Map&Make: Schema Guided Text to Table Generation

*Naman Ahuja [1] *Fenil Bardoliya[1] Chitta Baral[1] †Vivek Gupta[1]

[1]Arizona State University

{nahuja11, fbardoli, chitta, vgupt140}@asu.edu

Abstract

Transforming dense, unstructured text into interpretable tables—commonly referred to as Text-to-Table generation—is a key task in information extraction. Existing methods often overlook what complex information to extract and how to infer it from text. We present Map&Make, a versatile approach that decomposes text into atomic propositions to infer latent schemas, which are then used to generate tables capturing both qualitative nuances and quantitative facts. We evaluate our method on three challenging datasets: Rotowire, known for its complex, multi-table schema; Livesum which requires numerical aggregation; and Wiki40 which require open text extraction from mulitple domains. By correcting hallucination errors in Rotowire, we also provide a cleaner benchmark. Our method shows significant gains in both accuracy and interpretability across comprehensive comparative and referenceless metrics. Finally, ablation studies highlight the key factors driving performance and validate the utility of our approach in structured summarization. Code and data are available at: https://coral-lab-asu.github.io/map-make.

1 Introduction

Driven by the exceptional success of Large Language Models (LLMs) in tasks such as question answering (Chen et al., 2020; Zhu et al., 2024), text summarization (Wiseman et al., 2017; Wang et al., 2020a), and text data mining (Li et al., 2023a; Sui et al., 2024), recent works have shifted to explore the structured summarization of text. Reading lengthy texts is time-consuming, making it challenging to extract key information efficiently. Tables, as a widely used structured format, can organize data in a clear and interpretable manner, facilitating information retrieval and comprehension.

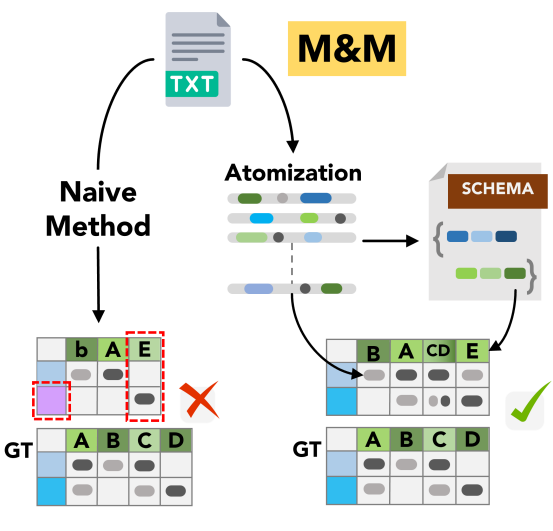


Figure 1: Comparison between Naive methods and our method for Text-to-Table generation.

Early works (Wu et al., 2022; Li et al., 2023b) treat text-to-table generation as an information extraction problem, viewing the table and its corresponding text representation as essentially similar. The pioneering work by (Deng et al., 2024) extends beyond simple extraction, integrating and categorizing information in complex scenarios. However, to the best of our knowledge, all existing studies perform this task under a predefined schema, where information about the schema is either provided during fine-tuning (Li et al., 2023b; Wu et al., 2022) or included in prompts for few-shot methods (Deng et al., 2024; Tang et al., 2024). This reliance on predefined schemas limits adaptability to open-domain text, where tabular structures may need to be inferred dynamically.

For more generalized systems, a hybrid approach is needed: leveraging schema induction techniques to recognize patterns in structured representations while retaining the flexibility to handle novel text narratives. This ensures robustness in managing

*These authors contributed equally to this work and share first authorship.

†This author supervised the research and serves as the corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30249–30262

both expected and unseen table structures. Nevertheless, tabular summarization of dense information without any prior knowledge of the underlying structure remains largely unexplored.

Regarding methodology, there are contrasting findings on the effect of fine-tuning LLMs versus zero-shot or few-shot prompting strategies across different datasets. Extensive benchmarks indicate that LLMs often perform suboptimally in zero-shot settings, missing relevant information and introducing unattested data or hallucinations (Deng et al., 2024). More sophisticated prompting techniques have been proposed to mitigate these issues (Wei et al., 2022; Khot et al., 2022). In-context learning (Brown et al., 2020), combined with chain-of-thought (CoT) prompting (Wei et al., 2022) offers some improvements but can overfit to the provided shots (Perez et al., 2021), reducing effectiveness on unseen data. Some studies (Tang et al., 2024; Sundar et al., 2024) report performance gains from fine-tuning, whereas others (Deng et al., 2024) observe no or negative improvements for table generation tasks.

Hence, there is a need for a generalized, schema-agnostic tabular summarization framework that can effectively *"mine"* the underlying structure from free-form text and produce comprehensive information under a given instruction. In this work, we aim to address the aforementioned challenges with the following contributions:

1. We introduce a generalized notion of Structured Summarization, evaluating LLMs' capabilities on planning table schemas in Zero-Shot and One-Shot settings to summarize information under a given instruction exhaustively.

2. We propose a generalizable framework, *Map&Make (M&M)*, applicable to any Instruction-Driven Tabular Summarization task extending beyond simple extraction tasks.

3. We present a manually corrected version of the Rotowire Benchmark, a popular text-to-table benchmark. This correction addresses hallucination and fidelity issues in previous versions, ensuring a fair evaluation setting for our methods.

4. We conduct comprehensive experiments on multiple closed-source and open-source state-of-the-art LLMs on a diverse suite of metrics encompassing both table quality and information coverage. Results show that M&M outperforms existing methods and exhibits robust generalization across different Tabular Summarization paradigms. Additionally, M&M maintains stable performance on large text corpora, where other methods often degrade.

