Tree-of-Text: A Tree-based Prompting Framework for Table-to-Text Generation in Sports Game Reports

Speaker: Shang-Hsuan Chiang

Advisor: Wen-Chih Peng

Date: 2025/07/21

國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Outline

1. Introduction
2. Related Work
3. Problem
4. Solution
5. Experiment
6. Conclusion
7. Q&A

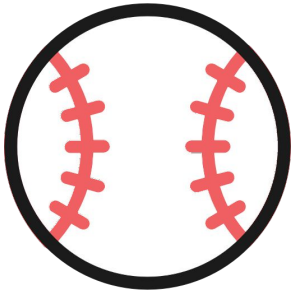
1. Introduction

Introduction

There are many kinds of sports games.

Badminton

Basketball



Baseball

Even more!

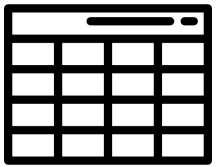
Introduction

How to watch a sports game?

Video

Easy to understand 👍

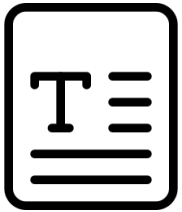
Time-consuming 👎



Table

Time-saving 👍

Hard to understand 👎



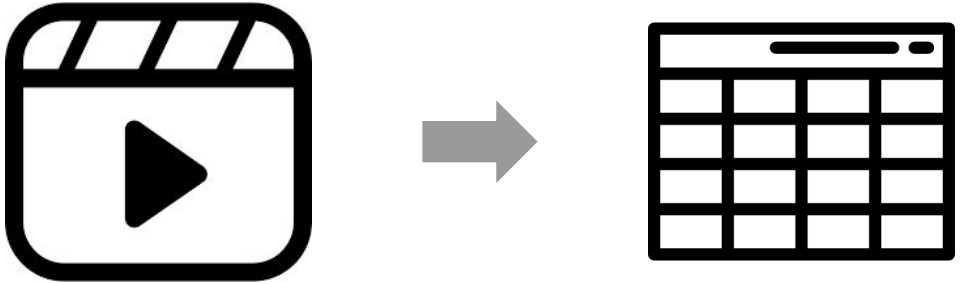
Text

Easy to understand 👍

Time-saving 👍

Introduction

How to convert a video into a table?

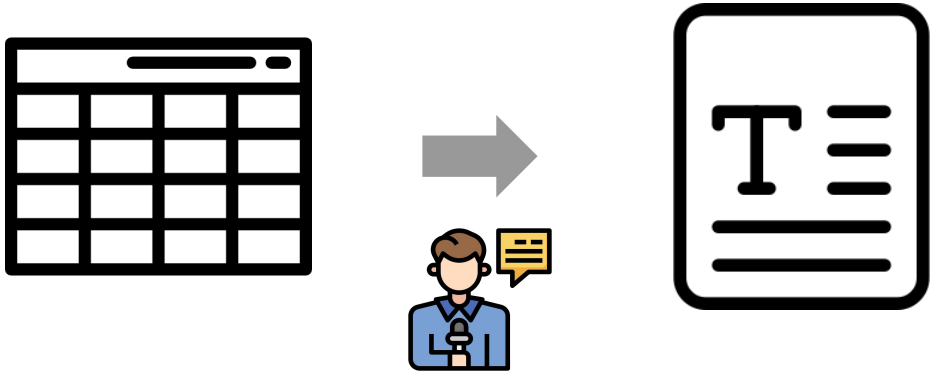


ShuttleSet [1] has already completed this task.

[1] ShuttleSet: A Human-Annotated Stroke-Level Singles Dataset for Badminton Tactical Analysis [KDD 2023]

Introduction

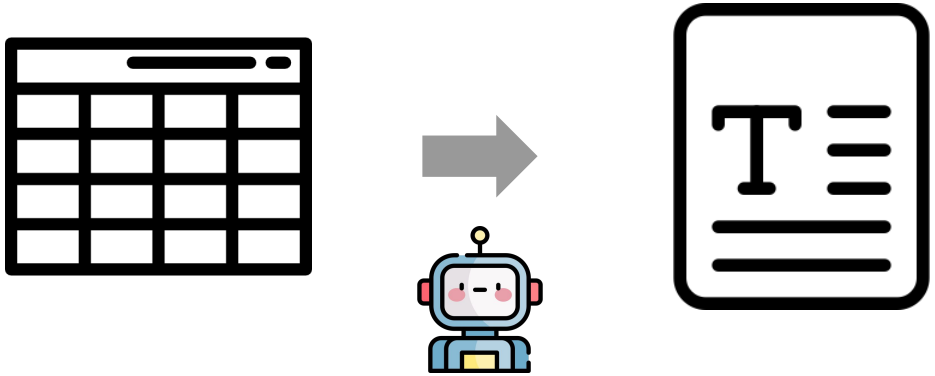
How to convert a table into a text?



