full accuracy on the development set. We tuned a few hyperparameters on development data, while most remained at default settings (see B.2 for details).

Table 3.1 summarizes the performance across different subsets of content selection values, with the final row showing the results when FCR was applied. Without FCR, the most important content values were those directly extracted from the table (TAB). The best results overall were achieved when all values were used, although the inclusion of AUX values did not provide much improvement

Model	Sketch	Full
No content selection (TlT_{noCS})	15.0	4.9
AUX	14.0	6.2
INF	28.7	11.0
TAB	42.6	27.3
TAB, INF	56.5	39.3
TAB, AUX	44.3	28.6
TAB, INF, AUX	**58.5**	38.9
TAB, INF, AUX + FCR (TlT)	56.0	**46.5**

3.1 Table – Content Selection ablation results for Table2Logic. We report accuracy (%) over Chen et al. (2020d)'s development set, evaluating both sketch and full versions of gold LFs for different subsets of content selection (CS) and False Candidate Rejection (FCR), as described in Section 3.3.

(in fact, excluding AUX values led to marginally better results).

The use of FCR significantly boosted the accuracy of full LFs, demonstrating that this method helps filter out "False" LFs that would otherwise lead to incorrect statements.

Although the overall accuracy of TlT might appear low, it is important to remember that the gold LFs represent only a subset of the possible correct LFs. As we will show in later sections, the LFs produced by TlT are of high quality, despite their lower measured accuracy.

Additionally, we performed an ablation where table information was removed, providing the model only with the content selection data. Both sketch accuracy and full accuracy dropped significantly (to 50.3% and 42.7%, respectively), highlighting the importance of including table data, even when content selection is available.

3.3.4 Automatic Evaluation

For the automatic evaluation, we compared the generated descriptions to the reference descriptions in the test split using n-gram overlap metrics. Table 3.2 presents the results for BLEU-4 (B-4) (Papineni et al., 2002), ROUGE-1, 2, and L (R-1, R-2, and R-L) (Lin, 2004), as well as two metrics that capture semantic similarity: BERTscore (BERTs) (Zhang et al., 2019) and BARTscore (BARTs) (Yuan et al., 2021).

Model	B-4	R-1	R-2	R-L	BERTs	BARTs
T2T$_{noCS}$	16.8	37.7	19.3	31.6	88.8	-4.04
TlT_{noCS}	15.6	39.0	18.9	32.2	87.9	-4.03
T2T	26.8	55.2	31.5	45.7	91.9	**-2.98**
TlT (ours)	**27.2**	**56.0**	**33.1**	**47.7**	**92.0**	-2.99
TlT_{gold}	31.7	62.4	38.7	52.8	93.1	-2.65
TlT_{gold}*	31.4*	64.2*	39.5*	54.0*	-	-

3.2 Table – Automated n-gram similarity metrics for generated textual descriptions on the test set. Metrics include BLEU-4 (B-4), ROUGE-1, 2, and L (R-1, R-2, R-L), BERTscore (BERTs), and BARTscore (BARTs). The last two rows represent upper-bound results, which use manual LFs. Results marked with * are from Chen et al. (2020d). Both BERTs and BARTs reflect f1 scores, with higher BARTscore values indicating better performance.

The results show that both the baseline (T2T$noCS$) and our system ($TlTnoCS$) perform poorly without content selection. However, when content selection is incorporated, performance improves by around 10 points in all metrics for both T2T and TlT. The use of automatically generated LFs in TlT provides additional gains over the T2T system, yielding at least one point higher across all metrics. If TlT had access to the correct LFs, the results would improve by an additional four points, as shown by the $TlTgold$ results. Notably, our $TlTgold$ results closely match those reported by Chen et al. (2020d), with only minor variations, likely due to weight differences between the model released by the authors and ours.

3.3.5 Human Fidelity Evaluation

To assess the fidelity of the generated descriptions, we conducted a manual evaluation with three models: the baseline T2T, our TlT model, and the upper-bound TlT_{gold}. We randomly selected 90 tables from the test set, generating a description for each table with all three models. Each evaluator was given 30 sentences, along with the corresponding table and caption, and asked to determine whether the description was true, false, or nonsensical in relation to the table and caption. A group of eighteen volunteer researchers, who were not involved in the project, performed the evaluations. Fleiss' kappa (Fleiss, 1971), a statistical measure for assessing agreement among multiple raters, was used to evaluate inter-rater con-

sistency. The kappa value of 0.84 indicated strong agreement among the evaluators, and examples with disagreements were discarded.

The results of the human fidelity evaluation are presented in Table 3.3. Our findings show that the fidelity results for $TlTgold$ are consistent with the values reported by Chen et al. (2020d). For completeness, we also include the results for T2T$noCS$ from their paper, which are comparable to the results presented here.

Human fidelity evaluations showed much larger differences between models than automatic metrics, which can be attributed to the limitations of n-gram overlap metrics (such as BLEU and ROUGE) in evaluating the semantic and pragmatic quality of text. These metrics often fail to capture deeper meaning, leading to scenarios where a model's output may have high overlap with the reference text but still be factually incorrect (Zhang et al., 2019). Furthermore, such metrics may not correlate well with human judgments, which can result in high scores for grammatically correct but semantically flawed text (Moramarco et al., 2022).

From these results, we can estimate the individual contributions of various model components to fidelity:

- **Manual content selection** adds 24 points (T2T$noCS$ vs. T2T);
- **Automatic LFs** contribute an additional 30 points (T2T vs. TlT);
- **Manual LFs** provide a further 7 points (TlT vs. $TlTgold$);
- **Perfect Logic2Text generation** could yield 18 more points (TlT_{gold} vs. 100%).

These results confirm the significance of automatically generating LFs to boost fidelity, with the largest improvement being 30 points, leading to a 67% improvement over models that do not use LFs. The remaining gaps highlight areas for future research, such as improving automatic content selection (24 points), enhancing Logic2Text generation (18 points), and refining Table2Logic parsing (7 points). In the following section, we explore the errors in the latter two components in more detail.

3.3.6 Qualitative Analysis

We also performed a qualitative analysis of the failure cases from both the Table2Logic and Logic2Text modules, as well as examined instances where factually correct descriptions were generated from LFs that differed from the gold LFs.

Model	Faithful	Unfaithful	Nonsense
T2T$_{noCS}$*	20.2*	79.8*	-
T2T	44.9	49.3	5.8
TlT (ours)	**75.0**	**20.3**	**4.7**
TlT_{gold}	82.4	13.51	4.1

3.3 Table – Human evaluation of fidelity across three model configurations using 90 test samples. The table shows the percentage of generated sentences classified as Faithful, Unfaithful, or Nonsense by human evaluators. Cases with complete disagreement between evaluators were discarded. Results marked with * are from Chen et al. (2020d).

Table2Logic

We analyzed the LFs generated by TlT in the development set that did not match their gold LF counterparts. It is important to note that a generated LF can still be valid even if it does not match the gold LF. For this analysis, we traversed each LF left to right, identifying the first node that deviated from the gold standard. Table 3.4 lists the most frequent discrepancies, ordered by frequency.

The most common differences involved *Stat* nodes, where the generated comparison was different from the gold reference. Column and row selections were also frequently mismatched, even when the system had access to content selection values. These three types of nodes were responsible for the majority of deviations. Less frequent differences involved generating alternative comparison or arithmetic operations.

Logic2Text

In cases where descriptions were generated from gold LFs (TlT_{gold}), the faithfulness score was 82%. To better understand the 18% of cases that were incorrect, we sampled examples for further analysis. Full examples, including the table, caption, gold LF, and generated description, can be found in Appendix B.4. The errors were grouped into three main categories:

Comparative arithmetic: In 40% of errors, Logic2Text incorrectly represented comparative arithmetic rules. For example, it might incorrectly state that a value was smaller than another when the LF indicated it was larger. Additionally, Logic2Text often omitted qualifiers such as "roughly" or "most", leading to in-

	Fr.	Total	Confusions
Stat	0.38	0.13	greater → less all equals → most equals equals → and
C	0.25	0.19	column 3 → column 0 column 1 → column 0
Row	0.16	0.02	row 0 → row 2 row 2 → row 0 row 2 → row 1
View	0.11	0.20	filter_greater → filter_less filter_greater → filter_eq filter_eq → all_rows
N	0.05	0.03	sum → avg avg → sum
Obj	0.03	0.26	str_hop → num_hop num_hop → str_hop
V	0.01	0.16	value 72 → value 73 value 70 → value 71
I	0.01	0.01	1 → 0

3.4 Table – Distribution of node type discrepancies between TlT and gold LFs. "Fr." indicates the frequency of node types in mismatched LFs, while "Total" represents their overall frequency in gold LFs. The rightmost column lists the most frequent confusions (i.e., nodes generated by TlT compared to their gold LF counterparts).

correct statements of equality.

We hypothesise that these errors may be linked to the limited number of parameters in the model. With 774M paremeters, GPT-2 large was starting to be considered medium size by the time we conducted this study, however althought to the tiem of writing, current LLMs with 7B, 13B, and even 70B parameters still struggle with mathematical reasoning. Thus we this might not be related as much to the parameter count but more to the architecture or pre-training objective itself.

We originaly hypothesized that these errors may have been linked to the limited number of parameters in the model. However, at the time of writting this thesis, GPT-2 large, with 774 million parameters, is already considered relatively small compared to newer large language models (LLMs) with 7B, 13B, or even 70B parameters. Despite their increased size, even these more recent models still face challenges in mathematical reasoning. Suggesting that the issue may not be tied strictly to model size, but rather to architectural limitations of LLMs or the objectives used during pre-training, which may not sufficiently capture the nuances of comparative arithmetic reasoning. Another contributing factor could be

LF difference	Sentences
Similar structure, semantically equivalent	TlT: In the list of Appalachian regional commission counties, Schoharie has the highest unemployment rate. **Human**: The appalachian county that has the highest unemployment rate is Schoharie.
Similar structure, semantically different	TlT: Dick Rathmann had a lower rank in 1956 than he did in 1959. **Human**: Dick Rathmann completed more laps in the Indianapolis 500 in 1956 than in 1959.
Different structure, semantically different	TlT: Most of the games of the 2005 Houston Astros' season were played in the location of arlington. **Human**: Arlington was the first location used in the 2005 Houston Astros season.
Simpler structure, more informative	TlT: Aus won 7 events in the 2006 asp world tour. **Human**: Seven of the individuals that were the runner up were from aus.

3.5 Table – Examples of faithful sentences generated by TlT from intermediate LFs that do not match the corresponding gold LF.

the relatively low frequency of certain comparative rules in the dataset, with only 44% of LFs involving any of the 22 comparative arithmetic rules. Importantly, similar errors also occur in models that do not use LFs, suggesting these issues are not exclusive to our system.

LF omission: In 33% of cases, Logic2Text omitted entire branches of the LF, resulting in incomplete or inaccurate sentences. This often caused the generated sentence to incorrectly refer to all data points, rather than the subset specified in the LF.

Verbalization: Verbalization errors accounted for 27% of the mistakes, including misspellings or incorrect word choices. For example, Logic2Text might generate a name like *foulisco* instead of the correct *francisco*.

These errors likely stem from the fact that Logic2Text is based on a general-purpose language model (GPT-2). While these models excel at generating fluent text, they may not always faithfully reflect the data in the input LF, even after fine-tuning. Some of these issues may also arise from the low frequency of certain operations in the training set. While the 18% error rate for TlT_{gold} is lower than that of non-LF models, it suggests there is still room for improvement.

Implications of Divergent LF Production from Gold Reference LF

Although our Table2Logic system achieved only 46% accuracy when compared to gold LFs (as shown in Table 3.1), the descriptions generated from these LFs were highly faithful, with a 75% fidelity score, only 7 points lower than descriptions based on gold LFs. This may seem counterintuitive, but it demonstrates that a system can generate correct and faithful LFs that differ from the gold reference by focusing on a different aspect of the table data.

To further investigate, we manually examined cases where TlT generated faithful descriptions using LFs that deviated from the gold standard. In all such cases, the generated LFs were correctly structured and faithfully represented the table data. Table 3.5 provides a sample of these outputs, with full examples available in B.5.

We categorized the discrepancies as follows: 69% of cases involved LFs with a similar structure to the gold references, but with key differences in *Value* or *Column* nodes. In 15% of cases, the LFs were semantically equivalent to the gold references, despite structural differences. The remaining 16% involved LFs with a different structure that still faithfully represented the table data.

This analysis highlights the limitations of reference-based evaluations, where LFs that diverge from the gold standard may still produce accurate and useful descriptions. As such, the 46% accuracy score underestimates the true quality of the generated LFs and the corresponding descriptions. In some instances, the descriptions generated by TlT were more concise and informative than those based on gold LFs, further demonstrating the potential of the system.

3.4 Conclusions

In this chapter, we introduced TlT , a system that, given a table and selected content, first generates logical forms and then produces a textual statement. This work demonstrates, for the first time, that the automatic generation of LFs enhances performance across multiple automatic evaluation metrics, and significantly improves factual correctness based on human evaluation. We also conducted a detailed analysis to separate the contributions of content selection and the formalization of outputs as LFs, finding that the latter has a more pronounced impact on fidelity.

This work contributes to the field by enabling table-to-text applications to leverage the benefits of automatically generated, factually verifiable logical forms without the need for creating them manually. These benefits include a 67% im-

provement in fidelity compared to baseline models, as well as the introduction of an intermediate formal representation in the text generation process. This intermediate step allows for the automated validation of factual accuracy prior to generating the final natural language output, which is crucial for many table-to-text applications where maintaining faithfulness is crucial.

Our analysis also revealed that the most significant potential for performance improvement lies in the content selection process. Enhancements in logic-to-text generation and table-to-logic generation would yield additional, though smaller, gains in fidelity. Future research will focus on developing automated content selection, which we believe can be learned from patterns in user preferences found in training data. Furthermore, recent advances in semantic parsing, such as the use of larger pre-trained language models (BigScience Workshop, 2022; Zhang et al., 2022; Touvron et al., 2023), can be integrated into our system to further enhance the role of LFs in improving fidelity. Finally, enhancing the model's capacity for mathematical reasoning, particularly to improve its handling of comparative clauses would also be an interesting line for future research.

4. CHAPTER

Pixel-based Table-To-Text Generation

4.1 Motivation and Contributions

The findings in our previous chapter proved the use of logical forms to be useful and promising when applied to regular tables. However, after examining numerous tables from different datasets, we discovered that many real-world tables do not conform to such table formats but rather to irregular ones. Therefore, in order to develop a system applicable to all tables, we needed to explore a new paradigm for representing them.

We follow the intuition that tables found in documents and web-pages are ultimately intended to be consumed visually. Authors often take advantage of the structural and formatting freedom of these domains to convey information in ways that go beyond the limitations of regular table formats (See Figure 4.1 for a comparison between regular and irregular tables). As a result, many tables deviate significantly from the traditional matrix-like format, making traditional approaches either impractical or suboptimal.

Table-to-text generation models typically transform tables into text sequences through linearization, a method that inevitably introduces redundancies and results in excessively long inputs (Figure 4.2 offers an example on how a table is typically linearized as text). This issue has been recognized in numerous prior works, ranging from early template-based methods (Wiseman et al., 2018) to more advanced neural models that attempt to respect the table's structure through explicit content planning or contrastive learning techniques (Su et al., 2021; An et al., 2022; Chen

Place	Player	Country	Score
1	Willie Park, Jr.	Scotland	151
2	Harry Vardon	Jersey	154
T3	Thomas Renouf	Jersey	156
T3	J.H. Taylor	England	156
T5	Harold Hilton	England	157
T5	David Kinnell	Scotland	157
T7	James Kinnell	Scotland	158
T7	Freddie Tait	Scotland	158
9	Sandy Herd	Scotland	159
10	David Herd	Scotland	160

(a) Table with a Regular Structure

Club	Season	League			Continental		Other	
		Division	Apps	Goals	Apps	Goals	Apps	Goals
RubinKazan	2011-12	Russian Premier League	0	0	0	0		
	2012-13		0	0			1	0
	Total		**0**	**0**	**0**	**0**	**0**	**0**
Neftekhimik Nizhnekamsk (loan)	2012-13	Russian FNL	6	0			6	0
	2013-14		13	1			15	0
	Total		**19**	**1**	**0**	**0**	**0**	**0**
Khimik Dzerzhinsk	2014-15	Russian FNL	9	1			11	1
Lokomotiv Plovdiv	2016-17	First League	33	3			36	3
	2017-18		29	3			31	3
	2018-19		2	0			2	0
	Total		**64**	**6**				
Career total			**83**	**7**	**0**	**0**	**0**	**0**

(b) Table with an Irregular Structure

4.1 Figure – Comparison between regular and irregular table formats.

et al., 2023b). However, despite these advancements, most approaches still treat tables as sequences of tokens, failing to fully leverage their structural and spatial properties. Previous efforts to treat table-to-text generation as a multimodal task have been limited, often focusing on converting table images into textual tokens using OCR-based systems (Dash et al., 2023), a noisy process that reduces the table to a secondary modality.

Vision language models (VLMs) have made significant progress in handling visually-grounded text understanding. Pix2Struct (Lee et al., 2023), for instance, has demonstrated how visual models can be applied to language tasks without relying on OCR, showing significant improvements in tasks like chart question answering (Masry et al., 2022) or Visual Question Answering. However, their use in structured data generation remained underexplored.

Given the limitations of text-based approaches and the increasing capabilities of VLMs in processing visually represented language, we proposed to rethink table-to-text generation as a visual recognition task. This shift allowed us to avoid the inefficiencies of string-based table representations and instead treat tables as visual objects, which naturally preserves their structure and compactness. Our work builds on recent advancements in VLMs, such as Pix2Struct (Lee et al., 2023), by extending their applicability to the table-to-text domain. By leveraging pixel-based models, we introduce PixT3, a multimodal table-to-text generation model that operates directly on the visual representation of tables. This approach addresses the challenges posed by linearization and context size limitations, making it suitable for both open-ended and controlled generation settings.

In this work, we present PixT3, the first table-to-text generation model to generate text directly from tables represented as images. This pixel-based approach

allows us to bypass the limitations of linearized table representations, improving the model's ability to handle larger and more complex table structures.

To further enhance the model's understanding of table layouts and their content, we propose a new self-supervised learning objective that reinforces structure-awareness of visually represented tables during training. This ensures that the model captures essential relationships within tables without requiring explicit content selection.

Through extensive experiments on the ToTTo (Parikh et al., 2020) and Logic2Text (Chen et al., 2020e) datasets, we demonstrate that PixT3 outperforms state-of-the-art models in tasks where the table is part of the input, in both open-ended and controlled scenarios, while remaining competitive when only provided with the context-selected values of the table.

Finally, we introduce a new dataset derived from Logic2Text, specifically designed to evaluate the generalization abilities of table-to-text models. This dataset enables testing of new approaches on the complex logical reasoning within tables.

4.2 Methodology

4.2.1 Problem Formulation

As we described in the previous section, we define the task of table-to-text generation as the task that involves converting a structured table $\boldsymbol{t}$ into a natural language description $\boldsymbol{y} = [y_1, \ldots, y_k]$, where k is the length of the description. Typically, the table $\boldsymbol{t}$ is reformatted as a sequence of textual records $\boldsymbol{t} = [t_{1,1}, t_{1,2}, \ldots, t_{m,n}]$, with m and n representing the number of rows and columns, respectively.

In this occasion, we approached this task from a visual recognition perspective, where the input table is treated as an image $\boldsymbol{x}$.

The input image is reshaped into a sequence of patches, analogous to linguistic tokens. More formally, for an input image $\boldsymbol{x} \in R^{H \times W \times C}$ and patch size p, we create N image patches denoted as $x_p \in R^{N \times (P^2 \cdot C)}$. Here, (H, W) represents the resolution of the original image, C is the number of channels, (P, P) represents the resolution of each image patch, and $N = \frac{HW}{P^2}$ is the resulting number of patches, which effectively serves as the input sequence length. The model was designed to autoregressively estimate the conditional probability of a text sequence given the source image, as described by the following equation:

Table Title: Shuttle America
Section Title: Fleet

Aircraft	Total	Orders	Passengers				Operated for	Notes
			F	Y+	Y			
Embraer E170	5	-	6	16	48	70	United Express	transferred to Republic Airline
	14	-	9	12		69	Delta Connection Delta Shuttle	2 planes on wet lease from Republic Airline
Embraer E175	15	-	12	12	52	76		
Total	**35**	-						

Linearized Table: <page_title> Shuttle America <page_title> <section_title> Fleet <section_title> <table> <row> <cell> Aircraft <cell> <cell> Total <row_header> Aircraft <row_header> <cell> <cell> Orders <row_header> Aircraft <row_header> <row_header> Total <row_header> <cell> <cell> Passengers <row_header> Aircraft <row_header> <row_header> Total <row_header> <row_header> Orders <row_header> <cell> <cell> Operated For <row_header> Aircraft <row_header> <row_header> Total <row_header> <row_header> Orders <row_header> <row_header> Passengers <row_header> <cell> <cell> Notes <row_header> Aircraft <row_header> . . . . . .

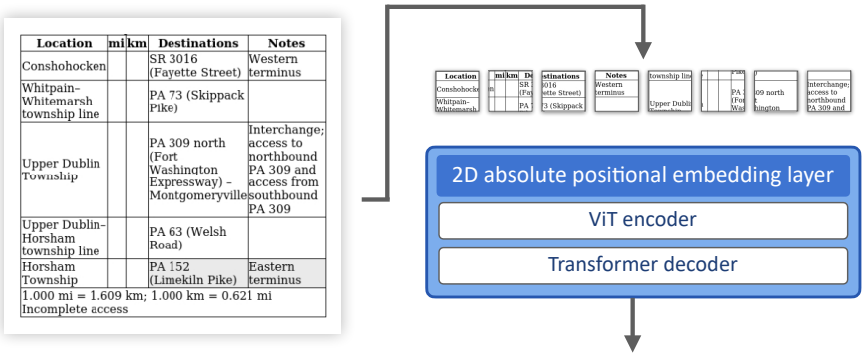
Target Description: Shuttle America operated the E-170 and the larger E-175 aircraft for Delta Air Lines.

4.2 Figure – Example of table-to-text generation taken from the ToTTo dataset (Parikh et al., 2020). In the controlled setting, a natural language description is generated only for highlighted (yellow) cells. The table is linearized by encoding each value as a (Column, Row, Value) tuple. We only show the first row, for the sake of brevity.

$$P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \prod_{i=1}^{n} P(y_i|\boldsymbol{y}_{<i},\boldsymbol{x};\boldsymbol{\theta}) \tag{4.1}$$

where $\boldsymbol{\theta}$ represents the transformer parameters, and $\boldsymbol{y}_{<i}$ refers to the words decoded so far.

As we mentioned in previous chapters, the task of table-to-text generation is challenging from a modeling perspective due to its inherent under-specification. Without a clear content selection signal, the range of possible verbalization is broad, making it difficult to determine which generated text is preferable as long as both remain faithful to the source data. To address this, the ToTTo dataset (Parikh et al., 2020) provides highlighted cells within the table, indicating the specific content that should be verbalized. In this work we followed follow ToTTo's approach and we defined three distinct generation settings to represent the infor-



4.3 Figure – Overview of PixT3 generation model.

mation provided to the model (see Appendix C.2 for visualization):

- **Tightly-controlled setting (TControl)**: The model is provided with only the highlighted cells, excluding the rest of the table. This setting has been commonly used in recent benchmarks (Wang et al., 2022b; An et al., 2022; Chen et al., 2023b; Su et al., 2021; Kale and Rastogi, 2020).
- **Loosely-controlled setting (LControl)**: The model is given both the highlighted cells *and* the entire table. This was the originally intended setting for the ToTTo dataset (Parikh et al., 2020).
- **Open-ended setting (OpenE)**: The model is provided with the entire table without any highlighted content.

4.2.2 The PixT3 Model

PixT3 is an image-encoder-text-decoder model that uses Pix2Struct (Lee et al., 2023) as its backbone. It was designed to process tables rendered as images and generate corresponding textual descriptions (see Figure 4.3). Pix2Struct is a ViT model pretrained on 80 million webpage screenshots extracted from URLs in the C4 corpus (Raffel et al., 2020). The model divides each input image into 16×16 pixel patches, creates embeddings for each patch, applies positional encodings, and feeds them into a Transformer encoder (Vaswani et al., 2017) to process the tabular data.

During the development of Pix2Struct, the model underwent an initial warm-up phase following a reading curriculum (Rust et al., 2023; Davis et al., 2022), where it learned to transcribe image-rendered text. This step helped instill basic text recognition capabilities and improved both training stability and fine-tuning performance. In the second, and main, pretraining phase, Pix2Struct was pre-trained with a screenshot parsing objective, generating simplified HTML subtrees for areas within a bounding box in the webpage screenshots. Additionally, a BART-like (Lewis et al., 2020) objective was incorporated, where 50% of the input text was masked, and the model had to reconstruct it during the HTML subtree generation process.

Beyond its text recognition proficiency, one of the key advantages of using Pix2Struct as the backbone for our model, rather than other end-to-end text recognition VLMs like Donut or Dessurt (Kim et al., 2022; Davis et al., 2022), is its ability to handle variable input resolutions and aspect ratios. Pix2Struct automatically resizes input images (up or down) to fit the maximum number of fixed-size patches within the sequence length, using 2-dimensional positional embeddings to accommodate different resolutions and aspect ratios without distorting the image. This flexibility is crucial when processing table images of varying sizes.

For training PixT3, we followed a similar curriculum. First, we initialized its weights using those from Pix2Struct. Next, we continued pretrained the model on a novel structure recognition task to improve its notion of table structures (more on this in Section 4.2.4). Finally, we fine-tuned the model on table-to-text generation datasets, such as ToTTo (Parikh et al., 2020), using a task-specific supervised objective.

4.2.3 Table-to-Image Rendering

To transform ToTTo serialized tables into images, we parsed tables into HTML format and rendered as images, including metadata such as Wikipedia page titles or section headers when available. We rendered each table into three images, each corresponding to one of the generation settings (TControl, LControl, OpenE) outlined earlier in Section 4.2.1 (see Appendix C.2 for examples).

While Pix2Struct supports variable input resolutions, handling extremely large images is computationally expensive. Following the approach in Lee et al. (2023), we limited the maximum input length to 2,048 patches (each of size 16×16 pixels), corresponding to a maximum image size of 524,288 pixels. In datasets like ToTTo, approximately 41.74% of the tables exceeded this size (see Figure C.1 in Appendix C.1), with 5% of the tables exceeding 8.3M pixels (32,768 patches). To

mitigate the performance issues caused by downscaling all large images, we introduced a truncation method based on a maximum downscaling factor γ. Images were first scaled down to $\gamma\%$ of their original size and then truncated from left to right to fit within the 2,048-patch limit. The optimal value for γ was determined empirically and was set to 0.39 (see Appendix C.3).

4.2.4 Structure Learning Curriculum

Pix2Struct, being a general-purpose visual language understanding model, lacks specific knowledge about table structures. Tables often exhibit a variety of visual presentations, such as spanning multiple rows or columns, irregular spacing and alignment, and diverse formatting styles. Additionally, tables follow structural conventions, where cells are generally related to others in the same row or column. These challenges have led to the development of dedicated table structure understanding techniques (Jin et al., 2023; Wang et al., 2022b) in text-based settings but cannot be directly applied to images.

To address this, we developed a structure learning curriculum by continuously pretraining PixT3 on an intermediate task that exposed the model to the conventions governing table layouts. Below, we outline this intermediate task, the corresponding dataset, and the self-supervised learning objective.

Dataset for Intermediate Training While existing datasets like ICDAR2021 (Kayal et al., 2021) and TableBank (Li et al., 2019) are representative of table parsing tasks, they primarily focus on scientific tables, which differ significantly from the Wikipedia tables found in ToTTo (Parikh et al., 2020), especially in terms of size and spanning cells. As a result, we created a synthetic image-to-text dataset tailored specifically for our task, using the table rendering pipeline described in Section 4.2.3. This process is flexible and can be adapted to other domains.

The structure of each table (size, column, and row spans) was randomly determined following on the distribution of these values in the ToTTo training set tables. Tables were limited to a maximum of 20 columns and 75 rows. Table cells were populated with random combinations of English alphabet characters and digits, serving as identifiers for the table cells rather than meaningful values (see Figure 4.4). The dataset consists of 135,400 synthetic tables: 120,000 for training, 7,700 for validation, and 7,700 for testing.

Table:

oY	io	HG	eG2S
Z4ikU	O1	aRU	mubk6
URa	dAF		I
I86	GAe	Ob	sUr5
L1	3	Vf1	Svaq2

Target:

<<<dAF><<<URa><I>>><<<io><O1><GAe>
<3>><<HG><aRU><Ob><Vf1>>>>

4.4 Figure – Synthetically generated table with a highlighted cell and corresponding pseudo-HTML target sequence (for self-supervised objective). Cells within the target sequence are highlighted in the table with a colored background. For details on the structure of the target, please refer to Appendix C.4.

Self-supervised Objective Common masking objectives used in language models, such as those found in BERT (Devlin et al., 2019a), do not transfer well to table-to-text tasks because table values are not naturally correlated with their neighboring cells. This makes it difficult to predict a masked cell based solely on its context. Early experiments with rearranging cells to create correlations did not improve downstream task performance (see Appendix C.4). Additionally, pretraining with table linearization scales poorly with large tables (Chen et al., 2023a), resulting in slow pretraining times.

In this work we proposed a self-supervised objective that encouraged PixT3 to capture relationships between cells in a table while generating a minimal number of tokens. Specifically, a random cell in a synthetic table was visually highlighted, and the model was trained to produce a sorted list of cells within the same row and column (see Figure 4.4). This objective captured a loosely notion of table structure, encouraging the model to focus on the spatial relationships between rows and columns. We used the same pseudo-HTML notation introduced in Pix2Struct to format the output sequence, helping the model transition from its screenshot parsing objective to this new task. For tables with heterogeneous structures, where cells spanned multiple rows or columns, the expected sequence contained all cells related to the highlighted one (see Figure 4.4).

4.2.5 PixT3 Fine-tuning

After the structural continuous pretraining, PixT3 was then fine-tuned on our image-rendered version of the ToTTo dataset (see Section 4.2.3). Although we chose the ToTTo dataset to perform our experiments, our method is not limited to any specific table format. Since our model processes unimodal input, we include table-related metadata (e.g., titles and section headers) as part of the table image itself, rendering them together as a single image (similar to the approach in Lee et al. (2023)).

4.3 Experiments

4.3.1 Experimental Setup

Model Configuration

All experiments in this work were conducted using the 282M parameter base Pix2Struct model (Lee et al., 2023). PixT3 variants were trained for the three table-to-text generation settings as defined in Section 4.2.1. For fine-tuning, PixT3 models were trained on the ToTTo dataset (Parikh et al., 2020) with tables rendered as images, following the procedure described in Section 4.2.3.

Fine-tuning was performed with a batch size of 8 and a gradient accumulation of 32 steps, using a single NVIDIA A100 80GB GPU. We selected checkpoints based on the best validation set performance. All models used an input sequence length of 2,048 patches and were optimized with the AdamW optimizer (Loshchilov and Hutter, 2017). The learning rate followed a schedule with a linear warmup over 1,000 steps up to 0.0001, after which it was decreased back to 0 following a cosine decay. The decoder's maximum sequence length was set to 50 tokens, covering 97.49% of the target descriptions in the training data. PixT3 was trained for 1,400 steps using the self-supervised objective from Section 4.2.4. Although we initially feared that exposing the model to the random text of our synthetic dataset would deteriorate the language modeling capabilities of Pix2Struct, our experiments showed that fully training the model produced better results than keeping the decoder weights fixed. Thus we performed this intermediate training without freezing the decoder weights. Detailed hyper-parameters are listed in Appendix C.8.

Datasets

We primarily evaluated PixT3 on ToTTo (Parikh et al., 2020), a large dataset with manually curated tables from various domains. To test the model's ability to generalize to out-of-distribution data, we also used Logic2Text (Chen et al., 2020e). This dataset contains 10,161 Wikipedia tables paired with human-written descriptions and logical forms. Unlike most examples in ToTTo, Logic2Text focuses on textual descriptions that require reasoning over tabular data. We used the logical form parsing and execution engine developed in our earlier work to automatically trace the cells involved in the reasoning process, using them as highlighted cells, similar to ToTTo (see Appendix C.5). Results are reported on the official Logic2Text test set, which includes 1,085 examples.

Model Comparison

PixT3 was compared against several text-only models of comparable parameter size. These included CoNT (An et al., 2022), the top-ranked model on the ToTTo leaderboard at the time of writing, which uses contrastive learning techniques and a global decoding strategy. CoNT, built on T5-base (220M parameters), expects input tables to be linearized into text sequences. We also compared against Lattice (Wang et al., 2022b), which encodes tables with layout awareness and position invariance, and vanilla T5-base, which performed competitively on the ToTTo leaderboard without specific modifications (Kale and Rastogi, 2020; An et al., 2022). All comparison models, along with PixT3, were fine-tuned on ToTTo for the three generation settings.

For out-of-domain experiments, we compared PixT3 against LLaVA-1.5 (Liu et al., 2023c), a large multimodal model (13B parameters) that combines the CLIP visual encoder (Radford et al., 2021) with the Vicuna-7B language model (Zheng et al., 2023). Although LLaVA is not specifically fine-tuned for table-to-text generation, its large scale made it an interesting point of comparison. Due to its architectural limitations, LLaVA can only process a single image per forward pass, limiting its ability to perform in-context learning. To simulate this setup, we provided LLaVA with an image, an instruction, and three example table descriptions for each generation setting (see Appendix C.6). A summary of the number of parameters for all models is shown in Table 4.2.

4.3.2 Results

In this section, we present the results of our experiments, evaluating PixT3 across the three established table-to-text generation settings. We detail our findings, comparing its performance against state-of-the-art models on both the ToTTo and Logic2Text datasets.

PixT3 Outperforms in Loosely Controlled and Open-Ended Settings

Table 4.1 presents the results of our experiments on ToTTo across the three generation settings. Model performance was evaluated using the same metrics as those on the ToTTo leaderboard: BLEU (Papineni et al., 2002) as a measure of fluency, PARENT (Dhingra et al., 2019), which takes table content into account to assess faithfulness, and BLEURT (Sellam et al., 2020), a composite metric that measures fluency and reference fidelity. ToTTo's test and development sets contain both overlapping and non-overlapping splits, where table headers may or may not appear in the training data.

In the tightly controlled generation setting (TControl), PixT3, as expected, did not outperform text-only models like CoNT or Lattice, since the highlighted cells provide limited visual information (see Appendix C.2, Figure C.2). PixT3 was outperformed by CoNT by 3.5 BLEU points on the development set and 3.7 points on the test set. However, in the loosely controlled setting (LControl), PixT3 excelled, showing nearly twice the improvement over CoNT and T5 models. In the challenging open-ended setting (OpenE), where models are required to select relevant content in addition to generating text, performance dropped across all models. However, PixT3 still outperformed CoNT, Lattice, and T5 in all metrics in this setting.

PixT3 Generalizes to Unseen Tables Requiring Reasoning

We tested PixT3's generalization capabilities by evaluating its performance on the Logic2Text dataset, which includes more complex reasoning tasks. Table 4.2 presents the results across the three generation settings. In the tightly controlled setting (TControl), PixT3's performance was relatively low, given that this setting only includes the highlighted cells and reasoning over the entire table is essential for producing accurate descriptions. However, in the loosely controlled setting (LControl), PixT3 performed significantly better, showing its ability to generate out-of-domain descriptions over entire tables. In the open-ended setting (OpenE),

		Dev		TestN		TestO		Test	
	Model	BL	PR	BL	PR	BL	PR	BL	PR
TControl	T5-base	47.7	57.1	38.9	51.2	55.4	61.1	47.2	56.2
	T5-3B	48.4	57.8	39.3	51.6	55.1	60.7	47.2	56.2
	Lattice	48.0	58.4	40.0	**53.8**	55.9	62.4	48.0	**58.1**
	CoNT	**49.0**	**58.6**	**40.6**	53.7	**56.7**	**62.5**	**48.7**	**58.1**
	PixT3	45.7	55.7	37.5	50.6	53.2	60.4	45.4	55.5
LControl	T5-base	24.5	27.2	19.4	23.9	29.4	30.3	24.5	27.1
	T5-3B	23.6	26.0	18.0	22.4	28.7	29.2	23.4	25.8
	Lattice	24.9	31.0	20.8	27.7	27.5	33.8	24.4	30.8
	CoNT	23.8	29.3	19.2	26.1	28.7	32.3	23.9	29.2
	PixT3	**46.2**	**55.1**	**38.1**	**50.3**	**52.7**	**59.0**	**45.4**	**54.7**
OpenE	T5-base	21.5	23.5	16.8	21.0	26.5	26.5	21.7	23.8
	T5-3B	20.8	22.9	16.7	20.3	25.5	25.5	21.2	22.9
	Lattice	20.9	26.1	17.6	24.3	23.7	27.6	20.8	25.9
	CoNT	21.7	25.8	16.9	23.2	26.3	28.3	21.6	25.8
	PixT3	**24.8**	**28.3**	**20.5**	**26.3**	**28.9**	**30.3**	**24.7**	**28.3**

4.1 Table – Evaluation results on ToTTo across three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE). We report BLEU (BL) and PARENT (PR) scores on the development (Dev) and test sets, including both overlapping (TestO) and non-overlapping (TestN) test splits. BLEURT scores are provided in Appendix C.5.

where models must autonomously select interesting content, PixT3 outperformed LLaVA and maintained parity with CoNT and Lattice.

Interestingly, a large multimodal model like LLaVA, could not match the performance of PixT3 or the T5-based models, indicating that fine-tuning for table-to-text tasks is more important than sheer parameter size. Output examples for Logic2Text are available in Appendix C.5.