2 Map&Make Framework

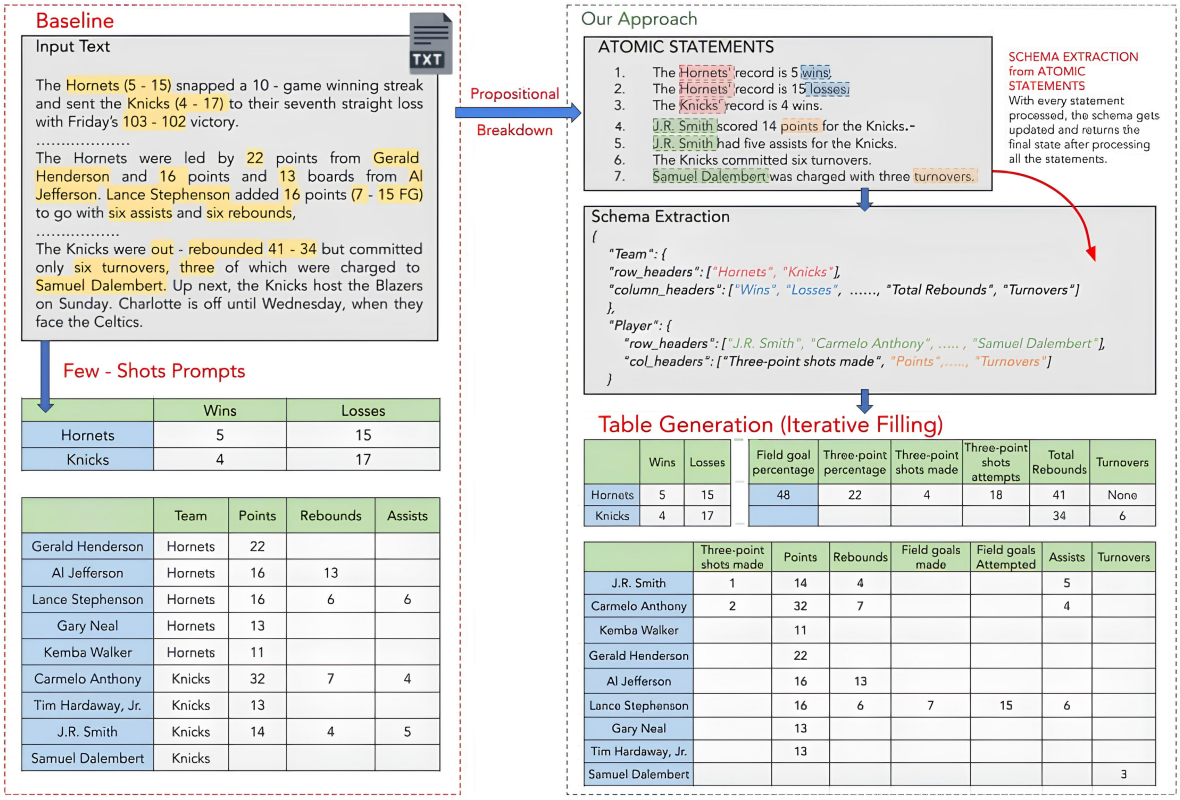
We propose a three-staged methodology applicable to any instruction-driven summarization task. Figure 2 illustrates our approach. Mirroring how humans construct tables—identifying key information, planning the layout, and filling in values—we employ a multi-agent prompting framework to enhance coverage and correctness.

2.1 Propositional Atomization

Propositional Atomization is the process of extracting atomic, self-contained facts from contextually embedded sentences. The benefits of this approach have been demonstrated in large-scale datasets and practical applications, such as in natural language inference and summary hallucination detection.(Chen et al., 2022) (Maynez et al., 2020). We design a prompt to transform input statements into **atomic statements** adhering to five key properties: **Well-formedness**, ensuring grammatical correctness and adherence to linguistic conventions; **Atomicity**, maintaining the smallest meaningful semantic unit; **Self-containedness (Decontextualization)**, making each proposition independently understandable without requiring external context; **Support**, ensuring all propositions are explicitly derived from the input text; and **Comprehensiveness**, collectively covering all latent claims and facts from the original text. This targeted parsing resolves ambiguous entity attributions and prevents the conflation of overlapping or nested entities. The exact prompt is given in Appendix **??**.

2.2 Schema Extraction

This part builds on propositional segmentation to extract tabular layouts. Directly planning schema results in incomplete coverage for inputs that require big tables to represent information. Hence, we iteratively build the table schema. The table schema is initialized as an empty list of row and column headers for each table. These lists are iteratively populated based on each atomic statement processed (as shown in Figure 2. Entities are mapped to row headers while their attributes



	Wins	Losses
Hornets	5	15
Knicks	4	17

	Team	Points	Rebounds	Assists
Gerald Henderson	Hornets	22		
Al Jefferson	Hornets	16	13	
Lance Stephenson	Hornets	16	6	6
Gary Neal	Hornets	13		
Kemba Walker	Hornets	11		
Carmelo Anthony	Knicks	32	7	4
Tim Hardaway, Jr.	Knicks	13		
J.R. Smith	Knicks	14	4	5
Samuel Dalembert	Knicks			

	Wins	Losses	Field goal percentage	Three-point percentage	Three-point shots made	Three-point shots attempts	Total Rebounds	Turnovers
Hornets	5	15	48	22	4	18	41	None
Knicks	4	17					34	6

	Three-point shots made	Points	Rebounds	Field goals made	Field goals Attempted	Assists	Turnovers
J.R. Smith	1	14	4			5	
Carmelo Anthony	2	32	7			4	
Kemba Walker		11					
Gerald Henderson		22					
Al Jefferson		16	13				
Lance Stephenson		16	6	7	15	6	
Gary Neal		13					
Tim Hardaway, Jr.		13					
Samuel Dalembert							3

Figure 2: An illustration of our approach. Propositional Breakdown segments the text to generate atomic statements b. Schema Extraction extracts the table structures to generate table schemas. c. Table Generation iteratively fills tables based on the atomic statements.

are added to column headers. This iterative approach allows headers to evolve dynamically, enabling adaptation to the changing structure of incoming data, and improving coverage. The exact prompt is given in Appendix **??**.

2.3 Table Generation

In the final step, we perform table-filling on empty tables defined by the extracted schema in the previous step. We again, process every statement iteratively to update or fill cell values as per the summarization instruction. For example, for counting events from live text as in the LIVESUM benchmark, cell values keep increasing with every statement processed. Contrastively, for static text such as in Rotowire, cell values are fixed once. We explicitly instruct LLMs to return statement-wise updates for every cell value to ensure transparency and avoid hallucinations. Refer Appendix **??** for the exact prompt.

3 Benchmarks

Pre-LLM methods for text-to-table generation (Wu et al., 2022), (Li et al., 2023b), (Pietruszka et al., 2024) have utilized four predominant benchmarks; Rotowire (Wiseman et al., 2017), E2E (Novikova et al., 2017), Wikibio (Lebret et al., 2016), WikiTableText (Bao et al., 2018). Initially proposed for **table-to-text** generation tasks, a majority of these benchmarks lack the structural complexity required to comprehensively evaluate state-of-the-art LLMs for information coverage in tabular summarization. In fact, E2E, WikiBio, and WikiTableText consist of only single tables with only two columns for every sample. Hence, we evaluate our framework on the repurposed version of the Rotowire proposed by (Wu et al., 2022), Wiki40B (Jain et al., 2024), and the recently released benchmark, Livesum (Deng et al., 2024).

Rotowire comprises dense post-match summaries of NBA games (2014–2017), where the task is to extract performance statistics to generate Player and Team Tables. In contrast, Livesum consists of live football commentary[†], requiring the aggregation of various events into team summary tables. The dataset also assigns a difficulty level: easy, medium, or hard to each column based on varying descriptions of events that influence

[†]https://www.bbc.com/sport/football

how challenging inferring the right value is. Each dataset presents distinct challenges: Rotowire's long text and multi-table structure make schema and sparse value extraction difficult, while Livesum demands identifying and aggregating events described in varying forms across the text.

While Rotowire and Livesum provide strong benchmarks within the sports domain, we further evaluate the generalization capability of Map&Make on Wiki40B—a large-scale, open-domain, multilingual corpus of cleaned and normalized Wikipedia articles spanning 40 languages. This dataset poses unique challenges for text-to-table generation due to its diverse topical coverage (e.g., historical revolutions to astrophysics), unstructured content, and varied narrative styles. Inspired by Jain et al. (2024), we sample 500 English articles containing at least 30 numerical values and 3+ sentences, ensuring sufficient complexity and coverage. Notably, Wiki40B also demands multi-table generation, making it a comprehensive testbed for evaluating the robustness and adaptability of text-to-table frameworks. Additional details on the benchmark datasets are provided in Appendix A.