1. Understand the structure of the table
2. Select relevant and important information
3. Write accurate and fluent text

Introduction

How to convert a table into a text?



1. Understand the structure of the table
2. Select relevant and important information
3. Write accurate and fluent text

2. Related Work

2.1. Table-to-Text Generation

Table-to-Text Generation

Convert **structured tables** into **unstructured texts**.

- **Content Planning (What to say):** means to analyze and filter given structured data, from which all or part of the data is selected for abstraction and association.
- **Content Generating (How to say):** refers to accurately and fluently describe the selected data through natural language.

[2] A Survey on Neural Data-to-Text Generation [IEEE Transactions on Knowledge and Data Engineering]

Table-to-Text Generation

- High data fidelity
- Long textual outputs

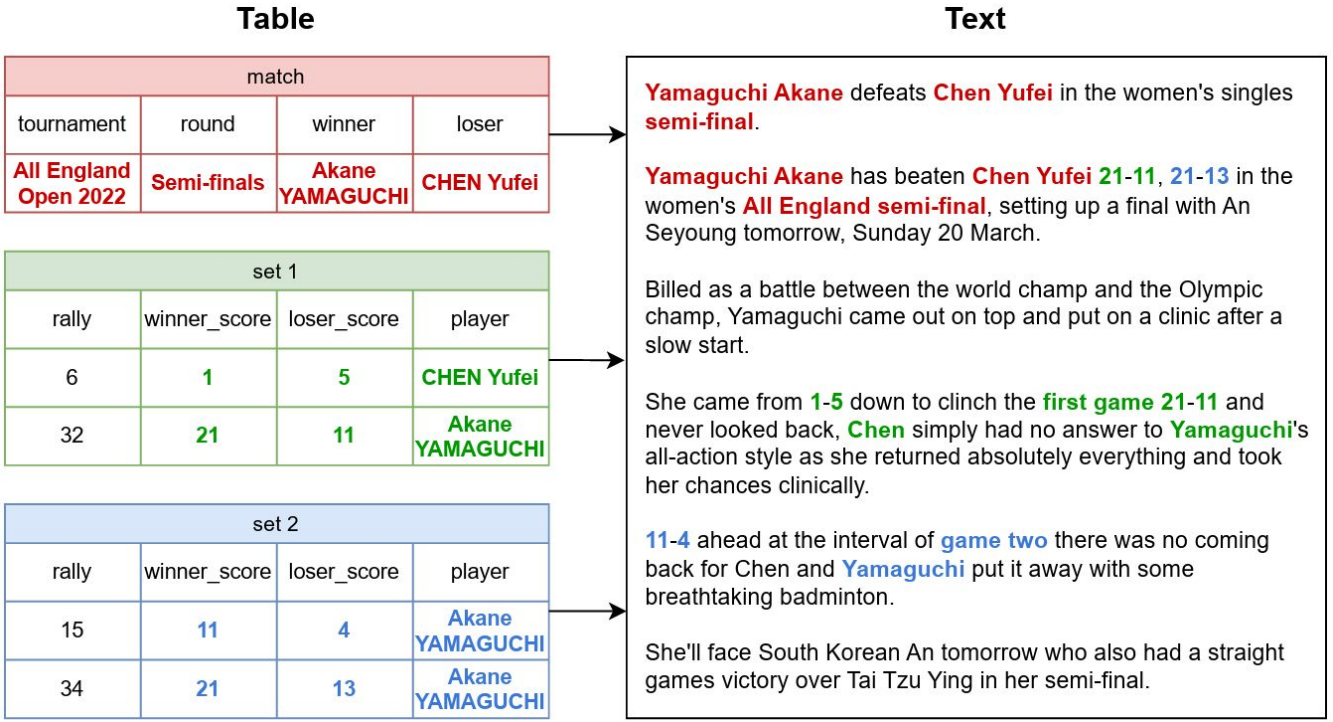
TABLE I
LIST OF DATASETS FOR D2T. "AVG.LEN" REFERS TO THE AVERAGE LENGTH OF SAMPLES