PixT3 Shows Robustness to Large Table Sizes

One important finding of this work is illustrated in Figure 4.5, where we analyze the impact of table size on model performance. As shown, T5, Lattice, and CoNT models struggle with larger tables, exhibiting reduced PARENT scores (BLEU scores also follow the same trend). PixT3, however, demonstrates greater robust-

		TControl		LControl		OpenE	
Model	Size	BLEU	PARENT	BLEU	PARENT	BLEU	PARENT
LLaVA	13B	12.6	34.36	5.9	23.18	6.7	20.14
T5-base	220M	16.8	55.97	11.5	40.02	7.9	30.67
T5-3B	3B	17.7	52.75	10.9	35.45	9.5	29.47
Lattice	220M	19.8	61.05	11.5	40.02	**11.7**	**38.12**
CoNT	220M	18.8	61.73	11.8	43.25	11.0	36.94
PixT3	282M	**20.6**	**61.86**	**21.5**	**56.45**	11.4	35.68

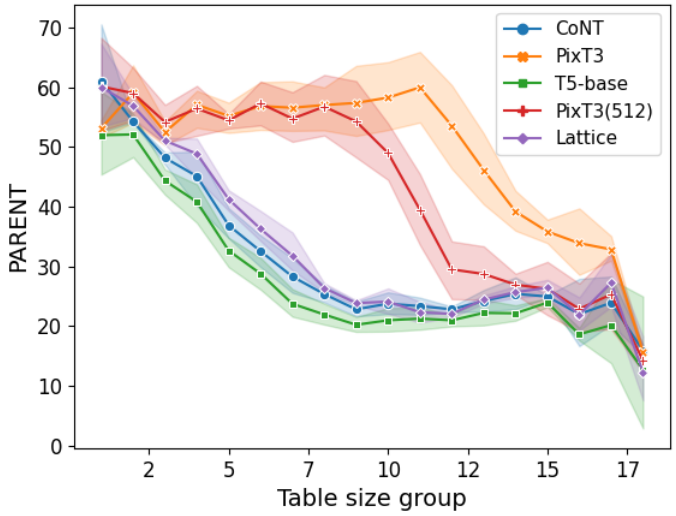
4.2 Table – Evaluation results on Logic2Text across the three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE). All models, except LLaVA, were fine-tuned on ToTTo and tested on Logic2Text. BLEURT scores are provided in Appendix C.5.

ness, with performance only decreasing for very large tables. As previously mentioned, token and patch context lengths are analogous, and while other models use a context length of 512 tokens, PixT3 uses a context length of 2048. To determine if the improved performance of our model is solely due to the increased context length, we also trained and evaluated a version of PixT3 with an input length of 512 patches. We can see how, while the smaller PixT3 model showed slightly lower performance, it still consistently outperformed the other models.

Structure Learning Curriculum Improves Generation

In Table 4.3, we present an ablation study comparing PixT3 with and without the structure learning curriculum and self-supervised objective (Section 4.2.4). Both models underwent the same fine-tuning process with the same hyper-parameters and rendered tables described in Section 4.2.3. The baseline PixT3 model (second row in Table 4.3) shows a significant improvement over the standard Pix2Struct model, which achieved a BLEU score of 0.2 and a PARENT score of 0.6 on the ToTTo development set. Adding the intermediate training curriculum further improves PixT3's performance, though the gains are modest across evaluation metrics.

A manual inspection of the descriptions generated by both PixT3 variants revealed that they are semantically equivalent to the target 43% of the time. However, the intermediate training curriculum greatly reduces structure-based faithfulness errors, especially in the OpenE setting. In a sample of 200 outputs ran-



4.5 Figure – Model performance (CoNT, T5, PixT3, Lattice, and PixT3 with 512-patch input size) in the loosely controlled setting across 18 table size groups (logarithmic scale). Shaded areas represent the upper and lower bounds for overlapping and non-overlapping ToTTo splits, while central points show overall results. Results are measured with PARENT; other metrics show similar trends. For more details, see Appendix C.1.

domly selected from the development set, 23% of the descriptions produced by the baseline PixT3 failed to accurately capture the table's structure or misinterpreted it. When PixT3 was trained with the structure learning curriculum, these errors dropped to just 7%.

PixT3 is Most Faithful in Loosely Controlled and Open-Ended Settings

We also conducted a human evaluation to assess the faithfulness of generated descriptions to the original table data. Participants[1] were asked to judge whether the descriptions were "True" or "False" based on the provided table information, that is, image of a table, and its Wikipedia page and section titles (see the complete instructions in Appendix C.7). PixT3 was compared to the top-performing text-only models, CoNT and Lattice, across 100 randomly selected table-description pairs

[1]Participants were recruited using the online platform Prolific. https://www.prolific.com

	Dev			Test		
Models	BL	PR	BRT	BL	PR	BRT
Pix2Struct	0.2	0.6	−1.433	—	—	—
PixT3 (W/o SLC)	38.7	46.0	−0.003	38.3	45.6	0.001
PixT3 (With SLC)	**39.2**	**46.5**	**0.008**	**38.7**	**46.3**	**0.007**

4.3 Table – Comparison of PixT3 with and without structure learning curriculum (SLC). Results are reported on the ToTTo development (Dev) and test sets, with BLEU (BL), PARENT (PR), and BLEURT (BRT) metrics averaged across the three generation settings.

from ToTTo (development set) and Logic2Text (test set). Overall we elicited 7,200 judgments (100 examples $\times$ 3 generation settings $\times$ 4 model descriptions $\times$ 3 participants $\times$ 2 datasets).

In Table 4.4 we see the results of the human evaluation, showing the proportion of descriptions rated as faithful. As expected, the human-authored reference descriptions were consistently rated as faithful across all generation settings. Following the results of the automatic evaluation, text-based models performed better in the TControl setting but showed a decline in faithfulness in the LControl and OpenE settings. We used paired bootstrap resampling to determine whether the differences between systems were statistically significant. In the TControl setting, PixT3 performed significantly worse than the reference human-made descriptions ($p < 0.05$), but there was no significant difference compared to CoNT or Lattice. In the LControl setting, all differences between systems were statistically significant ($p < 0.05$). In the OpenE setting, PixT3 was significantly different from CoNT and Lattice ($p < 0.05$), but not from the reference. The inter-evaluator agreement was moderate, with a Fleiss' Kappa coefficient of 0.55 (Fleiss, 1971).

4.4 Conclusions

In this work, we leveraged the strengths of Vision Transformers to redefine table-to-text generation as a visual recognition task, eliminating the need to linearize table inputs into a string format. Our proposed model, PixT3, introduces a novel training curriculum and a self-supervised learning objective designed to capture the structure of tables. Through experiments in both controlled and open-ended generation settings, PixT3 demonstrated its robustness across various table sizes,

	Model	TControl	LControl	OpenE
ToTTo	Reference	87	84	89
	Lattice	**79**	16	20
	CoNT	76	16	35
	PixT3	69	**72**	**78**
L2T	Reference	81	87	86
	Lattice	34	3	16
	CoNT	**35**	3	26
	PixT3	32	**40**	**60**

4.4 Table – Human evaluation results on ToTTo and Logic2Text. Proportion of descriptions rated as faithful for PixT3, CoNT, and the human-authored reference descriptions across three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE).

outperforming state-of-the-art models in tasks where the table is part of the input, in both open-ended and controlled scenarios, while remaining competitive when only provided with the context-selected values of the table. Furthermore, PixT3 showcases strong generalization to new domains, as shown by its performance on Logic2Text, a dataset we introduced to assess the ability of models to handle unseen tables.

There are several promising directions for future research. Expanding this pixel-based approach to develop a more general model that can handle a wide range of Table Understanding tasks would be a valuable direction of study. Additionally, exploring new training objectives and inductive biases to better capture table structure and inter-cell relationships in this modality would significantly contribute to advancements in the Table Understanding research area.

5. CHAPTER

Multimodal Table Understanding

5.1 Motivation and Contributions

Following the findings of our previous work, which explored table-to-text generation from a multimodal perspective, in this final contribution of the thesis we wanted to determine whether the benefits of treating tables as visual data could be extended to a broader set of Table Understanding (TU) tasks.

Previous attempts to tackle TU from a multimodal perspective have relied on text-based representations converted into images. This includes our previous work, in which we trained and evaluated our multimodal table-to-text model, PixT3, using image renders of serialized tables from the ToTTo and Logic2Text datasets. This approach stems from the fact that most commonly used tabular datasets serialize and store tables as text, making these textual representations the only available format. Even when other techniques convert these tables into a visual format, much of the original styling, formatting, and communicative design elements may already be lost during serialization, potentially discarding essential contextual information.

Meanwhile, pretraining objectives like next-token prediction and masking have traditionally helped Language Modeling approaches to capture generalistic language patterns and contextual relationships within text, enabling them to better understand and generate coherent and contextually relevant responses across a variety of tasks. However, these objectives are not well-suited to TU tasks because table values are not naturally correlated with their neighboring cells. Prior

work has thus incorporated objectives centered around Semantic Comprehension, Structural Awareness, and Relational Understanding of tables, but no consensus exists on the optimal tasks or combination of tasks for effective TU pretraining (see Appendix D.1 for a detailed list of objectives used in other works).

Therefore, our goal in this work was to create a dataset for TU that includes a diverse set of pretraining objectives and preserves the original visual representations of the tables. Rather than rendering the serialized versions of tables from current datasets, we traced each table back to its original source to extract its original, visually lossless representation. This approach allowed us to apply the multimodal method of PixT3, introduced in our previous work, to directly incorporate visual features, enabling models to leverage format and style cues without compromise while also retaining additional benefits demonstrated by PixT3, such as improved space efficiency.

In this work we introduce the first multimodal Table Understanding dataset containing original table images sourced from Wikipedia with 2.5 million instruction examples and 1.1 million unique table images.

5.2 Methodology

5.2.1 Dataset Overview

Given the advantages of training large language models (LLMs) with instruction-framed examples that frame each task as a question or command (Chung et al., 2022), we chose to frame all examples in our dataset as instructions. Our dataset is composed of instruction examples extracted from three established TU instruction datasets: TableInstruct (Zhang et al., 2024a), Docstruct4M (Hu et al., 2024), and MMTab (Zheng et al., 2024). Notably, none of the examples in these datasets are original; rather, they consist of examples from other datasets reframed as instructions. We refer to these three datasets as seed datasets, and collectively, our dataset includes instruction examples from 11 distinct seed datasets: TURL (Deng et al., 2020), ToTTo (Parikh et al., 2020), TabFact (Chen et al., 2020b), WikiTableQuestions (Pasupat and Liang, 2015), HybridQA (Chen et al., 2020c), NSF (National Science Foundation, National Center for Science and Engineering Statistics, 2019), StatCan (Statistics Canada, 2024), PubTabNet (Zhong et al., 2020), TABMWP (Lu et al., 2023a), TAT-QA (Zhu et al., 2021), and InfoTabs (Gupta et al., 2020).

Additionally, we augmented our dataset with instruction examples that we

generated from seed examples absent in the original instruction dataset instances that were not incorporated by the authors of those datasets. To generate these additional examples, we replicated the templates used within each instruction dataset, reframing the original input/output seed examples as instructions. This expansion not only increased the number of training examples available in our dataset but also enabled us to achieve a 1:1 alignment between our test sets and the seed test sets in terms of examples. This alignment allows our test sets to be directly comparable to those in other studies evaluated on the seed datasets. In total, our dataset includes 2,557,405 instruction examples paired with 1,142,250 tables.

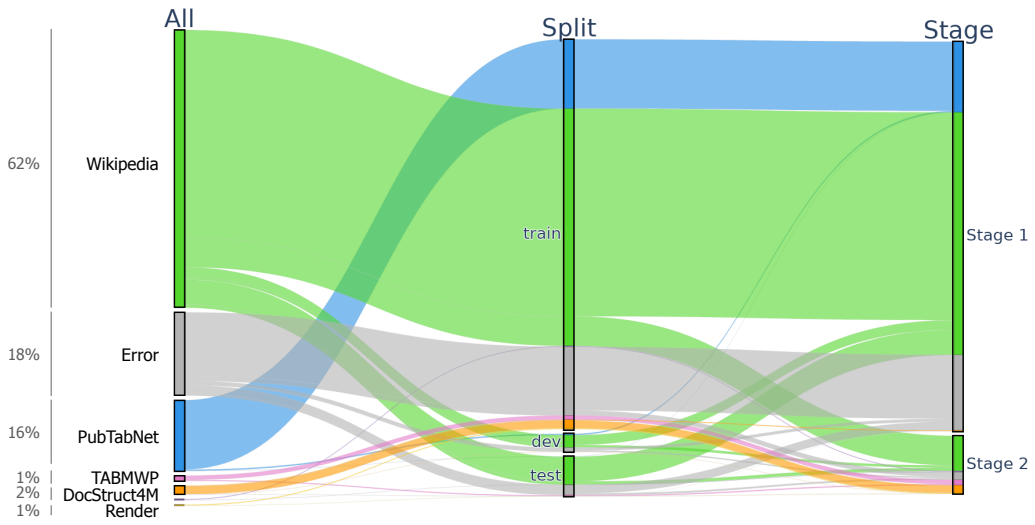
Original Images

When building our dataset, we focused exclusively on instructions whose tables could be traced back to their original visualizations, primarily from Wikipedia tables. Wikipedia tables generally follow an irregular table web format and are often rich in visual information. We traced each table in our seed dataset back to its original version in the corresponding Wikipedia article and stored a screenshot of it as the table's visual representation. Each screenshot serves as a lossless representation of the original table, preserving all information intended to be conveyed to the reader.

During the table retrieval process, we leveraged all available metadata from the seed datasets to locate the original Wikipedia articles as they existed at the time of dataset creation. A significant challenge in obtaining each table's original representation was that these datasets were constructed at different times, and Wikipedia articles are continually updated. To address this, we used Wikipedia's archiving system to retrieve each article as it appeared on the date of crawling. For seed datasets without a publicly available crawling date, we contacted the respective authors to obtain it.

From all tables in the retrieved Wikipedia article, we selected the table with the highest Levenshtein edit distance (Levenshtein, 1966) similarity to the serialized version in the dataset. We set a minimum similarity threshold of 0.70; if no tables met this threshold, the table was not retrieved.

While we place significant importance on Wikipedia tables due to their visual diversity, we also incorporate other table sources to enhance generalization of models trained with our dataset. Additional sources include scientific articles for PubTabNet tables and rendered tables from MMTab, TABMWP, DocStruct4M, as well as our own rendering engine. The distribution of table sources across each split and complexity level (See Section 5.2.1) is presented in Figure 5.1.



5.1 Figure – Dataset table source distribution.

Missing Tables

As mentioned, we were unable to retrieve 12.3% of tables, invalidating about 18% of our dataset's instruction examples (Appendix D.2 contains common error types). These missing tables rendered their associated instructions unusable for training, though we retained all instruction examples in the test set to ensure comparability with other works. Table 5.1 provides a summary of the final dataset composition, showing the number of instruction examples (instruction + table) available for each task.

The only task that originates directly from one of the instruction datasets (DocStruct4M) is Structure Aware Parsing. Specifically, this task requires the model to linearize a table into markdown format. By accessing the full set of original tables and replicating the instruction templates used in the instruction dataset, we were able to generate additional training examples for the training set. In contrast, the limited number of functional examples in the test set is not due to errors obtaining the associated tables but rather to the challenge of tracing back instruction examples based on TURL tables in this instruction dataset.

Task Breakdown

This dataset contains 12 Table Understanding (TU) tasks, which we classify into two complexity levels, referred to as stage 1 and stage 2, (each level is directly re-

Task	Train (%)	Dev (%)	Test (%)
Column Type Annotation	78.4	81.5	81.9
Entity Linking	75.1	76.1	73.7
Relation Extraction	83.8	83.4	84.2
FeTaQA	87.8	-	91.6
HiTab	96.0	-	95.9
TabFact	100.0	47.6	40.5
Structure Aware Parsing	*124.4	100.0	56.3
Table Numerical Reasoning	100.0	-	100.0
Infotabs	30.0	27.5	37.0
ToTTo	88.1	87.2	87.7
HybridQA	90.7	92.5	89.8
WikiTableQuestions	92.6	92.8	93.7

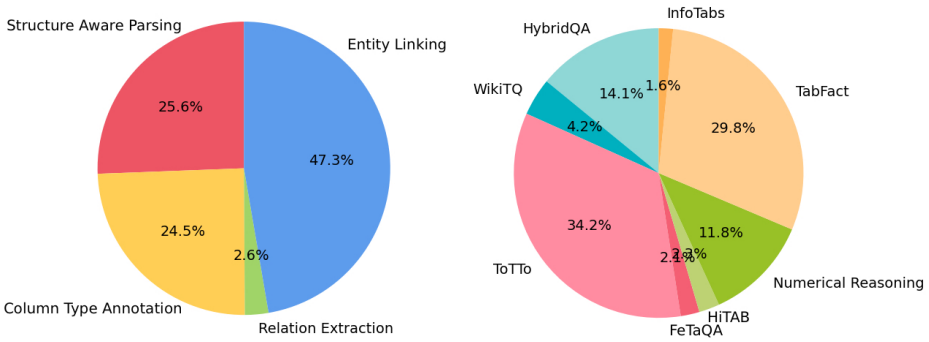
5.1 Table – Percentage of original tables obtained for each task. (*) additional training examples were created by replicating the dataset's template.

lated to training stages in current model training strategies). Stage 1 includes tasks based on fundamental Table Understanding principles, such as Semantic Comprehension, Structure Awareness, and Relational Understanding of tables and their entities. This stage contains the majority of examples (2.2 million). In stage 2, we include tasks that require not only a solid understanding of table mechanics but also additional processing skills, such as Table Question Answering (TableQA), table-to-text generation, Table Numerical Reasoning, and Table Fact Checking. There are 353,000 examples in stage 2.

Figure 5.2 shows the distribution of examples across tasks in the Stage 1 and Stage 2 training sets. Note that the number of examples per task in the test and development sets is less relevant, as each task is evaluated independently.

For a more detailed breakdown, the tasks in this dataset include:

- **Column Type Annotation**: In this task, the model is provided with a table and a set of 255 "data type" candidates, shuffled randomly for each example. Given the column name and cell values to avoid column name ambiguity, the model must select the correct data type for the values in the column. This task instills table Semantic Comprehension into the model. Instruction examples for this task were obtained from TableInstruct being TURL as the seed dataset. Our dataset includes 492,708 / 10,918 / 10,670 (train / dev / test) instructions for this task.



(a) Stage 1 **(b)** Stage 2

5.2 Figure – Task distribution across Stage 1 and Stage 2 complexity levels.

- **Entity Linking**: For this task, the model is given an entity from a table along with a set of candidate entities and descriptions and must identify which entity and description corresponds to the selected entity. This task promotes Semantic Comprehension. Instruction examples come from TableInstruct, with TURL as the seed dataset. Our dataset includes 949,376 / 58,417 / 166,465 (train / dev / test) instructions for this task.
- **Relation Extraction**: This task requires the model to select appropriate relations between two specified columns of a table from a set of candidates. Similar to Column Type Annotation, both column and entities are given to avoid column name ambiguities. Instruction examples are sourced from TableInstruct, with TURL as the seed dataset. Our dataset includes 52,725 / 1,813 / 1,745 (train / dev / test) instructions for this task.
- **Structure Aware Parsing**: Here, the model needs to parse the table into markdown format, fostering Table Structure Awareness. Instruction examples come from Docstruct4M, with PubTabNet as the seed dataset. Our dataset includes 513,412 / 9,115 / 1,102 (train / dev / test) instructions for this task.
- **Free-form Table Question Answering (FeTaQA)**: In this task, the model generates free-form answers to questions about Wikipedia tables, often requiring integration of information from discontinuous sections of the table. Unlike datasets with shorter text spans, FeTaQA emphasizes a higher-level understanding through long-form answers. This Stage 2 task requires proficiency in Semantic Comprehension, Relational Understanding, Structure

Awareness, Natural Language Generation, and Question Answering. Instruction examples come from TableInstruct based on FeTaQA. Our dataset contains 6,430 / 0 / 1,834 (train / dev / test) instructions for this task.

- **Hierarchical Table QA (HiTabQA)**: This question-answering task involves hierarchical tables (with different headers across the table) and sometimes includes numerical reasoning, such as sums, averages, maximum, minimum, and conting among others. It is a Satage 2 task requiring Semantic Comprehension, Relational Understanding, Table Structure Awareness, Numerical Reasoning, and Question Answering. Instructions were obtained from TableInstruct, with tables from seed datasets ToTTo, StatCan, and NSF. Our dataset contains 7,119 / 0 / 1,519 (train / dev / test) examples.

- **Table Fact Verification (TabFact, Infotabs)**: Also known as Table Entailment, this task involves classifying statements as supported or refuted based on table content. Solving this Stage 2 task requires Semantic Comprehension, Relational Understanding, Structure Awareness, Numerical Reasoning, and Symbolic Reasoning. Instruction examples come from TableInstruct and MMTab, based on examples from TabFact and Infotabs seed datasets. Our dataset includes 92,758 / 6,084 / 5,173 (train / dev / test) TabFact examples and 4,966 / 495 / 1,998 (train / dev / test) Infotabs examples.

- **Table Numerical Reasoning**: Given a table and a mathematical question, the model must answer using mathematical reasoning over table values. This Stage 2 task emphasizes Numerical Reasoning alongside Semantic Comprehension, Relational Understanding, and Table Structure Awareness. Instruction examples come from MMTab, with seed datasets TABMWP and TAT-QA. Our dataset includes 36,665 / 0 / 8,458 (train / dev / test) examples.

- **Table-to-Text (ToTTo)**: Known from our previous work, in this task the model needs to generate a description based on a Wikipedia table and a set of highlighted cells. Besides making the model contextualize specific data fields in the table, this Stage 2 task also requires proficiency in Natural Language Generation, Semantic Comprehension, Relational Understanding, and Table Structure Awareness. Instructions were generated by us following the test set template of TableInstruct, with original examples from ToTTo. Our dataset contains 106,414 / 6,715 / 6,750 (train / dev / test) examples.

- **Hybrid QA (HybridQA)**: This multi-hop question-answering task requires integrating structured table data and unstructured hyperlinked passages. That is, given a Wikipedia table and a set of contextual texts linked to the table's entities, the model needs answer a multi-hop question using information from both sources. This Stage 2 task demands Semantic Comprehension, Relational Understanding, Structure Awareness, Numerical Reasoning, Multimodal Reasoning, and Question Answering. Instructions were generated using the TableInstruct template, with original examples from HybridQA. Our dataset contains 43,737 / 2,485 / 3,111 (train / dev / test) examples.
- **Table QA (WikiTableQuestions)**: Given a Wikipedia table and a question, the model must answer based on table content. This Stage 2 task mainly focuses on Question Answering but also requires Semantic Comprehension, Relational Understanding, and Structure Awareness. Instruction examples are sourced from TableInstruct, with WikiTableQA as the seed dataset. Our dataset includes 13,098 / 3,284 / 4,070 (train / dev / test) instructions.

5.3 Experiments

5.3.1 Experimental Setup

Our dataset emphasizes the preservation of original visualizations of tables, differentiating it from other image-based Table Understanding approaches such as MMTab (Zheng et al., 2024). To demonstrate that the dataset is well-structured and effective for TU, it would ideally be used for both continuous pretraining stages. However, due to resource constraints, the Stage 1 pretraining was left for future work. This experiment focuses on Stage 2 fine-tuning of a backbone model to check two key aspects: 1) that our dataset enables the model to achieve better results compared to the backbone alone, and 2) that it compares favorably to state-of-the-art systems.

Model Configuration

For this experiment, we selected mPLUG-DocOwl 1.5 (Hu et al., 2024) as the backbone model due to its specialized architecture for encoding visually represented text, which is a fundamental characteristic when working with tabular data. Built on the strong foundation of mPLUG-Owl2 (Ye et al., 2023), which has

demonstrated significant performance in other Vision-Language Modeling tasks, mPLUG-DocOwl 1.5 incorporates a H-reducer module between its vision and text encoders. This module, which combines horizontally related image patches, allows for an efficient representation of images with written text in horizontally scripted languages. Furthermore, Hu et al. (2024) conducted a two-stage continuous pretraining of mPLUG-DocOwl 1.5 on their DocStruct4M dataset. This dataset includes tables, some of which are present in our dataset under the Structure Aware Parsing task, alongside additional training samples generated by replicating DocStruct4M's templates.

To test the efficacy of our dataset, we replaced mPLUG-DocOwl 1.5's Stage 2 fine-tuning with our own Stage 2 subset. We followed the training procedure outlined by Hu et al. (2024), replacing DocStruct4M with our dataset during Stage 2. That is, we initialized the model with the model parameters from the first stage of continuous pretraining and contunied the Stage 2 training with the examples from our dataset. Training was conducted across 8 nodes, each equipped with 4 NVIDIA Hopper H100 64GB GPUs, for 6,500 steps. A detailed list of hyperparameters is available in Appendix D.3.

Model Comparison

We compare our model with two state-of-the-art TU models: the multimodal image-based TableLLaVA (Zheng et al., 2024) and the unimodal text-based TableLlama (Zhang et al., 2024a). TableLLaVA is built on the LLaVA (Liu et al., 2023b) architecture, continuously pretrained on the MMTab dataset, while TableLlama is based on the LLaMA (Touvron et al., 2023) architecture, continuously pretrained on the TableInstruct dataset. Despite their differences in modality, both models are instruction-based TU models.

We evaluated all models using the test sets from the Stage 2 tasks in our dataset: FeTaQA, HybridQA, InfoTabs, TabFact, TABMWP, HiTabQA, and WikiTableQuestions (WikiTQ). For TAT-QA and ToTTo, we used their development sets instead of the test sets, as gold references were not available for the latter. We selected the most commonly used metrics to report performance in each dataset, namely BLEU4 (Papineni et al., 2002) for FeTaQA and ToTTo, and strict match accuracy for the others. For HybridQA, however, accuracy is calculated based on whether the reference text appears in the generated sequence rather than relying on exact match. We follow this approach to ensure a fair evaluation of other models' responses, as many were correct but did not adhere to the reference format, often being more verbose than the reference text.

Model	FeTaQA	HiTab	HybridQA	InfoTabs	TabFact	TaBMWP	TAT-QA	ToTTo	WikiTQ
Baseline	2.5*	17.6*	35.5*	29.9*	68.3	10.9*	12.7*	10.1*	**33.7**
Ours	**66.0**	**41.9**	**50.7**	**60.2**	**72.9**	**86.2**	**43.7**	**41.6**	32.2

5.2 Table – Evaluation results for mPLUG-DocOwl 1.5 (Baseline) and the same model architecture but replacing its Stage 2 training examples with the examples in our dataset (Ours). Metrics include BLEU4 for FeTaQA and ToTTo, and exact match accuracy for other tasks. (*) Indicates a dataset whose train set was not present in that model's training.

5.3.2 Results

Table 5.2 compares the original mPLUG-DocOwl 1.5, pretrained on both Stage 1 and Stage 2 of DocStruct4M, against a version where Stage 1 pretraining remains unchanged, but Stage 2 fine-tuning is performed using our dataset's Stage 2 examples. The results highlight that our model outperforms the baseline in 8 out of 9 evaluation tasks. Interestingly, the high performance of the baseline model on TabFact and WikiTQ can be attributed to the inclusion of these datasets during DocReason25K training, alongside DocStruct4M. Other datasets, marked with (*) were not present during the baseline's original training. While our model significantly outperformed MMTab in these tasks (See Table 5.3), the difference from the baseline was less pronounced, with a 4.6% accuracy improvement on TabFact and a 1.5% decline on WikiTQ.

To compare our model with other state-of-the-art approaches, we evaluated all models using the same examples as those in our test sets. As discussed in Section 5.2.1, our inability to retrieve all original table visualizations resulted in a reduced subset of effective examples (those containing both the table and instruction) in our test sets compared to the original datasets. Table 5.1 shows the proportion of effective examples from the original datasets included in our test sets. To ensure a fair comparison and preserve the original instruction format and table representation of each dataset, we evaluated all models on the equivalent examples from their respective datasets. Specifically, TableLLaVA was evaluated on MMTab's test sets, and TableLlama on TableInstruct's test sets, with any examples not present in our dataset's test sets filtered out.

The results in Table 5.3 show that our model outperforms Table-LLaVA and TableLlama in 8 and 7 out of 9 evaluation tasks, respectively. However, it still falls behind TableLlama in 2 of the three datasets included in TableLlama's training. While the quantity and diversity of examples in our dataset likely contributed to our model's superior performance over Table-LLaVA, we hypothesize that the pri-

Model	FeTaQA	HiTab	HyQA	InfoTabs	TabFact	TaBMWP	TATQA	ToTTo	WikiTQ
DocOwl1.5 (Ours)	**66.0**	41.9	**50.7**	60.2	72.9	**86.2**	**43.7**	**41.6**	**32.2**
Table-LLaVA (7B)	25.8	10.4	35.6*	**63.0**	53.7	57.9	16.7	26.1	11.1
TableLlama	39.1	**59.8**	36.5*	10.2*	**82.9**	11.2*	6.3*	21.5*	17.1*
Reported									
Table-LLaVA (7B)	25.6	10.9	-	65.3	59.8	57.8	12.8	23.0	18.4
Table-LLaVA (13B)	28.0	10.8	-	66.9	65.0	59.8	15.6	24.1	20.4
TableLlama	39.1	64.7	39.38*	-	82.6	-	-	20.8*	35.0*

5.3 Table – Evaluation results for mPLUG-DocOwl 1.5 fine-tuned on our Stage 2 dataset, compared with the state-of-the-art multimodal Table Understanding model TableLLaVA and the unimodal text-based model TableLlama. Results reported for these models in their original papers, evaluated over the full test set, are also included for reference. Metrics include BLEU4 for FeTaQA and ToTTo, and accuracy for other tasks. HybridQA (HyQA) accuracy is calculated based on whether the reference text is present in the generated sequence, rather than exact match. See Appendix D.4 for detailed results on exact match accuracy. Notably, exact match accuracy follows a similar trend, further highlighting the advantage of our model. (*) Indicates a dataset whose training set was not included in the model's training data.

mary factor is the proficiency of mPLUG-DocOwl 1.5 in handling visually represented text. Unlike Table-LLaVA's backbone model, LLaVA, which was primarily pretrained on natural images, mPLUG-DocOwl 1.5 was specifically adapted for Document Understanding tasks. Replacing Table-LLaVA's backbone with a more text-oriented multimodal model, such as LLaVAR (Zhang et al., 2023), could help determine whether the differences in performance are due to LLaVA's limited capabilities in visually situated text tasks.

Overall, our results demonstrate the high quality of our Stage 2 dataset, as training a baseline model on this dataset enables it to outperform current state-of-the-art VLMs across a wide range of tasks. Furthermore, it makes these models competitive with, and in some cases superior to, widely used text-based unimodal approaches.

5.4 Conclusions

Our dataset presents a unified, multimodal perspective on 11 widely-used table datasets and, at the time of writing, is the largest multimodal Table Understanding dataset available. It offers a broad range of examples and surpasses contemporary datasets, like MMTab, in scale. Additionally, our dataset includes original

images for Wikipedia-based tables and traces each example through the instruction dataset, back to the seed dataset, and ultimately to the original source. This traceability is a significant advantage, as most instruction datasets lack such detailed connections. Finally, we introduce a dedicated development set, a valuable contribution since most instruction datasets do not include a development set.

Empirical results demonstrate that our Stage 2 dataset is of high quality, as training a baseline model on it allows the model to outperform current state-of-the-art VLMs across a diverse set of tasks. Additionally, it enables these models to be competitive with, and sometimes exceed, the performance of widely used text-based unimodal approaches.

We leave the evaluation of the contributions of our Stage 1 subset for future work, which has the potential to push performance even further, as it contains more than six times the number of instructions in Stage 2. The publication of the dataset and related experiments is currently under preparation.

6. CHAPTER

Conclusions and Future Research

6.1 Conclusions

In this thesis, we introduced a comprehensive approach to address challenges in table-to-text generation, beginning with enhancing faithfulness through logical form generation for regularly structured tables. We then moved to irregularly structured tables by applying table image representations using Vision Transformers, and finally extended our findings in table representation to the broader field of Table Understanding by introducing a unified multimodal tabular dataset. These contributions represent significant advancements in improving factual accuracy, structural understanding, and dataset diversity for Table Understanding applications, with a particular focus on table-to-text. Below, we summarize the primary contributions of this research.

Automatic Logical Forms Improve Fidelity in Table-to-Text Generation The first contribution of this thesis is the development of the $T|T$ system, which enhances table-to-text generation by introducing an intermediate logical form generation stage. Our approach demonstrates that automatically generating logical forms substantially improves faithfulness and factual accuracy, showing a 67% increase in fidelity over baseline models. By separating content selection from LF generation, we further confirmed the advantages of incorporating logical forms in the generation pipeline, with LF-conditioned generation having the greatest impact on factual accuracy. This intermediate logical form representation enables

automated validation before natural language generation, adding a layer of robustness and reliability to the final output. This contribution marks a meaningful step toward more accurate and reliable table-to-text applications, with logical forms serving as a foundation for verifiable, factually consistent output.

Pixel-based Table-To-Text Generation Our second contribution explores the use of Vision Transformers to redefine table-to-text generation as a visual recognition task. Our model, PixT3, introduced a pixel-based approach that removes the need to linearize table inputs, a common limitation in previous models. By treating tables as visual entities and training PixT3 on a new image-based structure learning curriculum, we achieved robust structural understanding across a range of table sizes and formats. In fact, PixT3 outperforms other baselines in automatic metrics and human faithfulness evaluation. PixT3's strong performance on the Logic2Text dataset also demonstrates its adaptability to previously unseen tables. Overall, this contribution provides a foundation for developing more versatile and visually aware table-to-text models, highlighting its potential to support various multimodal applications that require an in-depth structural comprehension of table data.

A Multimodal Dataset for Table Understanding The third major contribution of this thesis is the creation of a multimodal, instruction-based Table Understanding dataset that includes 2.5 million examples and 1.1 million original table visualizations. This dataset, the largest of its kind, spans 11 widely-used Table Understanding tasks, offering diverse objectives of different complexities appropriate for the two typical model pre-training stages. It also offers traceability between examples, a property lacking in most derived datasets, linking each example back to its source to add transparency and reproducibility. Our dataset further includes a dedicated development set, another absent resource in most instruction-based TU datasets, thereby filling a gap in the current data landscape for multimodal Table Understanding tasks.

Publications Part of the research in this dissertation has been published in peer-reviewed journals and conferences. Notably, one paper was published in a *Journal Citation Report (JCR) Q1*-ranked journal, another in the 2024 *ACL main conference*, and a third one is currently under preparation. In addition to these publications, during the execution of this thesis, I also collaborated on two NLP-related research projects and was the first author of a peer-reviewed paper published in

another *JCR Q1*-ranked journal.

6.2 Future Work

This thesis has made strides in enhancing faithfulness, table representation, and dataset diversity in table-to-text generation and Table Understanding. However, several open questions and promising directions remain. This section outlines key areas for future research, aimed at addressing the current limitations and expanding the capabilities of the methods developed throughout the thesis.

Improving Logical Form-based Systems While logical forms have proven effective in reducing faithfulness errors in table-to-text generation, our implementation of LFs remains based on model conditioning, meaning that faithfulness errors may still occur during the interpretation or verbalization of the semantics represented by the LF. Future work could focus on enforcing LF semantics more directly within the generation model, potentially through stricter alignment techniques that ensure final outputs adhere to LF meaning. Additionally, the current LF grammar is tailored to the Logic2Text dataset, yet natural language encompasses a wider range of meanings. Expanding the LFs structure or adopting other standardized semantic grammars could allow models to handle more diverse reasoning and linguistic contexts.

Currently, LFs are designed for regular tables, however, tables in practice often present a wider spectrum of formats. Adapting LFs to work with irregularly structured tables would allow these benefits to extend across a broader array of tables and domains. Finally, considering the crucial role that content selection plays in table-to-text generation, future research could focus on leveraging user preferences within training data to allow automated content selection in future models.

Extending Vision Transformers to Table Understanding The results achieved with PixT3 suggest several promising directions for future work as well. Extending the model to support a wider array of Table Understanding tasks could transfer its benefits to other areas of Table Understanding. Additionally, refining training objectives to capture cell-to-cell relationships more effectively could further improve PixT3's understanding of table structures.

In our research, we have also demonstrated that current Vision-Language Models are capable of addressing complex tasks involving visually represented text.

Future studies could extend this work, moving beyond Table Understanding to explore the application of our approach in other areas, such as Chart Understanding.

Leveraging our Multimodal Table Understanding Dataset Finally, the dataset introduced in the last chapter also opens the door to exploring a wide range of research questions. Including, evaluating the impact of fully pretraining TU models on both Stage 1 and Stage 2 subsets of our dataset, identifying which Table Understanding tasks benefit most from representing tables as images, and assessing the extent to which preserving tables in their original visual form enhances performance on tasks initially designed for serialized data. Additionally, future research could also explore how multimodal models process table data, examine the impact of dataset contamination, and research the role of parametric knowledge in improving task performance on datasets based in real-world data.

Bibliography

Iñigo Alonso and Eneko Agirre. 2024. Automatic logical forms improve fidelity in table-to-text generation. *Expert Systems with Applications*, 238:121869.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 2197–2210. Curran Associates, Inc.

Anthropic. 2023. Claude: An ai assistant. Accessed: 2024-11-08.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR)*. ArXiv preprint arXiv:1409.0473.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2012. Abstract meaning representation (amr) 1.0 specification. In *Abstract meaning representation (amr) 1.0 specification*, volume Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL, pages 1533–1544.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings*

of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.

BigScience Workshop. 2022. Bloom (revision 4ab0472).

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Hong Kong. Association for Computational Linguistics.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Ursin Brunner and Kurt Stockinger. 2021. Valuenet: A natural language-to-sql system that learns from database information. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2177–2182.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2018. Language gans falling short. *arXiv preprint arXiv:1811.02549*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 128–135, New York, NY, USA. Association for Computing Machinery.