3.1 Corrections and Fidelity Issues:

In our initial explorations, we observed several discrepancies between the text and the ground truth tables in the data presented by (Wu et al., 2022). These discrepancies propagate on other benchmarks developed on this data (Tang et al., 2024) inhibiting fair and reliable evaluations of developed frameworks. We suspect these challenges stem from LLM-driven attribution methods, and existing problems in validating information (Adewumi et al., 2024), (Yue et al., 2023),(Patel et al., 2024). Thus the authors manually validate every sample in the test set. Keeping ground truth text the same, we attribute every row, column, and cell value to the text and update tables.

Our findings are reported in Table 1, where we compare our corrected version with the Original (Wu et al., 2022), and STRUCBENCH (Tang et al., 2024) benchmark which is another version of Rotowire. More details about the correction strategy can be found in Appendix A.2.1.

Table	*Cell*		*Row*		*Col*	
	H	**MI**	**H**	**MI**	**H**	**MI**
Original to Strucbench						
Team	1219	1271	8	8	627	626
Player	1390	1270	68	62	135	129
Original to Corrected						
Team	613	1137	21	50	329	528
Player	7310	1752	85	82	1000	188
Strucbench to Corrected						
Team	721	1247	21	50	385	585
Player	8104	2666	140	143	1077	271

Table 1: Total counts of corrected rows, columns, and cells across error types for Rotowire. **H** denotes Hallucination, and **MI** represents Missing Information. Here, a row or a column is flagged as hallucinated/missing if it contains at least one erroneous entry.

4 Experimental Setup

We describe the different prompting techniques, LLMs, and evaluation metrics employed in this work.

4.1 Models and Baselines

We experiment with M&M on both closed-source and open-source LLMs. Among proprietary models, we use Gemini 2.0 Flash experimental and GPT-4o (Hurst et al., 2024) (*gpt-4o-2024-08-06*), maintaining consistent configurations for *temperature*, *top_p*, and *top_k* across experiments. Additionally, we assess performance on open-source models such as Llama 3.3-70B Instruct. [†].

For baselines, we employ various prompting strategies. **Zero-Shot CoT** (Zhang et al., 2022) directly organizes and populates tables by leveraging step-by-step reasoning, while **One-Shot CoT** (Wei et al., 2022) enhances this by incorporating a single example in the prompt. We also compare against **Text-Tuple-Table (T^3)** (Deng et al., 2024), which extracts structured tuples (*subject-object-verb* or *subject-attribute-value*) before table generation. Originally schema-dependent, we adapt T^3 into a schema-agnostic setting, alongside its unified variant **T^3D**, which integrates tuple extraction and table construction in a single prompt.

To evaluate the efficacy of our framework, we experiment with two variants of our approach; **M&M - 3S**, performs each task sequentially, where as **M&M - U**; a unified variant of our approach, performs all the tasks in a single LLM call. We face several challenges on testing M&M on the Livesum dataset on smaller models due to the large input sizes, resulting in inefficient iterative table filling and same statements being duplicated multiple times. This aligns with the findings of the benchmark authors (Deng et al., 2024), where they face similar challenges in building prompting pipelines

[†]https://github.com/meta-llama/

for this dataset over smaller models. Moreover, the authors also show the no/negative improvements from fine-tuning on this dataset. Hence, we only experiment with prompting-based approaches for this problem.

4.2 Evaluation Metrics

Due to the high variance in table structures, where the same information can be presented in different formats, we utilize a diverse set of both LLM-based and Non-LLM-based metrics to ensure a comprehensive evaluation of correctness and competeness.

String Similarity Metrics: We utilize Exact Match (EM), CHRF (Popović, 2015), and BERTScore (Zhang* et al., 2020) as proposed by (Wu et al., 2022) for reference-based table evaluation. To measure information coverage, each (row, column, cell) tuple from the ground truth table is mapped to its most similar counterpart in the generated table, assigning the highest similarity score to that tuple. Rows Headers and Column Headers are evaluated in a similar fashion. The final EM, CHRF, and BERT scores are averaged over all tuples, as reported in Table 2. For Livesum, where the correctness of generated outputs is determined by the model counting the correct number of occurrences of an event, i.e numbers, metrics like CHRF and BERTScore are unsuitable. Instead, we use Root Mean Squared Error (RMSE) defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2}{n}}$$

where n denotes the total number of cells, and y_i and $\tilde{y}_i$ represent the content of the cell at index i in the ground truth table and the generated table, respectively. For 2D tables, the index i is determined by flattening the table into a 1D sequence. Additionaly, for exact cell value matching, we report Error Rate (ER %) as a percentage of erroneous cells.

Specialized Metrics: Evaluating table cells (or tuples) using similarity-based metrics independently without considering contextual information from the neighbouring cells can lead to incorrect penalization of good tables, or incorrect rewarding of bad tables. TabEval (Ramu et al., 2024) addresses these challenges by calculating table similarity by *unrolling* tables into a list of atomic statements and computes the entailment between statements generated from output tables and their ground truths. Additionally, we also utilize Auto-QA, an refernceless eval metric, as an information coverage metric (Jain et al., 2024) defined as:

$$\text{Cov}(\mathcal{T}) = \frac{\sum_{i=1}^{|G(\mathcal{S})|} E_{(q_i,a_i)}\left[Q(\{\mathcal{T}_j\}, q_i)\right]}{|G(\mathcal{S})|} \quad (1)$$

where $G(\mathcal{S})$ represents a set of Question-Answer pairs (q_i, a_i) generated by an LLM from the input text $\mathcal{S}$. The function $Q(\{\mathcal{T}_j\}, q)$ denotes the LLM's response to question q, derived from a set of generated tables $\{\mathcal{T}_j\}$. The term $E_{(q,a)}[x]$ evaluates whether the LLM's response x aligns with the reference answer a for the given question q. We report accuracy as the percentage of correctly answered questions based on the available tables.