Type	Sub.Type	Datasets	Domain	Language	Samples	Avg.Len	Plan	Year
Graph-to-text	KG-to-text	WebNLG	Wikipedia	English	9.6K	22.69	No	2017
		Enriched WebNLG	Wikipedia	English, German	32.9K	19.67	Yes	2018
		WebNLG+	Wikipedia	English, Russian	25.3K	20.57	Yes	2020
		AGENDA	Wikipedia	English	40.7K	141.2	No	2019
		WITA	Wikipedia	English	55.4K	18.8	No	2020
		DART	Wikipedia	English	82.2K	21.6	No	2021
		KGTEXT	Wikipedia	English	16M	20.2	No	2020
		GenWiki	Wikipedia	English	1336.7K	21.46	No	2020
		EventNarrative	Wikipedia (EventKG)	English	224.4K	50.58	No	2021
		TEKGEN	Wikipedia	English	5723K	21.2	No	2021
	AMR-to-text	AMR15	Discussion forum etc.	English	19.5K	21.3	No	2015
		AMR17	Discussion forum etc.	English	39.2K	20.4	No	2017
		AMR20	Discussion forum etc.	English	59.2K	16.9	No	2020
Table-to-Text	Domain-specific	RotoWire	Sports	English	4.9K	337.1	No	2017
		RotoWire-Modified	Sports	English	3.7K	384	No	2019
		RotoWire-FG	Sports	English	7.5K	205.9	No	2020
		ESPN	Sports	English	15.0K	9.5	No	2018
		MLB	Sports	English	26.3K	542.05	No	2019
		BioLeaflets	Biomedical	English	77.1K	412.9	No	2021
		numericNLG	Scientific	English	1.3K	94	No	2021
	Open-domain	WikiBio	Wikipedia	English	728.2K	26.1	No	2016
		WikiPerson	Wikipedia	English	311.5K	88.3	No	2018
		WikiTableText	Wikipedia	English	13K	13.91	No	2018
		WikiTablePara	Wikipedia	English	–	760	Yes	2018
		WikiTableT	Wikipedia	English	15K	115.9	No	2021
		Wiki3C	Wikipedia	English	10.2K	–	No	2021
		ToTTo	Wikipedia	English	136.0K	17.4	No	2020
		TabFact	Wikipedia	English	118K	13.8	No	2020
		TWT	Wikipedia	English	177.7K	16.4	No	2021
		LogicNLG	Wikipedia	English	37.0K	14.2	No	2020
		Logic2Text	Wikipedia (WikiTables)	English	10.7K	16.77	No	2020
MR-to-Text	Slot-value pairs	E2E	Restaurant	English	51.4K	22.41	No	2017
		Cleaned E2E	Restaurant	English	42.4K	22.9	No	2018
		Czech D2T	Restaurant	English, Czech	5.2K	–	No	2019
		ViGGO	Video Game	English	6.9K	25.01	No	2019
		KVRET	Multi-domain	English	3.0K	47.25	No	2017
		MultiWOZ	Multi-domain	English	10.4K	15.12	No	2018
	Attribute-value pairs	SUMTIME	Weather	English	1.2K	16.2	No	2008
		WEATHERGOV	Weather	English	22.1K	28.7	No	2009
		RoboCup	Sports	English	1.9K	5.7	No	2008
		CACAPO	Multi-domain	English, Dutch	21.0K	16.86	Yes	2020

"Plan" refers to whether it contains the content planning.

