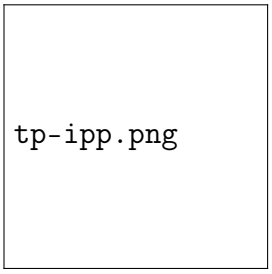

January, 2025

APM_5AI29_TP

Language Models and Structured Data

Project Report

Acronym of the Team: Table Turners
Name: Dmitry Timkin, Ivanina Ivanova, Mark Daychman, Renata Mindiiarova

---

Enhancing Text-to-Table Information Extraction with Updated BART Models

Abstract

In this paper, we revisit the proposed 'text-to-table' information extraction task, which converts unstructured text content into structured tables using sequence-to-sequence models. We explain the core idea of the original paper [?] to use table constraints and relation embeddings to improve accuracy. We test the approach on new datasets and develop an improved preprocessing and an ML-based error correction for an improved accuracy.

Problem Statement

Information extraction (IE) is one of the fundamental tasks in natural language processing. It aims to convert text into more dense and structured knowledge representations, making the information more accessible for further applications. One of the examples of such structured data type are tables. Before this paper was released, IE into table usually required predefined schemas, making it inflexible and domain specific. Accurate text-to-table generation, especially without a predefined schema, has many real-life applications across various domains, including automated structured reports from legal documents, financial statements, and medical records.

The paper formalizes this task as a sequence-to-sequence (seq2seq) problem and proposes a solution based on BART [?], which is known for its strong performance in text generation. The proposed method incorporates additional techniques such as table constraints and table relation embeddings to ensure that the generated tables are correctly structured.

Table constraints are implemented during the encoding process to enforce that the tables have a consistent structure. Specifically, the model determines the number of columns from the first generated header row and ensures that every generated row has the same number of columns.

This prevents incomplete rows or irregular table formats. This method improves the syntactic accuracy of the model's output.

Table relation embeddings help with the table cell alignment by incorporating their relationship with their respective row and column headers. During the table generation, the model uses row and column relation embeddings to identify which row each non-header cell belongs to. These embeddings are added as relational vectors in the self-attention function. This helps improve the table's coherence and accuracy.

On top of that the original authors of the paper correctly identify the greatest weakness of their proposed approach – the model performs worse with larger input texts. To help alleviate this weakness, we introduced 2 new approaches. We modified the preprocessing to also include a small summarization LLM to condense the text, so that the BART core model has a smaller input. We also added an extra validation step using a graph neural network (GNN) to ensure that the generated tables are correct.

Finally, we were also curious to see how the text-to-table approach generalizes with new data. We wanted to not only recreate the original results, but also use the preprocessing tools available in the repository to generate new data to test the model. Since, the reverse problem, i.e. 'table-to-text' has been fairly well-studied; one can use any existing table database, generate the textual summaries using one of the table-to-text models and use it as the input for the 'text-to-table' model. The accuracy can then be measured by comparing the original and reproduced tables.

Overall, in this paper we aim to address the following key questions:

- Does the original approach generalize well? Does it work equally well with technical jargon within areas like medicine and law?
- Does introducing an additional summarization preprocessing step boost the model's accuracy?
- Can we further improve the accuracy of the model by adding a validator?

Method/Overall Architecture

- We utilized a neural network-based summarization model to condense lengthy reports into concise and informative summaries. The model used is a DistilBART-CNN-12-6, which effectively reduces the length of reports while retaining critical information. In addition to summarizing, we integrated Named Entity Recognition (NER) to identify key entities such as players and teams within the text. To further enhance the quality of the data, we leveraged the Wikipedia API to fill in missing context or background knowledge, ensuring that all necessary details about teams and players were accurately captured.
- In addition to the first approach, we also experimented with an alternative method to further enhance the quality of the data and ensure it aligns with structured table generation. Specifically, we applied a multistep process to enhance the quality of the data and ensure that it aligns with structured table generation. First, we used sentence ranking to prioritize the most relevant sentences, ensuring that only the most important content was retained.

We also applied contrastive filtering to remove redundant or near-duplicate sentences, reducing noise in the dataset. To further refine the data, we used a Graph Neural Network (GNN) for attribute selection, which helped identify and retain the most relevant attributes for table generation, improving the overall accuracy and relevance of the output.

Example: Table Filtering for WikiTableText

Input Table (Before Filtering)

```
"table":
  - "title": "1978 Federation Cup (Tennis)"
  - "subtitle": "Qualifying Round"
  - "date": "19 August"
  - "winning team": "Philippines"
  - "score": "3-0"
  - "losing team": "Thailand"
```

Text: "Philippines won Thailand with 3–0 during 1978 Federation Cup."
Filtered Data:

```
[('subtitle', 'qualifying round'),
 ('winning team', 'Philippines'),
 ('score', '3-0'),
 ('losing team', 'Thailand')]
```

What Changed?

- The "title" and "date" were removed because they were not explicitly mentioned in the text.
- The "winning team", "score", and "losing team" were kept because they were directly referenced in the text.
- The "subtitle" was maintained for structural context, even though it was not directly mentioned in the text.

This process illustrates how sentence ranking, contrastive filtering, and GNN-based attribute selection help extract the most relevant information for structured table generation, ensuring a high level of accuracy and relevance in the resulting output.