5 Results and Discussion

Method	*EM*			*CHRF*			*BERT*		
	Cell	Row	Col	Cell	Row	Col	Cell	Row	Col
GPT-4o Zero-Shot									
CoT	**20.75**	**51.58**	30.13	39.62	**81.46**	49.05	38.25	62.31	59.59
T^3 - D	18.39	51.32	29.01	37.13	81.24	46.37	36.44	62.35	58.25
M&M - U	19.95	50.43	**30.39**	**41.99**	78.97	**53.12**	**42.90**	**62.57**	**64.82**
Gemini-2.0 Zero-Shot									
CoT	**21.90**	51.85	29.20	36.65	80.69	44.23	38.58	63.78	55.76
T^3 - D	20.77	**52.92**	22.83	32.38	**81.90**	38.80	37.54	**64.65**	54.05
M&M - U	19.76	48.05	**31.51**	**42.61**	75.00	**53.19**	**41.22**	60.63	**60.65**
Llama-3.3 70B Zero-Shot									
CoT	**21.44**	51.42	39.37	40.79	80.59	51.40	36.70	**63.45**	55.73
T^3 - D	21.05	51.57	**39.40**	**42.04**	**80.77**	**53.36**	**37.31**	62.93	**57.49**
M&M - U	19.26	**51.64**	27.52	35.63	79.86	45.45	36.95	63.28	55.30
GPT-4o One-Shot									
CoT	33.30	88.08	35.96	45.42	91.98	50.60	57.11	91.18	60.28
T^3 - D	34.53	85.98	38.36	47.96	91.35	53.31	57.83	89.79	61.90
T^3	17.92	53.92	34.43	40.94	82.69	52.25	39.21	64.49	61.77
M&M - U	**51.20**	**90.87**	**55.25**	**68.27**	**93.61**	**73.38**	**74.39**	**95.24**	**77.88**
M&M - 3S	35.24	64.06	52.72	61.89	85.06	72.49	57.90	73.30	76.80
Gemini-2.0 One-Shot									
CoT	40.65	**91.57**	43.21	55.19	**93.92**	60.10	64.60	**95.53**	66.22
T^3 - D	38.89	87.84	42.91	55.19	92.90	60.70	63.31	92.54	67.18
T^3	28.61	65.89	40.21	53.38	85.94	62.05	51.91	74.19	69.09
M&M - U	56.44	90.77	60.82	71.38	93.50	76.34	76.12	94.85	78.37
M&M - 3S	**61.78**	89.49	**67.37**	**76.12**	92.92	**81.57**	**79.48**	93.86	**82.69**
Llama-3.3 70B One-Shot									
CoT	39.57	91.82	41.52	53.08	93.77	57.41	62.69	94.61	64.09
T^3 - D	38.02	89.38	41.61	50.44	91.79	56.87	61.10	92.09	63.23
T^3	33.82	85.41	38.84	49.75	89.95	57.17	59.04	89.52	64.54
M&M - U	**44.27**	**92.78**	**47.27**	**60.54**	**94.85**	**65.62**	**66.96**	**95.77**	**70.25**
M&M - 3S	41.01	72.02	44.94	54.47	74.39	58.87	58.15	74.99	61.42

Table 2: Performance comparison of different methods on Rotowire across models using various string similarity metrics

Our experiments demonstrate Map&Make outperforms other prompting strategies improving information coverage and correctness significantly across all models and evaluation metrics, showing excellent generalization capabilities.

5.1 Performance on Rotowire

From Table 2, we observe that M&M outperforms other baselines such as CoT, T^3 consistently in the

Method	TabEval						Auto-QA
	Correctness		*Completeness*		*Overall*		*Accuracy*
	Team	Player	Team	Player	Team	Player	
GPT-4o Zero-Shot							
CoT	57.32	62.78	39.62	79.92	42.66	67.64	70.13
T^3 - D	**58.60**	**73.85**	**42.44**	79.37	**45.36**	**73.66**	66.69
M&M - U	43.74	68.40	41.74	**85.96**	38.27	73.20	**76.95**
Gemini-2.0 Zero-Shot							
CoT	**73.41**	66.94	36.27	**81.35**	**44.72**	**71.39**	60.52
T^3 - D	55.82	63.70	31.57	80.80	36.65	68.62	61.94
M&M - U	58.01	**68.93**	**44.19**	77.94	46.53	68.72	**64.63**
Llama-3.3 70B Zero-Shot							
CoT	**67.63**	70.57	30.66	13.35	38.09	16.21	54.24
T^3 - D	63.60	**74.88**	**33.43**	**34.23**	**40.56**	**33.14**	52.41
M&M - U	61.15	72.32	27.78	14.61	33.55	17.88	**59.17**
GPT-4o One-Shot							
CoT	80.23	86.74	56.26	57.74	64.34	65.57	41.42
T^3 - D	80.24	**87.56**	56.12	64.34	64.09	70.84	38.08
T^3	**81.65**	85.39	46.43	60.66	56.05	66.61	39.46
M&M - U	66.28	86.16	**77.65**	**88.15**	**69.56**	**85.85**	69.51
M&M - 3S	48.78	74.22	61.43	87.34	51.01	78.56	**80.38**
Gemini-2.0 One-Shot							
CoT	**82.91**	**91.51**	63.50	75.89	69.72	81.06	60.52
T^3 - D	81.86	91.08	65.18	76.27	70.01	80.92	47.94
T^3	75.70	87.22	61.75	82.07	64.93	82.68	56.58
M&M - U	75.27	91.17	77.47	92.92	**74.80**	**91.36**	64.88
M&M - 3S	65.89	83.54	**79.49**	**92.43**	69.95	86.71	**71.21**
Llama-3.3 70B One-Shot							
CoT	**80.93**	80.41	59.33	**47.79**	66.33	**47.29**	54.20
T^3 - D	69.54	**80.86**	48.57	45.32	55.21	45.92	46.57
T^3	65.31	70.66	64.38	36.38	61.13	33.03	53.49
M&M - U	74.33	76.47	**71.87**	43.34	**70.57**	39.53	55.96
M&M - 3S	47.92	57.12	53.59	34.04	48.21	31.34	**64.61**

Table 3: Performance comparison using TabEval and AutoQA on Rotowire across strategies on various models.

cell- and column-level metrics. In a zero-shot setting, M&M improves coverage by up to 32% (from CHRF) at the column level and 32% at the cell level. In a one-shot setting, M&M improves by about 29% at the cell level and by 27% at the column level. We see minimal to no improvement in the Row-level metrics across all metrics for both zero-shot and one-shot settings. The row headers in all the tables consist of proper nouns (Player Names, Team Names), and LLMs perform well in extracting named entities (Villena et al., 2024). At the table level, we observe up to 42% improvement in TabEval scores (Table 3) for the Team tables and 22% for the Player table. A bigger improvement is seen in the Team Tables as the global set of column headers for team tables is larger, introducing more variety in the team schemas. Also, we observe a decrease in the Tab-Eval Correctness. This is discussed in detail in the Appendix B.1, as the Ground Truth tables in the RotoWire dataset consist only of statistical values, and M&M being a loosely supervised method also extracts non-statistical columns (such as Injury Status and Average Player Performance, etc.), attributes not present in the Ground Truth, leading to decreased correctness scores. We validate the precision of our generated tables using the Auto-QA metric and observe an improvement of up to 15% in coverage from CoT baselines.

Method	Easy		Medium		Hard		Average	
	RMSE	ER	RMSE	ER	RMSE	ER	RMSE	ER
GPT-4o Zero-Shot								
CoT	**0**	**0**	1.55	47.71	1.84	79.84	1.55	48.14
T^{3}D	0.05	1.82	1.93	53.66	2.76	84.76	2.04	50.10
M&M - U	0.03	1.92	**0.80**	**26.56**	**1.50**	**55.19**	**0.97**	**24.63**
Gemini-2.0 Zero-Shot								
CoT	**0.05**	**2.51**	2.00	60.78	2.82	87	2.09	53.92
T^{3}D	0.05	2.82	3.45	60.20	4.39	93.26	3.34	54.44
M&M - U	0.08	3.82	**0.87**	**31.31**	**2.63**	**80.31**	**1.47**	**35.00**
GPT-4o One-Shot								
CoT	0.05	2.62	2.05	55.30	1.73	76.86	1.61	45.34
T^{3}D	0.08	**0.43**	1.46	46.62	2.73	84.73	1.75	44.91
T^3	0.23	16.5	1.48	34.16	2.39	51.91	1.76	35.62
M&M - U	**0.03**	1.85	0.81	31.06	1.96	68.10	1.19	32.94
M&M - 3S	0.07	3.44	**0.63**	**25.48**	**0.93**	**32.34**	**0.71**	**21.77**
Gemini-2.0 One-Shot								
CoT	0.01	0.63	1.69	58.31	2.23	82.80	1.68	48.32
T^{3}D	**0.01**	**0.54**	1.21	49.63	2.34	85.12	1.49	46.44
T^3	0.19	9.26	0.93	26.32	2.51	54.72	1.59	29.10
M&M - U	0.04	1.45	0.89	36.64	1.98	69.93	1.21	34.65
M&M - 3S	0.10	4.58	**0.60**	**23.72**	**0.89**	**33.13**	**0.70**	**21.37**