[2] A Survey on Neural Data-to-Text Generation [IEEE Transactions on Knowledge and Data Engineering]

Table-to-Text Generation



2.2. Model-based Method

Model-based Method

TABLE V
SUMMARY OF RECENT NEURAL MODULAR APPROACHES ON D2T

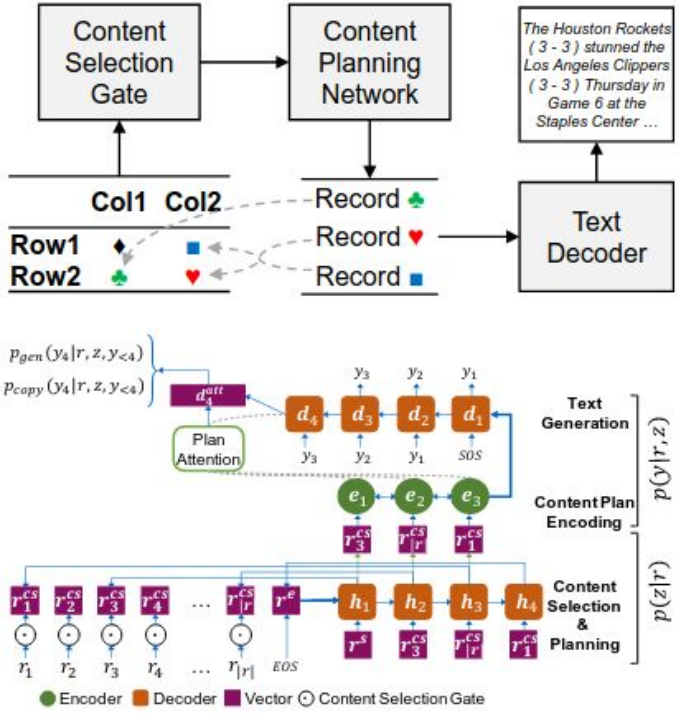
Work	Backbone	Pre-trained	Paradigm	Two-stage	Template	MTL	Year	Performance	Datasets
HSMM [22]	LSTM	✗	✗	✗	✓	✗	2018	B:59.8%, NIST:7.56 B:34.8%, NIST:7.59	E2E WikiBio
DCM [187]	LSTM	✗	✗	✗	✓	✗	2018	B:16.19%, CO:16.34%	RotoWire
NCP [18]	LSTM	✗	✗	✓	✗	✗	2019	B:16.50%, CO:18.58%	RotoWire
3-stages [48]	LSTM	✗	✗	✗	✓	✗	2019	B:33.3%	WikiTablePara
BestPlan [19]	LSTM	✗	✗	✓	✗	✗	2019	B:47.4%, CIDEr:2.692	WebNLG
PIVOT [188]	LSTM	✗	✗	✓	✗	✗	2019	B:27.34%, NIST:6.8763	WikiBio
CSP+TG [189]	Trans.	✗	✗	✓	✗	✓	2019	B:15.17%, CO:19.26%	RotoWire-Modified
Segment. [190]	LSTM	✗	✗	✓	✗	✗	2020	B:65.1%, Dist-3:911 B:46.1%, Dist-3:149	E2E WebNLG
Anc2Pro [191]	Trans.	✗	✗	✗	✓	✓	2020	B:49.9%	WebNLG
DUV [192]	LSTM	✗	✗	✓	✗	✓	2020	B:15.92%, CO:23.32% B:9.51%, CO:27.78%	RotoWire-Modified MLB
ITE [193]	T5	✓	Fine-tuning	✗	✓	✗	2020	B:57.1%	WebNLG
NDP [194]	LSTM	✗	✗	✓	✗	✗	2021	B:16.67%, CO:20.67%	RotoWire
Macro [195]	LSTM	✗	✗	✓	✗	✗	2021	B:15.46%, CO:17.7% B:12.62%, CO:21.8%	RotoWire MLB
PlanGen [196]	BERT/BART	✓	Fine-tuning	✓	✗	✗	2021	B:65.42% B:49.2%, PAT:58.7%, BRT:0.249	WebNLG ToTTo
Aug-plan [82]	BART	✓	Fine-tuning	✓	✗	✗	2021	B:31.16%, PAT:56.75%	WikiPerson
SANA [197]	Trans.	✗	✗	✓	✗	✓	2021	B:54.51%, PAT:61.01% B:30.29%, PAT:68.28%	WikiBio WikiPerson
P2G [24]	T5	✓	Prefix-tuning	✗	✓	✓	2021	B(500 e.g.):50.1%	WikiBio
3-STAGE [25]	BART	✓	✗	✗	✓	✗	2022	B:43.94%(Zero-Shot) B:36.04%(Zero-Shot)	WebNLG E2E

MTL: Multi-Task Learning. B: BLEU. BTS: BERTScore. PAT: PARENT. BRT: BLEURT.