Experimentation

Note: We faced several challenges running the original code from the paper due to its reliance on outdated Python versions and deprecated modules. The installation required extensive manual version tuning, use of an older pip version, and the manual cloning of certain packages from GitHub repositories to resolve compatibility issues.

Used Datasets

The original paper uses the following 4 datasets that we have all also included in our testing:

- **Rotowire** is a sports domain dataset containing basketball game reports. Each instance consists of a long text report and two tables representing team and player scores. This dataset is challenging due to the long-form text that includes irrelevant information, making information extraction difficult.
- **E2E** is a restaurant domain dataset where each instance is a short text description of a restaurant paired with an automatically constructed table summarizing its characteristics. It has a limited set of table texts, resulting in low diversity, which makes generalization difficult.
- **WikiTableText** is an open-domain dataset where each instance consists of a short text description and a table with row headers collected from Wikipedia. It captures structured information with a balance between textual descriptions and tabular data.
- **WikiBio** is extracted from Wikipedia biography pages. Each instance contains a biography introduction and a table from the infobox of the corresponding Wikipedia page. The text is significantly longer than the table and contains more information, making it useful for evaluating models that process rich textual content.

On top of that, we also introduce two new datasets:

- **Dataset 1** The Reported Financials dataset from Finnhub provides clean, comprehensive financial data sourced directly from SEC filings between 2010 and 2020. This dataset includes a wide range of financial information, such as income statements, balance sheets, and cash flow statements for various companies. We also trained a model on the Reported Financials dataset to extract relevant financial metrics such as revenue, profit, and assets from unstructured text and convert them into a structured tabular format.
- **Dataset 2** The MIMIC-III dataset is a large, open-access collection of anonymized clinical data from over 61,000 critical care admissions at a Boston teaching hospital, covering the period from 2001 to 2012. It includes 47 features such as demographics, vital signs, and lab test results, specifically for sepsis patients who meet the sepsis-3 definition criteria.

Accuracy Metric

To assess the accuracy of the generated tables, the paper employs precision, recall, and the F1 score. These metrics are applied to both headers and non-header cells to measure their correctness. Precision (P) is defined as the fraction of correctly predicted results among all predicted results:

$$P = \frac{1}{|y|} \sum_{y \in y} \max_{y^* \in y^*} O(y, y^*)$$

where $O(\cdot)$ denotes the similarity between a predicted and ground-truth value. Recall (R) measures the fraction of correct predictions relative to the total ground-truth entries:

$$R = \frac{1}{|y^*|} \sum_{y^* \in y^*} \max_{y \in y} O(y, y^*)$$

The F1 score is the harmonic mean of precision and recall:

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

For similarity computation $O(\cdot)$, three approaches are considered:

- **Exact Match:** Checks if the predicted text exactly matches the ground-truth text.
- **chrf Score:** Computes character-level n-gram similarity between predicted and ground-truth values.
- **BERTScore:** Measures similarity using contextual embeddings from BERT.

Non-header cells are evaluated using both their content and associated headers. This ensures that the cell belongs to the correct row and column. If the header text is slightly different but semantically equivalent, similarity is computed using the chrf or BERTScore measures rather than an exact match. Empty cells are ignored during evaluation as they do not contain useful information. The summary of the results can be found in Figure

Dataset	**Model**	**Row Header F1**			**Non-header Cell F1**		
		Exact	ChrF	BERT	Exact	ChrF	BERT
E2E	Original	91.23	92.40	95.34	90.80	90.97	92.20
	Summarization	99.62	99.69	99.88	97.87	97.99	98.56
	Summ. + Valid.	**99.63**	**99.69**	**99.88**	**97.88**	**98.00**	**98.57**
WikiTableText	Original	59.72	70.98	94.36	52.23	59.62	73.40
	Summarization	78.15	84.00	95.60	**59.26**	**69.12**	80.69
	Summ. + Valid.	**78.16**	**83.96**	**95.68**	59.14	68.95	**80.74**
WikiBio	Original	63.99	71.19	81.03	56.51	62.52	61.95
	Summarization	**80.53**	**84.98**	**92.61**	68.98	77.16	76.54
	Summ. + Valid.	80.52	84.96	92.60	**69.02**	**77.16**	**76.56**
Rotowire	Original	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX
	Summarization	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX
	Summ. + Valid.	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**
New Dataset 1	Original	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX
	Summarization	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX
	Summ. + Valid.	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**
New Dataset 2	Original	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX
	Summarization	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX	XX.XX
	Summ. + Valid.	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**	**XX.XX**

Table 1: Evaluation results on four existing and two newly introduced datasets.

Discussion

Comparison with expectations, limitations, lessons learned, and perspectives.

Further work

The original study evaluates the performance of BART-base and BART-large [**?**] from 2020, which were state of the art at the time. We think the results will significantly improve, because newer and larger models (e.g. T5 [**?**], Flan-T5 [**?**], ModernBERT [**?**]) are specifically optimized for long-text generation. Because of all the extra tokens that are needed to properly encode a table using text, the output is usually fairly long, so long-text generation models are expected to perform better.

We are planning to try more modern versions of the Bart model, but since the article code is implemented in the library fairseq, we encountered problems with implementing the new architecture into the code.

References