Table 4: Performance comparison using string similarity metrics across different categories showing Error Rates (in %) and RMSE scores of GPT-4o and Gemini-2.0-flash-exp for Livesum

5.2 Performance on Livesum

Table 4 compares the difficulty wise and overall performance of M&M with other techniques. M&M significantly enhances performance by reducing overall error rate by upto 35% and RMSE by upto 29% in zero shot setting, and by upto 55% in error and 57% in RMSE in one-shot setting. CoT also outperforms T^3-D in most experiments, signifying lack of generalization across a schema-agnostic setting. However we see consistent improvement across T^3 from CoT baselines, as it uses code-generation to integrate, or count the events before generating the final table. On comparing performance of zero-shot - vs one-shot methods, we see our methods consistently improve performance across all methods and models. This finding is consistent across both benchmarks, hence showcasing emergent capabilities as an adaptable one-shot framework.

5.3 Performance on Wiki40B

In the table 5, we report AutoQA scores on Wiki40B for different methods. Due to the absence of gold-standard tables, we adopt AutoQA, a reference-less evaluation metric that tests whether the generated table can accurately answer factoid questions derived from the original article.

Map&Make achieves a 14% improvement over CoT and a 9% gain over T^3, demonstrating its generalization capacity to open-domain, non-synthetic text. Moreover, as compared to generating a fixed number of tables, we observe our framework

dynamically adapts to multiple table generation (Range 1-13; Mean: 6.02; Std: 4.33) to improve comprehension.

Method	AutoQA-Score
Chain of Thoughts	63.68
Text-Tuple-Table	67.07
Map & Make (3-step)	**76.40**

Table 5: AutoQA scores using GPT-4o in zero-shot settings.

These results suggest that our modular framework, especially the atomization and iterative schema construction, scales effectively to structurally diverse open-domain text, making it a generalised and interpreted framework.

5.4 Performance Across LLMs:

In the Zero-Shot setting, GPT-4o and Gemini 2.0 Flash Exp demonstrate largely comparable performances across most metrics, with only minor variations. The largest observed difference is a 4.8-point delta in Column CHRF score for CoT prompting, where GPT slightly outperforms Gemini, and a 6-point delta in Column CHRF for M&M prompting, favoring Gemini. In the One-Shot setting, Gemini exhibits a slight edge, with improvements of up to 4.8 in CHRF Column score for CoT prompting, and 2.1 in Cell-level coverage for M&M Multistep. The overall TabEval score difference remains minimal, with a delta of just 0.2 in favor of Gemini. These results highlight the robustness and adaptability of our framework, maintaining stable and consistent performance across diverse datasets and evaluation metrics.

6 Discussion and Analysis

6.1 Error Analysis

To qualitatively analyze and pinpoint exactly where information loss occurs in table generation, we employ a table-transformation agent to transform CoT and M&M generated outputs to exactly match the Row and Column headers of the Ground Truth tables. The agent first maps the given rows and columns of the inputs with the Ground Truth schema and subsequently populates the table. Rows and Columns that occur in the model outputs but are not present in the ground truth are reported separately (extra information) and vice versa (missing information). This agent facilitates a more granular analysis of each table giving insights into what information a strategy it's misses.

	CoT	M&M
Extra Information		
Team Rows	0.05	0
Player Rows	0.41	0.06
Team Column	0	0.09
Player Columns	0.07	1.93
Missing Information		
Team Rows	0.019	0
Player Rows	0.11	0.04
Team Columns	1.04	0
Player Columns	2.6	0.21

Table 6: Average Per-Table Error Counts Across Rows and Columns for Rotowire

Rotowire: As shown in Table 6. Row-wise coverage improves from the CoT baseline when employing Map&Make reducing missing rows from 0.05 to 0 per table for Teams, and from 0.41 to 0.06 per table for players. We also observe a substantial improvement in Column level coverage. CoT baselines miss out on covering columns both in Team and Player tables, averaging 1.04 and 2.6 columns per table respectively. Map&Make, in contrast, misses out on 0 and 0.21 columns in the Team and Player tables respectively. Map&Make also produces extra columns for every table capturing the non-statistical aspects of entities, which are absent in the Ground Truths.

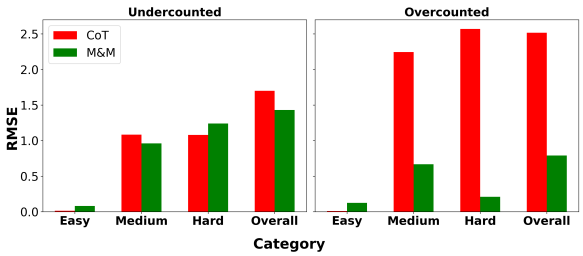


Figure 3: RMSE of Overcounting and Undercounting Instances for Livesum. Uncercounted refers to cell values less than the ground truth, Overcounted refers to cell values more than the ground truths.

Livesum: We further segregate instances of overcounting and undercounting of events separately to analyze where LLMs miss out on information (undercounting) and hallucinate (overcount). Figure 3 shows the segregated RMSE scores across both instances. We observe over 67% of the total errors for CoT are hallucinations. These findings align with other research the demonstrate LLM's struggle with counting tasks.(Ball et al., 2024);(Zhang et al., 2024). Contrastively, our methodology shows stable performance across all difficulty levels with overcounting RMSE decreasing by a big margin.

6.2 Analysis on Efficiency and Table Planning

Figure 4 describes the schemas coverage of CoT and our method's extracted schemas. We see a

very high correlation between missing columns and Ground Truth tables for CoT, showcasing poor performance as schema size increases. We show stable performance across larger table sizes for both player and team tables.

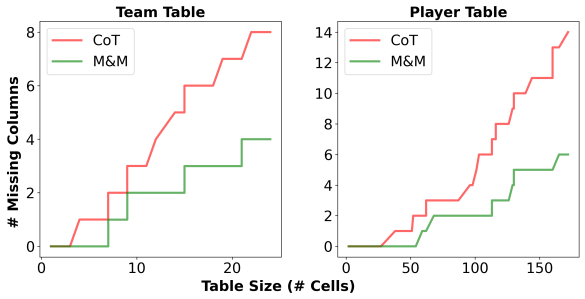


Figure 4: Comparison of Schema-Coverage with Increasing Table Sizes for Rotowire.