[2] A Survey on Neural Data-to-Text Generation [IEEE Transactions on Knowledge and Data Engineering]

NCP [3]

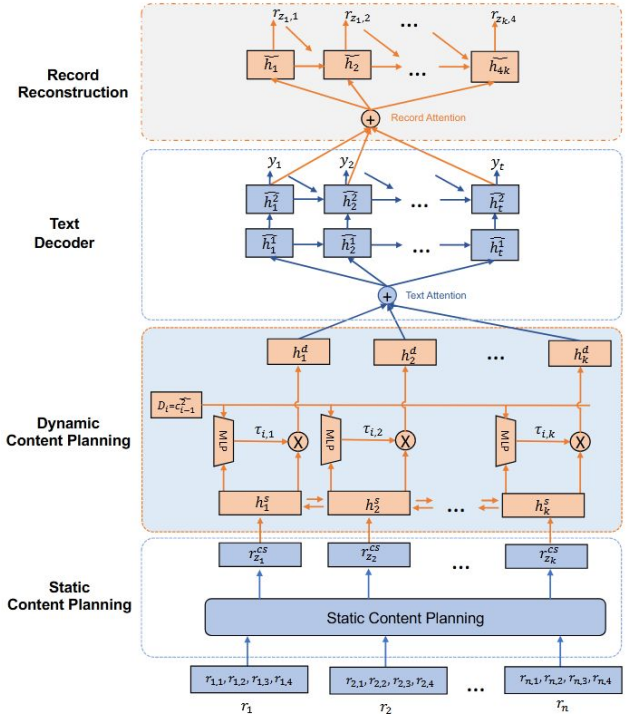
- **Content Selection (What to say)**
  - Content Selection Gate
- **Content Planning (What order)**
  - One-layer Pointer Network
- **Text Generation (How to say)**
  - Two-layer LSTM



[3] Data-to-Text Generation with Content Selection and Planning [AAAI 2019]

NDP [4]

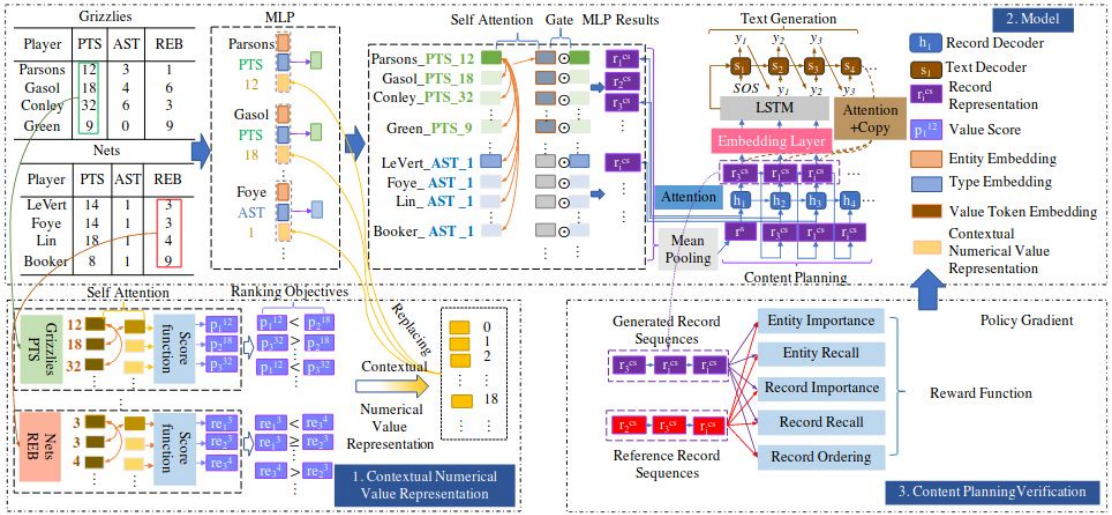
- Static Content Planning
- **Dynamic Content Planning**
- Text Decoder
- **Record Reconstruction**



[4] Neural data-to-text generation with dynamic content planning [Knowledge-Based Systems 2021]

DUV [5]