Leiyuan Chen, Chengsong Huang, Xiaoqing Zheng, Jinshu Lin, and Xuanjing Huang. 2023a. TableVLM: Multi-modal pre-training for table structure recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2437–2449, Toronto, Canada. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020c. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Xi Chen, Xinjiang Lu, Haoran Xin, Wenjun Peng, Haoyang Duan, Feihu Jiang, Jingbo Zhou, and Hui Xiong. 2023b. A table-to-text framework with heterogeneous multidominance attention and self-evaluated multi-pass deliberation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 607–620, Singapore. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020d. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111, Online. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020e. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111, Online. Association for Computational Linguistics.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Al-

bert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Eric Crestan and Patrick Pantel. 2011. Web-scale table census and classification. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, page 545–554, New York, NY, USA. Association for Computing Machinery.

Amanda Dash, Melissa Cote, and Alexandra Branzan Albu. 2023. Weathergov+: A table recognition and summarization dataset to bridge the gap between document image analysis and natural language generation. In *Proceedings of the ACM Symposium on Document Engineering 2023*, DocEng '23, New York, NY, USA. Association for Computing Machinery.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. 2022. End-to-end document recognition and understanding with dessurt. In *European Conference on Computer Vision*, pages 280–296. Springer.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 407–426, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: table understanding through representation learning. *Proceedings of the VLDB Endowment*, 14(3):307–319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. 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, Florence, Italy. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Julian Eisenschlos, Maharshi Gor, Thomas Müller, and William Cohen. 2021. MATE: Multi-view attention for table transformer efficiency. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7606–7619, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Lei Gao, Alan Zela, Shoaib Ahmed Siddiqui, Sumit Hasan, Yue Zhang, Venu Govindaraju, and Abdel Belaïd. 2019. Icdar 2019 competition on table detection and recognition (ctdar). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1510–1515. IEEE.

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.

Deepanway Ghosal, Preksha Nema, and Aravindan Raghuveer. 2023. ReTAG: Reasoning aware table to analytic text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6310–6324, Singapore. Association for Computational Linguistics.

Stefan Gobel, Tarek Hassan, Eduardo Oro, Giovanni Orsi, Jean-Yves Ramel, Emilie Ricci, Richard Zanibbi, Dimosthenis Karatzas, Harold Mouchere, Jean-Christophe Burie, et al. 2013. Icdar 2013 competition on table detection and structure recognition. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1449–1453. IEEE.

E. Goldberg, N. Driedger, and R.I. Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535, Florence, Italy. Association for Computational Linguistics.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

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

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mPLUG-DocOwl 1.5: Unified structure learning for OCR-free document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3096–3120, Miami, Florida, USA. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2020. Learning to select, track, and generate for data-to-text. *Journal of Natural Language Processing*, 27(3):599–626.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, Vancouver, Canada. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. 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. Association for Computational Linguistics.

Rihui Jin, Yu Li, Guilin Qi, Nan Hu, Yuan-Fang Li, Jiaoyan Chen, Jianan Wang, Yongrui Chen, and Dehai Min. 2024. Hgt: Leveraging heterogeneous graph-enhanced large language models for few-shot complex table understanding.

Rihui Jin, Jianan Wang, Wei Tan, Yongrui Chen, Guilin Qi, and Wang Hao. 2023. TabPrompt: Graph-based pre-training and prompting for few-shot table understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7373–7383, Singapore. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Pratik Kayal, Mrinal Anand, Harsh Desai, and Mayank Singh. 2021. Icdar 2021 competition on scientific table image recognition to latex. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*, pages 754–766. Springer.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Emiel Krahmer and Mariët Theune. 2010. *Empirical methods in natural language generation: Data-oriented methods and empirical evaluation*, volume 5790. Springer.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012a. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012b. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.

R. Lebret, D. Grangier, and M. Auli. 2016a. Neural Text Generation from Structured Data with Application to the Biography Domain . In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rémi Lebret, David Grangier, and Michael Auli. 2016b. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2019. Tablebank: A benchmark dataset for table detection and recognition.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023b. Table-gpt: Table-tuned gpt for diverse table tasks.

Wenwen Li, Xu Zhong, Zhixing Tang, Zhou Yu, Lei Cui, Furu Huang, and Ming Liu. 2020. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, pages 1918–1925.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore. Association for Computational Linguistics.

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

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023a. MatCha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770, Toronto, Canada. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *NeurIPS*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023c. Visual instruction tuning.

Jianfeng Liu, Haoran Wang, Xin Li, and Yu Zhang. 2021. Tabgnn: Table structure recognition with graph neural networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3756–3766. Association for Computational Linguistics.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations (ICLR)*.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023a. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*.

Qingyu Lu, Liang Ding, Liping Xie, Kanjian Zhang, Derek F. Wong, and Dacheng Tao. 2023b. Toward human-like evaluation for natural language generation with error analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5892–5907, Toronto, Canada. Association for Computational Linguistics.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14662–14684, Singapore. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. *Interspeech 2010*.

Tanushree Mitra, Clayton J Hutto, and Eric Gilbert. 2015. Comparing person- and process-centric strategies for obtaining quality data on amazon mechanical

turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1345–1354.

Francesco Moramarco, Alex Papadopoulos Korfiatis, Mark Perera, Damir Juric, Jack Flann, Ehud Reiter, Anya Belz, and Aleksandar Savkov. 2022. Human evaluation and correlation with automatic metrics in consultation note generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5739–5754, Dublin, Ireland. Association for Computational Linguistics.

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. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

National Science Foundation, National Center for Science and Engineering Statistics. 2019. Science and engineering indicators 2019. Accessed: 2024-10-28.

OpenAI. 2023. Gpt-4: Technical report. Accessed: 2024-11-08.

Vaishali Pal, Andrew Yates, Evangelos Kanoulas, and Maarten de Rijke. 2023. MultiTabQA: Generating tabular answers for multi-table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6322–6334, Toronto, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. 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. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6908–6915.

A Radford, J Wu, R Child, D Luan, and D Amodei.... 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A hierarchical model for data-to-text generation. In *European Conference on Information Retrieval*, pages 65–80. Springer.

Ehud Reiter. 2018a. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter. 2018b. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Claude E Shannon. 1951. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

K Simonyan and A Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society.

Statistics Canada. 2024. Statistics canada - the national statistical office of canada. Accessed: 2024-10-28.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chris Van Der Lee, Albert Gatt, Emiel Van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically

generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Doan Nam Long Vu, Nafise Sadat Moosavi, and Steffen Eger. 2022. Layer or representation space: What makes BERT-based evaluation metrics robust? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3401–3411, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022a. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022b. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048, Seattle, United States. Association for Computational Linguistics.

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

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. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Terry Winograd. 1972. *Understanding natural language*. Elsevier.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017a. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017b. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium. Association for Computational Linguistics.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

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

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Jing Zhang, Zhi Chen, Zhiyuan Liu, and Maosong Sun. 2020. Graph neural networks for table structure recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3610–3620. Association for Computational Linguistics.

Shuo Zhang and Krisztian Balog. 2017. Entitables: Smart assistance for entity-focused tables. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17. ACM.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024a. TableLlama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6024–6044, Mexico City, Mexico. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Weijia Zhang, Vaishali Pal, Jia-Hong Huang, Evangelos Kanoulas, and Maarten de Rijke. 2024b. Qfmts: Generating query-focused summaries over multi-table inputs.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023. Llavar: Enhanced visual instruction tuning for text-rich image understanding.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *LREC*.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022. ReasTAP: Injecting table reasoning skills during pre-training via synthetic reasoning examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9006–9018, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Ruizhe Chen, Xiangru Tang, Yumo Xu, Dragomir Radev, and Arman Cohan. 2023. QTSumm: Query-focused summarization over tabular data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1157–1172, Singapore. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9102–9124, Bangkok, Thailand. Association for Computational Linguistics.

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

Xu Zhong, Elaheh ShafieiBavani, and Richard Zanibbi. 2020. Image-based table recognition: data, model, and evaluation. *Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR)*, pages 847–854.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. 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 3277–3287, Online. Association for Computational Linguistics.

A. APPENDIX

Original papers

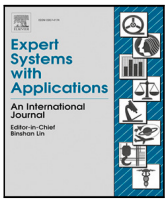
In this appendix, we provide the original papers included in this thesis, arranged in the recommended order for reading.

Expert Systems With Applications 238 (2024) 121869

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Automatic Logical Forms improve fidelity in Table-to-Text generation

Iñigo Alonso *, Eneko Agirre

HiTZ Basque Center for Language Technologies - Ixa NLP Group, University of the Basque Country (UPV/EHU), M. Lardizabal 1, Donostia, 20018, Basque Country, Spain

ARTICLE INFO

Keywords:
Natural Language Generation
Table-to-Text
Deep learning
Logical forms
Faithfulness
Hallucinations

ABSTRACT

Table-to-text systems generate natural language statements from structured data like tables. While end-to-end techniques suffer from low factual correctness (fidelity), a previous study reported fidelity gains when using manually produced graphs that represent the content and semantics of the target text called Logical Forms (LF). Given the use of manual LFs, it was not clear whether automatic LFs would be as effective, and whether the improvement came from the implicit content selection in the LFs. We present **TℓT**, a system which, given a table and a set of pre-selected table values, first produces LFs and then the textual statement. We show for the first time that automatic LFs improve the quality of generated texts, with a 67% relative increase in fidelity over a comparable system not using LFs. Our experiments allow to quantify the remaining challenges for high factual correctness, with automatic selection of content coming first, followed by better Logic-to-Text generation and, to a lesser extent, improved Table-to-Logic parsing.

1. Introduction

Data-to-text generation is the task of taking non-linguistic structured input such as tables, knowledge bases, tuples, or graphs, and automatically producing factually correct[1] textual descriptions of the contents of the input (Covington, 2001; Gatt & Krahmer, 2018; Reiter & Dale, 1997). Real-world applications include, among others, generating weather forecasts from meteorological data (Goldberg, Driedger, & Kittredge, 1994), producing descriptions from biographical information (Lebret, Grangier, & Auli, 2016), or generating sport summaries using game statistics (Wiseman, Shieber, & Rush, 2017). In these applications, the goal is to represent relevant information in the input data using natural language descriptions. Therefore, generating text that faithfully and accurately represents the underlying information in the source becomes critical. It should be noted that the task is underspecified, in the sense that the same table may be described by multiple textual descriptions, all of them correct, as each one can focus on different, relevant subsets of the input data. This makes the use of manual evaluation of fidelity key to measure the quality of the generated text. Our work focuses on how to improve faithfulness automatically.

Various Data-to-Text approaches have emerged to address this challenge. Methods include leveraging the structural information of the input data (Chen, Su, Yan, & Wang, 2020; Puduppully, Dong, & Lapata, 2019b; Wiseman et al., 2017), using neural templates (Wiseman, Shieber, & Rush, 2018), or focusing on content ordering (Puduppully, Dong, & Lapata, 2019a). Recent techniques (Aghajanyan et al., 2022; Chen, Chen, Su, Chen, & Wang, 2020; Chen, Chen, Zha et al., 2020; Kasner & Dusek, 2022) leverage large-scale pre-trained models (Devlin, Chang, Lee, & Toutanova, 2019), and report significant performance gains in terms of fluency and generalization with respect to previous work that did not use such models.

However, these end-to-end systems struggle with fidelity as they are still susceptible to produce hallucinations, i.e. they generate text that, despite its fluency, does not describe in a faithful way the input data (Koehn & Knowles, 2017; Maynez, Narayan, Bohnet, & McDonald, 2020).

In this context Chen, Chen, Zha et al. (2020) propose to reformulate Data-to-Text as a Logic-to-Text problem. Alongside the usual table information, the input to the language realization module in this approach also includes a tree-structured graph representation of the semantics of the target text called logical form (LF). Logical forms follow compositional semantics (Carnap, 1947) to formalize the underlying meanings represented in the target text. When provided alongside tables in this case, the meaning conveyed by LFs is related to a semantic context as defined in Wang, Liu, Ip, Zhang, and Deters (2014), Zhang (1994). In this case, the semantic context is given by the table. An example of how LFs represent this meaning can be seen in Fig. 2. Although the LFs were applied to tables in this paper, the proposal could be easily extended to other Data-to-Text problems.

* Corresponding author.
E-mail addresses: inigoborja.alonso@ehu.eus (I. Alonso), e.agirre@ehu.eus (E. Agirre).

[1] We use the terms factual correctness, faithfulness, and fidelity indistinctly.

https://doi.org/10.1016/j.eswa.2023.121869
Received 24 April 2023; Received in revised form 18 September 2023; Accepted 26 September 2023
Available online 12 October 2023

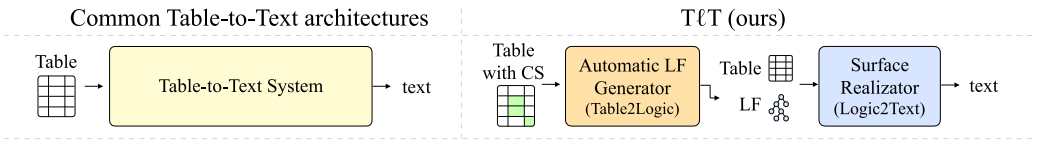


Fig. 1. Our proposed system to improve fidelity, **TℓT**, (right) alongside a typical Table-to-Text architecture (left).

With the use of manual LFs, Chen, Chen, Zha et al. (2020) report an increase in factual correctness from 20% to 82% compared to a system not using LFs. Manually produced LFs include, implicitly, a selection of the contents to be used in the description also referred as Content Selection (CS). Content Selection is the task of choosing the subset of the table that is to be communicated in the output (Duboue & McKeown, 2003). LFs inherently provide the content selection within themselves, and thus models based on manual LFs have an easier task and a lower probability of producing an unfaithful statement. The main shortcoming of this approach is that the manual production of LFs is very costly and it is not realistic to expect table producers to add formal semantic representations such us LFs for each table that they produce. Chen, Chen, Zha et al. (2020) left two open research questions: Firstly, the improvement in faithfulness could come from the implicit content selection alone, casting doubts about the actual contribution of LFs. Secondly, it is not clear whether a system using automatic LFs would be as effective as a system based on manual LFs. Our goal is to answer these two questions.

In this work we present **TℓT** (short from Table-to-Logic-to-Text), a two-step model that produces descriptions by, first, automatically generating LFs (Table-to-Logic parsing), and then producing the text from those LFs (Logic-to-Text generation). Our model (see Fig. 1) allows Table-to-Text generation systems to leverage the advantages of using LFs without requiring manually written LFs. We separate the content selection process from the logical form generation step, allowing to answer positively to the open questions mentioned above with experiments on the Logic2Text dataset (Chen, Chen, Zha et al., 2020). Although content selection alone improves results, the best results are obtained using automatic LFs, with noteworthy gains in fidelity compared to a system not using LFs. Our results and analysis allow to estimate the impact in fidelity of the remaining challenges, with automatic content selection coming first, followed by better Logic-to-Text generation and to a lesser extent Table-to-Logic parsing. We also provide qualitative analysis of each step.

All code, models and derived data are publicly available.[2]

2. Related work

Natural Language Generation from structured data is a long-established research line. Over time, multiple techniques have been developed to solve this task in different ways, such as leveraging the structural information of the input data (Chen, Su et al., 2020; Liu, Wang, Sha, Chang, & Sui, 2018; Puduppully et al., 2019b; Rebuffel, Soulier, Scoutheeten, & Gallinari, 2020; Wiseman et al., 2017), using neural templates (Li & Wan, 2018; Wiseman et al., 2018) or focusing on content ordering (Puduppully et al., 2019a; Sha et al., 2018; Su, Vandyke, Wang, Fang, & Collier, 2021). The use of pre-trained language models (Devlin et al., 2019; Radford, Wu, Child, Luan, & Amodei..., 2019) has allowed to improve text fluency compared to those early systems (Aghajanyan et al., 2022; Chen, Chen, Su et al., 2020; Kasner & Dusek, 2022); however, fidelity remains the main unsolved issue in all of the aforementioned systems.

A body of research has thus focused on improving factuality. Matsumaru, Takase, and Okazaki (2020) remove factually incorrect instances from the training data. Other proposals take control of the decoder by making it attend to the source (Tian, Narayan, Sellam, & Parikh, 2019), using re-ranking techniques (Harkous, Groves, & Saffari, 2020), or applying constrains that incorporates heuristic estimates of future cost (Lu et al., 2021). Alternatively, (Li & Rush, 2020; Shen, Chang, Su, Niu, & Klakow, 2020; Wang, Wang, An, Yu, & Chen, 2020) rely on heuristics, such as surface matching of source and target, to control generation.

In a complementary approach to improve factuality, Chen, Chen, Zha et al. (2020) propose reformulating Table-to-Text as a Logic-to-Text problem. They incorporate a tree-structured representation of the semantics of the target text, logical forms (LF), along with the standard table information. The logical form highly conditions the language realization module to produce the statement it represents, significantly improving fidelity results. However, the logical forms in this work are manually produced by humans, which is unrealistic and greatly reduces the applicability of this solution in a real-world scenario. Our work builds on top of this approach, adopting LFs and proposing to generate them automatically based on table data alone, with the goal of enabling practical use without sacrificing fidelity.

Automatically generating LFs requires of techniques capable of producing a formal representation from text, following a set of pre-defined grammar rules. This challenge is commonly addressed in so-called semantic parsing tasks (Radhakrishnan, Srikantan, & Lin, 2020; Yin & Neubig, 2017), but they have not been applied to table-to-text before. For instance, Guo et al. (2019) present IRNet, a NL-to-SQL semantic parser that generates grammatically correct SQL sentences based on their natural language descriptions. Valuenet, introduced by Brunner and Stockinger (2021), presents a BERT-based encoder (Devlin et al., 2019) in IRNet. In this work, we adapted the grammar-based decoder of Valuenet to produce LFs, which allowed us to show that we can produce high quality LFs.

3. Model

In this section we first introduce Logical Forms, and then the model that produces descriptions for tables via automatically produced Logical Forms.

3.1. Logical forms

The LFs used in this work are tree-structured logical representations of the semantics of a table-related statement, similar to AMR graphs (Banarescu et al., 2012), and follow the grammar rules defined by Chen, Chen, Zha et al. (2020). Each rule can be executed against a database, a table in this case, yielding a result based on the operation it represents. As these graphs represent factual statements, the root is a boolean operation that should return True upon execution. Fig. 2 shows an example of a table with its caption and logical form.

3.1.1. Logical form grammar

The grammar contains several non-terminals (nodes in the graph, some of which are illustrated in Fig. 2), as follows:

Stat represents boolean comparative statements such as greater than, less than, equals (shown as *eq* in the figure), not equals, most equals or all equals, among others. This is the root of the LF graph.

[2] https://github.com/alonsoapp/tlt

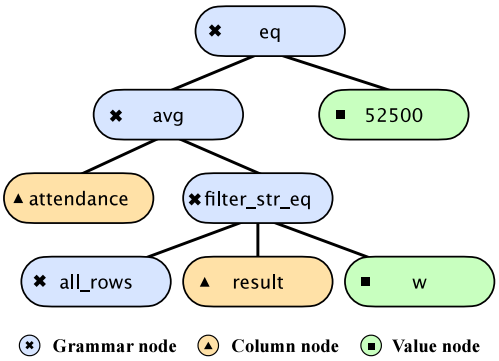
Caption:
1979 philadelphia eagles season

Table:

opponent	result	attendance
new york giants	w 23-17	67000
atlanta falcons	l 14-10	39700
new orleans saints	w 26-14	54000
new york giants	w 17-13	27500
pittsburgh steelers	w 17-14	61500

Statement: In the 1979 Philadelphia Eagles season there was an average attendance of 52500 in all winning games.

LF: eq { avg { filter_str_eq { all_rows ; result ; w } ; attendance } ; 52500 } = True



Content Selection values: 52500, w

Fig. 2. Example of a table with its caption, a logical form (in linearized and graph forms), its corresponding content selection values and the target statement. Note that *w* in the table stands for *win*. More details in the text.

C refers to an specific column in the input table (*attendance* and *result* in the figure).

V is used for specific values, which can be either values explicitly stated in the table (*w* in the figure) or arbitrary values used in comparisons or filters (*52500* in the figure).

View refers to a set of rows, which are selected according to a filter over all rows. The filters refer to specific conditions for the values in a specific column, e.g. *greater*. The figure shows *all_rows*, which returns all rows, and also *filter_str_eq* which returns the rows that contain the substring *"w"* in the *result* column.

N is used for operations that return a numeric value given a view and column as input, such as sums, averages (shown as *avg* in the figure), maximum or minimum values, and also counters.

Row is used to select a single row according to maximum or minimum values in a column.

Obj is used for operations that extract values in columns from rows (either views or specific rows). The most common operations are *hop* extractors that extract a unique value, for instance *str_hop_first* extracts a string from the first row of a given *View*.

I is used to select values from ordinal enumerations in *N* and *Row* rules, as for instance in order to select the "the 2nd highest" *I* would equal to 2.

Please refer to the Appendix C for full details. Keep in mind that *Stat*, *View*, *N*, *Row* and *Obj* are internal nodes that constitute the structure of the LF (shown in blue in the figure), while column *C*, value *V* and index *I* nodes are always leaf nodes.

We identified several ambiguities in the original grammar formulation that hindered the training of a semantic parser producing LFs.

The first one affects all functions that involve strings. Within the LF execution engine proposed by Chen, Chen, Zha et al. (2020), the implementation of those functions are divided into two: one that handles numeric and date-like strings, and a strict version for other string values. As a result, we explicitly represented these as two distinct functions within the grammar: a group for numerical and date-like values, and an additional group for other string values, denoted by the suffix "_str". The second issue addresses an inconsistency with the *hop* function. This function, when provided with a *Row*, returns the value associated to one of its columns. Although the grammar specifies that these functions are exclusively applied to *Row* objects, in 25% of the dataset examples, the function is used on a *View* object instead, which can encompass multiple rows. To address this, we defined a new function *hop_first* tailored to these specific situations.

The grammar in Appendix C contains the new rules that fix the ambiguity issues. We also converted automatically each LF in the dataset to conform to the unambiguous grammar. The conversion script is publicly available.

3.1.2. Content selection

To isolate the impact of content selection and full LFs, we extracted the LF values, allowing us to evaluate model performance with and without content selection. These extracted values include those explicitly stated in table cells, as well as other values existing in the LF but not explicitly present in the table, such as results of arithmetic operations. This set of values constitutes the supplementary input to the systems when using content selection (CS for short), categorized as follows:

- **TAB**: Values present in a table cell, verbatim or as a substring of the cell values.
  Fig. 2 shows an example, where "w" is a substring in several cells. 72.2% of the values are of this type.
- **INF**: Values not in the table that are inferred, e.g. as a result of an arithmetic operation over values in the table. For instance 52,500 in Fig. 2 corresponds to the average over attendance values. 20.8% of *Value* nodes are INF.
- **AUX**: Auxiliary values not in the table nor INF that are used in operations, e.g. to be compared to actual values in cells, as in "*All scores are bigger than 5.*". Only 7.1% are of type AUX.

In principle, one could train a separate model to select and generate all necessary content selection values for input into any Table-to-Text model, as follows: (1) Choose values from table cells, whether in full or as substrings (TAB); (2) Infer values through operations like average, count, or max (INF); (3) Induce values for use in comparisons (AUX). In order to separate the contribution of content selection and the generation of LFs, we chose to focus on using content selection and not yet on producing the actual values. Hence, we derive these values from the manual gold reference LFs, i.e., human-made reference logical forms provided in the dataset, and feed them to the models. The experiments will demonstrate that this content selection step is critical, and that current models fail without it. We leave the task of automatic content selection for further research.

3.2. Generating text via logical forms

Our Text-to-Logic-to-Text (**TℓT**) system has two main modules in a pipeline:

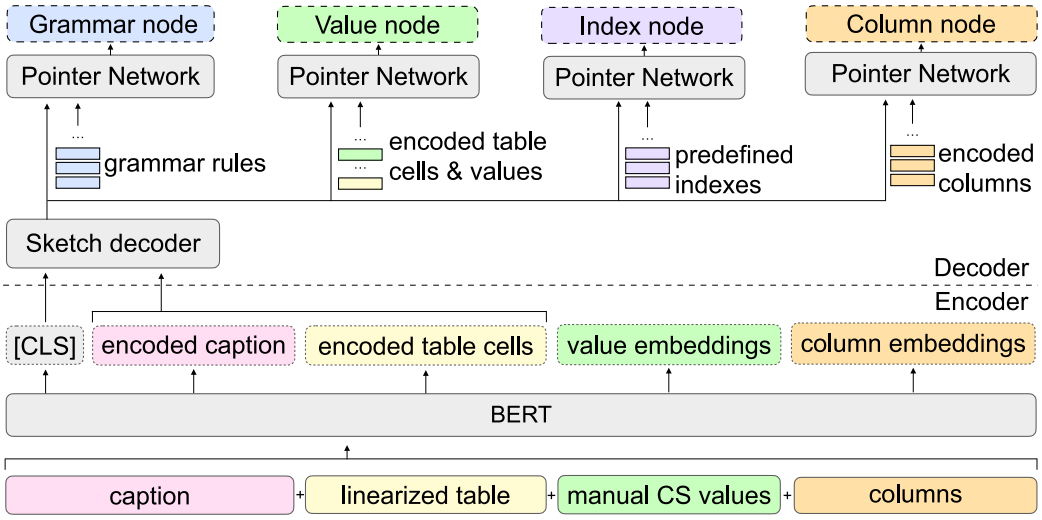


Fig. 3. Table2Logic architecture, with input in the top and output in the bottom. See text for details.

Given a table, its caption and, optionally, selected content, **Table2Logic** generates an LF; With the same table information, plus the generated LF, **Logic2Text** produces the statement text.

3.2.1. Table2Logic module

We frame this model as semantic parsing, adapting the IRNet grammar-based decoder by Guo et al. (2019) to LFs. More specifically, we follow the implementation of Valuenet by Brunner and Stockinger (2021), which is a more up to date revision of IRNet. Both models are NL-to-SQL semantic parsers that generate grammatically correct SQL sentences based on their descriptions. We adapted the system to produce logical forms instead of SQL. The architecture of Table2Logic is presented in Fig. 3.

We first feed a pre-trained BERT encoder (Devlin et al., 2019) with the concatenation of the following table data: the caption text, the table content in linearized form, the column names, and, in some of our model configurations, a set of content selection values manually extracted from the associated gold reference LF. The details about content selection values are presented in Section 3.1.2.

The output embeddings of the *CLS* token, the caption tokens and the linearized values in the table are fed into an LSTM decoder (Hochreiter & Schmidhuber, 1997). At each decoding step, the attention vector of the LSTM is used by four different pointer networks (Vinyals, Fortunato, & Jaitly, 2015). Each of these pointer networks specializes in generating one node type: *grammar*, *Value*, *Column* and *Index*. We follow a constrained decoding strategy where a pointer network is selected based on the node type that should follow the previously generated ones according to the grammar of LFs. Each of these pointer networks utilize the previously mentioned attention vector alongside a set of embeddings. In the case of *Value* and *Column* node types, these embeddings consist of the CS values and column encodings produced by the BERT model. On the other hand, *Index* and *grammar* node types use a separate set of predefined embeddings associated to each ordinal index and LF grammar rule respectively.

Following Guo et al. (2019), Table2Logic performs two decoding iterations. In a first iteration, a sketch LF is generated using the grammar pointer network. The sketch LF consisting only of grammar related nodes (e.g. those in blue in Fig. 2), where *Value*, *Column* and *Index* nodes are represented by placeholders that are filled in a second decoding iteration by the corresponding pointer network.

We follow a teacher-based training strategy to calculate the loss for each decoding iteration. In the first iteration the loss is calculated by accumulating the cross entropy loss for each generated grammar node given the previous gold reference nodes. The sketch is then used to calculate the cross entropy loss of generating *Value*, *Column* and *Index* nodes. The weights of the network are updated using the sum of both loss values.

During inference, we use beam search to produce a set of candidates. In addition, we explore a False Candidate Rejection (FCR) policy to filter out all LFs in the beam representing a *False* statement, as they would lead to a factually incorrect sentence. As previously mentioned in 3.1, the root node of each LF always consists of a boolean grammar rule. The structured nature of LFs enables us to automatically execute them against a data source, in this case, the table. Consequently, each LF yields either *True* or *False* based on the relationships between the various facts it encompasses. We exploit this property of LFs to discard all generated LFs that, despite their grammatical correctness, convey a *False* statement. Thus, only the candidate LF in the beam that executes to *True* with maximum beam probability is be selected. Section 4.3 reports experiments with FCR.

3.2.2. Logic2Text module

For the language realization model we use the top performer in Chen, Chen, Zha et al. (2020). This model consists on a GPT-2 (Radford et al., 2019) pre-trained large language model (LLM) fine-tuned to generate text from tables and human-produced logical forms. The implementation is rather simple; the input sequence is a concatenation of the table caption, table headers, and the linearized table content and logical form. The model, referred to as Logic2Text, receives this input and generates a sentence that is strongly conditioned by the semantic represented by the provided LF. The Logic2Text model enables us to produce natural language statements based on the automatic LFs produced by our Table2Logic model.

4. Experiments

In this section we report the results on text generation using the test split of the Logic2Text dataset. We first introduce the dataset, the different models, the automatic evaluation and the manual evaluation.

4.1. Dataset

We use the dataset introduced by Chen, Chen, Zha et al. (2020), a human-annotated dataset comprising 4992 open-domain tables obtained from the LogicNLG dataset (Chen, Chen, Su et al., 2020). Each table is paired with an average of 2 human-written statements describing facts within the table. Following a predefined questionnaire,

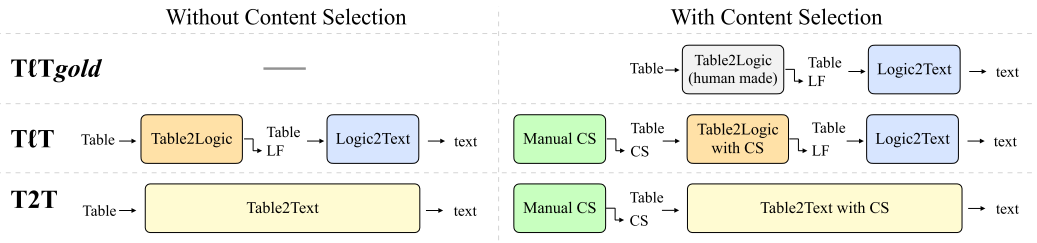


Fig. 4. Model configurations used in the main experiments.

each annotator describes the logic behind these statements. Subsequently, Chen, Chen, Zha et al. (2020) use the given answers to derive the LFs associated with each statement. The resulting dataset contains a total of 10,753 examples (8566 train, 1092 dev. and 1095 test) of high quality human-produced LFs alongside its corresponding statement and table information. We refer to these manually produced LFs as gold LFs, in contrast to the automatic LFs produced by our model. As mentioned in the introduction, Table-to-Text tasks are underspecified, allowing many other statements (and LFs) not provided in the dataset to be factually correct and equally informative as the ones in it.

4.2. Model configurations

The configuration of the different models are shown in Fig. 4. All models take as input the table information, including table caption, linearized table and column headers. In the top row, we include the upperbound system **T𝓁T**$_{gold}$, which takes the table plus the manually produced gold reference LF as input. In the middle row we include our system **T𝓁T**, which is composed by the Table2Logic module and the Logic2Text module. Both **T𝓁T** and **T𝓁T**$_{gold}$ use the same Logic2Text module, but while the first uses automatically produced LFs, the second uses manual LFs. **T𝓁T** is evaluated in two variants, with and without content selection (**T𝓁T** and **T𝓁T**$_{noCS}$, respectively). Logic2Text uses default hyperparameters (Chen, Chen, Zha et al., 2020).

The bottom row shows our baselines (T2T, short for Table2Text), which generate the text directly from table information, with and without content selection data. Since Logic2Text is based on state-of-the-art generation (Chen, Chen, Zha et al., 2020), and to ensure compatibility, both T2T and T2T$_{noCS}$ have the share codebase. That is, T2T uses the same GPT-2 model architecture as in Chen, Chen, Zha et al. (2020) but trained without LFs. Receiving only the linearized table (in case of T2T$_{noCS}$) and, in the case of T2T, the same list of manual CS values as **T𝓁T**.

4.3. Content selection ablation study

In order to develop Table2Logic, we examined the influence of content selection, along with the impact of rejecting LFs that evaluate to *False* (FCR) in development data. Accuracy was computed using strict equality with respect to any of the manual gold reference LFs. Both sketch accuracy (using placeholders for non-grammar nodes) and full accuracy are reported. As mentioned in the introduction, this task is underspecified, in that multiple LFs which are very different from the gold reference LFs could be also correct. Still, the accuracy is a good proxy of quality to discriminate between better and worse models. The results correspond to the checkpoints, out of 50 epochs, with the best full accuracy on development. We tuned some hyperparameters on development and used default values for the rest (see Appendix B for details).

Table 1 shows the results for different subsets of content selection values, with the last row reporting results when FCR is used. Without FCR, the most important set of values are those explicit in the table (TAB), and the best results correspond to the use of all values, although AUX values do not seem to help much (in fact, the best non-FCR full results are obtained without using AUX, by a very small margin). The last row reports a sizeable improvement in accuracy for full LFs when using FCR, showing that FCR is useful to reject faulty LFs that do not evaluate to True.

Table 1
Table2Logic: Accuracy (% on dev.) over sketch and full versions of gold LFs using different subsets of content selection (CS) and FCR in development. First row for **T𝓁T**$_{noCS}$, last row for **T𝓁T**, as introduced in Section 4.

Model	Sketch	Full
No content selection (**T𝓁T**$_{noCS}$)	15.0	4.9
AUX	14.0	6.2
INF	28.7	11.0
TAB	42.6	27.3
TAB, INF	56.5	39.3
TAB, AUX	44.3	28.6
TAB, INF, AUX	**58.5**	38.9
TAB, INF, AUX + FCR (**T𝓁T**)	56.0	**46.5**

Overall, the full accuracy of **T𝓁T** might seem low, but given that the gold reference LFs only cover a fraction of possible LFs they are actually of good quality, as we will see in the next sections.

We also performed an additional ablation experiment where we removed the table information from the system in the last row (**T𝓁T**). The sketch and full accuracies dropped (50.3 and 42.7 respectively), showing that access to table information is useful even when content selection is available.

4.4. Automatic evaluation

The automatic metrics compare the produced description with the reference descriptions in the test split. As shown in Table 2, we report the same n-gram similarity automatic metrics as in Chen, Chen, Zha et al. (2020), BLEU-4 (B-4) (Papineni, Roukos, Ward, & Zhu, 2002), ROUGE-1, 2, and L (R-1, R-2, and R-L for short) (Lin, 2004), along with two additional metrics BERTscore (BERTs) (Zhang, Kishore, Wu, Weinberger, & Artzi, 2019) and BARTscore (BARTs) (Yuan, Neubig, & Liu, 2021) which can capture the semantic similarity between the ground truth and generation results. The results show that generation without content selection is poor for both the baseline system and our system (T2T$_{noCS}$ and **T𝓁T**$_{noCS}$, respectively). Content selection is key for good results in both kinds of systems, which improve around 10 points in all metrics when incorporating content selection (T2T and **T𝓁T**). Automatic generation of LFs (**T𝓁T**) allows to improve over the system not using them (T2T) in at least one point. If **T𝓁T** had access to correct LFs it would improve 4 points further, as shown by the **T𝓁T**$_{gold}$ results. Observe that our results for **T𝓁T**$_{gold}$ are very similar to those reported in Chen, Chen, Zha et al. (2020), as shown in the last row. We attribute the difference to minor variations in the model released by the authors.

4.5. Human fidelity evaluation

Given the cost of human evaluation, we selected three models to manually judge the fidelity of the produced descriptions: the baseline

Table 2
Automated n-gram similarity metrics for textual descriptions (test). BLEU-4 (B-4), ROUGE-1, 2, and L (R-1, R-2, and R-L), BERTscore (BERTs) and BARTscore (BARTs). Bottom two rows are upperbounds, as they use manual LFs. See text for system description. Both BERTs and BARTs correspond to the f1 score. In case of the BARTscore higher is better.

Model	B-4	R-1	R-2	R-L	BERTs	BARTs
$T2T_{noCS}$	16.8	37.7	19.3	31.6	88.8	−4.04
$\mathbf{T\ell T}_{noCS}$	15.6	39.0	18.9	32.2	87.9	−4.03
T2T	26.8	55.2	31.5	45.7	91.9	**−2.98**
TℓT (ours)	**27.2**	**56.0**	**33.1**	**47.7**	**92.0**	−2.99
$\mathbf{T\ell T}_{gold}$	31.7	62.4	38.7	52.8	93.1	−2.65
$\mathbf{T\ell T}_{gold}$ [a]	31.4[a]	64.2[a]	39.5[a]	54.0[a]	-	-

[a] For results reported in Chen, Chen, Zha et al. (2020).

T2T model, our **TℓT** model and the upperbound with manual LFs, $\mathbf{T\ell T}_{gold}$. For this, we randomly selected 90 tables from the test set and generated a statement with each of the three models. In order to have two human judgments per example, we provided each evaluator with 30 sentences, along with the corresponding table and caption. The evaluators were asked to select whether the description is true, false or nonsense according to the caption and the table.This group of evaluators was comprised of eighteen volunteer researchers unrelated to this project. We use Fleiss' kappa coefficient (Fleiss, 1971) to measure the inter-evaluator agreement. This coefficient is a statistical measure used to assess the level of agreement among multiple raters when categorizing items into different classes. It takes into account both the observed agreement and the agreement expected by chance. It is a way to determine the extent to which the agreement among raters goes beyond what would be expected due to random chance alone. The coefficient ranges from −1 to 1, where higher values indicate better agreement beyond chance, while lower values indicate poor agreement. The evaluation concluded with a strong 0.84 Fleiss' kappa coefficient. We discarded examples where there was disagreement.