The facets of information coverage and accurate extraction in structured summarization tasks are challenging due to the high variability of linguistic structures, implicit relationships, and contextual dependencies present in unstructured text. These complexities, including syntactic ambiguities, coreference resolution challenges, and implicit entity attributions, make it difficult for models to reliably infer and represent structured information while preserving the fidelity of the original content. Rotowire tests the capabilities of models in planning exhaustive Table schemas from highly contexualised narratives where as Livesum tests exacting occurrences of different events. M&M prompting shows significant improvements across both datasets and hence, is a more reliable and exhaustive table generation strategy.

6.3 Ablation Study

M&M	Correctness		Completeness		Overall	
	Team	Player	Team	Player	Team	Player
Unified						
No Ablation	**75.27**	**91.17**	**77.47**	**92.92**	**74.80**	**91.36**
- Atomization	72.91	82.80	77.37	85.61	73.28	82.42
- Iterative						
Schema	69.74	80.45	70.75	86.42	68.50	81.73
Table	70.58	83.38	69.18	89.05	67.55	84.79
3-Step						
No Ablation	65.89	83.54	79.49	92.43	69.95	**86.71**
- Atomization	**73.43**	**84.39**	77.62	88.52	**73.41**	84.87
- Iterative						
Schema	69.48	80.72	**67.99**	**90.63**	65.85	83.91
Table	63.98	77.39	**80.26**	**93.69**	68.84	83.12

Table 7: Ablations on Rotowire.

To understand different components our method, we conducted some additional studies, where we focused on removing a key step from our method, showcased in Table 7 & 8.

Rotowire We observed that removing Atomization step leads to a drop in the Completeness of both the Team and the Player table across both Unified and 3-Step variation. This demonstrates that breaking down multi-entity complex statements is essential for information coverage. Ablating the Iterative Schema Generation step consistently degrades the performance, which illustrates that it is a crucial step in our method, as this is effective for tabular structure and enables better table filling as manifested. Furthermore, removing Iterative Table Generation step, we observe a significant drop in the Correctness and Overall metrics, however it is to be noted that there is slight increase in the Completeness for 3-Step. This suggests that without iterative updates there is less information captured and introduces redundancy.

M&M	Easy		Medium		Hard		Average	
	RMSE	ER	RMSE	ER	RMSE	ER	RMSE	ER
Unified								
No Ablation	0.08	3.84	0.87	31.34	2.63	80.29	1.47	35.00
- Atomization	0.05	2.73	0.53	18.58	1.15	44.37	0.75	21.06
- Iterative	0.02	0.93	1.52	54.33	3.41	86.20	2.08	48.94
3-Step								
No Ablation	0.1	4.54	0.6	23.74	0.89	33.14	0.70	21.37
- Atomization	0.05	2.90	0.64	20.36	1.59	47.89	1.01	22.96
- Iterative	0.08	3.73	1.47	37.44	2.81	65.67	1.85	36.07

Table 8: Ablations on Livesum

Livesum We perform the following ablations of our 3-step (M&M - 3S) and unified (M&M - U) approach. Since the schema for every table is the same for every test sample, we don't ablate the iterative structure step for this study. We see a significant decrease in performance without the iterative table-filling step across both Unified and 3-step settings. This underscores the importance of iterative table filling as models can more accurately track dynamic updates without hallucinating or missing out on events. Interestingly, we see a small performance increase after removing the unified variant's atomization step. This is because in a one-shot setting, removing the atomization part boils the problem down to a fixed schema updation.

6.4 Can Fine-Tuning Smaller Models Help?

We investigate whether our modular Map&Make framework can enable effective fine-tuning of smaller language models for structured generation. Using our pipeline, we applied **Gemini 2.0** to the ROTOWIRE training set to generate high-quality table-text supervision, which we used to fine-tune a **LLaMA 3 8B Instruct** model. Evaluation on the ROTOWIRE test set was conducted using AUTOQA, which assesses QA alignment, and TABEVAL (Recall), which measures table re-

construction. We compared the fine-tuned model to a Chain-of-Thought (CoT) baseline and our M&M (3-step) framework. Results are shown in Table 9.

Method	AutoQA	TabEval (Recall)	
		Team	Player
Fine-tuned LLaMA-3 8B	48.65	41.61	61.58
Chain-of-Thought (CoT)	41.42	56.26	57.74
M&M (3-step)	**80.38**	**61.43**	**87.34**

Table 9: Performance on AutoQA and TabEval metrics using Gemini 2.0-generated QA supervision.

ine-tuning with Gemini-generated supervision yields a clear gain in AutoQA (48.65 vs. 41.42) and a modest improvement in player-level recall (61.58 vs. 57.74) over CoT prompting. However, performance drops on team-level recall (41.61 vs. 56.26), indicating challenges in aggregating broader content. The full M&M framework consistently outperforms both alternatives across all metrics. While challenges such as formatting consistency and long-context reasoning persist, Map&Make remains effective both as an inference-time solution and as a scalable data generation engine for schema-agnostic table generation in low-resource and multilingual settings.

7 Related Works

With the advent of Large Language Models (LLMs), numerous methods for information extraction and summarization have been developed. For extraction tasks that involve structured data, two fields have been researched; Table-to-Text and Text-to-Table have been researched, with the latter being less explored.

Table-to-Text Prior works have been referred to as data-to-text in a more general sense. Former works in table-to-text traditionally uses sequence-to-sequence (seq2seq) models to generate table descriptions (Lebret et al., 2016), (Wiseman et al., 2017), (Liu et al., 2018), (Wang et al., 2020b). Since there exists a bidirectional relationship between structural data and unstructured text, the inverse task text-to-table, has also started gained attention.

Text-to-Table Recent studies investigates diverse methods to transform unstructured text into structured tables while addressing the underlying issues of schema inference, handling complex data relations, and knowledge integration. A key contribution to this area has been covered by (Wu et al., 2022), who introduced the approach of information extraction. They developed a seq2seq model with a fine-tuned version of BART for text-to-table translation using the Rotowire dataset. Additionally, studies like (Li et al., 2023b); (Sundar et al., 2024) have explored more sophisticated methods recognising challenges of interpreting 2-d structures as a linearized sequence object. While these methods outperforms traditional methods that employ relation extraction (Zheng et al., 2017), (Zeng et al., 2018), (Luan et al., 2019), (Zhong and Chen, 2021) and named entity recognition (NER) (Huang et al., 2015), (Ma and Hovy, 2016), (Lample et al., 2016), (Devlin et al., 2019), there has been less exploration of this task on leveraging the emergent capabilities of LLMs in developing a generalisable methodology. We introduce a schema-agnostic approach that can dynamically infer table structures from open-domain text. Our method aims to provide a more flexible and comprehensive solution to the text-to-table generation task.

8 Conclusion

We introduce Map&Make, a versatile approach to extract complex information from unstructured text. M&M tackles the challenges by breaking down the text into its atomic statements, this decomposition then helps extract the latent schema, which ultimately aids in populating the tables. M&M's core strength lies in its ability to dynamically infer schema from open-domain language, as well as its ability to handle complex information such as numerical aggregation and detailed qualitative descriptions. Furthermore, our corrections of the Rotowire benchmark provides a more reliable and fairer evaluation platform for future research in this domain. Our meticulous evaluation on Rotowire, Livesum, and Wiki40B showcases improvements in information coverage and localization. M&M mitigates information loss and maintains stable performance even with a larger input context length.