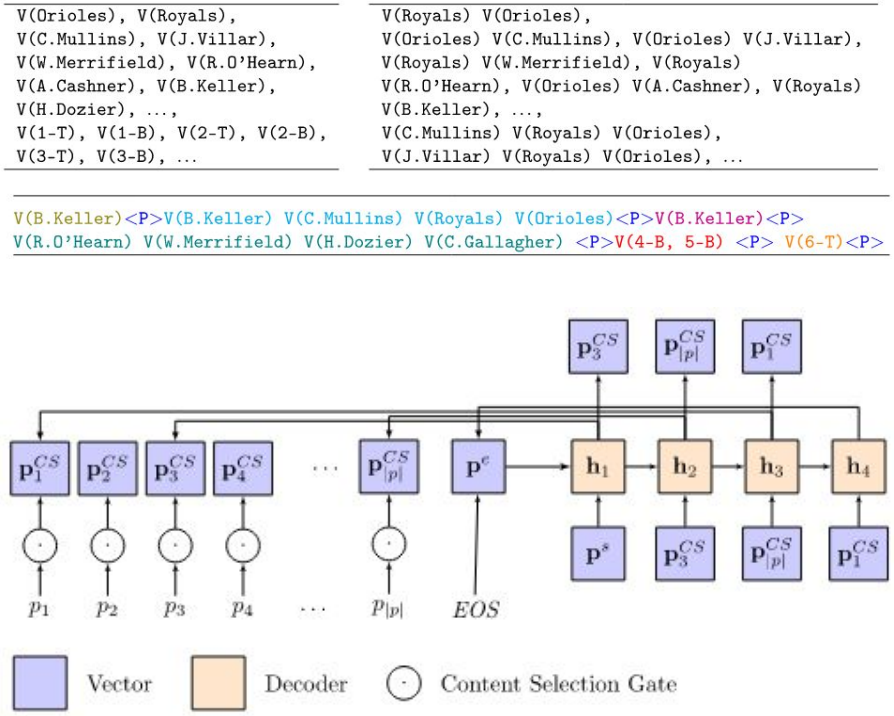
- Contextual Numerical Value Representation
- Content Planning Verification



[5] Enhancing Content Planning for Table-to-Text Generation with Data Understanding and Verification [EMNLP 2020]

Macro [6]

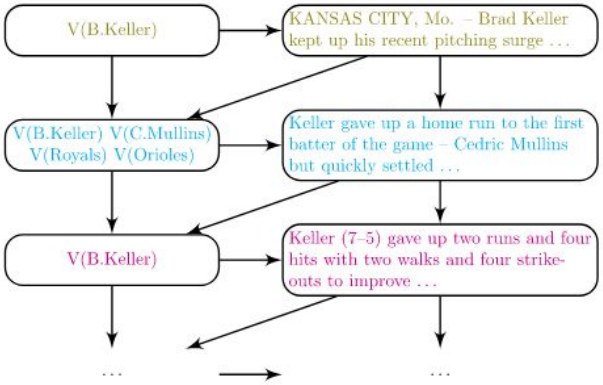
- **Macro Planning**
  - Entities
  - Events
  - Interactions
  - Paragraphs
- Text Generation

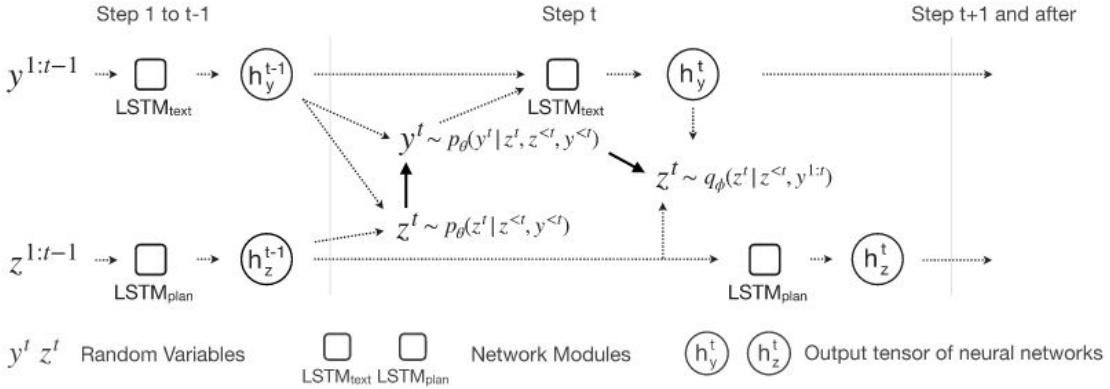


[6] Data-to-text Generation with Macro Planning [TACL 2021]

SeqPlan [7]