Table 3 shows the fidelity figures for the three models. After the evaluation, we noticed that the faithfulness results for $\mathbf{T\ell T}_{gold}$ in our experiment matched the figure reported by Chen, Chen, Zha et al. (2020), so we decided, for completeness, to include in the table their figures for $T2T_{noCS}$, which should be roughly comparable to the other results in the table.

In general, the differences in human fidelity evaluation are much higher than for automatic metrics, which we attribute to widely recognized issues of automatic metrics when evaluating text generation. In our case, the two most significant issues are the ones affecting n-gram overlaping metrics (e.g., BLUE, ROUGE). These automatic metrics exhibit insensitivity to semantic and pragmatic quality, making them fail to capture the semantic and pragmatic nuances of language. This can lead to models generating text that, despite being technically correct in terms of word overlap, can still be semantically inaccurate (Zhang et al., 2019). Furthermore, these metrics can also suffer from a lack of correlation with human judgment, leading to models that could generate text that is grammatically correct but incoherent and meaningfulness, yet receives a high score (Moramarco et al., 2022). From low to high, the results allow us to estimate the **separate contributions** of each component in absolute fidelity points:

- **Manual content selection** improves fidelity in 24 points ($T2T_{noCS}$ vs. T2T) ;
- **Automatic LFs** improve an additional 30 points (T2T vs. **TℓT**);
- **Manual LFs** give 7 points (**TℓT** vs. $\mathbf{T\ell T}_{gold}$);
- **Perfect Logic2Text** generation would yield 18 points ($\mathbf{T\ell T}_{gold}$ vs. 100%).

The figures confirm our contribution: it is possible to produce logical forms automatically, and they allow to greatly improve fidelity, with the largest fidelity improvement in the table, 30 absolute points, which correspond to a 67% improvement over the comparable system not using LFs. Note that the other improvements are actually gaps which allow us to prioritize the areas for further research: automatic content selection (24 pt.), better Logic2Text (18 pt.) and better Table2Logic (7 pt.). In the following section we analyze the errors in the two later modules.

Table 3
Human evaluation fidelity results. Given 90 test samples to three different model configurations, percentage of generated sentences identified as Faithful, Unfaithful or Nonsense by evaluators. Answer with full disagreement between evaluators are discarded.

Model	Faithful	Unfaithful	Nonsense
$T2T_{noCS}$ [a]	20.2[a]	79.8[a]	–
T2T	44.9	49.3	5.8
TℓT (ours)	**75.0**	**20.3**	**4.7**
$\mathbf{T\ell T}_{gold}$	82.4	13.51	4.1

[a] For results reported in Chen, Chen, Zha et al. (2020).

Table 4
Logic: Distribution of differing node types (**TℓT** vs. gold LFs). Fr. for frequency of node type in differing LFs, Total for overall frequency in gold. Rightmost column for most frequent confusions (**TℓT** → gold).

	Fr.	Total	Confusions
Stat	0.38	0.13	greater → less all equals → most equals equals → and
C	0.25	0.19	column 3 → column 0 column 1 → column 0
Row	0.16	0.02	row 0 → row 2 row 2 → row 0 row 2 → row 1
View	0.11	0.20	filter_greater → filter_less filter_greater → filter_eq filter_eq → all_rows
N	0.05	0.03	sum → avg avg → sum
Obj	0.03	0.26	str_hop → num_hop num_hop → str_hop
V	0.01	0.16	value 72 → value 73 value 70 → value 71
I	0.01	0.01	1 → 0

4.6. Qualitative analysis

We performed a qualitative analysis of failure cases in both Table2Logic and Logic2Text, as well as examples of factually correct descriptions generated from LFs different from gold LFs.

4.6.1. Table2Logic

We automatically compared the LFs generated by **TℓT** in the development set that did not match their corresponding gold LFs. Note that the produced LFs can be correct even if they do not match the gold LF. We traverse the LF from left to right and record the first

Table 5
Examples of faithful sentences produced by **TℓT** from intermediate LFs that do not match the gold LF.

LF difference	Sentences
Similar structure, semantically equivalent	**TℓT**: In the list of Appalachian regional commission counties, Schoharie has the highest unemployment rate. Human: The appalachian county that has the highest unemployment rate is Schoharie.
Similar structure, semantically different	**TℓT**: Dick Rathmann had a lower rank in 1956 than he did in 1959. Human: Dick Rathmann completed more laps in the Indianapolis 500 in 1956 than in 1959.
Different structure, semantically different	**TℓT**: Most of the games of the 2005 Houston Astros' season were played in the location of arlington. Human: Arlington was the first location used in the 2005 Houston Astros season.
Simpler structure, more informative	**TℓT**: Aus won 7 events in the 2006 asp world tour. Human: Seven of the individuals that were the runner up were from aus.

node that is different. Table 4 shows, in decreasing order of frequency, each grammar node type (cf. Section 3.1.1) with the most frequent confusions.

The most frequent differences focus on *Stat* nodes, where a different comparison is often generated. The next two frequent nodes are column and row selections, where **TℓT** selects different columns and rows, even if **TℓT** has access to the values in the content selection. The frequency of differences of these three node types is well above the distribution in gold LFs. The rest of differences are less frequent, and also focus on generating different comparison or arithmetic operations.

4.6.2. Logic2Text

The faithfulness score of descriptions generated from gold LFs (**TℓT**$_{gold}$) is 82%, so we analyzed a sample of the examples in this 18%. For the sake of space, we include full examples in Appendix D, which include table, caption, gold LF and generated description. We summarize the errors in three types:

Comparative arithmetic: Logic2Text miss-represented comparative arithmetic action rules in the LF in 40% of the cases. This resulted in cases where the output sentence declared that a given value was *smaller* than another when the LF stated it was *larger*. Logic2Text also seem to ignore *round* and *most* modifiers of comparison operations, producing sentences with strict equality and omitting qualifiers like "roughly" or "most". The absence of these qualifiers made the produced sentences factually incorrect.

The reason behind these types of errors remain uncertain. One plausible explanation could be linked to the limited number of parameters within the models of this architecture. While these models are capable of recognizing the need for a comparative rule at a given step, their size may still be insufficient for effectively distinguishing between two potential comparisons of the same category, e.g. *smaller* and *larger*. Another contributing factor may be related to the small amount of occurrences of each type of comparative rule within the training dataset. Only 44% of LFs in the training set contain any of the 22 comparative arithmetic action rules. Finally, we must also highlight that models that do not use LFs also incur in these kind of errors, showing that these are common errors across different model architectures and are not exclusive to our specific model.

LF omission: Logic2Text disregarded part of the LF (33% of errors), resulting in omissions that led to false sentences. Many of these errors involved omitting an entire branch of the LF, leading, for instance, to sentences wrongly referring to all the instances in the data instead of the subset described in the LF.

Verbalization: Logic2Text incurred in wrong verbalization and misspellings (27% of cases). For instance Logic2Text producing a similar but not identical name like in *foulisco* instead of *francisco*.

We attribute the errors to the fact that the generator is based on a general Language Model such as GPT-2. While these language models are excellent in producing fluent text, it seems that, even after fine-tuning, they have a tendency to produce sentences that do not fully reflect the data in the input logical form. It also seems that the errors might be explained by the lower frequency of some operations. The 18% gap, even if it is much lower than the gap for systems that do not use LFs, together with this analysis, show that there is still room for improvement.

4.6.3. Implications of divergent LF production from gold reference LF

The results in Table 1 show that our Table2Logic system has low accuracy when evaluated against gold logical forms (46%). On the contrary, the results in fidelity for the text generated using those automatically generated logical forms is very high, 75%, only 7 absolute points lower to the results when using gold logical forms. This high performance in fidelity for automatic LFs might seem counter-intuitive, but we need to note that it is possible to generate a correct and faithful LF that is completely different from the gold logical form, i.e. the system decides to produce a correct LF that focuses on a different aspect of the information in the table with respect to the gold LF.

In order to check whether this is actually the case, we manually examined the automatic LFs from **TℓT** that resulted in faithful sentences in the manual evaluation while being "erroneous", that is, different from their gold LF references. In all cases, such **TℓT** LFs are correctly formed and faithful, i.e. even if these LFs where "wrong" according to the strict definition of accuracy, the semantics they represent are informative and faithful to the source data. Table 5 shows a sample of the output sentence, with full details including table and LFs in Appendix E.

We categorized the samples as follows. 69% of them share a similar LF structure as their corresponding gold references, but with changes in key *Value* or *Column* nodes, making them semantically different. In 15% of the cases the LF had similar structure, but although there were differences, the LF was semantically equivalent to the gold LF. The rest of **TℓT** LFs (16%) had a different structure, and where semantically different from reference counterparts, while still being correct and faithful to the table. This reflects an interesting aspect of reference-based evaluation. In many cases, generating a sentence that diverges from the reference does not imply that such a sentence is less faithful, useful or informative. Thus, the accuracy evaluation with respect to gold LFs (cf. Table 1) provides an underestimate of the quality of the produced LFs and texts.

All in all the quality of LFs and corresponding text produced by **TℓT** for this sample is comparable to those of the gold LF, and in some cases more concise and informative. This analysis confirms that the quality of Table2Logic is well over the 46% accuracy estimate, and that it can be improved, as the produced text lags 7 points behind gold LFs.

5. Conclusions and future work

We have presented **TℓT** which, given a table and a selection of the content, first produces logical forms and then the textual statement. We show for the first time that automatic LFs improve results according to automatic metrics and, especially, manually estimated factual correctness. In addition, we separately study the contribution of content selection and the formalization of the output as an LF, showing a higher impact in fidelity of the later. In this paper, our focus is on tables. However, our findings and software can readily be extended to other structured inputs. Given that the grammar of LFs is independent of the table format, it can be easily adjusted for other common data-to-text inputs such as graphs or triplets by modifying its execution engine, keeping the LFs intact.

Our contribution enables future Data-to-Text applications to leverage the advantages of using factually verifiable logical forms, eliminating the need of manually constructed LFs. These advantages include a relative improvement in fidelity of 67% compared to baseline models, along with the ability to access an intermediate formal representation within the generation process. This facilitates the automated validation of a statement's factual accuracy before generating its corresponding natural language representation. The improvement in fidelity attained by our model is relevant for most Data-to-Text applications, where faithfulness is crucial.

The conducted analysis also enabled us to quantify that content selection would offer the most substantial performance improvement, followed to a lesser extent by improved logic-to-text generation, and, finally, improved table-to-logic generation. In the future, we plan to focus on automatic content selection, which we think can be largely learned from user preference patterns found in the training data. Recent advances in semantic parsing, e.g. the use of larger language models (BigScience Workshop, 2022; Raffel et al., 2020; Zhang et al., 2022), could also be easily folded in our system and would further increase the contribution of LFs. Finally, we also plan to make use of our qualitative analysis to explore complementary approaches for improving factual correctness in logic-to-text.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Readers can find the link to the publicly available data and code in the paper.

Acknowledgments

This work is partially funded by MCIN/AEI 10.13039/50110001 1033, Spain and by the European Union NextGenerationEU/PRTR, Spain, as well as the Basque Government, Spain IT1570-22.

Appendix A. Training procedure

All experiments where carried out in a machine with a GPU NVIDIA TITAN Xp 12 GB. The average training runtime for all Table2Logic based models is 19 h. For the Logic2Text presented models, it averaged 10 h. Both Table2Logic and Logic2Text models have a very similar amount of parameters (117M).

Appendix B. Model hyper-parameters

We keep Logic2Text's hyper-parameters the same as Chen, Chen, Zha et al. (2020). We refer the reader to the paper. Regarding the Table2Logic model in **TℓT**, which is based on Brunner and Stockinger (2021)'s Valuenet, we changed the grammar and added additional input data, as well as changing the code accordingly to our use case. We use the same hyper-parameters as stated in the paper, with the exception of the base learning rate, beam size, number epochs, and gradient clipping. This is the list of hyper-parameters used by Table2Logic for the model **TℓT**:

Random seed: 90	Attention vector size: 300
Maximum sequence lengthy: 512	Grammar type embedding size: 128
Batch size: 8	Grammar node embedding size: 128
Epochs: 50	Column node embedding size: 300
Base learning rate: $5 * 10^{-5}$	Index node embedding size: 300
Connection learning rate: $1 * 10^{-4}$	Readout: 'identity'
Transformer learning rate: $2 * 10^{-5}$	Column attention: 'affine'
Scheduler gamma: 0.5	Dropout rate: 0.3
ADAM maximum gradient norm: 1.0	Largest index for I nodes: 20
Gradient clipping: 0.1	Include OOV token: True
Loss epoch threshold: 50	Beam size: 2048
Sketch loss weight: 1.0	Max decoding steps: 50
Word embedding size: 300	False Candidate Rejection: True
Size of LSTM hidden states: 300	

Appendix C. Logical form grammar

See Fig. C.5

Appendix D. Logic2text errors

This section shows examples of error cases where the logic-to-text stage of the pipeline failed to produce faithful sentences given a gold LF. We include one example for each error type, including table, caption, gold logical form and generated description. See Section 4.6.2 for more details.

D.1. Comparative arithmetic

See Table D.1.
Logical Form:

```
and
├── only
│   └── filter_greater
│       ├── 0
│       ├── all_rows
│       └── bronze
└── str_eq
    ├── austria
    └── str_hop_first
        ├── filter_greater
        │   ├── 0
        │   ├── all_rows
        │   └── bronze
        └── nation
```

TℓT sentence: austria was the only country to win 0 bronze medals at the fil world luge championships.
Gold sentence: austria was the only country to have bronze medals in the luge championship in 1961.

D.2. LF omission

See Table D.2.
Logical Form:

```
eq
├── 8
└── count
    └── filter_str_eq
        ├── all_rows
        ├── forest steppe
        └── habitat type
```

TℓT sentence: there are 8 habitats that can be found in moldova.
Gold sentence: 8 land formations are classified with a habitat type of forest steppe.

```
Stat ::= only View | and Stat Stat | greater Obj Obj | less Obj Obj | eq Obj Obj |
         str_eq Obj Obj | not_eq Obj Obj | not_str_eq Obj Obj | round_eq Obj Obj |
         all_eq View C Obj | all_str_eq View C Obj | all_not_eq View C Obj |
         all_str_not_eq View C Obj | all_less View C Obj | all_less_eq View C Obj |
         all_greater View C Obj | all_greater_eq View C Obj | most_eq View C Obj |
         most_str_eq View C Obj | most_not_eq View C Obj |
         most_str_not_eq View C Obj | most_less View C Obj | most_less_eq View C Obj |
         most_greater View C Obj | most_greater_eq View C Obj
View ::= all_rows | filter_eq View C Obj | filter_str_eq View C Obj |
         filter_not_eq View C Obj | filter_str_not_eq View C Obj |
         filter_less View C Obj | filter_greater View C Obj | filter_greater_eq View C Obj |
         filter_less_eq View C Obj | filter_all View C
   N ::= count View | avg View C | sum View C | max View C | min View C |
         nth_max View C I | nth_min View C I
 Row ::= argmax View C | argmin View C | nth_argmax View C I | nth_argmin View C I
 Obj ::= str_hop Row C | num_hop Row C | str_hop_first View C |
         num_hop_first View C | diff Obj Obj | N | V
   C ::= column
   I ::= index
   V ::= value
```

Fig. C.5. The logical form Grammar after fixing the ambiguity issues in the original version (Chen, Chen, Zha et al., 2020). We follow the same notation as in IRNet and Valuenet. The tokens to the left of the ::= represent non-terminals (node types in the graph). Tokens in italics represent the possible rules for each node, with pipes (|) separating the rules. The rules added to the original grammar in order to fix ambiguity issues are highlighted in green.

Table D.1
Table example titled "Fil world luge championships 1961.".

Rank	Nation	Gold	Silver	Bronze	Total
1	austria	0	0	3	3
2	italy	1	1	0	2
3	west germany	0	2	0	2
4	poland	1	0	0	1
5	switzerland	1	0	0	1

D.3. Verbalization

See Table D.3.

Logical Form:

```
greater
├── num_hop_first
│   ├── filter_str_eq
│   │   ├── all_rows
│   │   ├── francisco elson
│   │   └── player
│   └── years
└── num_hop_first
    ├── filter_str_eq
    │   ├── all_rows
    │   ├── pervis ellison
    │   └── player
    └── years
```

TℓT sentence: foulisco elson played for the supersonics after pervis ellison.
Gold sentence: francisco elson played 8 years later thanpervis ellison.

Appendix E. Examples of faithful TℓT sentences where LF is different to gold

This section shows examples of automatic LFs from **TℓT** that resulted in faithful sentences in the manual evaluation while being different from their gold LF references. Each example extends the information shown in Table 5.

E.1. Similar structure, semantically equivalent

Table E.1.

TℓT Logical Form:

```
str_eq
├── schoharie
└── str_hop
    ├── county
    └── nth_argmax
        ├── 1
        ├── all_rows
        └── unemployment rate
```

Gold Logical Form:

```
str_eq
├── schoharie
└── str_hop
    ├── argmax
    │   ├── all_rows
    │   └── unemployment rate
    └── county
```

TℓT sentence: in the list of appalachian regional commission counties, schoharie has the highest unemployment rate.
Human sentence: the appalachian county that has the highest unemployment rate is schoharie.

E.2. Similar structure, semantically different

Table E.2.

TℓT Logical Form:

```
less
├── num_hop_first
│   ├── filter_str_eq
│   │   ├── 1956
│   │   ├── all_rows
│   │   └── year
│   └── rank
└── num_hop_first
    ├── filter_str_eq
    │   ├── 1959
    │   ├── all_rows
    │   └── year
    └── laps
```

Table D.2
Table example titled "Geography of moldova.".

Land formation	Area, km square	Of which currently forests, km square	% forests	Habitat type
northern moldavian hills	4630	476	10.3%	forest steppe
dniester - rāut ridge	2480	363	14.6%	forest steppe
middle prut valley	2930	312	10.6%	forest steppe
bălţ steppe	1920	51	2.7%	steppe
ciuluc - soloneţ hills	1690	169	10.0%	forest steppe
cornești hills (codru)	4740	1300	27.5%	forest
lower dniester hills	3040	371	12.2%	forest steppe
lower prut valley	1810	144	8.0%	forest steppe
tigheci hills	3550	533	15.0%	forest steppe
bugeac plain	3210	195	6.1%	steppe
part of podolian plateau	1920	175	9.1%	forest steppe
part of eurasian steppe	1920	140	7.3%	steppe

Table D.3
Table example titled "Seattle supersonics all - time roster.".

Player	Nationality	Jersey number (s)	Position	Years	From
craig ehlo	united states	3	sg	1996–1997	washington state
dale ellis	united states	3	sg/sf	1986–1991 1997–1999	tennessee
pervis ellison	united states	29	c	2000	louisville
francisco elson	netherlands	16	c	2008	california
reggie evans	united states	34 , 30	pf	2002–2006	iowa
patrick ewing	united states	33	center	2000–2001	georgetown

Table E.1
Table example titled "List of appalachian regional commission counties.".

County	Population	Unemployment rate	Market income per capita	Poverty rate	Status
allegany	49927	5.8%	16850	15.5%	- risk
broome	200536	5.0%	24199	12.8%	transitional
cattaraugus	83955	5.5%	21285	13.7%	transitional
chautauqua	136409	4.9%	19622	13.8%	transitional
chemung	91070	5.1%	22513	13.0%	transitional
chenango	51401	5.5%	20896	14.4%	transitional
cortland	48599	5.7%	21134	15.5%	transitional
delaware	48055	4.9%	21160	12.9%	transitional
otsego	61676	4.9%	21819	14.9%	transitional
schoharie	31582	6.0%	23145	11.4%	transitional
schuyler	19224	5.4%	21042	11.8%	transitional
steuben	98726	5.6%	28065	13.2%	transitional
tioga	51784	4.8%	24885	8.4%	transitional

Gold Logical Form:

```
greater
├── num_hop_first
│   ├── filter_str_eq
│   │   ├── 1956
│   │   ├── all_rows
│   │   └── year
│   └── laps
└── num_hop_first
    ├── filter_str_eq
    │   ├── 1959
    │   ├── all_rows
    │   └── year
    └── laps
```

TℓT sentence: dick rathmann had a lower rank in 1956 than he did in 1959.

Human sentence: dick rathmann completed more laps in the indianapolis 500 in 1956 than in 1959.

Table E.2
Table example titled "Dick rathmann.".

Year	Qual	Rank	Finish	Laps
1950	130.928	17	32	25
1956	144.471	6	5	200
1957	140.780	withdrew	withdrew	withdrew
1958	145.974	1	27	0
1959	144.248	5	20	150
1960	145.543	6	31	42
1961	146.033	8	13	164
1962	147.161	13	24	51
1963	149.130	14	10	200
1964	151.860	17	7	197

E.3. Different structure, semantically different

Table E.3.

Table E.3
Table example titled "2005 houston astros season.".

Date	Winning team	Score	Winning pitcher	Losing pitcher	Attendance	Location
may 20	texas	7 - 3	kenny rogers	brandon backe	38109	arlington
may 21	texas	18 - 3	chris young	ezequiel astacio	35781	arlington
may 22	texas	2 - 0	chan ho park	roy oswalt	40583	arlington
june 24	houston	5 - 2	roy oswalt	ricardo rodriguez	36199	houston
june 25	texas	6 - 5	chris young	brandon backe	41868	houston

Table E.4
Table example titled "2006 asp world tour.".

Location	Country	Event	Winner	Runner - up
gold coast	australia	roxy pro gold coast	melanie redman - carr (aus)	layne beachley (aus)
tavarua	fiji	roxy pro fiji	melanie redman - carr (aus)	layne beachley (aus)
teahupoo , tahiti	french polynesia	billabong pro tahiti women	melanie redman - carr (aus)	chelsea georgeson (aus)
itacarã	brazil	billabong girls pro	layne beachley (aus)	jessi miley - dyer (aus)
hossegor	france	rip curl pro mademoiselle	chelsea georgeson (aus)	melanie redman - carr (aus)
manly beach	australia	havaianas beachley classic	stephanie gilmore (aus)	layne beachley (aus)
sunset beach , hawaii	united states	roxy pro	melanie bartels (haw)	stephanie gilmore (aus)
honolua bay , hawaii	united states	billabong pro	jessi miley - dyer (aus)	keala kennelly (haw)

T&T Logical Form:

```
most_str_eq
├── all_rows
├── arlington
└── location
```

Gold Logical Form:

```
str_eq
├── arlington
└── str_hop
    ├── argmin
    │   ├── all_rows
    │   └── date
    └── location
```

T&T sentence: most of the games of the 2005 houston astros' season were played in the location of arlington.
Human sentence: arlington was the first location used in the 2005 houston astros season.

E.4. Simpler, more informative semantic

Table E.4.

T&T Logical Form:

```
eq
├── 7
└── count
    └── filter_str_eq
        ├── all_rows
        ├── aus
        └── winner
```

Gold Logical Form:

```
eq
├── 7
└── count
    └── filter_str_eq
        ├── all_rows
        ├── aus
        └── runner - up
```

T&T sentence: aus won 7 events in the 2006 asp world tour.
Human sentence: seven of the individuals that were the runner up were from aus.

Appendix F. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2023.121869.

References

Aghajanyan, A., Okhonko, D., Lewis, M., Joshi, M., Xu, H., Ghosh, G., & Zettlemoyer, L. (2022). HTLM: Hyper-text pre-training and prompting of language models. In *International conference on learning representations*. URL https://openreview.net/forum?id=P-pPW1nxf1r.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., & Schneider, N. (2012). Abstract meaning representation (amr) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL* (pp. 1533–1544).

BigScience Workshop (2022). *BLOOM (revision 4ab0472)*. Hugging Face, http://dx.doi.org/10.57967/hf/0003, URL https://huggingface.co/bigscience/bloom.

Brunner, U., & Stockinger, K. (2021). ValueNet: A natural language-to-SQL system that learns from database information. In *2021 IEEE 37th international conference on data engineering* (pp. 2177–2182). http://dx.doi.org/10.1109/ICDE51399.2021.00220.

Carnap, R. (1947). *Meaning and necessity: a study in semantics and modal logic*. Chicago: University of Chicago Press.

Chen, W., Chen, J., Su, Y., Chen, Z., & Wang, W. Y. (2020). Logical natural language generation from open-domain tables. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7929–7942). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.708, Online. URL https://aclanthology.org/2020.acl-main.708.

Chen, Z., Chen, W., Zha, H., Zhou, X., Zhang, Y., Sundaresan, S., & Wang, W. Y. (2020). Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 2096–2111). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.findings-emnlp.190, Online. URL https://aclanthology.org/2020.findings-emnlp.190.

Chen, W., Su, Y., Yan, X., & Wang, W. Y. (2020). KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 8635–8648). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.emnlp-main.697, Online. URL https://aclanthology.org/2020.emnlp-main.697.

Covington, M. A. (2001). Building natural language generation systems. *Language*, *77*(3), 611–612. http://dx.doi.org/10.1353/lan.2001.0146.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). 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)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423, URL https://aclanthology.org/N19-1423.

Duboue, P. A., & McKeown, K. R. (2003). Statistical acquisition of content selection rules for natural language generation. In *Proceedings of the 2003 conference on empirical methods in natural language processing* (pp. 121–128). URL https://aclanthology.org/W03-1016.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*(5), 378.

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65–170. http://dx.doi.org/10.1613/jair.5477.

Goldberg, E., Driedger, N., & Kittredge, R. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, *9*(2), 45–53. http://dx.doi.org/10.1109/64.294135.

Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J.-G., Liu, T., & Zhang, D. (2019). Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4524–4535). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-1444, URL https://aclanthology.org/P19-1444.

Harkous, H., Groves, I., & Saffari, A. (2020). Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th international conference on computational linguistics* (pp. 2410–2424). Barcelona, Spain (Online): International Committee on Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.coling-main.218, URL https://aclanthology.org/2020.coling-main.218.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Kasner, Z., & Dusek, O. (2022). Neural pipeline for zero-shot data-to-text generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 3914–3932). Dublin, Ireland: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.acl-long.271, URL https://aclanthology.org/2022.acl-long.271.

Koehn, P., & Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the first workshop on neural machine translation* (pp. 28–39). Vancouver: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/W17-3204, URL https://aclanthology.org/W17-3204.

Lebret, R., Grangier, D., & Auli, M. (2016). Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1203–1213). Austin, Texas: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D16-1128, URL https://aclanthology.org/D16-1128.

Li, X. L., & Rush, A. (2020). Posterior control of blackbox generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 2731–2743). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.243, Online. URL https://aclanthology.org/2020.acl-main.243.

Li, L., & Wan, X. (2018). Point precisely: Towards ensuring the precision of data in generated texts using delayed copy mechanism. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1044–1055). Santa Fe, New Mexico, USA: Association for Computational Linguistics, URL https://aclanthology.org/C18-1089.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics, URL https://aclanthology.org/W04-1013.

Liu, T., Wang, K., Sha, L., Chang, B., & Sui, Z. (2018). Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the AAAI conference on artificial intelligence. Vol. 32. no.1*. http://dx.doi.org/10.1609/aaai.v32i1.11925.

Lu, X., Welleck, S., West, P., Jiang, L., Kasai, J., Khashabi, D., Bras, R. L., Qin, L., Yu, Y., & Zellers, R. (2021). Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. arXiv preprint arXiv:2112.08726.

Matsumaru, K., Takase, S., & Okazaki, N. (2020). Improving truthfulness of headline generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1335–1346). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.123, Online. URL https://aclanthology.org/2020.acl-main.123.

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1906–1919). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.173, Online. URL https://aclanthology.org/2020.acl-main.173.

Moramarco, F., Papadopoulos Korfiatis, A., Perera, M., Juric, D., Flann, J., Reiter, E., Belz, A., & Savkov, A. (2022). Human evaluation and correlation with automatic metrics in consultation note generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 5739–5754). Dublin, Ireland: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.acl-long.394, URL https://aclanthology.org/2022.acl-long.394.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318). Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/1073083.1073135, URL https://aclanthology.org/P02-1040.

Puduppully, R., Dong, L., & Lapata, M. (2019a). Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence. Vol. 33. no. 01* (pp. 6908–6915). http://dx.doi.org/10.1609/aaai.v33i01.33016908, URL https://ojs.aaai.org/index.php/AAAI/article/view/4668.

Puduppully, R., Dong, L., & Lapata, M. (2019b). Data-to-text generation with entity modeling. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2023–2035). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-1195, URL https://aclanthology.org/P19-1195.

Radford, A., Wu, J., Child, R., Luan, D., & Amodei..., D. (2019). *Language models are unsupervised multitask learners*. OpenAI Blog, URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Radhakrishnan, K., Srikantan, A., & Lin, X. V. (2020). ColloQL: Robust text-to-SQL over search queries. In *Proceedings of the first workshop on interactive and executable semantic parsing* (pp. 34–45). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.intexsempar-1.5, URL https://aclanthology.org/2020.intexsempar-1.5.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67, URL http://jmlr.org/papers/v21/20-074.html.

Rebuffel, C., Soulier, L., Scoutheeten, G., & Gallinari, P. (2020). A hierarchical model for data-to-text generation. In *European conference on information retrieval* (pp. 65–80). Springer, http://dx.doi.org/10.1007/978-3-030-45439-5_5.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(1), 57–87. http://dx.doi.org/10.1017/S1351324997001502.

Sha, L., Mou, L., Liu, T., Poupart, P., Li, S., Chang, B., & Sui, Z. (2018). Order-planning neural text generation from structured data. In *AAAI'18/IAAI'18/EAAI'18, Proceedings of the thirty-second AAAI conference on artificial intelligence and thirtieth innovative applications of artificial intelligence conference and eighth AAAI symposium on educational advances in artificial intelligence*. AAAI Press.

Shen, X., Chang, E., Su, H., Niu, C., & Klakow, D. (2020). Neural data-to-text generation via jointly learning the segmentation and correspondence. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7155–7165). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.641, Online. URL https://aclanthology.org/2020.acl-main.641.

Su, Y., Vandyke, D., Wang, S., Fang, Y., & Collier, N. (2021). Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the association for computational linguistics: EMNLP 2021* (pp. 895–909). Punta Cana, Dominican Republic: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.findings-emnlp.76, URL https://aclanthology.org/2021.findings-emnlp.76.

Tian, R., Narayan, S., Sellam, T., & Parikh, A. P. (2019). Sticking to the facts: Confident decoding for faithful data-to-text generation. arXiv preprint arXiv:1910.08684.

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems. Vol. 28*. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf.

Wang, J., Liu, D., Ip, W. H., Zhang, W., & Deters, R. (2014). Integration of system-dynamics, aspect-programming, and object-orientation in system information modeling. *IEEE Transactions on Industrial Informatics*, *10*(2), 847–853. http://dx.doi.org/10.1109/TII.2014.2300703.

Wang, Z., Wang, X., An, B., Yu, D., & Chen, C. (2020). Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1072–1086). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.101, Online. URL https://aclanthology.org/2020.acl-main.101.

Wiseman, S., Shieber, S., & Rush, A. (2017). Challenges in data-to-document generation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2253–2263). Copenhagen, Denmark: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D17-1239, URL https://aclanthology.org/D17-1239.

Wiseman, S., Shieber, S., & Rush, A. (2018). Learning neural templates for text generation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3174–3187). Brussels, Belgium: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D18-1356, URL https://aclanthology.org/D18-1356.

Yin, P., & Neubig, G. (2017). A syntactic neural model for general-purpose code generation. In *The 55th annual meeting of the association for computational linguistics*. Vancouver, Canada: URL https://arxiv.org/abs/1704.01696.

Yuan, W., Neubig, G., & Liu, P. (2021). Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, *34*, 27263–27277.

Zhang, W. (1994). *An integrated environment for CAD/CAM of mechanical systems* (Ph.D. thesis), TU Delft.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., & Zettlemoyer, L. (2022). OPT: Open pre-trained transformer language models. http://dx.doi.org/10.48550/ARXIV.2205.01068, arXiv. URL https://arxiv.org/abs/2205.01068.

PixT3: Pixel-based Table-To-Text Generation

Iñigo Alonso,
HiTZ Center - Ixa,
University of the
Basque Country UPV/EHU
inigoborja.alonso@ehu.eus

Eneko Agirre
HiTZ Center - Ixa,
University of the
Basque Country UPV/EHU
e.agirre@ehu.eus

Mirella Lapata
Institute for Language,
Cognition and Computation,
University of Edinburgh
mlap@inf.ed.ac.uk

Abstract

Table-to-text generation involves generating appropriate textual descriptions given structured tabular data. It has attracted increasing attention in recent years thanks to the popularity of neural network models and the availability of large-scale datasets. A common feature across existing methods is their treatment of the input as a string, i.e., by employing linearization techniques that do not always preserve information in the table, are verbose, and lack space efficiency. We propose to rethink data-to-text generation as a visual recognition task, removing the need for rendering the input in a string format. We present PixT3, a multimodal table-to-text model that overcomes the challenges of linearization and input size limitations encountered by existing models. PixT3 is trained with a new self-supervised learning objective to reinforce table structure awareness and is applicable to open-ended *and* controlled generation settings. Experiments on the ToTTo (Parikh et al., 2020a) and Logic2Text (Chen et al., 2020c) benchmarks show that PixT3 is competitive and, in some settings, superior to generators that operate solely on text.[1]

1 Introduction

Generating text from structured inputs such as tables, tuples, or graphs, is commonly referred to as data-to-text generation (Reiter and Dale, 1997; Covington, 2001; Gatt and Krahmer, 2018). This umbrella term includes several tasks ranging from generating sport summaries based on boxscore statistics (Wiseman et al., 2017), to producing fun facts from superlative Wikipedia tables (Korn et al., 2019), and creating textual descriptions given biographical data (Lebret et al., 2016). From a modeling perspective, data-to-text generation is challenging as it is not immediately obvious how to best describe the given input. For instance, the table in Figure 1 can be verbalized in different ways, depending on the specific content we choose to focus on. In *controlled* data-to-text generation (Parikh et al., 2020a), models are expected to generate descriptions for pre-selected parts of the input (see the *highlighted* cells in Figure 1).

Regardless of the generation setting, numerous approaches have emerged in recent years with different characteristics. A few exploit the structural information of the input (Puduppully et al., 2019; Chen et al., 2020b; Wang et al., 2022), use neural templates (Wiseman et al., 2018), or resort to content planning (Su et al., 2021; Puduppully et al., 2022). While others (Chen et al., 2020a,c; Aghajanyan et al., 2022; Kasner and Dusek, 2022) improve on fluency and generalization by leveraging large-scale pre-trained language models (Devlin et al., 2019; Raffel et al., 2020). A common feature across these methods is their treatment of tabular input as a string, following various linearization methods. As an example, Figure 1 shows the representation of tabular data (top) as a sequence of (Column, Row, Value) tuples (bottom).

Problematically, representing tabular information as a linear sequence results in a verbose representation that often exceeds the context window limit of popular Transformer models (Vaswani et al., 2017). The challenge of processing such long sequences has fostered the development of even more controlled methods which refrain from encoding the table as a whole, concentrating exclusively on highlighted content (e.g., *only* the yellow cells in Figure 1). Unfortunately, models trained on abridged input have difficulty generalizing to new domains while being practically ineffective in scenarios where content selection is not provided.

In this paper we propose to rethink data-to-text generation as a visual recognition task, allowing us to represent and preserve tabular information compactly. Vision Transformers (ViTs; Dosovitskiy et al. 2021) have significantly advanced

[1]Our code, models, and data are available at https://github.com/alonsoapp/PixT3.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6721–6736

Table Title: Shuttle America
Section Title: Fleet

Aircraft	Total	Orders	Passengers				Operated for	Notes
			F	Y+	Y			
Embraer E170	5	-	6	16	48	70	United Express	transferred to Republic Airline
	14	-	9	12		69	Delta Connection Delta Shuttle	2 planes on wet lease from Republic Airline
Embraer E175	15	-	12	12	52	76		
Total	**35**	-						

Linearized Table: <page_title> Shuttle America <page_title> <section_title> Fleet <section_title> <table> <row> <cell> Aircraft <cell> <cell> Total <row_header> Aircraft <row_header> <cell> <cell> Orders <row_header> Aircraft <row_header> <row_header> Total <row_header> <cell> <cell> Passengers <row_header> Aircraft <row_header> <row_header> Total <row_header> <row_header> Orders <row_header> <cell> <cell> Operated For <row_header> Aircraft <row_header> <row_header> Total <row_header> <row_header> Orders <row_header> <row_header> Passengers <row_header> <cell> <cell> Notes <row_header> Aircraft <row_header> . . . . . .

Target Description: Shuttle America operated the E-170 and the larger E-175 aircraft for Delta Air Lines.

Figure 1: Example of table-to-text generation taken from the ToTTo dataset (Parikh et al., 2020a). In the controlled setting, a natural language description is generated only for highlighted (yellow) cells. The table is linearized by encoding each value as a (Column, Row, Value) tuple. We only show the first row, for the sake of brevity.

the field of visual language understanding (Kim et al., 2022; Davis et al., 2022) demonstrating proficiency in various tasks, including language modeling (Rust et al., 2023), visual document understanding (Huang et al., 2022), and visual question answering (Masry et al., 2022). Our work builds on Pix2Struct (Lee et al., 2023), a pretrained image-to-text model which can be fine-tuned for visually-situated language tasks. We recast data-to-text generation as an image-to-text problem and present PixT3, a **Pix**el-based **T**able-**t**o-**T**ext model, which is generally applicable to open-ended and controlled generation settings, overcoming the challenges of linearization and input size limitations encountered by existing models.