Future work can explore richer evaluation methodologies, including human-centered assessments, to better capture the nuanced utility and faithfulness of the generated tables. Another exciting direction is extending Map&Make to support structurally complex tables, such as those with hierarchical headers or merged cells (Cheng et al., 2021), which would further expand the applicability of schema-guided text-to-table generation in diverse domains.

Limitations

While Map&Make demonstrates significant advancements in schema-guided text-to-table generation, it is important to acknowledge certain limitations.

Reliance on LLMs M&M's performance is tied to the text generation capabilities of the LLMs, the quality of atomization, schema extraction, and table generation/filling is directly proportional to the LLM's reasoning abilities. This leads to several dependencies such as handling highly ambiguous text, grammatical errors, and contradictory information.

Computational Cost While the iterative nature helps in information coverage as seen in Table 7 and Table 3, but becomes computationally expensive due to lengthy outputs, such as those in Livesum. We explore fine-tuning smaller models like Llama 8B to reduce computational overhead, and we find that, while it improves performance, there is a significant gap between smaller fine-tuned models and larger SoTA LLMs like GPT-40, Gemini-2.0-flash.

Ethics Statement

The authors ensure that this work meets the highest ethical standards in research and publication. We have carefully addressed ethical considerations to guarantee appropriate behavior and fair use of computational linguistics methods. Our assertions are consistent with experimental data. While some stochasticity is predicted with black-box Large Language Models, we minimize it by keeping a constant temperature, top_p, top_k. Furthermore, the use of LLMs such as GPT-4o, Gemini, and Llama in this study adheres to their policies of usage. We have used AI assistants (Grammarly and ChatGPT) to address the grammatical errors and rephrase the sentences. Finally, to the best of our knowledge, we believe that this work introduces no additional risk.

Acknowledgments

We thank the Complex Data Analysis and Reasoning Lab at Arizona State University for computational support. We thank the anonymous reviewers for their helpful feedback. Lastly, we thank our lab cat, Coco, for ensuring our professor stayed just the right amount of insane during deadlines.

References

Tosin Adewumi, Nudrat Habib, Lama Alkhaled, and Elisa Barney. 2024. On the limitations of large language models (llms): False attribution. *arXiv preprint arXiv:2404.04631*.

Thomas Ball, Shuo Chen, and Cormac Herley. 2024. Can we count on llms? the fixed-effect fallacy and claims of gpt-4 capabilities. *arXiv preprint arXiv:2409.07638*.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2022. Propsegment: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Annual Meeting of the Association for Computational Linguistics*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*.

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

Zheye Deng, Chunkit Chan, Weiqi Wang, Yuxi Sun, Wei Fan, Tianshi Zheng, Yauwai Yim, and Yangqiu Song. 2024. Text-Tuple-Table: Towards Information Integration in Text-to-Table Generation via Global Tuple Extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9300–9322, Miami, Florida, USA. Association for Computational Linguistics.

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.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Parag Jain, Andreea Marzoca, and Francesco Piccinno. 2024. STRUCTSUM generation for faster text comprehension. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7876–7896, Bangkok, Thailand. Association for Computational Linguistics.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

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.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023a. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263*.

Tong Li, Zhihao Wang, Liangying Shao, Xuling Zheng, Xiaoli Wang, and Jinsong Su. 2023b. A sequence-to-sequence&set model for text-to-table generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5358–5370.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. 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 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. 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.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Nilay Patel, Shivashankar Subramanian, Siddhant Garg, Pratyay Banerjee, and Amita Misra. 2024. Towards improved multi-source attribution for long-form answer generation. 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 3906–3919, Mexico City, Mexico. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems*.

Michał Pietruszka, Michał Turski, Łukasz Borchmann, Tomasz Dwojak, Gabriela Nowakowska, Karolina Szyndler, Dawid Jurkiewicz, and Łukasz Garncarek. 2024. Stable: Table generation framework for encoder-decoder models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2454–2472.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Pritika Ramu, Aparna Garimella, and Sambaran Bandyopadhyay. 2024. Is this a bad table? a closer look at the evaluation of table generation from text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22206–22216, Miami, Florida, USA. Association for Computational Linguistics.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Anirudh Sundar, Christopher Richardson, and Larry Heck. 2024. gtbls: Generating tables from text by conditional question answering. *arXiv preprint arXiv:2403.14457*.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34, Mexico City, Mexico. Association for Computational Linguistics.

Fabián Villena, Luis Miranda, and Claudio Aracena. 2024. llmner:(zerol few)-shot named entity recognition, exploiting the power of large language models. *arXiv preprint arXiv:2406.04528*.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Douglas Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, et al. 2020a. Cord-19: The covid-19 open research dataset. *ArXiv*.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020b. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*.

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.

Xueqing Wu, Jiacheng Zhang, and Hang Li. 2022. Text-to-Table: A New Way of Information Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2518–2533, Dublin, Ireland. Association for Computational Linguistics.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *arXiv preprint arXiv:2305.06311*.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xiang Zhang, Juntai Cao, and Chenyu You. 2024. Counting ability of large language models and impact of tokenization. *arXiv preprint arXiv:2410.19730*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Fengbin Zhu, Ziyang Liu, Fuli Feng, Chao Wang, Moxin Li, and Tat Seng Chua. 2024. Tat-llm: A specialized language model for discrete reasoning over financial tabular and textual data. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 310–318.

Appendices

A Datasets

In this section, we present more details about the datasets used in this study.

A.1 Livesum

This benchmark was created by scraping live commentory of football games played in the English Premier League. Player Names and Team Names were anonymised and summary tables were annotated manually. The difficulty level of each event (columns in a table) is determined by the number of different descriptions that are present for each event, as shown in figure 5. Goals and Red Cards are classified as easy due to their rare occurrences and straightforward descriptions. Shots and Fouls are classified as Hard as they have the most varying descriptions. The remaining four events are classified as Medium difficulty.

Figure 5: Eight types of event information (inner circle) that require summarization in Livesum dataset, along with their common expressions (outer circle) in the commentary. (Deng et al., 2024)

A.2 Rotowire

The Rotowire dataset was originally introduced by (Wiseman et al., 2017) for table-to-text generation, comprising NBA game summaries paired with detailed box and line scores. It was later repurposed by (Wu et al., 2022) for text-to-table generation by filtering the original tables to retain only records that could be grounded in the textual summaries. For each sample summary, a team-level and a player-level table were constructed. The benchmark consists of 3,398 training, 727 validation, and 728 test samples. During our experiments, we observed multiple table entries with no grounding in the corresponding text, which we corrected to have a clean and reliable benchmark.

A.2.1 Dataset Corrections

The match summaries often contain information about player and team performances outside the scope of the match in question. For example, the recent performance trends of teams, upcoming fixtures, players on impressive streaks, etc. To distill relevant information, we operate the instruction to keep only *statistical values of Teams and Players pertinent to the match in question.*
First, we extract the global column headers from the complete test set for the Team and Player tables. We further add two column headers to the global set "Points in the Paint" and "Half-Time Score". The former represents the points made by the team through 2-point shots, and the latter represents the score of both teams at half-time, hence relevant to the team performance summary. Row Headers require no correction at the global level as they contain Player Names and Team Names, hence are validated for every sample individually.