Our contributions can be summarized as follows: (a) we introduce the first pixel-based model for table-to-text generation and showcase its robustness across generation settings with varying table sizes; (b) we propose a new training curriculum and self-supervised learning objective to reinforce table structure awareness; (c) automatic and human evaluation results on the ToTTo benchmark (Parikh et al., 2020b) show that PixT3 excels in open-ended generation, leading to improved faithfulness and generation quality, while being competitive with existing methods in controlled scenarios; and (d) we present a new dataset based on Logic2Text (Chen et al., 2020c), which allows us to evaluate generalization capabilities of current table-to-text models.

2 Related Work

The bulk of previous work treats tables as textual objects. Several techniques have been developed to extract accurate information from them (Puduppully et al., 2019; Chen et al., 2020b) using templates (Wiseman et al., 2018), enforcing table structure awareness (Mahapatra and Garain, 2021; Wang et al., 2022), applying contrastive learning (An et al., 2022; Chen et al., 2023b) or focusing on content planning (Su et al., 2021; Puduppully et al., 2022). Other techniques (Chen et al., 2020a,c; Aghajanyan et al., 2022; Kasner and Dusek, 2022) improve fluency and generalization by leveraging large-scale pretrained language models (Devlin et al., 2019; Raffel et al., 2020). Tables are generally linearized, even when special-purpose techniques are developed for encoding table structure (Wang et al., 2022). Dedicated table understanding techniques (Wang et al., 2021; Jin et al., 2023) eschew linearization but have not been integrated with generation tasks.

Previous attempts to address table-to-text generation from a visual recognition perspective (Dash et al., 2023; Srihari et al., 2003) have relied on OCR methods which first extract text from the image and then feed it as a string to a generation model. Aside from being noisy, these techniques typically embrace a text-centric point of view, treating the image as a limitation rather than an informative modality. Our work builds on recent visual language understanding models (Kim et al., 2022; Davis et al., 2022; Lee et al., 2023) which are based exclusively on pixels and have managed to outperform OCR methods in several natural language processing tasks (Rust et al., 2023; Huang et al., 2022; Masry et al., 2022; Salesky et al., 2023).

The field of Vision Language Models (VLMs)

has also experienced significant growth in recent years (Liu et al., 2023; Ye et al., 2023b; Bai et al., 2023; Wang et al., 2023; Alayrac et al., 2022). While most of them focus primarily on natural images, a few are starting to explore the application of dual encoder architectures to visually represented language (Ye et al., 2023a; Zhang et al., 2023). However, these architectures are not parameter lean (with increased model size of a factor of 40 or more compared to Pix2Struct), and some continue to rely on fixed resolution images which can be particularly problematic when processing tabular data.

A few other efforts have recently explored multimodal approaches to processing tables for various tasks, including table-to-text generation. Dash et al. (2023) convert images into HTML tokens which are subsequently linearized and processed by a traditional text-to-text model. Other work (Chen et al., 2023a) focuses on recognizing the structure of tables from images as an independent task. It also leverages multimodal pretraining and unsupervised table structure learning objectives, but ignores the content of table cells and their relations. To the best of our knowledge, our work is the first to conceptualize data-to-text generation as a visually-situated language understanding problem.

3 Problem Formulation

The task of table-to-text generation aims to take a structured table $\boldsymbol{t}$ as input and output a natural language description $\boldsymbol{y} = [y_1, \ldots, y_k]$ where k is the length of the description. Table $\boldsymbol{t}$ is typically reformatted as a sequence of textual records $\boldsymbol{t} = [t_{1,1}, t_{1,2}, \ldots, t_{i,j}, \ldots, t_{m,n}]$ where m and n respectively denote the number of rows and columns of $\boldsymbol{t}$.

We approach this task from a visual recognition perspective, and expect the input table to be an image $\boldsymbol{x}$. The image is reshaped into a sequence of patches analogous to linguistic tokens. More formally, for an input image $\boldsymbol{x} \in R^{H \times W \times C}$ and patch size p, we create N image patches denoted as $x_p \in R^{N \times (P^2 \cdot C)}$. (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ the resulting number of patches, which serves effectively as the input sequence length. Our proposed model learns to autoregressively estimate the conditional probability of a text sequence from a source image as:

$$P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta}) = \prod_{i=1}^{n} P(y_i|\boldsymbol{y}_{<i}, \boldsymbol{x}; \boldsymbol{\theta}) \quad (1)$$

where $\boldsymbol{\theta}$ are transformer parameters and $\boldsymbol{y}_{<i}$ the words decoded thus far.

We further define three generation settings, which manipulate the information provided to the model in terms of content selection (see Appendix B for visualization). In the *tightly-controlled* setting (TControl), the model is given highlighted cells only, ignoring the table. Most recent approaches benchmark model performance in this setting (Wang et al., 2022; An et al., 2022; Chen et al., 2023b; Su et al., 2021; Kale and Rastogi, 2020). In the *loosely controlled* setting (LControl), the model is given highlighted cells *and* the entire table. This is the original setting for which the ToTTo dataset (Parikh et al., 2020a) was constructed. Finally, we introduce the *open-ended* setting (OpenE), where the model is given the table without any highlighting.

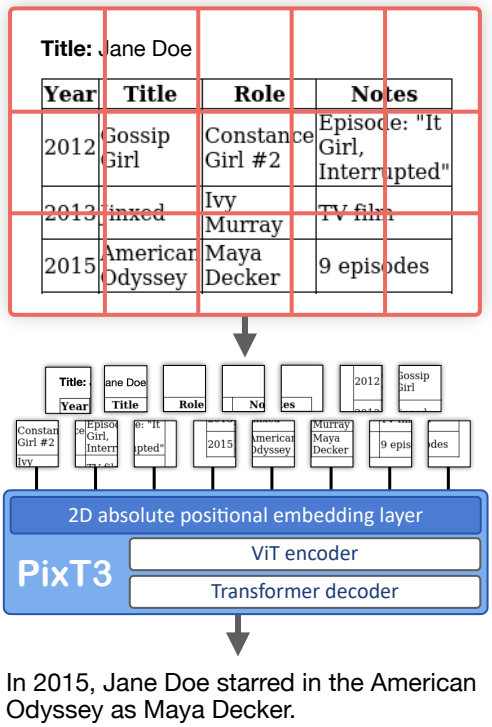


Figure 2: Overview of PixT3 generation model.

4 The PixT3 Model

PixT3 is an image-encoder-text-decoder model based on Pix2Struct (Lee et al., 2023). It expects

image rendered tables and generates descriptions thereof (see Figure 2). Pix2Struct is a Vision Transformer model pretrained on 80M screenshots of web pages extracted from URLs in the C4 corpus (Raffel et al., 2020). It splits input images into patches of 16×16 pixels (see Figure 2), linearly embeds each patch, adds position embeddings, and feeds the resulting sequence of vectors to a standard Transformer encoder (Vaswani et al., 2017).

Pix2Struct was first warmed up with a reading curriculum (Rust et al., 2023; Davis et al., 2022), to improve training stability and fine-tuning performance and then pretrained with a screenshot parsing objective; specifically, it generates a simplified version of an HTML subtree that represents a highlighted area of a web page screenshot. It also adds a BART-like (Lewis et al., 2020) learning signal to pretraining by masking 50% of the text in the input and then requiring the model to produce the entire subtree. Importantly for our table-to-text generation task, Pix2Struct supports variable image resolution and multiple aspect ratios. It first re-scales the input (up or down) to extract the maximal number of fixed-size patches that fit within a given sequence length and then replaces the typical 1-dimensional absolute positional embedding with a 2-dimensional one, which adds resolution flexibility and removes any aspect ratio distortion.

We initialize PixT3's model weights with Pix2Struct; we next adopt a curriculum training strategy which instills in our model knowledge about tables and their structure (see Section 4.2); and finally, we fine-tune on table-to-text generation datasets such as ToTTo (Parikh et al., 2020a) with a task-specific supervised objective.

4.1 Table-to-Image Rendering

We parse tables to HTML, and subsequently render them into images. We also render table metadata (e.g., Wikipedia page and section title), if it exists, as part of the image, adding it on top of the table. Tables are rendered into three different images corresponding to the generation settings defined in Section 3 (see Appendix B, Figure 6).

Although Pix2Struct can handle variable resolutions and input patches, very long inputs are nevertheless computationally expensive. Following Lee et al. (2023), we set the maximum input length to 2,048 patches (of 16×16 pixels) which corresponds to a maximum image size of 524,288 pixels. 41.74% of the tables in a dataset like ToTTo (Parikh et al., 2020a) exceed this size (see Figure 5 in Appendix A), with 5% being larger than 8.3M pixels (32,768 patches). Indiscriminately down-scaling *all* images exceeding the maximum input length would negatively affect performance, especially for very big tables, effectively rendering them unreadable (we showcase how image size affects model performance in Figure 4). To avoid this as much as possible, we truncate the image to fit within a maximum down-scaling factor γ. In other words, images are first compressed to $\gamma\%$ of their original size and then truncated from left to right until they fit into 2,048 patches. The optimal value for γ is determined empirically (see Appendix C).

4.2 Structure Learning Curriculum

Pix2Struct is a general-purpose visual language understanding model, and as such it is not particularly knowledgeable about tables and their structure. Tables can be presented in a variety of ways visually, such as spanning multiple columns or rows, with or without horizontal and vertical lines, non-standard spacing and alignment, and text formatting. Aside from presentation, there are various conventions about the underlying semantics of tables and their structure, e.g., each cell is only related to cells in the same column and row. These challenges have led to the development of dedicated table understanding techniques (Jin et al., 2023; Wang et al., 2022) in the domain of text but cannot be readily ported to images.

Instead, we encourage PixT3 to adhere to tabular conventions, by first training it on an intermediate supporting task. This acts as a structure learning curriculum, exposing the model to the rules governing tables. We next elaborate on the intermediate task, its corresponding dataset, and the proposed self-supervised objective.

Dataset for Intermediate Training Existing datasets like ICDAR2021 (Kayal et al., 2021) and TableBank (Li et al., 2019) are representative of the task of parsing table images into their structure and, in theory, could be used for our intermediate training purposes. However, they focus on scientific tables which do not follow the typical distribution of Wikipedia tables found in ToTTo (Parikh et al., 2020a), e.g., in terms of size and cells spanning across rows and columns. We instead propose to create a synthetic image-to-text dataset, making use of the table rendering pipeline described in Section 4.1. Although we generate tables specifically tailored for our use-case, the generation process is

Table:

oY	io	HG	eG2S
Z4ikU	O1	aRU	mubk6
URa	dAF		I
I86	GAe	Ob	sUr5
L1	3	Vf1	Svaq2

Target:

<<<dAF>><<<URa><I>>><<<io><O1><GAe><3>><<<HG><aRU><Ob><Vf1>>>>>

Figure 3: Synthetically generated table with a highlighted cell and corresponding pseudo-HTML target sequence (for self-supervised objective). Cells within the target sequence are highlighted in the table with a colored background. For details on the structure of the target, please refer to Appendix D.

flexible and can be adapted to other domains with different characteristics.

We determine the structure of each table (size, column, and row spans) randomly, following ToTTo's training set distribution. We cap the generation process at a maximum of 20 columns and 75 rows. Table cells are filled with synthetic values consisting of a random combination of one to five random English alphabet characters and digits, functioning as identifiers rather than meaningful values (see Figure 3 for an example). Our dataset contains 135,400 synthetic tables, 120,000 for training, 7,700 for validation, and 7,700 for testing.

Self-supervised Objective While masking is a widely adopted learning objective (Devlin et al., 2019), it does not naturally transfer to our table-to-text generation task; table values are not naturally correlated to neighboring values and thus a masked cell cannot be easily predicted from other cells in its context. Table values could be rearranged so that they correlate to their neighbors, however, early experiments showed that this type of objective does not improve downstream task performance (see Appendix D for details). Another common pretraining objective is table linearization (Chen et al., 2023a), which, however, scales poorly with table size, leading to slow pretraining.

We propose a self-supervised objective that encourages PixT3 to capture the relations between cells within a table while generating a small amount of tokens. Specifically, we highlight a random cell in a synthetically generated table, and train the model to produce a sorted list of cells within the same column and row (see Figure 3). Our objective encapsulates a loose notion of table structure, nudging the model to pay attention to the arrangement of columns and rows around a cell. We follow the same pseudo HTML notation introduced in Pix2Struct to format our output sequence, easing the model's transition from its original screenshot parsing objective to this new one. Note that we consider tables with a heterogeneous structure where cells can span across multiple columns and rows. In such cases, the expected sequence will contain all cells in related rows and columns surrounding the highlighted cell (see Figure 3).

4.3 PixT3 Fine-tuning

The intermediately pre-trained PixT3 is subsequently fine-tuned on an image-rendered dataset (see Section 4.1). In experiments, we use ToTTo (Parikh et al., 2020b), however, our approach is not tied to a particular style of tables. Due to our model's requirement for unimodal input, we treat table-related information (such as its title) as part of the table itself and render them both as one image (see Lee et al. 2023 for a similar approach).

5 Experimental Setup

Model Configuration All our experiments were conducted with the *base* pretrained Pix2Struct[2] model (282M parameters). We trained PixT3 variants for the three table-to-text generation settings defined in Section 3. All PixT3 models were fine-tuned on ToTTo (Parikh et al., 2020a) with tables rendered as images following the procedure outlined in Section 4.1. The maximum down-scaling factor γ was set to 0.39.

PixT3 models were fine-tuned with a batch size of 8 and a gradient accumulation of 32 steps on a single NVIDIA A100 80GB GPU. Checkpoints were selected according to best performance on the validation set. All models used an input sequence length of 2,048 patches and were optimized with AdamW (Loshchilov and Hutter, 2017). We used a learning rate scheduler with a linear warmup of 1,000 steps to 0.0001, followed by cosine decay to 0. The decoder maximum sequence length was set to 50 tokens, which covers 97.49% of the target descriptions in the training data. PixT3 was trained for 1.4k steps with the self-supervised objective described in Section 4.2. Our decoder was not frozen during intermediate training, as initial

[2] https://github.com/google-research/pix2struct

experiments showed that a fully trained model outperformed one with frozen decoder weights. A full list of fine-tuning hyper-parameters can be found in Appendix H.

Datasets We evaluated our model on ToTTo (Parikh et al., 2020a), a large-scale, manually curated dataset representative of several domains and types of tables. We also assessed the generalization capabilities of PixT3 on out-of-distribution tables. We created an out-of-domain benchmark with content selection annotations similar to ToTTo based on Logic2Text (Chen et al., 2020c), an existing dataset which contains a total of 10,161 Wikipedia tables, paired with human-authored descriptions and logical forms. Logic2Text differs from ToTTo in that descriptions are not simple verbalisations of table rows and columns, but require some form of reasoning (e.g., comparisons or counting operations). We were able to automatically trace values mentioned in the logical form back to the cells of the input tables (Alonso and Agirre, 2023), thus obtaining highlighted cell annotations similar ToTTo's (see Appendix E for an example). We report results on the official test set (1,085 examples).

Model Comparison We evaluated PixT3 against several text-only models with similar parameter sizes. These include CoNT (An et al., 2022), the top performant (published) model in the ToTTo leaderboard.[3] CoNT is a text-to-text generation model which makes use of contrastive learning, through improved selection of contrastive examples, a new contrastive loss, and a global decoding strategy. CoNT expects the input table to be converted to a string, and is built on top of T5-base (220M parameters). We also compared against Lattice (Wang et al., 2022), a model which enforces awareness of table layout though pruning the attention flow and encoding cells in a way that is invariant to their relative position in a sequence. This model also uses T5-base and expects linearized input. In addition, we report results with vanilla T5-base which performed competitively on the ToTTo leaderboard without any task specific modifications (Kale and Rastogi, 2020; An et al., 2022). All comparison models and PixT3, were trained on the ToTTo training set in our three generation settings.[4]

		Dev		TestN		TestO		Test	
	Model	BL	PR	BL	PR	BL	PR	BL	PR
TControl	T5-base	47.7	57.1	38.9	51.2	55.4	61.1	47.2	56.2
	T5-3B	48.4	57.8	39.3	51.6	55.1	60.7	47.2	56.2
	Lattice	48.0	58.4	40.0	**53.8**	55.9	62.4	48.0	**58.1**
	CoNT	**49.0**	**58.6**	**40.6**	53.7	**56.7**	**62.5**	**48.7**	**58.1**
	PixT3	45.7	55.7	37.5	50.6	53.2	60.4	45.4	55.5
LControl	T5-base	24.5	27.2	19.4	23.9	29.4	30.3	24.5	27.1
	T5-3B	23.6	26.0	18.0	22.4	28.7	29.2	23.4	25.8
	Lattice	24.9	31.0	20.8	27.7	27.5	33.8	24.4	30.8
	CoNT	23.8	29.3	19.2	26.1	28.7	32.3	23.9	29.2
	PixT3	**46.2**	**55.1**	**38.1**	**50.3**	**52.7**	**59.0**	**45.4**	**54.7**
OpenE	T5-base	21.5	23.5	16.8	21.0	26.5	26.5	21.7	23.8
	T5-3B	20.8	22.9	16.7	20.3	25.5	25.5	21.2	22.9
	Lattice	20.9	26.1	17.6	24.3	23.7	27.6	20.8	25.9
	CoNT	21.7	25.8	16.9	23.2	26.3	28.3	21.6	25.8
	PixT3	**24.8**	**28.3**	**20.5**	**26.3**	**28.9**	**30.3**	**24.7**	**28.3**

Table 1: Automatic evaluation results on ToTTo in three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE). We report BLEU (BL) and PARENT (PR) results on the development (Dev) and Test sets, including the overlapping (TestO) and non-overlapping (TestN) test set splits. BLEURT results are in Appendix E.

For our out-of-domain experiments, we also compare against LLaVA-1.5 (Liu et al., 2023), a large pretrained multimodal model (13B parameters) which is built on top of the CLIP visual encoder (Radford et al., 2021) and the Vicuna-7B language model (Zheng et al., 2023), and fine-tuned on vision-language instructions. LLaVA has not been fine-tuned specifically for table-to-text generation, however, it is interesting to see if sufficiently large scale is all it takes to do well on the table-to-text generation task. LLaVA can only handle a *single* image at each forward pass. This limitation prevents it from performing inference in an in-context learning setting, where the model has access to multiple input-output examples at the same time. To approximate in-context learning as closely as possible, we provided LLaVA with an image, an instruction, and three table descriptions as output examples for each generation setting (see Appendix F for details). We summarize the number of parameters for all comparison models in Table 2.

we do still provide a few description examples in our prompt to ensure a fair zero-shot comparison. All prompts used for LLaVA in this evaluation can be found in Appendix F.

[3] A model named SKY appears to slightly outperform CoNT in the leaderboard, however, at the time of writing, we were not able to verify this, i.e., by finding a publication or preprint describing this model.

[4] Comparison models were trained with the authors' publicly available scripts.

6 Results

PixT3 is the best performing model in loosely controlled and open-ended generation settings. Table 1 summarizes our results on ToTTo in our three generation settings. We evaluated model performance automatically with the same metrics used to rank participant systems in the ToTTo leaderboard. These include BLEU (Papineni et al., 2002) which is as a proxy for fluency, PARENT (Dhingra et al., 2019), a metric proposed specifically for data-to-text evaluation that takes the table into account, serving as a proxy of faithfulness, and BLEURT (Sellam et al., 2020); the latter is a composite metric that takes a reference and model output as input, and returns a score that indicates the extent to which the output is fluent and conveys the meaning of the reference. Note that ToTTo features two splits in the development/test set containing tables whose header values are present (overlapping split) and absent (non-overlapping split) in the training set. Results on the test set, which is not publicly available, were obtained via submitting to the ToTTo leaderboard.

We first discuss our results on the tightly controlled generation setting (TControl) where models are not given the full table, just the highlighted cells. We would not expect PixT3 to excel at this setting, which is better suited to text-to-text models (highlighted cells make for non-descriptive images, see Appendix B, Figure 6). PixT3 is indeed unable to outperform CoNT, Lattice, and related T5 variants, falling 3.5 BLEU points behind on the development set and 3.7 on the test set. However, LControl, the loosely controlled generation setting, better showcases the advantages of PixT3, which in this case demonstrates almost a two times improvement over CoNT and T5 models. Performance degrades drastically for all systems in the open-ended setting (OpenE) which is challenging; models are expected to perform content selection in addition to text generation, and could produce table descriptions which are valid but different from the reference. Automatic metrics based on n-gram overlap are particularly punitive in this case. Nevertheless, PixT3 is superior to CoNT, Lattice, and T5 across evaluation metrics.

PixT3 generalizes to out-of-domain tables which require reasoning skills. We next evaluate whether PixT3 generalizes to unseen tables, outside ToTTo's distribution. Table 2 shows our results on Logic2Text (Chen et al., 2020c), again following the three generation settings. Compared to ToTTo, Logic2Text is a more challenging dataset as most descriptions rely on reasoning over the entire table. This results in poor model performance in the TControl setting which does not include the table as input. Nonetheless, we observe that PixT3 excels at the LControl setting, even though it has to process and reason over the entire table. The OpenE setting is challenging for all models as they are asked to identify interesting cells to talk about in *out-of-domain* tables. PixT3 still maintains an edge over T5 and LLaVA, performing on par with CoNT and Lattice. We observe that LLaVA cannot match the performance of PixT3 and T5-based models. This underscores the importance of task-specific fine-tuning over parameter size. We present output examples in Appendix E.

	Model	Size	BLEU	PARENT
TControl	LLaVA	13B	12.6	34.36
	T5-base	220M	16.8	55.97
	T5-3B	3B	17.7	52.75
	Lattice	220M	19.8	61.05
	CoNT	220M	18.8	61.73
	PixT3	282M	**20.6**	**61.86**
LControl	LLaVA	13B	5.9	23.18
	T5-base	220M	11.5	40.02
	T5-3B	3B	10.9	35.45
	Lattice	220M	11.5	40.02
	CoNT	220M	11.8	43.25
	PixT3	282M	**21.5**	**56.45**
OpenE	LLaVA	13B	6.7	20.14
	T5-base	220M	7.9	30.67
	T5-3B	3B	9.5	29.47
	Lattice	220M	**11.7**	**38.12**
	CoNT	220M	11.0	36.94
	PixT3	282M	11.4	35.68

Table 2: Automatic evaluation results on Logic2Text in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE). All models (except LLAVA) were fine-tuned on ToTTo and tested on Logic2Text. BLEURT results are in Appendix E.

PixT3 is robust against table input size. In Figure 4, we analyze the effect of table size on model performance. As can be seen, T5, Lattice, and CoNT are severely affected: the bigger the table, the less accurate the generated description. PixT3 is evidently more robust, showing degradation in performance only for very big tables. We also examined whether PixT3 has an edge because of its ability to encode longer inputs. Recall that CoNT, Lattice, and T5-base utilize a fixed input length

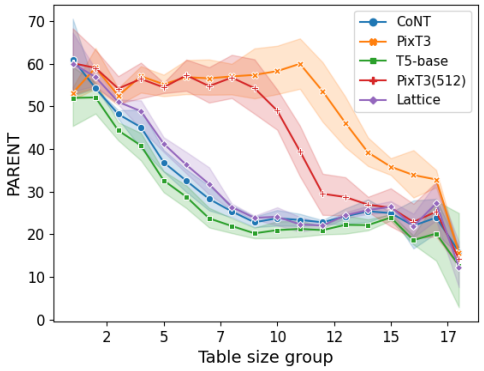


Figure 4: Model performance (CoNT, T5, PixT3, Lattice, and PixT3 with 512 patch input size) in the LControl setting across 18 table size groups (logarithmic scale). Upper and lower bounds in shaded areas correspond to results for the overlapping and non-overlapping ToTTo splits, while central points correspond to results overall. We report results with PARENT, other metrics show similar tendencies. We refer to Appendix A for further details.

of 512 tokens, while PixT3 uses 2,048 patches. We thus trained a PixT3 variant with input length set to 512 patches. As shown in Figure 4, the more constrained PixT3 model is slightly worse and more likely to degrade with increased table size but consistently outperforms CoNT, Lattice, and T5.

The structure learning curriculum improves generation quality across metrics. In Table 3 we perform an ablation study comparing PixT3 with and without our structure learning curriculum and self-supervised objective (Section 4.2). For both models we follow the same fine-tuning process: we render tables into images, identify the optimal point of image compression and truncation (see Section 4.1), and perform hyper-parameter search to optimize Pix2Struct-base for our task. Vanilla PixT3 (second row in Table 3) shows a substantial improvement over an out-of-the-box Pix2Struct model which achieves a BLEU score of 0.2 and PARENT score of 0.6 on the ToTTo development set. Adding the intermediate training curriculum (second row in Table 3) slightly improves vanilla PixT3 across evaluation metrics.

Manual inspection of the descriptions produced by the two PixT3 model variants reveals they are often semantically equivalent to the target (43% of the time). Nevertheless, the intermediate training curriculum substantially reduces structure-based faithfulness errors, especially in the OpenE setting. On a sample of 200 outputs (randomly selected

	Dev			Test		
Models	BL	PR	BRT	BL	PR	BRT
Pix2Struct	0.2	0.6	−1.433	—	—	—
PixT3 (W/o SLC)	38.7	46.0	−0.003	38.3	45.6	0.001
PixT3 (With SLC)	**39.2**	**46.5**	**0.008**	**38.7**	**46.3**	**0.007**

Table 3: PixT3 with and without structure learning curriculum (SLC); we report results on the ToTTo development (Dev) and Test set with BLEU (BL), PARENT (PR), and BLEURT (BRT), averaged across the three generation settings.

from the development set), we found that 23% of the descriptions produced by vanilla PixT3 disregard or misinterpret the structure of the table. Structural faithfulness errors reduce to 7% when PixT3 is trained with our structure learning curriculum.

PixT3 is most faithful in loosely controlled and open-ended generation settings. We further conducted a human evaluation study to quantify the extent to which the generated descriptions are faithful to the table. We evaluated PixT3, and the two best performing text-only systems (CoNT, and Lattice) on two sets of 100 randomly selected table-description pairs from ToTTo (development set) and Logic2Text (test set), in the three generation settings. Crowdworkers were presented with an uncompressed image of a table, its page and section title, and a model generated description. As an upper bound, we also elicited judgments for the human curated reference descriptions for the same ToTTo and Logic2Text examples. Participants were asked to determine whether a description was "True" or "False" based on the information provided in the table and/or its title and subtitle (see instructions in Appendix G). Overall we elicited 7,200 judgments (100 examples $\times$ 3 generation settings $\times$ 4 model descriptions $\times$ 3 participants $\times$ 2 datasets). Crowdworkers were recruited using the online platform Prolific.[5]

Table 4 shows the results of the human evaluation, specifically the proportion of descriptions deemed faithful. As expected, the human authored Reference description is consistently faithful across generation settings. CoNT is more faithful in TControl but deteriorates in the LControl and OpenE settings. We further examined whether differences among systems are statistically significant using paired bootstrap resampling. PixT3 is significantly worse ($p < 0.05$) than the Reference in TControl

[5] https://www.prolific.com

	Model	TControl	LControl	OpenE
ToTTo	Reference	87	84	89
	Lattice	**79**	16	20
	CoNT	76	16	35
	PixT3	69	**72**	**78**
L2T	Reference	81	87	86
	Lattice	34	3	16
	CoNT	**35**	3	26
	PixT3	32	**40**	**60**

Table 4: Human evaluation results on ToTTo and Logic2Text (L2T). Proportion of descriptions rated as faithful for PixT3, CoNT, and Reference in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE).

but not CoNT or Lattice. In LControl all differences between systems are statistically significant ($p < 0.05$). In OpenE, PixT3 is significantly different ($p < 0.05$) from CoNT and Lattice but not from the Reference. Inter-rater agreement was moderate with a Fleiss' Kappa coefficient of 0.55 (Fleiss, 1971).

7 Conclusion

In this paper, we leverage the capabilities of Vision Transformers to recast table-to-text generation as a visual recognition task, removing the need for rendering the input in a string format. Our model, PixT3, introduces a new training curriculum and self-supervised learning objective in order to capture the structure and semantics of tables. Experiments across constrained and open-ended generation settings show it is robust to different table sizes, performing competitively and often better than state-of-the-art models. PixT3 is also able to handle new domains with unseen tables, as evidenced by our results on Logic2Text, a new dataset which we propose for assessing the generalization capabilities of table-to-text generation models.

Avenues for future research are many and varied. There are several downstream tasks which stand to benefit from a pixel-based view of textual information, including multilingual table-to-text generation, and semantic parsing. We would also like to investigate additional objectives and inductive biases that can better capture the structure of tables and inter-cell dependencies.

8 Limitations

While PixT3 shows promising results, its performance is affected by the dimension of the input tables (for instance, 16% of the Wikipedia tables in ToTTo remain too big for PixT3 to represent effectively). It would be interesting to look into alternative ways of preprocessing very large tables, e.g., by rendering them via multiple images. While our proposed intermediate training methodology mitigates faithfulness errors, the model still struggles with hallucinations, falling short of human-level performance.

Finally, PixT3, as well as other comparison systems, have limited reasoning capabilities, e.g., they cannot infer information which is not explicitly stated in the table or make logical connections between concepts. PixT3's superior performance in terms of faithfulness on Logic2Text (see Table 4) is due to generating simpler sentences rather than superior reasoning skills. Thus, aside from new training objectives, a promising direction would be to combine the visual representations with an intermediate planning component that encourages the model to reason about the input while generating the output.

Acknowledgements

We thank the meta-reviewer and anonymous reviewers for their constructive feedback. The authors also thank Ander Salaberria for his insightful comments on earlier versions of this work. We gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1), the Basque Government (Research group funding IT-1805-22), MCIN/AEI/10.13039/501100011033 project AWARE (TED2021-131617B-I00), European Union NextGenerationEU/PRTR, and the LUMINOUS project (HORIZON-CL4-2023-HUMAN-01-21-101135724).

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2022. HTLM: Hyper-text pre-training and prompting of language models. In *International Conference on Learning Representations*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022.

Flamingo: a visual language model for few-shot learning.

Iñigo Alonso and Eneko Agirre. 2023. Automatic logical forms improve fidelity in table-to-text generation. *Expert Systems with Applications*, page 121869.

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 2197–2210. Curran Associates, Inc.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Leiyuan Chen, Chengsong Huang, Xiaoqing Zheng, Jinshu Lin, and Xuanjing Huang. 2023a. TableVLM: Multi-modal pre-training for table structure recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2437–2449, Toronto, Canada. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Xi Chen, Xinjiang Lu, Haoran Xin, Wenjun Peng, Haoyang Duan, Feihu Jiang, Jingbo Zhou, and Hui Xiong. 2023b. A table-to-text framework with heterogeneous multidominance attention and self-evaluated multi-pass deliberation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 607–620, Singapore. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020c. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111, Online. Association for Computational Linguistics.

Michael A Covington. 2001. Building natural language generation systems. *Language*, 77(3):611–612.

Amanda Dash, Melissa Cote, and Alexandra Branzan Albu. 2023. Weathergov+: A table recognition and summarization dataset to bridge the gap between document image analysis and natural language generation. In *Proceedings of the ACM Symposium on Document Engineering 2023*, DocEng '23, New York, NY, USA. Association for Computing Machinery.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. 2022. End-to-end document recognition and understanding with dessurt. In *European Conference on Computer Vision*, pages 280–296. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. 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, Florence, Italy. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA. Association for Computing Machinery.

Rihui Jin, Jianan Wang, Wei Tan, Yongrui Chen, Guilin Qi, and Wang Hao. 2023. TabPrompt: Graph-based pre-training and prompting for few-shot table understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7373–7383, Singapore. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of*

the 13th International Conference on Natural Language Generation, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Zdeněk Kasner and Ondrej Dusek. 2022. Neural pipeline for zero-shot data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3914–3932, Dublin, Ireland. Association for Computational Linguistics.

Pratik Kayal, Mrinal Anand, Harsh Desai, and Mayank Singh. 2021. Icdar 2021 competition on scientific table image recognition to latex. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*, pages 754–766. Springer.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*.

Flip Korn, Xuezhi Wang, You Wu, and Cong Yu. 2019. Automatically generating interesting facts from wikipedia tables. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 349–361, New York, NY, USA. Association for Computing Machinery.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2019. Tablebank: A benchmark dataset for table detection and recognition.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Joy Mahapatra and Utpal Garain. 2021. Exploring structural encoding for data-to-text generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 404–415, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020a. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020b. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of EMNLP*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Ratish Puduppully, Yao Fu, and Mirella Lapata. 2022. Data-to-text generation with variational sequential planning. *Transactions of the Association for Computational Linguistics (to appear)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the

limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023. Multilingual pixel representations for translation and effective cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Sargur N. Srihari, Ajay Shekhawat, and Stephen W. Lam. 2003. *Optical Character Recognition (OCR)*, page 1326–1333. John Wiley and Sons Ltd., GBR.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048, Seattle, United States. Association for Computational Linguistics.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogvlm: Visual expert for pretrained language models.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1780–1790, New York, NY, USA. Association for Computing Machinery.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium. Association for Computational Linguistics.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, Qin Jin, Liang He, Xin Lin, and Fei Huang. 2023a. UReader: Universal OCR-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2841–2858, Singapore. Association for Computational Linguistics.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023b. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023. Llavar: Enhanced visual instruction tuning for text-rich image understanding.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

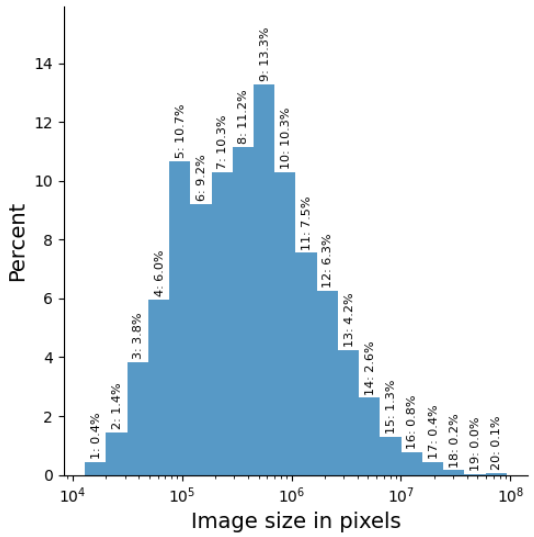


Figure 5: Proportion of ToTTo examples (development set) per table size (shown in logarithmic scale).

A Table Size Distribution in ToTTo

We measure the size of a table by the total amount of pixels in its corresponding rendered image. We then calculate the distribution of each size, and group tables into 20 buckets accordingly. Each bucket covers a logarithmically increasing amount of table sizes. Figure 5 shows the resulting buckets and the proportion of ToTTo examples in each (development set). The quality of descriptions generated within each group, are evaluated in Section 6, see Figure 4.

B Table-to-Text Generation Settings

Figure 6 illustrates how the image input to PixT3 differs according to three generation settings: tightly controlled (the model is given only highlighted cells, no table), loosely controlled (the model is given the table and highlighted cells), and open-ended (the model is given the table without any highlighting).

C Image Truncation and Down-scaling

We explored the impact of down-scaling on model performance and its tradeoff with truncation. We conducted a series of experiments wherein PixT3 models were trained on versions of ToTTo with varying down-scaling factor γ: 0.87, 0.58, 0.39, 0.26, and 0.00. Note that γ=0.00 corresponds to a setting where no truncation takes place, only down-scaling. According to the results shown in Table 5, it is best to combine truncation with down-scaling,

TControl

Title: Huracán (TV series)
Section: International release
Highlights: Canal de las Estrellas // October 13, 1997 // Huracán // Monday to Friday

LControl

Title: Huracán (TV series)
Section: International release

Country	Network(s)	Series premiere	Series finale	Title	Weekly schedule	Timeslot
Mexico	Canal de las Estrellas	October 13, 1997	March 27, 1998	Huracán	Monday to Friday	21:30
United States	Univision	April 13, 1998	June 8, 1998	Huracán	Monday to Friday	14:00

OpenE

Title: Huracán (TV series)
Section: International release

Country	Network(s)	Series premiere	Series finale	Title	Weekly schedule	Timeslot
Mexico	Canal de las Estrellas	October 13, 1997	March 27, 1998	Huracán	Monday to Friday	21:30
United States	Univision	April 13, 1998	June 8, 1998	Huracán	Monday to Friday	14:00

Reference

On October 13, 1997, Canal de las Estrellas started broadcasting Huracán on weekdays.

Figure 6: PixT3 input image examples (and reference) in three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE).

none of the extreme settings (no truncation vs too much truncation) are beneficial. The optimal γ value is 0.39.

D Intermediate Training

Synthetic Dataset Generation In this section we provide a more detailed description regarding the generation of synthetic tables for intermediate training. As our goal was to generate tables with a structure similar to ToTTo, we first measured the probability distribution of columns, rows, column spans and row spans for the tables in the training set to avoid over-fitting and contamination. We observed that the distribution of columns (up to 20 columns) remained almost constant across tables, and did not affect the probability distribution of rows. As a result, we aggregated row numbers across columns and computed a single distribution for rows to simplify our generation task, using discrete probability distributions. In order to limit the size of the generated tables we cap the number of columns and rows to 20 and 75, respectively. For

Epoch \ γ	0.00	0.26	0.39	0.57	0.87
16	28.71	29.13	29.47	**29.58**	27.47
17	28.99	29.53	**29.99**	29.70	27.69
18	29.67	30.04	**30.55**	30.21	28.13
19	29.98	30.04	**30.63**	30.54	28.33
20	29.83	30.21	**30.68**	30.53	29.39

Table 5: Evaluation results (BLUE) for PixT3 model in tightly controlled generation setting for different γ down-scaling factors. We show the Last five epochs on the ToTTo training set.

the synthetic text within the cells, we randomly generated digits in the [1–5] range and character sequences from [A–Z, a–z] which gave us a total of 776,520,240 permutations of possible unique cell values.