Every table follows the same structure with Row Headers as the teams (Home Team, Away Team) and 8 columns, one for each event. As our problem statement involves generating summary tables without any schematic constraints, we find extra events such as **Passes, Assists, Through Balls** that are contextually relevant but cannot be evaluated using the ground truth. Hence, we employ a paraphrasing agent to transform our outputs as per the ground truth's structure. Full prompt used for paraphrasing can be found in appendix **??**

B Additional Results

B.1 Rotowire

Method	*EM*			*CHRF*			*BERT*		
	Cell	Row	Col	Cell	Row	Col	Cell	Row	Col
					GPT-4o Zero-Shot				
CoT	18.52	47.28	27.89	34.10	77.60	46.06	34.92	57.19	66.58
M&M - U	15.07	46.43	21.73	29.92	74.59	41.48	34.34	57.54	59.56
					GPT-4o One-Shot				
CoT	56.62	88.44	61.20	76.16	92.36	79.50	83.01	91.41	87.20
M&M - U	45.41	88.39	50.38	61.24	91.38	68.77	68.52	93.37	74.50
M&M - 3S	23.30	58.33	34.69	40.17	79.36	53.71	43.05	67.41	60.87
					Gemini-2.0 Zero-Shot				
CoT	33.56	21.90	51.85	37.71	76.49	49.09	40.46	58.64	79.93
M&M - U	16.85	45.41	28.03	36.39	71.97	48.98	36.91	56.66	61.78
					Gemini-2.0 One-Shot				
CoT	56.27	91.79	59.85	75.56	94.13	79.10	84.18	95.95	85.90
M&M - U	57.00	90.32	61.61	71.77	93.20	77.64	77.90	94.76	80.38
M&M - 3S	49.90	86.11	55.95	62.49	89.82	71.86	69.86	91.52	74.65

Table 10: Rotowire Correctness Scores

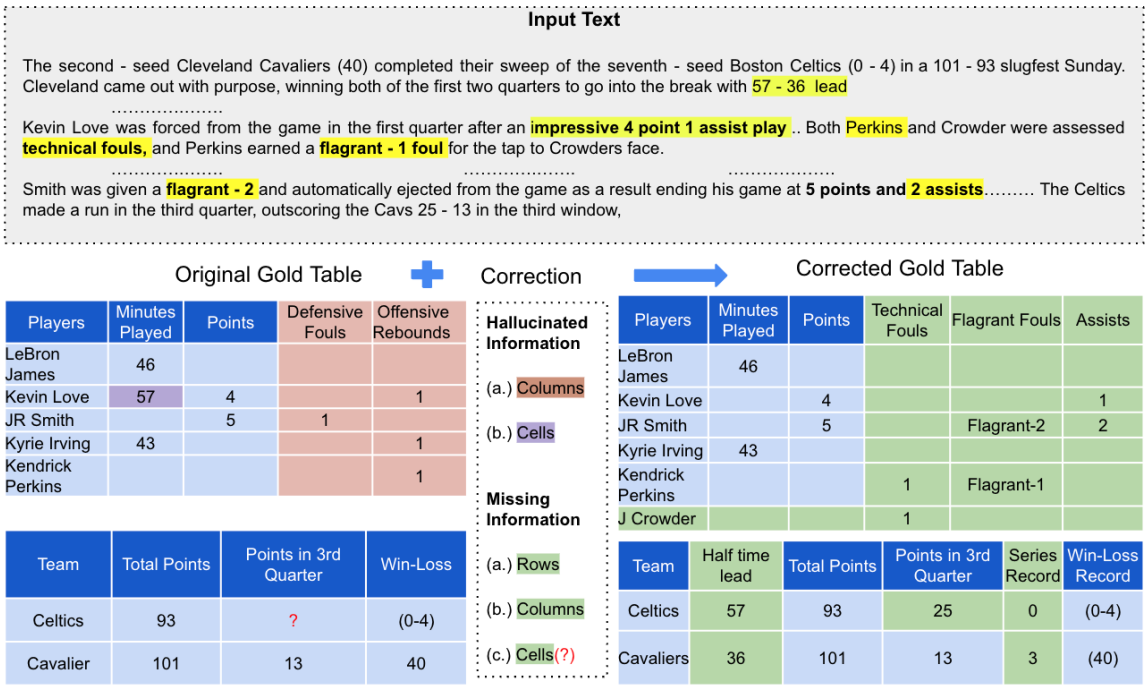


Figure 6: Instances of hallucinations and missing information in Rotowire.

Method	*EM*			*CHRF*			*BERT*		
	Cell	Row	Col	Cell	Row	Col	Cell	Row	Col
GPT-4o Zero-Shot									
CoT	18.93	48.71	28.36	35.56	78.90	46.87	35.49	58.92	61.74
M&M - U	16.61	47.79	24.69	33.72	76.16	45.87	37.20	59.25	61.29
GPT-4o One-Shot									
CoT	38.93	87.99	42.22	53.22	91.95	58.93	65.69	91.09	69.47
M&M - U	46.85	89.05	51.71	63.08	92.01	70.21	70.33	93.92	75.47
M&M - 3S	26.66	60.28	40.20	46.60	81.37	60.41	48.05	69.48	66.81
Gemini-2.0 Zero-Shot									
CoT	20.72	49.02	30.30	35.68	77.91	45.55	38.33	60.34	64.07
M&M - U	17.71	46.09	28.91	38.00	72.82	50.24	37.97	57.81	60.42
Gemini-2.0 One-Shot									
CoT	45.65	91.36	48.71	61.78	93.75	66.89	71.66	95.59	73.46
M&M - U	56.03	90.24	60.63	70.75	93.08	76.56	76.44	94.66	78.89
M&M - 3S	53.63	87.14	59.89	66.88	90.82	75.61	73.32	92.30	77.74

Table 11: Rotowire Overall Scores

In Table 10 and 11, we provide the Correctness (Precision) and Overall Scores (F1) for Rotowire on GPT-4o, and Gemini-2.0-flash-exp. We see minimal to no improvement in correctness scores across similarity metrics due to the lack of any statistical attributes in the ground-truths. Map & Make, being an unsupervised designed to enhance coverage of information produces descriptive columns such as **Injury Status**, categorical columns such as **Starting 5 player/ Off the Bench Player** (used to describe whether a player is present in the game from the start of the match or is substituted mid-game). Also, columns that represent attributes not directly relevant to the game such as **Average Points Scored past 3 games**, **Win Streak** have no reference in ground truths which leads to penalisation of our methodology. However, to validate the correctness of the extra columns generated in our tables we employ AutoQA (shown in Table 3). M&M's consistent improvement on this metric validates the faithfulness of extra information to the gold text.

C M&M Sample Outputs

The sample outputs of all the prompts in our framework (on Gemini-2.0-flash) can be found here: https://github.com/coral-lab-asu/map-make/tree/main/code/sample-outputs

D Prompts

All the prompts used for baselines and Map&Make can be found here:https://github.com/coral-lab-asu/map-make/tree/main/code/prompts. These contains prompts for:

- Chain of Thoughts
- Text Tuple Table (Merged and 3-step)
- Map&Make (Propositional Atomization, Schema Extraction, Iterative Table Filling, and Unified) -0.5em.