Overall, we generated 120K tables accompanied with target pseudo HTML descriptions. The latter were on average 121 tokens long, with the longest sequences containing 877 tokens. In experiments, we observed that text size affects mainly the average count of tokens, whereas the number of table columns and rows influences the length of the target sequences. The sequences follow a hierarchical structure defined by the characters < and >. In the first hierarchical level, one container can be found for each highlighted cell in the table. Each container includes, in the following order, the highlighted cell, the cells in all related columns, and all cells in all related rows. This structure can represent multiple related columns and rows per highlighted cell, as well as multiple highlighted cells per table.

Alternative Objectives We conducted a set of experiments to identify the best self-supervised objective for our structure learning curriculum. In addition to the objective presented in Section 4.2, we also experimented with a masking objective. Specifically, given a randomly generated table, we filled each cell with text indicative of its position in the table. We then masked random cells and the model was trained to predict the missing cell values (see Figure 7 for an example). We empirically observed that this objective led to worse performance compared to PixT3, even though it resulted in relatively fast training, since the table can be converted into a sequence with a small number of tokens. We hypothesize that this objective only weakly enforces table structure learning as the model does not need to pay attention to all the cells in a column and row to guess the missing value but simply rely on its closest neighbors. We also experimented with a combination of the masking objective discussed here and the structure learning objective described in Section 4.2. However, this model still lagged behind PixT3.

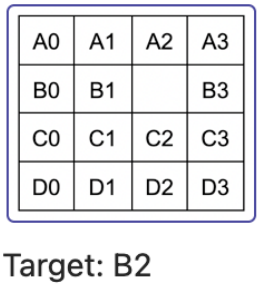


Figure 7: Synthetically generated table with masked cell. Filled cell values denote position in the table.

		Dev Set (All)	Test Set (Non)	Test Set (Over)	Test Set (All)
	Model	BLEURT	BLEURT	BLEURT	BLEURT
TControl	T5-base	0.233	0.106	0.354	0.230
	T5-3B	0.228	0.104	0.344	0.224
	Lattice	0.226	0.103	0.348	0.226
	CoNT	**0.240**	0.116	0.364	0.240
	PixT3	0.178	0.044	0.312	0.178
LControl	T5-base	−0.298	−0.395	−0.191	-0.293
	T5-3B	−0.309	−0.416	−0.194	-0.305
	Lattice	−0.287	−0.382	−0.195	-0.288
	CoNT	−0.293	−0.387	−0.190	-0.289
	PixT3	**0.169**	**0.047**	**0.287**	**0.167**
OpenE	T5-base	−0.371	−0.458	−0.278	-0.368
	T5-3B	−0.385	−0.456	−0.301	-0.378
	Lattice	−0.377	−0.451	−0.302	-0.377
	CoNT	−0.370	−0.452	−0.281	-0.366
	PixT3	**−0.332**	**−0.414**	**−0.258**	**−0.336**

Table 6: BLEURT results on ToTTo for T5, PixT3, Lattice, and CoNT in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE). In the TControl setting, T5 results are taken from Kale and Rastogi (2020) and CoNT results from An et al. (2022). This table complements results reported in Table 1.

E Additional Results and Examples

In addition to BLEU and PARENT reported in Tables 1 and 2, we also present results with BLEURT in Table 6 and Table 7. We further show example output on the Logic2Text dataset (zero-shot setting) in Figure 8. In the TControl setting, CoNT struggles to produce a coherent sentence, while PixT3 generates a faithful but not very informative one. This is not surprising as the models receive nothing but the title and highlighted cells, making it extremely difficult to generate the target sentence. In LControl, both models have access to the entire table; however, they still produce a false statement,

	Model	BLEURT
TControl	LLaVA	−1.230
	T5-base	−1.086
	T5-3B	−1.079
	Lattice	**−1.060**
	CoNT	−1.103
	PixT3	−1.104
LControl	LLaVA	−1.189
	T5-base	−1.147
	T5-3B	−1.167
	Lattice	−1.147
	CoNT	−1.159
	PixT3	**−1.073**
OpenE	LLaVA	**−1.184**
	T5-base	−1.237
	T5-3B	−1.196
	Lattice	−1.231
	CoNT	−1.231
	PixT3	−1.213

Table 7: Automatic evaluation results on Logic2Text in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE). All models (except LLAVA) were fine-tuned on ToTTo and tested on the Logic2Text. This table complements results reported in Table 2.

most likely a consequence of the zero-shot nature of our generation task. Finally, in the less constrained OpenE setting, PixT3 generates a coherent and faithful sentence. While CoNT also produces a fluent sentence, it incurs a faithfulness error when mentioning "(+5)" instead of "(-5)". This is likely due to the performance degradation this model experiences when provided with the full table.

F LLaVA promts

As mentioned in Section 5, our zero-shot experiments involved comparisons against LLaVA-1.5 (Liu et al., 2023), a large pretrained multimodal model (13B parameters). We devised the following prompts for each generation setting:

TControl "Here are some descriptions based on other highlights of other tables 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description based on the following highlighted cells extracted form a table."

LControl "Here are some descriptions based on the highlights of other tables not present in the input: 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description based on the highlighted cells in this table following the same style as the example descriptions."

OpenE "Here are some descriptions from other tables not present in the input: 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description stating something from this table following the same style as the example descriptions."

G Human Evaluation Guidelines

We provide the full set of instructions presented to crowdworkers for the human evaluation study. Our participants were native English speakers from the United Kingdom and the United States of America, with a 50/50 equal gender split between male and female.

Thank you for taking part in our experiment! You will be presented with a table and a computer-generated description of its content. Your task is to determine whether each description is "True" or "False" based on the information provided in the table and/or its title and subtitle (you will see examples later-on). No expert knowledge is required to perform this task. You should evaluate the descriptions given the information presented in the table, without taking any other information into account (e.g., based on your own knowledge or the web).

Here are some guidelines to help you with your evaluation:

Acronyms: tables often have acronyms which the descriptions might spell out. For example, if the table mentions "TD" and the description correctly spells it out as "touch down," you should not consider this "False" (although the description might be false for other reasons).

Implicit information: the description might mention information that can be inferred but is not explicitly spelled-out in the table. For example, it could mention "steam engines" when the table lists theirs names without explicitly

Title: 1973 u.s. open (golf)

place	player	country	score	to par
1	gary player	south africa	67 + 70 = 137	- 5
2	jim colbert	united states	70 + 68 = 138	- 4
t3	jack nicklaus	united states	71 + 69 = 140	- 2
t3	johnny miller	united states	71 + 69 = 140	- 2
t3	bob charles	new zealand	71 + 69 = 140	- 2
t6	gene borek	united states	77 + 65 = 142	e
t6	julius boros	united states	73 + 69 = 142	e
t6	tom weiskopf	united states	73 + 69 = 142	e
t6	arnold palmer	united states	71 + 71 = 142	e
t6	lee trevino	united states	70 + 72 = 142	e

- **Reference:** Jim Colbert has the second best number of strokes to par.
- **CoNT (TControl):** Jim Colbert led the 1973 U.S. open (golf course) with a score of to par.
- **PixT3 (TControl):** Jim Colbert took part in the 1973 U.S. open (golf) tournament.
- **CoNT (LControl):** At the 1973 U.S. open (golf), Jim Colbert shot a record of 267 (+1) and finished four strokes ahead of runner-up Lee Janzen.
- **PixT3 (LControl):** Jim Colbert had a score of 142.
- **CoNT (OpenE):** Gary Player scored 137 (+5) and finished five strokes ahead of runner-up Jim Colbert.
- **PixT3 (OpenE):** Gary Player won the 1973 U.S. Open (golf) with a score of 137.

Figure 8: Logic2Text table and model output in three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE).

talking about steam engines. In this case, the description should not be considered "False".

- You should evaluate each description independently.

- If the description does not make sense and is impossible to evaluate (usually when summarizing very large tables), you should consider it as "False".

We suggest starting by reading the description and then referring to the table to verify if it aligns with its claims.

This data elicitation study is performed by researchers at [REDACTED]. If you have any questions, feel free to contact [REDACTED]. Participation in this research is voluntary. You have the right to withdraw from the experiment at any time. The collected data will be used for research purposes only. We will not collect any personal information. Your responses will be linked to your anonymous Prolific ID for the exclusive purpose of conducting our experiment.

H PixT3 Fine-tuning Hyper-parameters

PixT3 models across all three settings (TControl, LControl, OpenE) were fine-tuned using the same hyper-parameters. To prevent over-fitting, we employed early stopping based on the BLEU score computed on the validation set every 250 steps. Table 8 enumerates the specific hyper-parameter values used in PixT3, with all remaining parameters set to the default values defined in Pix2Struct (Lee et al., 2023).

Hyperparameter	Value
Optimizer	AdamW
Learning rate	0.0001
Warm-up steps	1000
Max. input patches	2048
Shuffle train data	False
Epochs	30
Train batch size	8
Gradient accum. steps	32
Mixed precision	fp16
Evaluation batch size	32
Eval freq. steps	250
Inf. beam search	8 beams

Table 8: Hyperparameters used in PixT3.

Artificial Intelligence In Medicine 155 (2024) 102938

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Research Paper

MedExpQA: Multilingual benchmarking of Large Language Models for Medical Question Answering

Iñigo Alonso, Maite Oronoz, Rodrigo Agerri *

HiTZ Center - Ixa, University of the Basque Country UPV/EHU, Spain

ARTICLE INFO

Keywords:
Large Language Models
Medical Question Answering
Multilinguality
Retrieval Augmented Generation
Natural Language Processing

ABSTRACT

Large Language Models (LLMs) have the potential of facilitating the development of Artificial Intelligence technology to assist medical experts for interactive decision support. This potential has been illustrated by the state-of-the-art performance obtained by LLMs in Medical Question Answering, with striking results such as passing marks in licensing medical exams. However, while impressive, the required quality bar for medical applications remains far from being achieved. Currently, LLMs remain challenged by outdated knowledge and by their tendency to generate hallucinated content. Furthermore, most benchmarks to assess medical knowledge lack reference gold explanations which means that it is not possible to evaluate the reasoning of LLMs predictions. Finally, the situation is particularly grim if we consider benchmarking LLMs for languages other than English which remains, as far as we know, a totally neglected topic. In order to address these shortcomings, in this paper we present MedExpQA, the first multilingual benchmark based on medical exams to evaluate LLMs in Medical Question Answering. To the best of our knowledge, MedExpQA includes for the first time reference gold explanations, written by medical doctors, of the correct and incorrect options in the exams. Comprehensive multilingual experimentation using both the gold reference explanations and Retrieval Augmented Generation (RAG) approaches show that performance of LLMs, with best results around 75 accuracy for English, still has large room for improvement, especially for languages other than English, for which accuracy drops 10 points. Therefore, despite using state-of-the-art RAG methods, our results also demonstrate the difficulty of obtaining and integrating readily available medical knowledge that may positively impact results on downstream evaluations for Medical Question Answering. Data, code, and fine-tuned models will be made publicly available.[1]

1. Introduction

We are currently seeing a dramatic increase in research on how to apply Artificial Intelligence (AI) to the medical domain with the aim of generating decision support tools to assist medical experts in their everyday activities. This has been further motivated by rather strong claims about Large Language Models (LLMs) in medical Question Answering (QA) tasks, such as that they obtain passing marks for medical licensing exams like the United States Medical Licensing Examination (USMLE) [1,2].

Assisting medical experts by answering their medical questions is a natural way of articulating human-AI interaction as it is usually considered that Medical QA involves processing, acquiring and summarizing relevant information and knowledge and then reasoning about how to apply the available knowledge to the current context given by a clinical case. For example, a resident medical doctor preparing for the licensing exams may want to know what and why is the correct treatment or diagnosis in the context of a clinical case [3,4]. This means that a LLM should be able to automatically identify, access and correctly apply the relevant medical knowledge, and that it will be capable of elucidating between the variety of symptoms, each of which may be indicative of multiple diseases. Finally, it is also assumed that the model will interact with the resident medical doctor in a natural manner, ideally using natural language. Therefore, developing the required AI technology to help, for example, resident medical doctors to prepare their licensing exams remains a far from trivial endeavour.

Nonetheless, and as a crucial first step to address this challenge, the AI ecosystem has seen an explosion of LLMs (both general purpose and specific to the medical domain) reporting high accuracy results on Medical QA tasks thereby demonstrating that LLMs are somewhat capable of encoding clinical knowledge [1]. State-of-the-art models include

* Corresponding author.
E-mail addresses: inigoborja.alonso@ehu.eus (I. Alonso), maite.oronoz@ehu.eus (M. Oronoz), rodrigo.agerri@ehu.eus (R. Agerri).

[1] https://huggingface.co/datasets/HiTZ/MedExpQA.

https://doi.org/10.1016/j.artmed.2024.102938
Received 1 April 2024; Received in revised form 8 June 2024; Accepted 26 July 2024
Available online 31 July 2024

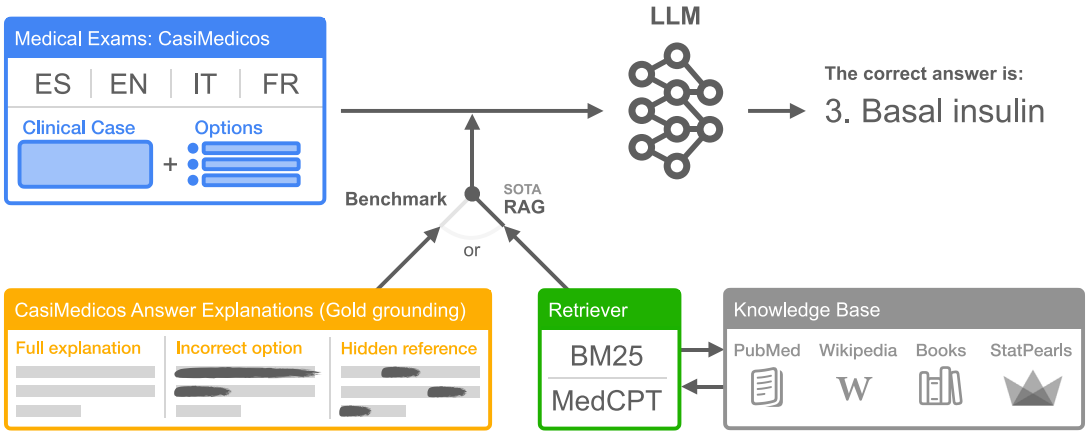


Fig. 1. Graphical description of the MedExpQA benchmark in which various types of gold and external medical knowledge are added to Large Language Models in order to find the correct answer in the CasiMedicos dataset.

publicly available ones such as LLaMA [5] and the medical-specific PMC-LLaMA [6], Mistral [7] and its medical version BioMistral [8], and proprietary models such as MedPaLM [9] and GPT-4 [2], among many others.

While their published high-accuracy scores on Medical QA may seem impressive, these LLMs still present a number of shortcomings. First, LLMs usually generate factually inaccurate answers that seem plausible enough for a non-medical expert (known as hallucinations) [10,11]. Second, their knowledge might be outdated as the pre-training data used to train the LLMs may not include the latest available medical knowledge. Third, the Medical QA benchmarks [1, 11] on which they are evaluated do not include gold reference explanations generated by medical doctors that provide the required reasoning to support the model's predictions. Finally, and to the best of our knowledge, evaluations have only been done for English, which makes it impossible to know how well these LLMs fare for other languages.

Retrieval Augmented Generation (RAG) techniques have been specifically proposed to address the first two issues, namely, the lack of up-to-date medical knowledge and the tendency of these models to hallucinate [11]. Their MedRAG approach obtains clear zero-shot improvements for two of the five datasets on their MIRAGE benchmark, while for the rest the obtained gains are rather modest. Still, MedRAG proves to be an effective technique to improve Medical QA by incorporating external medical knowledge [11].

In this paper we present MedExpQA (Medical Explanation-based Question Answering), which is, to the best of our knowledge, the first multilingual benchmark for Medical QA. Furthermore, and unlike previous work, our new benchmark also includes gold reference explanations to justify why the correct answer is correct and also to explain why the rest of the options are incorrect. Written by medical doctors, these high-quality explanations help to assess the model's decisions based on complex medical reasoning. Moreover, our MedExpQA benchmark leverages the reference explanations as *gold knowledge* to establish various upperbounds for comparison with results obtained when applying automatic MedRAG methods. By doing so, we aim to address all four shortcomings of LLMs for Medical QA listed above.

Although by design independent of the specific source data used, for this work we leverage the Antidote CasiMedicos dataset [4,12], which consist of Resident Medical Exams or *Médico Interno Residente* in Spanish, an exam similar to other licensing examinations such as USMLE, to setup MedExpQA. In addition to a short clinical case, a question and the multiple-choice options, CasiMedicos includes gold reference explanations regarding both the correct and incorrect options. Originally in Spanish, CasiMedicos was translated and annotated in English, French and Italian [4].

Fig. 1 provides an overview of the MedExpQA benchmark. Taking CasiMedicos as the data source, the basic input, without any additional

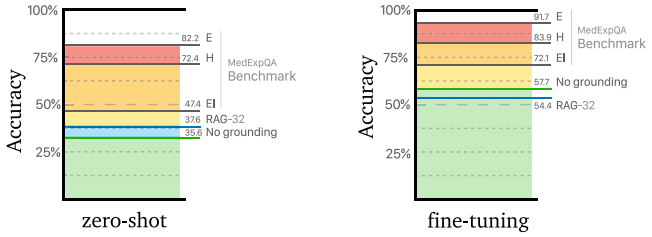


Fig. 2. Overview of averaged results in MedExpQA for gold and automatically knowledge grounding based on Retrieval Augmented Generation (RAG). *E*: gold explanations written by medical doctors; *H*: E with explicit references to the possible answers hidden; and *EI*: gold explanations about the incorrect options; *RAG-32*: automatically retrieved knowledge grounding (details in Section 5); *no-grounding*: baseline model with no external knowledge.

knowledge, to the LLM consists of a clinical case and the multiple-choice options. Furthermore, the model can also be provided with three types of gold reference explanations (or gold knowledge grounding) extracted from the CasiMedicos explanations: (i) the full gold explanation as written by the medical doctors; (ii) only the explanations regarding the incorrect answers and, (iii) the full gold explanation with explicit references to the possible answers hidden. Finally, we can also apply automatic knowledge retrieval approaches such as MedRAG to provide LLMs with automatically obtained up-to-date medical knowledge. Thus, in MedExpQA it is possible to compare not only whether the MedRAG methods improve over the basic input with no external knowledge added, but also to establish the differences in performance of LLMs (with or without RAG) with respect to results obtained when gold reference explanations are available. An additional benefit of MedExpQA being multilingual is that we get to compare LLMs performance not only for English, but also on popular languages such as Spanish, French or Italian.

Fig. 2 shows that comprehensive multilingual experimentation on MedExpQA using four state-of-the-art LLMs including LLaMA [5] PMC-LLaMA [6], Mistral [7] and BioMistral [8], demonstrate that LLMs performance, even when improved with external knowledge from MedRAG (corresponding to RAG-32 in Fig. 2), still has a long way to go to get closer to the performance obtained when the external knowledge available to the LLM is based on gold reference explanations (*E* and *H* in Fig. 2). Another interesting point is that fine-tuning results in huge performance increases across settings and models but at the cost of making MedRAG redundant. In other words, MedRAG only has a positive impact in zero-shot settings. We believe that this illustrates the difficulty of automatically retrieving and integrating readily available knowledge in a way that may positively impact final downstream results on Medical

QA. Finally, results are substantially lower for French, Italian and Spanish, which suggests that more work is needed to improve LLMs performance for languages different to English. Summarizing, the main contributions of our work are the following:

1. MedExpQA: the first multilingual benchmark for MedicalQA including gold reference explanations.
2. Comprehensive study on the role of medical knowledge to answer medical exams by leveraging gold reference explanations written by medical doctors as upper bound with respect to automatically retrieved knowledge using state-of-the-art RAG techniques.
3. Experimental results demonstrate that fine-tuning clearly outperforms querying the LLMs in zero-shot, making redundant the external knowledge obtained via RAG.
4. Overall performance of LLMs with or without RAG still has large room for improvement when compared with any of the results obtained using gold reference explanations.
5. Performance for French, Italian and Spanish substantially lower for every LLM in every evaluation setting, which stresses the urgent need of advancing the state-of-the-art for Medical QA in languages different to English.
6. Data, code and fine-tuned models available to facilitate reproducibility of results and benchmarking of LLMs in the medical domain[2].

In the rest of the paper we first discuss the related work and then in Section 3 we describe the Large Language Models (LLM) and the Retrieval Augmented Generation method used for experimentation. Section 4 provides a description of the MedExpQA benchmark, including the Antidote CasiMedicos dataset. The experimental setup is explained in Section 5 and results are reported in Section 6. Section 7 offers a discussion of the main issues raised by the empirical results obtained. We finish with some concluding remarks and future work in Section 8.

2. Related work

We are currently seeing a vertiginous rhythm in the development of Large Language Models (LLMs) which is having a great impact on Natural Language Processing for the medical domain. This is particularly true on Medical Question Answering tasks where LLMs have been successfully applied to generate answers to highly specialized medical questions. Thus, the performance improvements on Abstractive Medical Question Answering of general purpose LLMs such GPT-4 [2] and GPT-3 [13], PaLM [14], LLaMa [5] or Mistral [7], has resulted in a huge interest to adapt or to generate LLMs specialized for medical text processing.

Some of these models are based on the encoder–decoder architecture, such as SciFive [15], and English T5 model adapted to the scientific domain, or Medical-mT5, a multilingual model built by fine-tuning mT5 on a multilingual corpus of 3B tokens [16]. However, the large majority of the LLMs specially generated for medical applications are autorregresive decoder models such as BioGPT [17], ClinicalGPT [18], Med-PaLM [1], MedPaLM-2 [9], PMC-LLaMA [6], and more recently, BioMistral [8].

These models have been reporting high-accuracy scores on various medical QA benchmarks, which generally consist of exams or general medical questions. Several of the most popular Medical QA datasets [19–24] have been grouped into two multi-task English benchmarks, namely, MultiMedQA [1] and MIRAGE [11] with the aim of providing an easier comprehensive experimental evaluation benchmark of LLMs for Medical QA.

Despite recent improvements on these benchmarks that had led to claims about the capacity of LLMs to encode clinical knowledge [1], these models remain hindered by well known issues related to: (i) their tendency to generate plausible-looking but factually inaccurate answers and, (ii) working with outdated knowledge as their pre-training data may not be up-to-date to the latest available medical progress; (iii) the large majority of these benchmarks do not include gold reference explanations to help evaluate the reasoning capacity of LLMs to predict the correct answers; (iv) they have mostly been developed for English, which leaves a huge gap regarding the evaluation of the abilities of LLMs for other languages.

Regarding the first issue listed above, it should be considered that these LLMs are not restricted to the input context to generate the answer as they are able to produce word by word output by using their entire vocabulary in an auto-regressive manner [25]. This often results in answers that are apparently plausible and factually correct, when in fact they are not always factually reliable. With respect point (ii), while LLMs are pre-trained with large amounts of texts, they may still lack the specific knowledge required to answer highly specialized questions or it may simply be in need of an update.

Recent work [26] has proposed Retrieval Augmented Generation (RAG) [27] to mitigate these limitations. This method involves incorporating relevant external knowledge into the input of these LLMs with the aim of improving the final generation. By doing so, it increases the probability of generated responses being grounded in the automatically retrieved evidence, thereby enhancing the accuracy and quality of the output. Some of the most common retrieval methods employed include TF-IDF, BM25 [28], and others more specific to the medical domain such as MedCPT [29]. With the aim of providing an exhaustive evaluation of RAG methods for the medical domain, the MIRAGE benchmarch includes 5 well-known English Medical QA datasets which are used to compare zero-shot performance of various LLMs whenever automatically retrieved knowledge is available via their MedRAG method or in the absence of it. According to the authors, MedRAG not only helps to address the problem of hallucinated content by grounding the generation on specific contexts, but it also provides relevant up-to-date knowledge that may not be encoded in the LLM [11]. By employing MedRAG they are able to clearly improve the zero-shot results of some of the LLMs tested, although for others results are rather mixed.

Finally, and to the best of our knowledge, no Medical QA benchmark currently addresses the last two shortcomings, namely, the lack of gold reference explanations and multilinguality. Motivated by this, we propose MedExpQA, a multilingual benchmark including gold reference explanations written by medical doctors that can be leveraged to setup various upperbound results to be compared with the performance of LLMs enhanced by automatic RAG methods.

3. Materials and methods

In this section we describe the main resources used in our experimentation with MedExpQA, namely, the Large Language Models (LLMs) tested on our benchmark and MedRAG, the Retrieval Augmented Generation method proposed by Xiong et al. [11] to automatically retrieve medical knowledge.

3.1. Models

We selected two open source state-of-the-art LLMs in the MedicalQA domain at the time of writing: PMC-LLaMA [6] and BioMistral [8].

PMC-LLaMA is based on LLaMA [5], one of the most popular LLMs currently available. PMC-LLaMA is an open-source language model specifically designed for medical applications. This model was first pre-trained on a combination of PubMed-related English academic papers from the S2ORC corpus [30] and from medical textbooks. It was then further fine-tuned on a dataset of instruction-based medical texts. For our experiments we pick the 13B parameter variant of this model which

[2] https://huggingface.co/datasets/HiTZ/MedExpQA

outperforms LLaMA-2 [5], Med-Alpaca [31], and Chat-Doctor [32] in various Medical QA tasks including MedQA [23], MedMCQA [24], and PubMedQA [19].

BioMistral [8] is a suite of open-source models based on Mistral [7] further pre-trained using English textual data from PubMed Central Open Access[3]. They released a set of 7b parameter models following merging techniques like TIES [33], DARE [34], and SLERP [35]. In this paper we use the DARE variant of BioMistral as it is the best performing model on the MedQA benchmark, outscoring other state-of-the-art LLMs on Medical QA evaluations, including PMC-LLaMA.

Additionally, and in order to contrast their performance against their general purpose counterparts, we also test LLaMA-2 and Mistral. Thus, for both PMC-LLaMa and LLaMA-based models we use the 13 billion parameter variants. As BioMistral is only available in the 7b version, we also pick the Mistral model of 7b parameters.

Every zero-shot and fine-tuning experiment with LLMs are performed via the HuggingFace API [36].

3.2. Retrieval-augmented generation (RAG)

We apply MedRAG as the Retrieval-Augmented Generation (RAG) state-of-the-art technique especially developed for the medical domain [11]. RAG approaches are mostly composed of three components: the LLM, the retrieval method and the data source from which to retrieve the knowledge. MedRAG includes four retrievers and four different corpora as data sources.

With respect the retrievers, we use both BM25 [28] and MedCPT [29] to perform the retrieval and fuse the retrieved candidate lists into one using Reciprocal Rank Fusion (RRF) [37]. BM25 is a ranking function used in Information Retrieval to rank documents based on their relevance to a given query. It combines Term Frequency (TF) and Inverse Document Frequency (IDF) to calculate the relevance score of a document to a query taking into account the document length for normalization. MedCPT is a Contrastive Pre-trained Transformer model trained with PubMed search logs for zero-shot biomedical information retrieval. This model retrieves the relevant documents in the knowledge base considering relationships between different medical entities and concepts in the query.

Regarding the data sources, we use MedCorp, a combination of the four corpora available in MedRAG: PubMed, Textbooks [23] for domain-specific knowledge, StatPearls[4] for clinical decision support, and Wikipedia for general knowledge. According to the MIRAGE results [11], using MedCorp was the only realistic option for MedRAG to systematically improve results over the baseline for most of the LLMs and retriever methods evaluated.

4. MedExpQA: A new multilingual benchmark for medical QA

Although independently designed with respect to any specific dataset, in this paper we setup MedExpQA, introduced in Section 4.2, on the Antidote CasiMedicos dataset [4,12], which is described in detailed in Section 4.1.

4.1. Antidote CasiMedicos dataset

Every year the Spanish Ministry of Health releases the previous year's Resident Medical exams or *Médico Interno Residente* (MIR) which, as depicted in Table 1, include a clinical case (**C**), the multiple choice options (**O**), and the correct answer (**A**). The MIR exams are then commented every year by the CasiMedicos MIR Project 2.0[5] which means that CasiMedicos medical doctors voluntarily write gold reference explanations (full gold explanation **E** in Table 1) providing reasons for both correct (**EC**) and incorrect options (**EI**).

The Antidote CasiMedicos dataset [4,12] consists of the original Spanish commented exams which were cleaned, structured and manually annotated to link the relevant textual parts in the gold reference explanation (**E**) with the correct (**EC**) or incorrect options (**EI**). Once the Spanish version of the dataset was created, parallel translated annotated versions were generated for English, French, and Italian.

A quantitative description of the multilingual Antidote CasiMedicos dataset is given in Table 2. The average number of tokens in the clinical cases is 137, being quite similar for Spanish and Italian (140.3 and 142.2 respectively), while for English the average is smaller (115.4 tokens) while the French one is the largest (150.1 tokens). The average length in tokens of the multiple choice options (79.6 tokens in average) is quite high but with a high variability. The multiple choice options may consist of short drug names (the minimum number of words is around 15–17) to long descriptions of treatments or medical claims as illustrated by the example shown in Table 3. The full gold reference explanations that professional medical doctors write can be quite long (170.25 tokens in average) but it should be noted that some documents lack the explanation about the correct answer.

The complexity of some of the clinical case questions can be appreciated in the example shown in Table 3 where the possible answers (section **O**) describe disorders (option (1)), treatments (options (2) and (3)) or medical statements (options (4) and (5)). Furthermore, while in the majority of the cases the question is about the correct answer, sometimes the required option is the incorrect one, as shown in Tables 1 and 3.

The final Antidote CasiMedicos Dataset consists of 622 documents per language [4,12]. The dataset official distribution already provide train, validation and test splits[6] (depicted in Table 4), which we use for the all the experiments presented in Section 6.

Finally, we examined the distribution of correct answers in each of the three splits (train, validation and test) to consider the possibility that an unbalanced distribution might condition the results of the tested models. Fig. 3 shows that, although most of the exams have the option 3 as the correct answer, the distribution among the correct answers in the three subsets is quite balanced. This suggests that this particular issue should not influence the final experimental results.

4.2. The MedExpQA benchmark

MexExpQA is a multilingual benchmark to evaluate LLMs in Medical Question Answering. Unlike previous work, MedExpQA includes reference gold explanations written by medical doctors which are leveraged to setup a benchmark with three types of gold knowledge: (i) the full gold reference explanation (part **E** in Table 1); (ii) the full gold reference explanation corresponding to the incorrect options only (**EI**) and (iii), the full gold reference explanation masking the explicit references in the text to the multiple-choice options.

In other words, and as illustrated in Fig. 1, we use these three types of high-quality explanations written by medical doctors as a proxy of relevant gold knowledge that may be used by LLMs to answer medical questions. Thus, the results obtained by LLMs with each type of gold knowledge can be seen as the upperbound results provided by our benchmark to establish how well LLMs can perform according to the different types of specialized gold knowledge readily available. In the following we describe in detail each of the three types of gold reference explanations that we generate to setup our benchmark.

[3] PMC Open Access Subset. Available from https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

[4] https://www.statpearls.com/

[5] https://www.casimedicos.com/mir-2-0/

[6] https://huggingface.co/datasets/HiTZ/casimedicos-exp

Table 1
Document in the Antidote CasiMedicos dataset with the correct and incorrect explanations manually annotated. C: Clinical case and question; O: Multiple-choice options; A: Correct answer; E: Full gold reference explanation written by medical doctors; EC: Explanation about the correct answer; EI: Explanation about the incorrect answers.

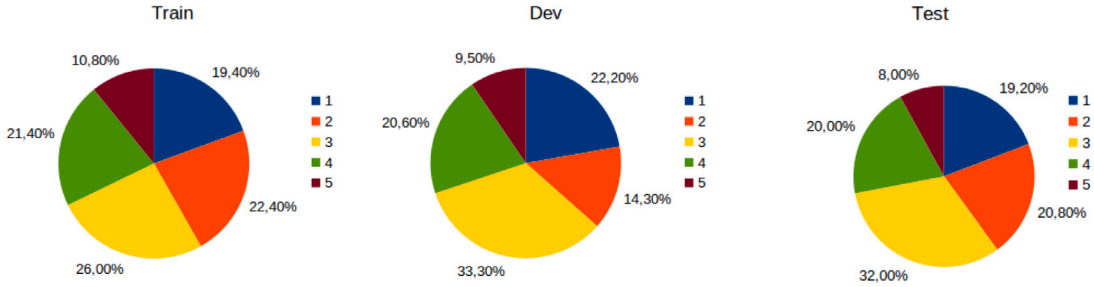
C	30-year-old man with no past history of interest. He comes for consultation due to the presence of small erythematous-violaceous lesions that on palpation appear to be raised in the pretibial region. The analytical study shows a complete blood count and coagulation study without alterations, and in the biochemistry, creatinine and ions are also within the normal range. The urinary sediment study shows hematuria, for which the patient had already been studied on other occasions, without obtaining a definitive diagnosis. Regarding the entity you suspect in this case, it is FALSE that
O	(1) In 20 to 50% of cases there is elevation of serum IgA concentration. (2) In the renal biopsy the mesangial deposits of IgA are characteristic. (3) It is frequent the existence of proteinuria in nephrotic range. (4) It is considered a benign entity since less than 1/3 of patients progress to renal failure. (5) The cutaneous biopsy allows to establish the diagnosis in up to half of the cases.
A	**3**
E	They are talking to us with high probability of a mesangial IgA glomerulonephritis or Berger's disease. Therefore, we are going to discard options one by one: 1: True. Serum IgA elevation is found in up to 50% of cases. 2: True. Mesangial IgA deposits are present in almost 100% of cases. 3: This option is false, because this glomerulonephritis is classically manifested with nephritic and not nephrotic syndrome (although in some rare cases proteinuria in nephrotic range does appear, but in the MIR they do not ask about these rare cases). 4: At the beginning this option generated doubts in me, but looking in the literature, it is true that the evolution to renal failure (according to last series) occurs in about 25% of the cases, so this option is true. 5: Skin biopsy, because it is easier to perform than renal biopsy, is the diagnostic technique of choice (the skin lesions that constitute Schonlein-Henoch purpura, so frequently associated with this entity and which the patient in the case presents, are biopsied).
EC	3: This option is false, because this glomerulonephritis is classically manifested with nephritic and not nephrotic syndrome (although in some rare cases proteinuria in nephrotic range does appear, but in the MIR they do not ask about these rare cases).
EI	1: True. Serum IgA elevation is found in up to 50% of cases. 2: True. Mesangial IgA deposits are present in almost 100% of cases. 4: At the beginning this option generated doubts in me, but looking in the literature, it is true that the evolution to renal failure (according to last series) occurs in about 25% of the cases, so this option is true. 5: Skin biopsy, because it is easier to perform than renal biopsy, is the diagnostic technique of choice (the skin lesions that constitute Schonlein-Henoch purpura, so frequently associated with this entity and which the patient in the case presents, are biopsied).

Table 2
Quantitative description of the multilingual CasiMedicos dataset. Number of tokens in the clinical case including: the question (C), the multiple-choice options (O), the explanation about the correct answer (EC) and the full gold reference explanation (E) including argumentation about the correct and incorrect answers.

	Number of tokens	Average	Min	Max
Spanish	Clinical Case (C)	140.3 ± 62.4	41	504
	Multiple choice options (O)	77.0 ± 47.0	15	297
	Explanation about the correct (EC)	58.9 ± 37.7	0	483
	Full explanation (E)	174.1 ± 147.8	9	982
English	Clinical Case (C)	115.4 ± 52.8	34	419
	Multiple choice options (O)	64.7 ± 37.1	15	217
	Explanation about the correct (EC)	47.3 ± 30.4	0	382
	Full explanation (E)	139.1 ± 117.7	4	784
Italian	Clinical Case (C)	142.2 ± 64.5	35	539
	Multiple choice options (O)	79.0 ± 50.1	17	284
	Explanation about the correct (EC)	60.6 ± 38.4	0	500
	Full explanation (E)	179.1 ± 150.6	8	1013
French	Clinical Case (C)	150.1 ± 68.6	39	586
	Multiple choice options (O)	83.0 ± 52.8	16	319
	Explanation about the correct (EC)	63.9 ± 41.2	0	535
	Full explanation (E)	188.7 ± 158.9	8	1076
Avg. ALL	Clinical Case (C)	137		
	Multiple choice options (O)	79.6		
	Explanation about the correct (EC)	57.6		
	Full explanation (E)	170.25		

Table 3
Example of a document in the CasiMedicos dataset with very different types of response options. (1) diagnosis; (2) and (3) treatments; and (4) and (5) correspond to medical statements.

	Example of a document from the CasiMedicos Dataset
C	A 63-year-old woman comes to the emergency department reporting severe headache with signs of meningeal irritation, bilateral visual disturbances and ophthalmoplegia. A CT scan showed a 2 cm space-occupying lesion in the sella turcica compatible with pituitary adenoma with signs of intratumoral hemorrhage, with deviation of the pituitary stalk and compression of the glandular tissue. Mark which of the following answers is WRONG:
O	(1) Diagnostic suspicion is pituitary apoplexy. (2) Treatment with high-dose corticosteroids should be initiated and the evolution observed, since this treatment could reduce the volume of the lesion and avoid intervention. (3) Treatment with glucocorticoids should be considered to avoid secondary adrenal insufficiency that would compromise the patient's vital prognosis. (4) The presence of ophthalmoplegia and visual defects are indications for prompt intervention by urgent surgical decompression. (5) After resolution of the acute picture, the development of panhypopituitarism is frequent.
A	**4**



(a) Distribution of correct answers in the **train** dataset.

(b) Distribution of correct answers in the **validation** dataset.

(c) Distribution of correct answers in the **test** dataset.

Fig. 3. Distribution of correct answers in the train, validation and test splits. The percentage in blue indicates the proportion of exams with the first option, number 1, as correct answer; orange corresponds to option 2; yellow to option 3; green to option 4; and brown to option 5. Note that not every document includes 5 possible options. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Number of documents in CasiMedicos train, validation and test splits.

	Train	Validation	Test
Clinical cases	434	63	125
Total	622		

4.2.1. *Full reference gold explanations*

The full explanation (**E**) about the correct and incorrect answers is given as context to the LLM, in what we assume to be gold specific knowledge for the model to answer the medical questions of CasiMedicos. Being the full gold reference explanation, we consider this to be the best possible form of gold knowledge that we can provide the LLM with. In other words, the performance obtained in MedExpQA using this type of knowledge will mark the upperbound for this particular benchmark. Table 5 provides an example of the full gold reference explanation for the same document already discussed in Table 1.

4.2.2. *Explanation of the incorrect options*

As shown in Table 6, for this particular type of gold knowledge we only use the part of the full gold reference explanation corresponding to the explanations about the incorrect options (**EI**). This type gold knowledge aims to test the capacity of LLMs to correctly answer the medical question by knowing which options are incorrect.

Depending on the nature of the question, sometimes medical doctors consider sufficient to only explain the correct answer. Thus, it should be noted that not every document in CasiMedicos includes the gold reference explanations about the incorrect options. On average, 20.5% of the explanations correspond in their entirety to the correct answer (17.7% in the train set, and 22.2% and 21.6% in the validation and test, respectively), while 26.7 include the explanations for all the possible options. Obviously, as CasiMedicos is a multilingual parallel dataset, this phenomenon occurs across the four languages: English, French, Italian and Spanish.

4.2.3. *Full gold explanation with explicit references hidden*

As it can be appreciated in the full gold reference explanations discussed above, most of the time medical doctors provide explicit textual references regarding the correct or incorrect options. In order to analyze the impact of these explicit signals or patterns on the LLMs performance, we decided to mask those explicit references to establish how well LLMs could answer with actual gold knowledge but without the easy clues in the text pointing to the correct or incorrect answers.

In order to avoid the manual annotation of 2488 documents, we prompt GPT-4[7] [38] with a set of rules and in-context-learning examples to automatically mask the specific areas of text that may point the model at the correct or incorrect answer without any further reasoning. The prompt can be found in A, Fig. A.10.

A small manual analysis of a subset of GPT-4-generated texts revealed a strong correlation with human annotations. To further validate the efficacy of our method, we randomly selected 80 documents (20 per

[7] `gpt-4-1106-preview`

Table 5
Full explanation (E) of the example in Table 1. The explanation about the correct answer is marked in blue and the remaining 4 explanations for the incorrect options in green.

E	They are talking to us with high probability of a mesangial IgA glomerulonephritis or Berger's disease. Therefore, we are going to discard options one by one: 1: True. Serum IgA elevation is found in up to 50% of cases. 2: True. Mesangial IgA deposits are present in almost 100% of cases. 3: This option is false, because this glomerulonephritis is classically manifested with nephritic and not nephrotic syndrome (although in some rare cases proteinuria in nephrotic range does appear, but in the MIR they do not ask about these rare cases). 4: At the beginning this option generated doubts in me, but looking in the literature, it is true that the evolution to renal failure (according to last series) occurs in about 25% of the cases, so this option is true. 5: Skin biopsy, because it is easier to perform than renal biopsy, is the diagnostic technique of choice (the skin lesions that constitute Schonlein-Henoch purpura, so frequently associated with this entity and which the patient in the case presents, are biopsied).

Table 6
Explanation of the Incorrect Options (EI) which corresponds to the full explanation (E) of the example in Table 1 with the explanation of the correct answer removed.

EI	They are talking to us with high probability of a mesangial IgA glomerulonephritis or Berger's disease. Therefore, we are going to discard options one by one: 1: True. Serum IgA elevation is found in up to 50% of cases. 2: True. Mesangial IgA deposits are present in almost 100% of cases. 5: Skin biopsy, because it is easier to perform than renal biopsy, is the diagnostic technique of choice (the skin lesions that constitute Schonlein-Henoch purpura, so frequently associated with this entity and which the patient in the case presents, are biopsied).

Table 7
Full gold reference explanation with explicit references hidden (H). Process performed by GPT-4 with the prompt in A Fig. A.10. In this example the segments *'This option is false'*, *'so this option is true'* and *'is the diagnostic technique of choice'* are hidden.

H	They are talking to us with high probability of a mesangial IgA glomerulonephritis or Berger's disease. Therefore, we are going to discard options one by one: 1: True. Serum IgA elevation is found in up to 50% of cases. 2: True. Mesangial IgA deposits are present in almost 100% of cases. 3: [HIDDEN], because this glomerulonephritis is classically manifested with nephritic and not nephrotic syndrome (although in some rare cases proteinuria in nephrotic range does appear, but in the MIR they do not ask about these rare cases). 4: At the beginning this option generated doubts in me, but looking in the literature, it is true that the evolution to renal failure (according to last series) occurs in about 25% of the cases, [HIDDEN]. 5: Skin biopsy, because it is easier to perform than renal biopsy, is the [HIDDEN] (the skin lesions that constitute Schonlein-Henoch purpura, so frequently associated with this entity and which the patient in the case presents, are biopsied).

language) and measured performance across the four languages. This resulted in an average F1 score of 0.85 with a standard deviation of 0.02.

Thus, this method allowed us to perform this rather precise multilingual redacting process over the 2488 documents in a fast and cost effective manner. Table 7 shows how every explicit reference to the correct or incorrect answers discussed previously now appear as **[HIDDEN]**.

The results obtained by LLMs in MedExpQA using the three types of gold knowledge described above can then be compared with other automatic knowledge retrieval approaches based, for example, on Retrieval-Augmented Generation techniques for the medical domain such as MEDRAG, introduced in the previous section. Furthermore, we should stress that MedExpQA as a benchmark is independent of any dataset, as the only requirement is for it to include gold reference explanations of the possible answers.

5. Experimental setup

For our experiments we selected top performing state-of-the-art models for Medical Question Answering described in Section 3.1, namely, PMC-LLaMA, LLaMA-2, BioMistral, and Mistral.

We test these models in both zero-shot (see prompts in Figs. A.6–A.9) and fine-tuned settings to contrast their out-of-the-box performance against a more adjusted performance to our dataset. The models were fine-tuned using Low-Rank Adaptation (LoRA) [39], using adapters with a rank of 8 and a scaling factor (alpha) of 16 across all models (details about parameters used with LoRA are provided in C).

The choice of hyperparameters was based on previous work using the same LLMs we use in this papers. Moreover, satisfactory results were confirmed in a preliminary round of experiments. Although these models would benefit from an exhaustive grid search of hyperparameters tailored to each model and evaluation setting, the compute required to do so exceeds the capacity of our lab. Full details of hyperparameter settings are available in B. Each model was fine-tuned for 10 epochs, with checkpoints saved at the end of each. Experiments were undertaken in a NVIDIA A100 GPU (C offers information about computation times). At the end of the fine-tuning process, the checkpoint with the highest performance was selected. All models underwent monolingual training using the dataset corresponding to each specific language. We will measure the impact on MedExpQA of the different types of knowledge that LLMs may use:

(i) Gold grounding knowledge:

  (1) **E**: Full gold reference explanations as written by the medical doctors.
  (2) **EI**: Gold explanations about the Incorrect Options.
  (3) **H**: Full gold explanations with [HIDDEN] explicit references to the multiple-choice options.

(ii) Automatically obtained grounding knowledge:

  (1) **None**: Answering the medical question with no additional external knowledge.
  (2) **RAG-7**: Automatically obtained knowledge by applying MEDRAG to retrieve the k=7 most relevant documents.
  (3) **RAG-32**: Automatically obtained knowledge by applying MEDRAG to retrieve the k=32 most relevant documents.

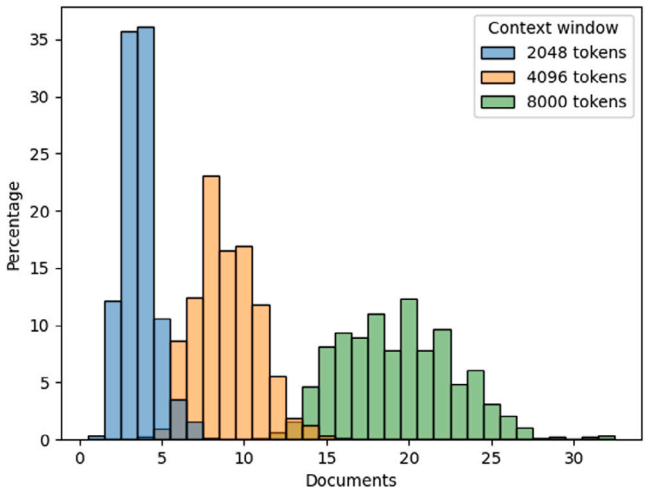


Fig. 4. Distribution of retrieved documents across different context windows. Three different histograms are shown that depict the maximum number of documents that can be accommodated within various context windows across dataset examples: 2,048 tokens (PMC-LLaMA), 4,096 tokens (LLaMA2), and 8,192 tokens (Mistral and BioMistral).

We use the entire clinical case, question, and multiple-choice options to generate the query for all 6 different evaluation settings. Gold knowledge grounding is leveraged as explained in the previous section. With respect to the methods to automatically obtained external knowledge, we take into account the results obtained in the MIRAGE benchmark [11] and apply MEDRAG by using the RRF-2 of two retrieval algorithms, namely, BM25 and MedCPT, over the MEDCORP corpus. We use the entire clinical case, question, and multiple-choice options to generate the query to retrieve the k = 7 most relevant documents. We define k = 7 by computing the average token length of MedCorp documents; if we consider that 85% of our prompts can be represented under 400 tokens, this leaves 1648 tokens for knowledge grounding, which amounts to 7 documents on average. This configuration is used to define **RAG-7**.

Furthermore, as MEDRAG obtained best results for most of the benchmarks when retrieving at most 32 documents, we also experimented with this setting. Nevertheless, it should be considered that the context window of each model, namely, the maximum amount of word tokens that each LLM can pay attention to in the input, will determine how many of these documents are actually fed into the LLM at each forward pass. Hence, when the combination of both the retrieved documents and the prompt exceed the context window, then we truncate the amount of documents to ensure that the prompt is not affected. Fig. 4 illustrates the distribution of documents corresponding to different context window sizes. Specifically, it shows the number of examples in the dataset that align with varying numbers of retrieved documents for context windows of 2048, 4096, and 8000 tokens. In the results reported in the next section, **RAG-32** for both zero-shot and fine-tune settings helps us to evaluate the impact of retrieving more or less relevant documents as external knowledge.

5.1. Evaluation

We ask LLMs to generate not only the index number of the predicted correct option but also the full textual answer. However, accuracy is calculated by comparing the first generated character after the prompt following *"The correct answer is: "*[8]. We verify that this character always corresponds to one of the options in the exams' possible answers. A provides an example of the prompts used for each language and for every model.

[8] And equivalent prompts for French, Italian and Spanish.

6. Results

We report the main results of the experiments performed in the MedExpQA benchmark in Table 8 for zero-shot while the fine-tuning accuracy scores are presented in Table 9.

Zero-shot results. They show that Mistral consistently achieves the highest accuracy across every evaluation setting and language, even outscoring the medical specific BioMistral. Among the gold knowledge results, we can see that removing the explanation of the correct answer (**EI**) really hinders performance. However, using the full gold reference answer helps LLMs to obtain excellent marks. Moreover, differences between using **E** and **H** are quite large, especially for languages different to English.

It should be noted that the best automatic method still fares very badly with respect to any of the gold knowledge results, which shows that retrieval methods for the medical domain still have large room for improvement. While the best automatic method corresponds to **RAG-7**, differences in performance are not that great with respect to **None** or **RAG-32**.

We hypothesize that the lack of substantial improvement when using 32 snippets for knowledge grounding may indicate that a saturation point may be reached beyond which additional snippets do not provide any additional benefit. To analyze this more precisely, we conducted an evaluation of the zero-shot performance of the 4 LLMs when feeding the model from 0 to up to 32 snippets, following a power of two sequence of snippets. Thus, Fig. 5 illustrates that a positive trend exists when increasing the number of snippets. However, we can see how this improvement tanks at around 8 snippets in most of the models. This result correlates to our findings in Tables 8 and 9.

Finally, performance on English was substantially higher for every models and RAG configurations. This manifests the English-centric focus of most LLMs while showcasing the urgent need of dedicating resources and effort to developing multilingual LLMs which could then compete across all languages included in multilingual benchmarks such as MedExpQA.

Fine-tuning results. They show that fine-tuning the LLMs on the CasiMedicos dataset help to greatly increase performance for every evaluation setting, language and LLM. BioMistral seems to obtain the best overall scores but that is due to its high scores on the full gold reference explanation setting (**E**). Thus, if we look at the rest of the evaluation settings, Mistral, as it happened in the zero-shot scenario, remains the best performing LLM on the MedExpQA benchmark.

The superior results of **None** with respect to RAG scores demonstrate that fine-tuning makes any external knowledge automatically retrieved using RAG methods redundant. Finally, while scores for French, Italian and Spanish remain lower than those obtained for English, performance for those languages greatly benefit from fine-tuning, especially if we compare them with their zero-shot counterpart results.

Overall results. Overall, results demonstrate that the gold reference explanations leveraged as knowledge for Medical QA help LLMs to obtain almost perfect scores, especially when fine-tuning the models. Fine-tuning particularly benefits **EI**, which obtains as good results as **H** applied in zero-shot settings.

Our results allow us to draw several more conclusions. First, that despite using state-of-the-art RAG methods for the medical domain [11], their results are rather disappointing. Both in zero-shot when compared with the results based on any kind of gold knowledge, and in fine-tuning in which RAG methods score worse than not using any additional knowledge.

Second, our MedExpQA benchmark suggests that overall performance of even powerful LLMs such as Mistral still have a huge room for improvement to reach scores comparable to those obtained when gold knowledge is available.

We calculated a McNemar [40] test of statistical significance to establish whether the *RAG-7* and *RAG-32* results were significantly

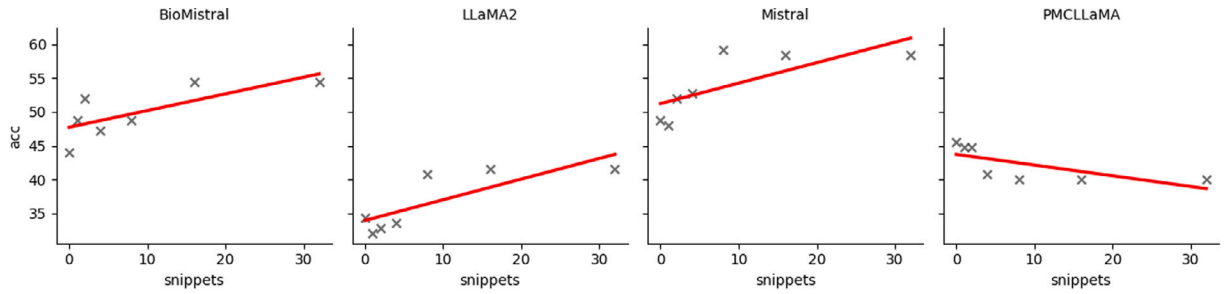


Fig. 5. Performance of different models in a zero-shot setting with up to 0, 2, 4, 8, 16, and 32 retrieved snippets.

Table 8
Zero-shot results. E: Full gold explanation. EI: Gold Explanations of the Incorrect Options; H: Full gold explanation with Hidden explicit references to the correct/incorrect answer; None: model without any additional external knowledge; RAG-7: Retrieval Augmented Generation with k = 7; RAG-32: Retrieval Augmented Generation with k = 32; underline: best result per type of knowledge; **bold**: best result overall.

	PMC-LLaMA (13B)				LLaMA2 (13B)				Mistral (7B)				BioMistral (7B)				Avg.
	EN	ES	IT	FR	EN	ES	IT	FR	EN	ES	IT	FR	EN	ES	IT	FR	ALL
E	83.2	77.6	76.8	80.0	81.6	77.6	77.6	75.2	**89.6**	**88.0**	**87.2**	**88.0**	88.8	83.2	80.8	80.8	**82.2**
EI	60.0	42.4	43.2	46.4	44.0	31.2	39.2	44.8	59.2	53.6	52.0	52.8	50.4	44.0	46.4	49.6	47.4
H	78.4	63.2	72.0	70.4	68.8	64.8	63.2	65.6	82.4	75.2	77.6	78.4	80.8	74.4	69.6	74.4	72.4
None	45.6	36.8	33.6	30.4	34.4	18.4	12.8	27.2	48.8	41.6	40.8	39.2	44.0	39.2	35.2	41.6	35.6
RAG-7	40.0	30.4	28.0	24.8	42.4	36.0*	30.4*	32.0	55.2	44.0	38.4	42.4	44.8	40.0	40.8	36.8	37.9
RAG-32	40.0	30.4	28.0	24.8	41.6	31.2*	32.8*	26.4	58.4*	41.6	41.6	42.4	54.4	37.6	31.2	39.2	37.6
Avg.	57.9	46.8	46.9	46.1	52.1	43.2	42.7	45.2	**65.6**	**57.3**	**56.3**	**57.2**	60.5	53.1	50.7	53.7	–

* Results that are statistically significant at $\alpha = .05$ wrt to their None baseline.

Table 9
Fine-tuning results. E: Full gold explanation. EI: Gold Explanations of the Incorrect Options; H: Full gold explanation with Hidden explicit references to the multiple choice options; None: model without any additional external knowledge; RAG-7: Retrieval Augmented Generation with k = 7; RAG-32: Retrieval Augmented Generation with k = 32; underline: best result per type of knowledge; **bold**: best result overall.

	PMC-LLaMA (13B)				LLaMA2 (13B)				Mistral (7B)				BioMistral (7B)				Avg.
	EN	ES	IT	FR	EN	ES	IT	FR	EN	ES	IT	FR	EN	ES	IT	FR	ALL
E	92.0	89.6	89.6	88.8	90.4	90.4	89.6	92.0	**94.4**	92.8	91.2	92.8	**94.4**	**93.6**	**92.0**	**93.6**	**91.7**
EI	69.6	67.2	67.2	68.0	73.6	70.4	66.4	70.4	81.6	78.4	75.2	76.8	73.6	72.0	71.2	71.2	72.1
H	82.4	76.0	80.0	82.4	83.2	85.6	84.0	81.6	88.0	84.8	88.8	88.0	83.2	82.4	86.4	84.8	83.9
None	58.4	48.8	49.6	53.6	57.6	50.4	53.6	54.4	68.0	63.2	56.8	66.4	61.6	58.4	56.8	65.6	57.7
RAG-7	56.8	35.2	44.8	38.4	60.8	56.8	48.8	51.2	69.6	59.2	56.8	64.8	64.8	57.6	61.6	59.2	55.4
RAG-32	56.8	35.2	44.8	38.4	60.8	52.0	51.2	49.6	75.2	55.2	52.0	60.0	65.6	57.6	55.2	60.8	54.4
Avg.	69.3	58.7	62.7	61.6	71.1	67.6	65.6	66.5	**79.5**	**72.3**	70.1	**74.8**	73.9	70.3	**70.5**	72.5	–

better than their respective *None* baselines. As it can be seen in Tables 8 and 9, only five zero-shot scores (out of 64) marked with an asterisk in Table 8 are statistically significant at $\alpha = .05$. Finally, performance for languages different to English is much lower for every model and evaluation setting. This points out to an urgent necessity to invest in the development and research of LLMs which may be optimized not only for English, but for other world languages too. Obviously, the evaluation of such LLMs would in turn require multilingual evaluation benchmarks which may be deployed to provide a comprehensive and realistic overview of their performance. We hope that contributing MedExpQA may serve as encouragement to the AI and medical research communities to generate more benchmarks of its kind for many of the world languages.

7. Discussion

The results discussed in the previous section show that even when performing fine-tuning with the full gold reference explanations LLMs still remain several points below perfect scores. Furthermore, the statistical analysis of the obtained results indicates that, despite differences compared to the **None** models, the performance gains (when that is the case) of models using *RAG-7* or *RAG-32* are, in 61 out 64 cases, not statistically significant. In contrast, the statistical analysis found out that the results using gold knowledge (**E**, **EI**, **H**) were all statistically significant at $\alpha = .05$

Apart from the evaluation results, and in order to better understand the dataset on which the MedExpQA is setup, we performed several analysis regarding the quality and quantity of the explanations provided by the CasiMedicos medical doctors.

Regarding the quality of the explanations, we found several examples such as the one depicted in Table 10. Instead of directly answering the question, the medical doctor (psychiatry resident) writing the explanation gives information that is not relevant to explain the correct answer (marked in red). We hypothesize that such explanations, which lack any relevant medical information, may have a negative impact on the final LLMs performance.

It should be noted that, despite CasiMedicos being a high-quality dataset written voluntarily by medical doctors, sometimes (i) their explanations may not follow a repetitive formal structure and, (ii) they are not always subjected to a second review by an auditor as it usually happens in specialized textual books.

Regarding the quantity of the explanations, around 5% of the full gold reference explanations in the CasiMedicos dataset do not contain any explicit explanation regarding the correct answer. Sometimes the

Table 10
Example of a gold full explanation (E) with irrelevant and not medical comments.

E	Another simple question with an immediate answer, which offers no doubt. It describes a patient worried about a non-existent physical defect, whose concern distresses him and prevents him from leaving the house. As a psychiatry resident, I wish the MIR questions in my specialty were a bit more thought-provoking and in-depth, although I know that the seconds you will have saved by marking **the fourth** one directly are very valuable.

medical doctor explains the incorrect options, hoping that the reader may indirectly reach the correct conclusion, or sometimes they are cases such as the one discussed above.

In any case, while it is possible to filter out such examples, we thought it useful to leave them with the aim of analyzing in the future the performance of LLMs and RAG methods for these specific cases. After all, we would like LLMs to be able to also generalize in situations in which the knowledge is provided in a non-standard structured manner, as it is the case in the large majority of the full gold reference explanations provided in CasiMedicos.

We would like to give a final word on multilinguality. Results have shown that performance for French, Italian and Spanish is worse across the board and we believe that this topic has a lot of interesting questions for future research. Are these results a consequence of the pre-training of the LLMs? For the RAG experiments, how much, positive or negative, influence has the fact that the extracted knowledge from MedCorp is in English? Would it be better to prompt the model only in English and then translate the answers into each of the target languages, in what is usually known as a *translate-test* approach? We believe that a benchmark such as MedExpQA would help to investigate these research questions which may be crucial to develop robust multilingual medical QA approaches.

8. Concluding remarks

In this paper we present MedExpQA, the first multilingual benchmark for Medical QA. As a new feature, our new benchmark also includes gold reference explanations to justify why the correct answer is correct and also to explain why the rest of the options are incorrect. The high-quality gold explanations have been written by medical doctors and they allow to test the LLMs when different types of gold knowledge is available. Comprehensive experimentation has demonstrated that automatic state-of-the-art RAG methods still have a long way to go to get near the scores obtained by LLMs when fed with gold knowledge. Furthermore, our benchmark has made explicit the lower overall performance of LLMs for languages other than English for Medical QA.

We think that MedExpQA may contribute to the development of AI tools to assist medical experts in their everyday activities by providing a robust multilingual benchmark to evaluate LLMs in Medical QA. Future work may involve evaluating LLMs not only regarding their accuracy in predicting the correct answer, but also on the quality of the explanations generated to justify such prediction. Of course, these approaches may pose new evaluation challenges that have not been yet contemplated in this work.

CRediT authorship contribution statement

Iñigo Alonso: Writing – original draft, Visualization, Validation, Software, Investigation, Data curation, Conceptualization. **Maite Oronoz:** Writing – original draft, Visualization, Supervision, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Rodrigo Agerri:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Methodology, Investigation, Funding acquisition, Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Rodrigo Agerri reports financial support was provided by Spain Ministry of Science and Innovation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the CasiMedicos Proyecto MIR 2.0 for their permission to share their data for research purposes. This work has been partially supported by the HiTZ Center and the Basque Government, Spain (Research group funding IT1570-22). We are also thankful to several MCIN/AEI/10.13039/501100011033 projects: (i) Antidote (PCI2020-120717-2), and by European Union NextGenerationEU/PRTR; (ii) DeepKnowledge (PID2021-127777OB-C21) and ERDF A way of making Europe; (iii) Lotu (TED2021-130398B-C22) and European Union NextGenerationEU/PRTR; (iv) EDHIA (PID2022-136522OB-C22); (v) DeepMinor (CNS2023-144375) and European Union NextGenerationEU/PRTR. We also thank the European High Performance Computing Joint Undertaking (EuroHPC Joint Undertaking, EXT-2023E01-013) for the GPU hours.

Appendix A. Prompts

In this appendix, we provide the specific prompts used to interact with the Large Language Models of this work.

Appendix B. Hyperparameters

In this appendix we list some of the hyperparameters used in this work (see Table B.11).

Appendix C. Efficiency metrics

In this work we only use or apply the LLMs to establish our benchmark, be that in zero-shot or fine-tuning. As such, we do not perform any modification in the way the LLMs work. Therefore, for efficiency and architectural issues the original papers of Llama2, PMC-Llama, Mistral and BioMistral could be inspected. Our contributions are focused on (i) establishing a multilingual benchmark for Medical QA, (ii) experimenting with state-of-the-art RAG methods and (iii) providing gold reference explanations as a form of "gold" RAG that can be used to compare the LLMs with. Having said that, below we offer detailed information about some efficiency metrics. All the metrics have been calculated using a NVIDIA A100 Graphics Processing Unit (GPU).

- The total number of parameters updated through Low Rank Adaptation (LoRA) during Parameter-Efficient Fine-Tuning (PEFT) are the reported in Table C.12.
- Table C.13 shows the number of **samples per second** processed when using Mistral (7B) and LLaMA2 (13B) in a NVIDIA A100 GPU. The performance in the other two models, BioMistral (7B) and PMC-LLaMA (13B) is the same.
- Table C.14 shows **the time in minutes and hours** when processing data with Mistral (7B) and LLaMA2 (13B). The other two models, BioMistral (7B) and PMC-LLaMA (13B), showcase the same times.

```
======================== Prompt English ========================

You are a helpful medical expert, and your task is to answer a
multi-choice medical question using the relevant documents. Please
choose the answer from the provided options. Your responses will
be used for research purposes only, so please have a definite
answer.
Here are the relevant documents:
{context}
Here is the question:
{question}
Here are the potential choices:
{options}
The correct answer is:
```

Fig. A.6. Prompt used for models in English.

```
======================== Prompt Spanish ========================

Eres un experto médico y tu tarea consiste en responder a una
pregunta médica de test utilizando tu conocimiento y los
siguientes documentos relevantes. Por favor, elige la respuesta
entre las opciones proporcionadas. Tus respuestas se utilizarán
únicamente con fines de investigación, así que te rogamos que
proporciones una respuesta definitiva.
Estos son los documentos relevantes:
{context}
Aquí está la pregunta:
{question}
Aquí están las posibles opciones:
{options}
La opción correcta es:
```

Fig. A.7. Prompt used for models in Spanish.

```
======================== Prompt Italian ========================

Sei un medico esperto e il tuo compito consiste nel rispondere a
una domanda di test medico utilizzando le tue conoscenze e i
documenti successivi rilevanti. Per favore, scegli la risposta tra
le opzioni fornite. Le tue risposte verranno utilizzate
esclusivamente con fini di indagine, quindi ti chiediamo di
fornirti una risposta definitiva.
Questi sono i documenti rilevanti:
{context}
Ecco la domanda:
{question}
Ecco le opzioni possibili:
{options}
L'opzione corretta è:
```

Fig. A.8. Prompt used for models in Italian.

```
======================== Prompt French ========================

Vous êtes un expert en médecine et votre tâche consiste à répondre
à une question d'examen médical en utilisant vos connaissances et
les documents suivants. Veuillez choisir la réponse parmi les
options proposées. Vos réponses seront utilisées uniquement à des
fins de recherche, veuillez donc fournir une réponse claire.
Voici les documents pertinents:
{context}
Voici la question:
{question}
Voici les options possibles:
{options}
La bonne option est:
```

Fig. A.9. Prompt used for models in French.

```
======================== Prompt Redacting ========================
In the following text, remove all references that clearly state
that any of the options 1, 2, 3, {"4 or 5" if
example_contains_5_options else "or 4"} are either correct or
false. Don't change the original text and don't write linebreaks;
only replace with the tag [HIDDEN] the text that says that
something is the correct or incorrect option if there is are any.
Don't replace text that doesn't specifically imply that certain
something is the right or wrong answer. For example: the text
"option {correct_option_index} is correct." should be "[HIDDEN]",
the text "Option {random.choice(incorrect_option_indexes)} is less
likely because this and that" should be "[HIDDEN] this and that",
the text "answer blablabla is the right answer because whatever"
should be "answer blablabla is [HIDDEN] whatever". Here is the
text: {full_answer}
```

Fig. A.10. Prompts to remove explicit references to the multiple-choice options.

Table B.11
Hyperparameters used in the configuration of the experiments.

Hyperparameter	Value
Optimizer	adamw_torch_fused
Learning rate	0.00015
Weight decay	0.0
ADAM ϵ	1e−7
Epochs	10
Train batch size	16
Evaluation batch size	8
Floating Point 16-bit precision training	False
Brain Float 16-bit precision training	True
Maximum #tokens in input	
PMCLLaMA	2048
LLaMA2	4096
Mistral	8000
BioMistral	8000
Maximum #tokens in generation	
PMCLLaMA	2048
LLaMA2	4146
Mistral	8050
BioMistral	8050
Low-Rank Adaptation (LoRA)	
R parameter	8
LoRA α	16
LoRA Dropout	0.05

Table C.12
Trainable parameters: Number of parameter in training using the LoRA model; All parameters: total of parameters used in the LoRA model; Trainable %: number of trainable parameters of the total number of parameters in the LoRA model.

	7B parameter models		
	Trainable parameters	All parameters	Trainable %
Mistral and BioMistral	20,971,520	3,773,042,688	0.555825
	13B parameter models		
	Trainable parameters	All parameters	Trainable %
LLaMA2	31,293,440	6,703,272,960	0.466838
PMC-LLaMa	31,293,440	6,703,283,200	0.466838

Table C.13
Samples processed by second in a NIVIDIA A100 GPU. E: Full gold explanation. H: Full gold explanation with Hidden explicit references to the correct/incorrect answer; None: model without any additional external knowledge; RAG-7: Retrieval Augmented Generation with k = 7; RAG-32: Retrieval Augmented Generation with k = 32.

Samples per second	Train		Inference	
	7B	13B	7B	13B
E	1.981	1.270	7.681	4.757
H	1.998	1.282	7.676	4.76
None	3.248	2.116	11.375	6.956
RAG-7	1.031	0.629	3.637	2.081
RAG-32	0.191	0.281	0.744	1.013

Table C.14
Time in minutes (m) and hours (h) when processing data in a NIVIDIA A100 GPU. E: Full gold explanation. H: Full gold explanation with Hidden explicit references to the correct/incorrect answer; None: model without any additional external knowledge; RAG-7: Retrieval Augmented Generation with k = 7; RAG-32: Retrieval Augmented Generation with k = 32.

Time for training	7B	13B
E	1 h 4 m	2 h 1 m
H	1 h 9 m	2 h 9 m
None	47 m	1 h 39 m
RAG-7	1 h 42 m	3 h 2 m
RAG-32	7 h 34 m	5 h 31 m

References

[1] Singhal K, Azizi S, Tu T, Mahdavi SS, Wei J, Chung HW, et al. Large language models encode clinical knowledge. Nature 2023;620(7972):172–80.

[2] Nori H, King N, McKinney SM, Carignan D, Horvitz E. Capabilities of gpt-4 on medical challenge problems. 2023, arXiv preprint arXiv:2303.13375.

[3] Safranek CW, Sidamon-Eristoff AE, Gilson A, Chartash D. The role of large language models in medical education: Applications and implications. JMIR Med Educ 2023;9:e50945.

[4] Goenaga I, Atutxa A, Gojenola K, Oronoz M, Agerri R. explanatory argument extraction of correct answers in resident medical exams. 2023.

[5] Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al. Llama 2: Open foundation and fine-tuned chat models. 2023, arXiv:2307.09288.

[6] Wu C, Lin W, Zhang X, Zhang Y, Wang Y, Xie W. PMC-LLaMA: Towards Building Open-source Language Models for Medicine. 2023, arXiv:2304.14454.

[7] Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas Ddl, et al. Mistral 7B. 2023, arXiv preprint arXiv:2310.06825.

[8] Labrak Y, Bazoge A, Morin E, Gourraud P-A, Rouvier M, Dufour R. BioMistral: A Collection of Open-Source Pretrained Large Language Models for Medical Domains. 2024, arXiv:2402.10373.

[9] Singhal K, Tu T, Gottweis J, Sayres R, Wulczyn E, Hou L, et al. Towards expert-level medical question answering with large language models. 2023, arXiv preprint arXiv:2305.09617.
[10] Xie Q, Schenck EJ, Yang HS, Chen Y, Peng Y, Wang F. Faithful AI in medicine: A systematic review with large language models and beyond. medRxiv 2023.
[11] Xiong G, Jin Q, Lu Z, Zhang A. Benchmarking Retrieval-Augmented Generation for Medicine. 2024, arXiv preprint arXiv:2402.13178.
[12] Agerri R, Alonso I, Atutxa A, Berrondo A, Estarrona A, García-Ferrero I, et al. HiTZ@Antidote: Argumentation-driven Explainable Artificial Intelligence for Digital Medicine. In: SEPLN 2023: 39th International Conference of the Spanish Society for Natural Language Processing. 2023.
[13] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst 2020;33:1877–901.
[14] Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. PaLM: Scaling Language Modeling with Pathways. J Mach Learn Res 2022;24:240:1–240:113.
[15] Phan LN, Anibal JT, Tran H, Chanana S, Bahadroglu E, Peltekian A, et al. SciFive: a text-to-text transformer model for biomedical literature. 2021, CoRR arXiv:2106.03598.
[16] García-Ferrero I, Agerri R, Salazar AA, Cabrio E, de la Iglesia I, Lavelli A, et al. Medical mT5: An Open-Source Multilingual Text-to-Text LLM for the Medical Domain. In: Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING). 2024.
[17] Luo R, Sun L, Xia Y, Qin T, Zhang S, Poon H, et al. BioGPT: generative pre-trained transformer for biomedical text generation and mining. Brief Bioinform 2022;23(6).
[18] Wang G, Yang G, Du Z, Fan L, Li X. ClinicalGPT: Large Language Models Finetuned with Diverse Medical Data and Comprehensive Evaluation. 2023, arXiv preprint arXiv:2306.09968.
[19] Jin Q, Dhingra B, Liu Z, Cohen W, Lu X. PubMedQA: A Dataset for Biomedical Research Question Answering. In: Inui K, Jiang J, Ng V, Wan X, editors. Proceedings of EMNLP-IJCNLP. Association for Computational Linguistics; 2019, p. 2567–77.
[20] Abacha AB, Shivade C, Demner-Fushman D. Overview of the MEDIQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering. In: Proceedings of the 18th bioNLP workshop and shared task. 2019, p. 370–9.
[21] Vilares D, Gómez-Rodríguez C. HEAD-QA: A Healthcare Dataset for Complex Reasoning. In: Proceedings of the ACL. Florence, Italy: Association for Computational Linguistics; 2019, p. 960–6.
[22] Abacha AB, Mrabet Y, Sharp M, Goodwin TR, Shooshan SE, Demner-Fushman D. Bridging the Gap Between Consumers' Medication Questions and Trusted Answers. In: MedInfo. 2019, p. 25–9.
[23] Jin D, Pan E, Oufattole N, Weng W-H, Fang H, Szolovits P. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. Appl Sci 2021;11(14):6421.
[24] Pal A, Umapathi LK, Sankarasubbu M. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In: Flores G, Chen GH, Pollard T, Ho JC, Naumann T, editors. Proceedings of the conference on health, inference, and learning. Proceedings of machine learning research, Vol. 174, PMLR; 2022, p. 248–60.
[25] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 2020;21(1):5485–551.
[26] Zakka C, Shad R, Chaurasia A, Dalal AR, Kim JL, Moor M, et al. Almanac — Retrieval-augmented language models for clinical medicine. NEJM AI 2024;1(2). AIoa2300068.
[27] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Adv Neural Inf Process Syst 2020;33:9459–74.
[28] Robertson S, Zaragoza H. The probabilistic relevance framework: BM25 and beyond. Found Trends Inf Retr 2009;3(4):333–89.
[29] Jin Q, Kim W, Chen Q, Comeau DC, Yeganova L, Wilbur WJ, et al. MedCPT: Contrastive Pre-trained Transformers with large-scale PubMed search logs for zero-shot biomedical information retrieval. Bioinformatics 2023;39(11):btad651.
[30] Lo K, Wang LL, Neumann M, Kinney R, Weld D. S2ORC: The semantic scholar open research corpus. In: Jurafsky D, Chai J, Schluter N, Tetreault J, editors. Proceedings of the ACL. Association for Computational Linguistics; 2020, p. 4969–83.
[31] Han T, Adams LC, Papaioannou J-M, Grundmann P, Oberhauser T, Löser A, et al. MedAlpaca–An Open-Source Collection of Medical Conversational AI Models and Training Data. 2023, arXiv preprint arXiv:2304.08247.
[32] Li Y, Li Z, Zhang K, Dan R, Jiang S, Zhang Y. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. Cureus 2023;15(6).
[33] Yadav P, Tam D, Choshen L, Raffel C, Bansal M. TIES-Merging: Resolving Interference When Merging Models. In: Thirty-seventh conference on neural information processing systems. 2023.
[34] Yu L, Yu B, Yu H, Huang F, Li Y. Language models are super mario: Absorbing abilities from homologous models as a free lunch. 2023, arXiv preprint arXiv:2311.03099.
[35] Shoemake K. Animating rotation with quaternion curves. In: Proceedings of the 12th annual conference on computer graphics and interactive techniques. SIGGRAPH '85, New York, NY, USA: Association for Computing Machinery; 1985, p. 245–54.
[36] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: State-of-the-art natural language processing. In: Liu Q, Schlangen D, editors. Proceedings of EMNLP: System Demonstrations. Association for Computational Linguistics; 2020, p. 38–45.
[37] Cormack GV, Clarke CLA, Buettcher S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. SIGIR '09, New York, NY, USA: Association for Computing Machinery; 2009, p. 758–9.
[38] OpenAI, Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, et al. GPT-4 Technical Report. 2024, arXiv:2303.08774.
[39] Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: Low-Rank Adaptation of Large Language Models. In: International Conference on Learning Representations. 2022.
[40] Dietterich TG. Approximate statistical test for comparing supervised classification learning algorithms. Neural Comput 1998;10(7):1895–923.

B. APPENDIX

Improving faithfulness in Table-to-Text Generation

B.1 Training Procedure

All experiments were conducted on a machine equipped with an NVIDIA TITAN Xp GPU with 12GB of memory. The average training time for models based on Table2Logic was 19 hours, while for the Logic2Text models, the average was 10 hours. Both model types, Table2Logic and Logic2Text, contain a similar number of parameters, approximately 117 million.

B.2 Model hyper-parameters

We maintain the same hyper-parameters for Logic2Text as used by Chen et al. (2020d) and direct readers to their paper for further details. For the Table2Logic model in TlT, which is based on Valuenet from Brunner and Stockinger (2021), we made modifications to the grammar, incorporated additional input data, and adjusted the code to fit our specific use case. The hyper-parameters largely follow those outlined in the original paper, with changes to the base learning rate, beam size, number of epochs, and gradient clipping. Below is the list of hyper-parameters used for Table2Logic in the TlTmodel:

Random seed: 90
Maximum sequence lengthy: 512
Batch size: 8
Epochs: 50
Base learning rate: $5 * 10^{-5}$
Connection learning rate: $1 * 10^{-4}$
Transformer learning rate: $2 * 10^{-5}$
Scheduler gamma: 0.5
ADAM maximum gradient norm: 1.0
Gradient clipping: 0.1
Loss epoch threshold: 50
Sketch loss weight: 1.0
Word embedding size: 300
Size of LSTM hidden states: 300
Attention vector size: 300
Grammar type embedding size: 128
Grammar node embedding size: 128
Column node embedding size: 300
Index node embedding size: 300
Readout: 'identity'
Column attention: 'affine'
Dropout rate: 0.3
Largest index for I nodes: 20
Include OOV token: True
Beam size: 2048
Max decoding steps: 50
False Candidate Rejection: True

B.3 Logical Form grammar

```
Stat ::= only View | and Stat Stat | greater Obj Obj | less Obj Obj | eq Obj Obj |
         str_eq Obj Obj | not_eq Obj Obj | not_str_eq Obj Obj | round_eq Obj Obj |
         all_eq View C Obj | all_str_eq View C Obj | all_not_eq View C Obj |
         all_str_not_eq View C Obj | all_less View C Obj | all_less_eq View C Obj |
         all_greater View C Obj | all_greater_eq View C Obj | most_eq View C Obj |
         most_str_eq View C Obj | most_not_eq View C Obj |
         most_str_not_eq View C Obj | most_less View C Obj | most_less_eq View C Obj |
         most_greater View C Obj | most_greater_eq View C Obj
View ::= all_rows | filter_eq View C Obj | filter_str_eq View C Obj |
         filter_not_eq View C Obj | filter_str_not_eq View C Obj |
         filter_less View C Obj | filter_greater View C Obj | filter_greater_eq View C Obj |
         filter_less_eq View C Obj | filter_all View C
   N ::= count View | avg View C | sum View C | max View C | min View C |
         nth_max View C I | nth_min View C I
 Row ::= argmax View C | argmin View C | nth_argmax View C I | nth_argmin View C I
 Obj ::= str_hop Row C | num_hop Row C | str_hop_first View C |
         num_hop_first View C | diff Obj Obj | N | V
   C ::= column
   I ::= index
   V ::= value
```

B.1 Figure – Logical form grammar, after resolving the ambiguity issues in the original definition (Chen et al., 2020d). We adhere to the same notation used in IRNet and Valuenet. Non-terminals (node types in the graph) are represented by the tokens to the left of ::=, while the possible rules for each node are shown in italics, with pipes (|) separating the different rules. The rules added to the original grammar to address ambiguity issues are marked in green.

B.4 Logic2Text errors

This section provides examples of error cases where the logic-to-text stage of the pipeline failed to generate accurate sentences from a gold logical form (LF). For each error type, we present one example, including the table, caption, gold logical form, and the generated description. For more detailed information, refer to Section 3.3.6.

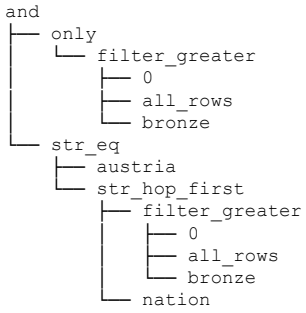
B.4.1 Comparative arithmetic

Caption: fil world luge championships 1961

Table:

rank	nation	gold	silver	bronze	total
1	austria	0	0	3	3
2	italy	1	1	0	2
3	west germany	0	2	0	2
4	poland	1	0	0	1
5	switzerland	1	0	0	1

Logical Form:



TlT **sentence:** austria was the only country to win 0 bronze medals at the fil world luge championships .

Gold sentence: austria was the only country to have bronze medals in the luge championship in 1961 .

B.4.2 LF omission

Caption: geography of moldova

Table:

land formation	area , km square	of which currently forests , km square	% forests	habitat type
northern moldavian hills	4630	476	10.3 %	forest steppe
dniester - rāut ridge	2480	363	14.6 %	forest steppe
middle prut valley	2930	312	10.6 %	forest steppe
bălţi steppe	1920	51	2.7 %	steppe
ciuluc - soloneţ hills	1690	169	10.0 %	forest steppe
corneşti hills (codru)	4740	1300	27.5 %	forest
lower dniester hills	3040	371	12.2 %	forest steppe
lower prut valley	1810	144	8.0 %	forest steppe
tigheci hills	3550	533	15.0 %	forest steppe
bugeac plain	3210	195	6.1 %	steppe
part of podolian plateau	1920	175	9.1 %	forest steppe
part of eurasian steppe	1920	140	7.3 %	steppe

Logical Form:

```
eq
├── 8
└── count
    └── filter_str_eq
        ├── all_rows
        ├── forest steppe
        └── habitat type
```

TlT **sentence:** there are 8 habitats that can be found in moldova .

Gold sentence: 8 land formations are classified with a habitat type of forest steppe .

B.4.3 Verbalization

Caption: seattle supersonics all - time roster

Table:

player	nationality	jersey number (s)	position	years	from
craig ehlo	united states	3	sg	1996 - 1997	washington state
dale ellis	united states	3	sg / sf	1986 - 1991 1997 - 1999	tennessee
pervis ellison	united states	29	c	2000	louisville
francisco elson	netherlands	16	c	2008	california
reggie evans	united states	34 , 30	pf	2002 - 2006	iowa
patrick ewing	united states	33	center	2000 - 2001	georgetown

Logical Form:

```
greater
├── num_hop_first
│   ├── filter_str_eq
│   │   ├── all_rows
│   │   ├── francisco elson
│   │   └── player
│   └── years
└── num_hop_first
    ├── filter_str_eq
    │   ├── all_rows
    │   ├── pervis ellison
    │   └── player
    └── years
```

TlT **sentence:** foulisco elson played for the supersonics after pervis ellison .

Gold sentence: francisco elson played 8 years later thanpervis ellison .

B.5 Examples of faithful TlT sentences where LF is different to gold

This section presents examples of automatic logical forms (LFs) from TlT that produced accurate sentences during manual evaluation, despite differing from their corresponding gold LF references. Each example provides additional details beyond what is shown in Table 3.5.

B.5.1 Similar structure, semantically equivalent

Caption: list of appalachian regional commission counties

Table:

county	population	unemployment rate	market income per capita	poverty rate	status
allegany	49927	5.8 %	16850	15.5 %	- risk
broome	200536	5.0 %	24199	12.8 %	transitional
cattaraugus	83955	5.5 %	21285	13.7 %	transitional
chautauqua	136409	4.9 %	19622	13.8 %	transitional
chemung	91070	5.1 %	22513	13.0 %	transitional
chenango	51401	5.5 %	20896	14.4 %	transitional
cortland	48599	5.7 %	21134	15.5 %	transitional
delaware	48055	4.9 %	21160	12.9 %	transitional
otsego	61676	4.9 %	21819	14.9 %	transitional
schoharie	31582	6.0 %	23145	11.4 %	transitional
schuyler	19224	5.4 %	21042	11.8 %	transitional
steuben	98726	5.6 %	28065	13.2 %	transitional
tioga	51784	4.8 %	24885	8.4 %	transitional

TlT **Logical Form:**

```
str_eq
├── schoharie
└── str_hop
    ├── county
    └── nth_argmax
        ├── 1
        ├── all_rows
        └── unemployment rate
```

Gold Logical Form:

```
str_eq
├── schoharie
└── str_hop
    ├── argmax
    │   ├── all_rows
    │   └── unemployment rate
    └── county
```

TlT **sentence:** in the list of appalachian regional commission counties , schoharie has the highest unemployment rate .

Human sentence: the appalachian county that has the highest unemployment rate is schoharie .

B.5.2 Similar structure, semantically different

Caption: dick rathmann

Table:

year	qual	rank	finish	laps
1950	130.928	17	32	25
1956	144.471	6	5	200
1957	140.780	withdrew	withdrew	withdrew
1958	145.974	1	27	0
1959	144.248	5	20	150
1960	145.543	6	31	42
1961	146.033	8	13	164
1962	147.161	13	24	51
1963	149.130	14	10	200
1964	151.860	17	7	197

TlT **Logical Form:**

```
less
├── num_hop_first
│   ├── filter_str_eq
│   │   ├── 1956
│   │   ├── all_rows
│   │   └── year
│   └── rank
└── num_hop_first
    ├── filter_str_eq
    │   ├── 1959
    │   ├── all_rows
    │   └── year
    └── laps
```

Gold Logical Form:

```
greater
├── num_hop_first
│   ├── filter_str_eq
│   │   ├── 1956
│   │   ├── all_rows
│   │   └── year
│   └── laps
└── num_hop_first
    ├── filter_str_eq
    │   ├── 1959
    │   ├── all_rows
    │   └── year
    └── laps
```


TlT **sentence:** dick rathmann had a lower rank in 1956 than he did in 1959 .

Human sentence: dick rathmann completed more laps in the indianapolis 500 in 1956 than in 1959 .

B.5.3 Different structure, semantically different

Caption: 2005 houston astros season

Table:

date	winning team	score	winning pitcher	losing pitcher	attendance	location
may 20	texas	7 - 3	kenny rogers	brandon backe	38109	arlington
may 21	texas	18 - 3	chris young	ezequiel astacio	35781	arlington
may 22	texas	2 - 0	chan ho park	roy oswalt	40583	arlington
june 24	houston	5 - 2	roy oswalt	ricardo rodriguez	36199	houston
june 25	texas	6 - 5	chris young	brandon backe	41868	houston

TlT **Logical Form:**

```
most_str_eq
├── all_rows
├── arlington
└── location
```

Gold Logical Form:

```
str_eq
├── arlington
└── str_hop
    ├── argmin
    │   ├── all_rows
    │   └── date
    └── location
```

TlT **sentence:** most of the games of the 2005 houston astros ' season were played in the location of arlington .

Human sentence: arlington was the first location used in the 2005 houston astros season .

B.5.4 Simpler, more informative semantic

Caption: 2006 asp world tour

Table:

location	country	event	winner	runner - up
gold coast	australia	roxy pro gold coast	melanie redman - carr (aus)	layne beachley (aus)
tavarua	fiji	roxy pro fiji	melanie redman - carr (aus)	layne beachley (aus)
teahupoo , tahiti	french polynesia	billabong pro tahiti women	melanie redman - carr (aus)	chelsea georgeson (aus)
itacarã	brazil	billabong girls pro	layne beachley (aus)	jessi miley - dyer (aus)
hossegor	france	rip curl pro mademoiselle	chelsea georgeson (aus)	melanie redman - carr (aus)
manly beach	australia	havaianas beachley classic	stephanie gilmore (aus)	layne beachley (aus)
sunset beach , hawaii	united states	roxy pro	melanie bartels (haw)	stephanie gilmore (aus)
honolua bay , hawaii	united states	billabong pro	jessi miley - dyer (aus)	keala kennelly (haw)

TlT **Logical Form:**

```
eq
├── 7
└── count
    └── filter_str_eq
        ├── all_rows
        ├── aus
        └── winner
```

Gold Logical Form:

```
eq
├── 7
└── count
    └── filter_str_eq
        ├── all_rows
        ├── aus
        └── runner - up
```

TlT **sentence:** aus won 7 events in the 2006 asp world tour .

Human sentence: seven of the individuals that were the runner up were from aus .

C. APPENDIX

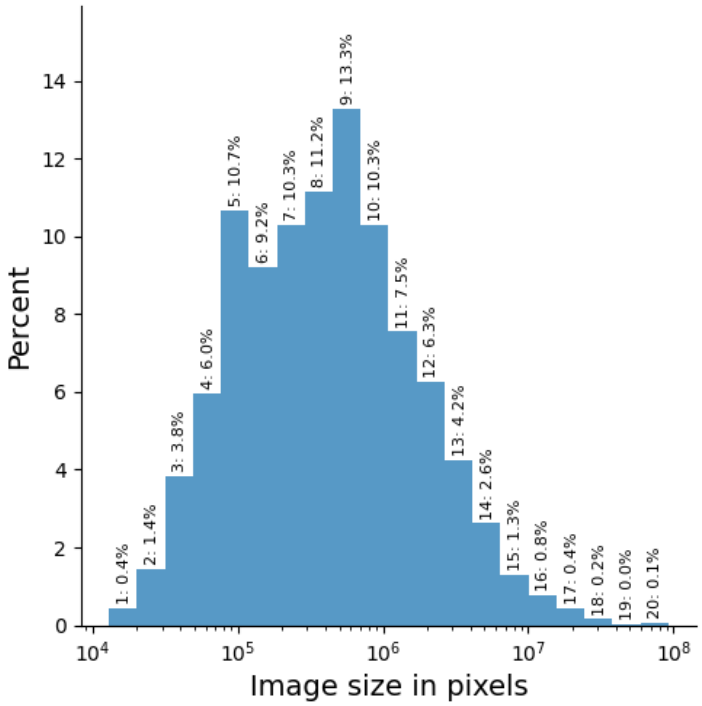
Pixel-based Table-To-Text Generation

C.1 Table Size Distribution in ToTTo

We measured table size based on the total number of pixels in the rendered table images. The size distribution was calculated and tables were grouped into 20 buckets, with each bucket representing a range of table sizes increasing logarithmically. Figure C.1 presents the distribution of ToTTo examples across these buckets for the development set. The performance of generated descriptions within each size group is discussed in Section 4.3.2, and results are displayed in Figure 4.5.

C.2 Table-to-Text Generation Settings

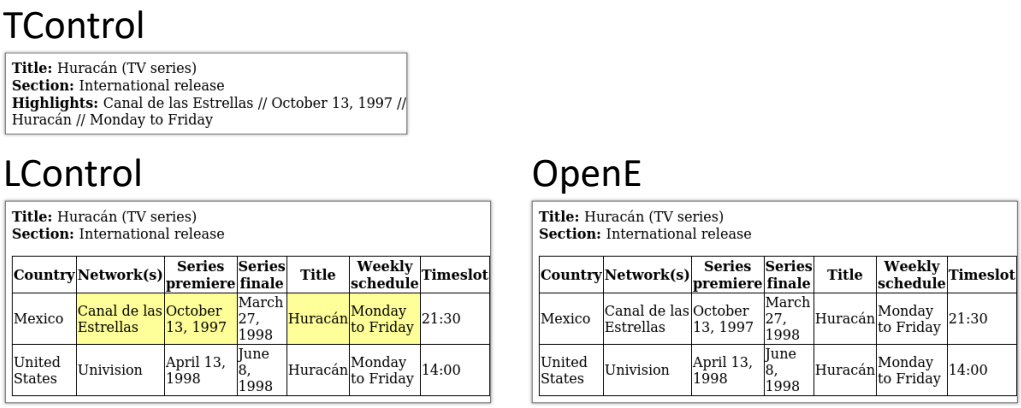
Figure C.2 demonstrates how the input to PixT3 varies depending on the generation setting. These settings include tightly controlled (where only highlighted cells are provided, without the table), loosely controlled (where both the table and highlighted cells are provided), and open-ended (where the entire table is provided without any highlighted content).



C.1 Figure – Distribution of ToTTo examples (development set) by table size (shown on a logarithmic scale).

C.3 Image Truncation and Down-scaling

We analysed how downscaling affects model performance, balancing it against the potential drawbacks of truncation. To do this, we trained PixT3 models on different versions of ToTTo, each with varying downscaling factors γ: 0.87, 0.58, 0.39, 0.26, and 0.00. Note that $\gamma = 0.00$ means no truncation occurred, and only downscaling was applied. Based on the results in Table C.1, a combination of truncation and downscaling yielded the best results, with extreme settings (either no truncation or excessive truncation) proving suboptimal. The optimal downscaling factor was found to be 0.39.



C.2 Figure – Examples of PixT3 input images (and reference) across three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE).

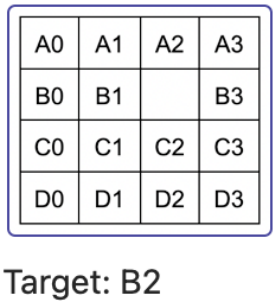
C.4 Intermediate Training

Synthetic Dataset Generation In this section, we provide additional details on how synthetic tables were generated for intermediate training. To ensure that the synthetic tables were similar to those in ToTTo, we first analyzed the probability distributions of columns, rows, column spans, and row spans from the ToTTo training set, minimizing the risk of overfitting and contamination. We observed that the number of columns (up to 20) remained fairly constant across tables, so we simplified the generation process by aggregating row numbers across columns and using a single probability distribution for rows. The number of columns and rows was capped at 20 and 75, respectively. For the content within the cells, we randomly generated combinations of digits (1-5) and characters (A-Z, a-z), yielding a total of approximately 776 million possible unique cell values.

In total, we generated 120,000 tables, each paired with a target pseudo-HTML description. On average, these descriptions were 121 tokens long, with the longest containing 877 tokens. During our experiments, we found that text length was influenced mainly by the number of columns and rows in the table, with larger tables leading to longer target sequences. The target sequence follows a hierarchical structure, with each highlighted cell acting as a container that includes all related

Epoch \ γ	0.00	0.26	0.39	0.57	0.87
16	28.71	29.13	29.47	**29.58**	27.47
17	28.99	29.53	**29.99**	29.70	27.69
18	29.67	30.04	**30.55**	30.21	28.13
19	29.98	30.04	**30.63**	30.54	28.33
20	29.83	30.21	**30.68**	30.53	29.39

C.1 Table – Evaluation results (BLEU scores) for the PixT3 model in the tightly controlled setting across different γ downscaling factors. Results are shown for the last five epochs on the ToTTo training set.



C.3 Figure – Example of a synthetically generated table with a masked cell. Filled cell values indicate their position within the table.

cells from the same rows and columns.

Alternative Objectives We conducted experiments to identify the most effective self-supervised objective for structure learning. In addition to the primary objective described in Section 4.2.4, we tested a masking objective. For this, we generated tables filled with text indicating the position of each cell, then masked random cells and trained the model to predict the missing values (see Figure C.3 for an example). While this approach resulted in faster training, it led to worse performance compared to PixT3. We believe this is because the model only needed to consider nearby cells to predict the masked value, rather than fully understanding the table's structure. We also tried combining the masking objective with the structure learning objective from Section 4.2.4, but the performance was still below that of PixT3.

		Dev Set (All)	Test Set (Non)	Test Set (Over)	Test Set (All)
	Model	BLEURT	BLEURT	BLEURT	BLEURT
TControl	T5-base	0.233	0.106	0.354	0.230
	T5-3B	0.228	0.104	0.344	0.224
	Lattice	0.226	0.103	0.348	0.226
	CoNT	**0.240**	0.116	0.364	0.240
	PixT3	0.178	0.044	0.312	0.178
LControl	T5-base	−0.298	−0.395	−0.191	-0.293
	T5-3B	−0.309	−0.416	−0.194	-0.305
	Lattice	−0.287	−0.382	−0.195	-0.288
	CoNT	−0.293	−0.387	−0.190	-0.289
	PixT3	**0.169**	**0.047**	**0.287**	**0.167**
OpenE	T5-base	−0.371	−0.458	−0.278	-0.368
	T5-3B	−0.385	−0.456	−0.301	-0.378
	Lattice	−0.377	−0.451	−0.302	-0.377
	CoNT	−0.370	−0.452	−0.281	-0.366
	PixT3	**−0.332**	**−0.414**	**−0.258**	**−0.336**

C.2 Table – BLEURT evaluation results for T5, PixT3, Lattice, and CoNT across three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE). The T5 results in the TControl setting are sourced from Kale and Rastogi (2020), and the CoNT results are from An et al. (2022). This table provides additional information to complement the results presented in Table 4.1.

C.5 Additional Results and Examples

Alongside BLEU and PARENT scores reported in Tables 4.1 and 4.2, we also include BLEURT results in Table C.2 and Table C.3. Furthermore, Figure C.4 provides example outputs on the Logic2Text dataset in a zero-shot setting. In the tightly controlled (TControl) setting, CoNT struggles to generate a coherent sentence, while PixT3 produces a faithful but somewhat generic description. In the loosely controlled (LControl) setting, both models have access to the entire table, yet both produce incorrect statements, likely due to the zero-shot nature of the task. In the open-ended (OpenE) setting, PixT3 generates a coherent and accurate sentence, while CoNT introduces an error by mentioning "(+5)" instead of "(-5)," likely due to performance degradation when given the entire table.

	BLEURT		
Model	TControl	LControl	OpenE
LLaVA	−1.230	−1.189	**−1.184**
T5-base	−1.086	−1.147	−1.237
T5-3B	−1.079	−1.167	−1.196
Lattice	**−1.060**	−1.147	−1.231
CoNT	−1.103	−1.159	−1.231
PixT3	−1.104	**−1.073**	−1.213

C.3 Table – Automatic evaluation results on the Logic2Text dataset across three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE). All models, except LLAVA, were fine-tuned on ToTTo and then tested on Logic2Text. This table provides additional metrics to complement the results shown in Table 4.2.

C.6 LLaVA promts

As part of the zero-shot experiments outlined in Section 4.3.1, we compared our models with LLaVA-1.5 (Liu et al., 2023c), a large pretrained multimodal model with 13 billion parameters. The following prompts were used for each generation setting:

TControl "Here are some descriptions based on other highlights of other tables 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description based on the following highlighted cells extracted form a table."

LControl "Here are some descriptions based on the highlights of other tables not present in the input: 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description based on the highlighted cells in this table following the same style as the example descriptions."

Title: 1973 u.s. open (golf)

place	player	country	score	to par
1	gary player	south africa	67 + 70 = 137	- 5
2	jim colbert	united states	70 + 68 = 138	- 4
t3	jack nicklaus	united states	71 + 69 = 140	- 2
t3	johnny miller	united states	71 + 69 = 140	- 2
t3	bob charles	new zealand	71 + 69 = 140	- 2
t6	gene borek	united states	77 + 65 = 142	e
t6	julius boros	united states	73 + 69 = 142	e
t6	tom weiskopf	united states	73 + 69 = 142	e
t6	arnold palmer	united states	71 + 71 = 142	e
t6	lee trevino	united states	70 + 72 = 142	e

- **Reference:** Jim Colbert has the second best number of strokes to par.
- **CoNT (TControl):** Jim Colbert led the 1973 U.S. open (golf course) with a score of to par.
- **PixT3 (TControl):** Jim Colbert took part in the 1973 U.S. open (golf) tournament.
- **CoNT (LControl):** At the 1973 U.S. open (golf), Jim Colbert shot a record of 267 (+1) and finished four strokes ahead of runner-up Lee Janzen.
- **PixT3 (LControl):** Jim Colbert had a score of 142.
- **CoNT (OpenE):** Gary Player scored 137 (+5) and finished five strokes ahead of runner-up Jim Colbert.
- **PixT3 (OpenE):** Gary Player won the 1973 U.S. Open (golf) with a score of 137.

C.4 Figure – Logic2Text table and model output across three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE).

OpenE "Here are some descriptions from other tables not present in the input: 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description stating something from this table following the same style as the example descriptions."

C.7 Human Evaluation Guidelines

We provided detailed instructions to crowdworkers participating in our human evaluation study. Our participants were native English speakers from the UK and the USA, with an equal gender split (50% male, 50% female).

Thank you for taking part in our experiment! You will be presented with a table and a computer-generated description of its content. Your task is to determine whether each description is "True" or "False" based on the information provided in the table and/or its title and subtitle (you will see examples later-on). No expert knowledge is required to perform this task. You should evaluate the descriptions given the information presented in the table, without taking any other information into account (e.g., based on your own knowledge or the web).

Here are some guidelines to help you with your evaluation:

***Acronyms**: tables often have acronyms which the descriptions might spell out. For example, if the table mentions "TD" and the description correctly spells it out as "touch down," you should not consider this "False" (although the description might be false for other reasons).*

***Implicit information**: the description might mention information that can be inferred but is not explicitly spelled-out in the table. For example, it could mention "steam engines" when the table lists theirs names without explicitly talking about steam engines. In this case, the description should not be considered "False".*

- You should evaluate each description independently.

- If the description does not make sense and is impossible to evaluate (usually when summarizing very large tables), you should consider it as "False".

We suggest starting by reading the description and then referring to the table to verify if it aligns with its claims.

This data elicitation study is performed by researchers at [REDACTED]. If you have any questions, feel free to contact [REDACTED]. Participation in this research is voluntary. You have the right to withdraw from the experiment at any time. The collected data will be used for research purposes only. We will not collect any personal information. Your responses will be linked to your anonymous Prolific ID for the exclusive purpose of conducting our experiment.

C.8 PixT3 Fine-tuning Hyper-parameters

PixT3 models for all three generation settings (TControl, LControl, OpenE) were fine-tuned using the same set of hyperparameters. To prevent overfitting, we applied early stopping based on the BLEU score computed on the validation set every 250 steps. Table C.4 outlines the specific hyperparameters used in PixT3, with all other parameters set to the default values from Pix2Struct (Lee et al., 2023).

Hyperparameter	**Value**
Optimizer	AdamW
Learning rate	0.0001
Warm-up steps	1000
Max. input patches	2048
Shuffle train data	False
Epochs	30
Train batch size	8
Gradient accum. steps	32
Mixed precision	fp16
Evaluation batch size	32
Eval freq. steps	250
Inf. beam search	8 beams

C.4 Table – Hyperparameters used in PixT3.

D. APPENDIX

Multimodal Table Understanding

D.1 Table Understanding Pre-Training Objectives

In Table Understanding, a variety of pre-training objectives have been proposed to equip models with the ability to comprehend and manipulate tables. These objectives span a range of tasks, from fundamental operations like masked language modeling and entity linking to more complex tasks such as schema augmentation and table summarization. Each task is designed to target specific aspects of table-based reasoning, enhancing the model's capabilities in relational inference, data population, semantic parsing, and more. By applying these objectives during pre-training, recent models have demonstrated significant improvements in table-specific tasks. The following section details key pre-training objectives used across various state-of-the-art models in table understanding:

- **Masked Language Modeling**: This objective requires the model to predict masked tokens or entire cells within tables or associated text, helping it to understand contextual relationships between cells, columns, and surrounding text. Applications of this objective include: TaPas ((Herzig et al., 2020)), TaBERT ((Yin et al., 2020)), TUTA ((Wang et al., 2021)), OmniTab ((Jiang et al., 2022)), TURL ((Deng et al., 2020)), Table-GPT ((Li et al., 2023b)).
- **SQL Executor Model**: The model is trained to execute SQL operations directly over tables, simulating a SQL execution engine. This task enables

models to interpret and process structured queries, improving their comprehension of table semantics and logical structure. This technique was introduced in TAPEX ((Liu et al., 2022)).

- **Table Entailment**: By classifying statements as either supported or refuted based on table content, this objective builds a model's inferencing capabilities for truth-value assessments in tabular data. Implemented in (Eisenschlos et al., 2020) and TableLlama ((Zhang et al., 2024a)).

- **Masked Column Prediction**: This task involves predicting the names and data types of masked columns, which helps the model develop schema understanding and familiarity with common data type conventions. Used in TaBERT ((Yin et al., 2020)).

- **Entity Linking**: Given a selected cell and a set of entities, that is, information about a certain something or someone, make the model choose the entity that corresponds to the cell. Used in TURL ((Deng et al., 2020)) and TableLlama ((Zhang et al., 2024a)).

- **Relation Extraction**: Given two pair of column names and a set of possible relations (i.e. 'goverment.politician.party') the model needs to choose the correct relation between the columns. Explored in TURL ((Deng et al., 2020)) and TableLlama ((Zhang et al., 2024a)).

- **Row Population**: This objective requires the model to arrange a set of pre-selected cell values within a designated column. Introduced in (Zhang and Balog, 2017) and used by TURL ((Deng et al., 2020)) and TableLlama ((Zhang et al., 2024a)).

- **Schema Augmentation**: The model rearranges shuffled column headers to align with a coherent schema, which builds the model's ability to recognize correct schema configurations. Introduced in (Zhang and Balog, 2017).

- **Cell Value Recovery**: This objective involves using an extra two layer neural network to encode the value of a cell into an embedding and then making the model to recover the value of the cell based on the embedding and the rest of the table data. This technique is used in TaBERT ((Yin et al., 2020)).

- **Cell Type Classification**: Given a predefined cell type taxonomy, the model needs to identify the structural types of cells in the table. Applied in TUTA

((Wang et al., 2021)), HGT ((Jin et al., 2024)), and TabPrompt ((Jin et al., 2023)).

- **Table Type Classification**: This objective involves categorizing tables based on predefined taxonomies ((Crestan and Pantel, 2011)), enhancing the model's comprehension of different table structures and purposes. Found in TUTA ((Wang et al., 2021)), HGT ((Jin et al., 2024)), and TabPrompt ((Jin et al., 2023)).

- **Table Row Classification**: In this task, the model needs to clasify whether a given header is a header row or a data row, supporting the understanding of hierarchical structures within tables. Deployed in HGT ((Jin et al., 2024)).

- **Table Column Classification**: Find the semantic type of a column, from a given list of choices, these choices can either follow a given taxonomy or directly be a estanadrised Wikipedia type. Implemented in TURL ((Deng et al., 2020)), Table-GPT ((Li et al., 2023b)), and TableLlama ((Zhang et al., 2024a)).

- **Cell-level Cloze**: The model predicts masked cell values from multiple-choice options to imrpove its contextual understanding of cell-level information. Applied in TUTA ((Wang et al., 2021)).

- **Table Context Retrieval**: This objective requires the model to retrieve relevant table metadata, such as titles and descriptions, from provided text snippets, improving model's context retrieval and table comprehension capabilities. Found in TUTA ((Wang et al., 2021)).

- **TableQA**: Given a table and a question in natural language, the model must answer based on table content. Works using this objective include WikiTableQuestions ((Pasupat and Liang, 2015)), OmniTab ((Jiang et al., 2022)), Table-GPT ((Li et al., 2023b)), and TableLlama ((Zhang et al., 2024a)).

- **Table Cell Matching**: The model gets the encoding of table cells and a list of shuffled cell text contents. The model needs to pair each cell encoding with its corresponding text in the list. Applied in HGT ((Jin et al., 2024)).

- **Table Context Generation**: Given a table cell, the model needs to generate the context around that table. Found in HGT ((Jin et al., 2024)).

- **Masked Entity Recovery**: Recover masked cells based on surrounding cells and table meta-data. Used in TURL ((Deng et al., 2020)).
- **Highlighted Cells QA**: This task involves answering questions based on highlighted table cells, aligning question-answering with specific data fields. Implemented in FeTaQA ((Nan et al., 2022)) and used in TableLlama ((Zhang et al., 2024a)).
- **Hierarchical Table QA**: Models learn to answer questions over tables with hierarchical structures, such as multi-column or multi-row spans. Used in HiTab ((Cheng et al., 2022)) and TableLlama ((Zhang et al., 2024a)).
- **Table Grounded Dialogue Generation**: Generates conversational responses grounded in table content, supporting table-centric dialog tasks. Used in TableLlama ((Zhang et al., 2024a)).
- **Hybrid Table Context QA**: Given a table and a set of contextual texts linked to the table's entities, the model needs answer a multi-hop question using information from both sources. Introduced by HybridQA ((Chen et al., 2020c)) and applied in TableLlama ((Zhang et al., 2024a)).
- **Highlighted Cells Description**: In this task the model needs to generate a description based on a Wikipedia table and a set of highlighted cell, making the model contextualize specific data fields. Introduced in ToTTo ((Parikh et al., 2020)) and applied in TableLlama ((Zhang et al., 2024a)).
- **Missing-value Identification**: In this task the model needs to identify the position of missing cells within tables. Implemented in Table-GPT ((Li et al., 2023b)).
- **Column-finding**: Identify the column name of a specific value that appears only once in a given table. Used in Table-GPT ((Li et al., 2023b)).
- **Row-to-row Transform**: Given a table and a transformed version without a random missing value (could be transposed or columns merged) make the model infer the transformation and fill in the missing value. Applied in Table-GPT ((Li et al., 2023b)).
- **Entity Matching**: In this objective models need to match rows from different tables that refer to the same real-world entity. Found in Table-GPT ((Li et al., 2023b)).

- **Schema Matching**: Get different rows of the same table, paraphrase the column name of one of them, shuffle, and make the model match them. Used in Table-GPT ((Li et al., 2023b)).
- **Error Detection**: Detect data values in a table that is a likely error from misspelling. Implemented in Table-GPT ((Li et al., 2023b)).
- **List Extraction**: The model reconstructs column separators in unformatted table. Applied in Table-GPT ((Li et al., 2023b)).
- **Head Value Matching**: This objective requires models to pair column headers with their corresponding data values. Found in Table-GPT ((Li et al., 2023b)).
- **Table Semantic Parsing**: Transforms natural language questions into SQL statements for a given table. Used in Table-GPT ((Li et al., 2023b)).
- **Table Summarization**: Generate natural language summaries of table content. Found in Table-GPT ((Li et al., 2023b)).
- **Column Augmentation**: Given a table with masked columns, requires the model to generate them. Applied in Table-GPT ((Li et al., 2023b)).
- **Row Augmentation**: Given a table with masked rows, requires the model to generate them. Used in Table-GPT ((Li et al., 2023b)).
- **Row/Column Swapping**: Prompts the model to execute swap operations on rows or columns and generate the resulting modified table structure. Implemented in Table-GPT ((Li et al., 2023b)).
- **Row/Column Filtering**: Requieres the model to remove columns or rows at a certain index. Applied in Table-GPT ((Li et al., 2023b)).
- **Row/Column Sorting**: Sort rows or columns according to specific criteria. Found in Table-GPT ((Li et al., 2023b)).
- **Table Numerical Reasoning**: Given a table and a mathematical question, the model must answer using mathematical reasoning over table values. Found in TABMWP ((Lu et al., 2023a)), TAT-QA ((Zhu et al., 2021)), and MMTab ((Zheng et al., 2024)). Works like (Liu et al., 2023a) also apply this objective to visually represented tables in MatCha.

- **Financial QA**: The model answers finance-related questions, often with complex financial vocabulary. Found in FINQA ((Chen et al., 2021)).

D.2 Table Retrieval Errors

Distribution of table retrieval errors across seed datasets.

Dataset	Total Errors	NO (%)	Sim (%)	NTF (%)	Other (%)
TURL	138413 (23.9%)	3.0%	19.4%	0.9%	2.7%
ToTTo	16282 (12.0%)	1.1%	9.3%	0.3%	1.2%
TabFact	9345 (55.7%)	1.5%	48.3%	0.5%	5.3%
InfoTabs	1926 (70.8%)	15.5%	41.5%	11.3%	2.5%
HybridQA	1223 (9.9%)	0.8%	6.3%	0.1%	2.7%
WikiTQ	158 (7.5%)	0.0%	4.7%	1.3%	1.6%

D.1 Table – Table retrieval error distribution per seed dataset. **Total Errors**: Total number of tables not obtained and their share of the total number of tables in the seed dataset. **NO**: No Wikipedia article was found. **Sim**: None of the tables in the Wikipedia article were similar enough to the serialized table in the seed dataset. **Other**: Other types of errors.

D.3 Stage 2 Training Hyperparameters

The Stage 2 training of the model mPLUG-DocOwl 1.5 (Hu et al., 2024) was carried out in 4x NVIDIA Hopper H100 64GB GPU over 6,500 steps. We follow the same training hyperparameters as in Hu et al. (2024) with an effective batch size of 256 (batch 8 x 32 GPUs), and a maximum learning rate of 2e-5 after a warm-up of 195 steps followed by a cosine decay.

D.4 HybridQA exact match accuracy

Metrics in Section 5.3.2, Table 5.3 include BLEU4 for FeTaQA and ToTTo, and accuracy for other tasks. However, HybridQA accuracy is calculated based on whether the reference text is present in the generated sequence, rather than exact match to fairly evaluate other model's responses. In this section we show the results on exact match accuracy and accuracy based on the presence of the reference

text within the generated sequence. Notably, exact match accuracy follows a similar trend, further highlighting the advantage of the model trained with our Stage 2 dataset.

Model	Exact match (acc)	Contains (acc)
DocOwl1.5	29.8	35.5
DocOwl1.5 (Ours)	**46.1**	**50.7**
Table-LLava (7B)	0.0	35.6
Table llama	7.5	36.5

D.2 Table – Accuracy results for the HybridQA dataset evaluation, including exact match accuracy and accuracy based on the presence of the reference text within the generated sequence.