- **Interleaving planning and generation steps**
- Sequential latent variable
- High sample efficiency in low-resource settings





[7] Data-to-text Generation with Variational Sequential Planning [TACL 2022]

2.3. Prompt-based Method

Zero-shot, One-shot, and Few-shot [8]

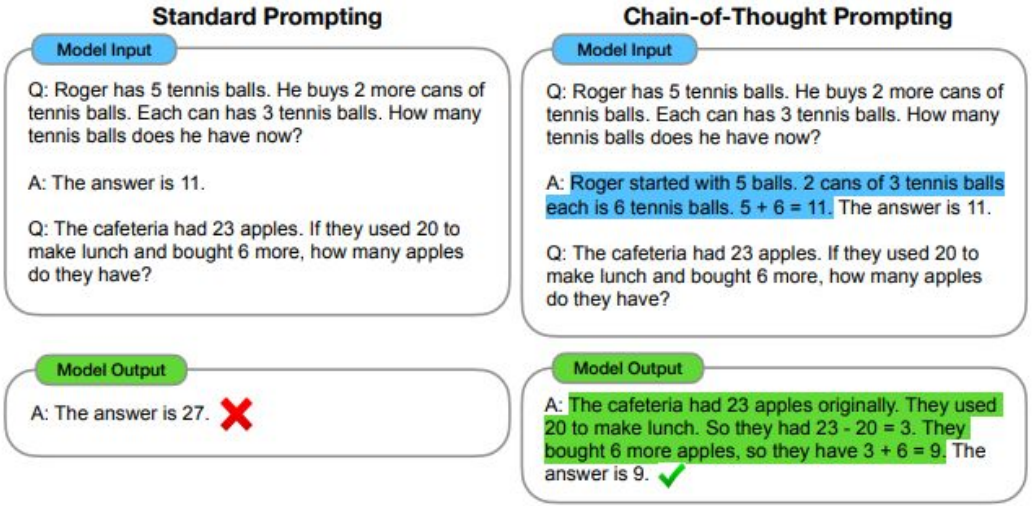
- Zero reference example
- One reference example
- Few reference examples

Learning Type	Description	Example Prompt	Example Output
Zero-Shot	The model performs tasks without prior specific examples.	"Classify this review: I loved this movie! Sentiment:"	Sentiment: Positive
One-Shot	The model learns from a single example.	"Classify this review: I loved this movie! Sentiment: Positive Classify this review: I don't like this chair. Sentiment:"	Sentiment: Negative
Few-Shot	The model learns from a few examples.	"Classify this review: I loved this movie! Sentiment: Positive Classify this review: I don't like this chair. Sentiment: Negative Classify this review: Who would use this product? Sentiment:"	Sentiment: [Positive/Negative/ Neutral]

[8] Language Models are Few-Shot Learners [NeurIPS 2020]

Chain-of-Thought [9]

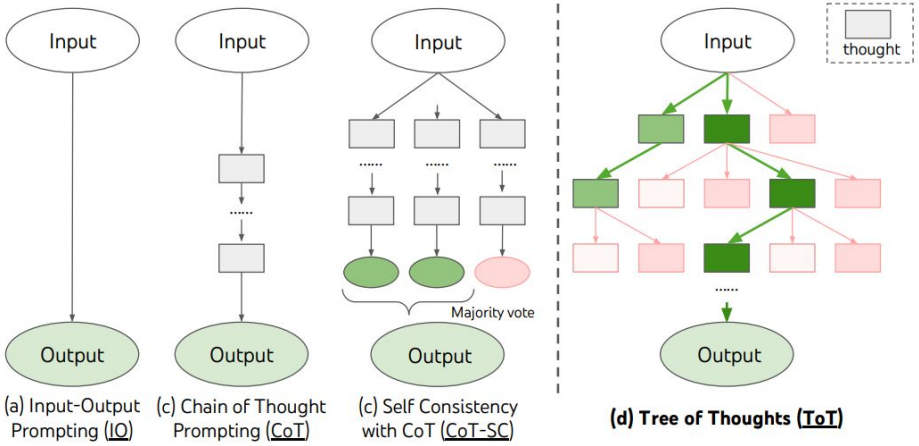
- Let think step by step!



[9] Chain-of-Thought Prompting Elicits Reasoning in Large Language Models [NeurIPS 2022]

Tree-of-Thought [10]

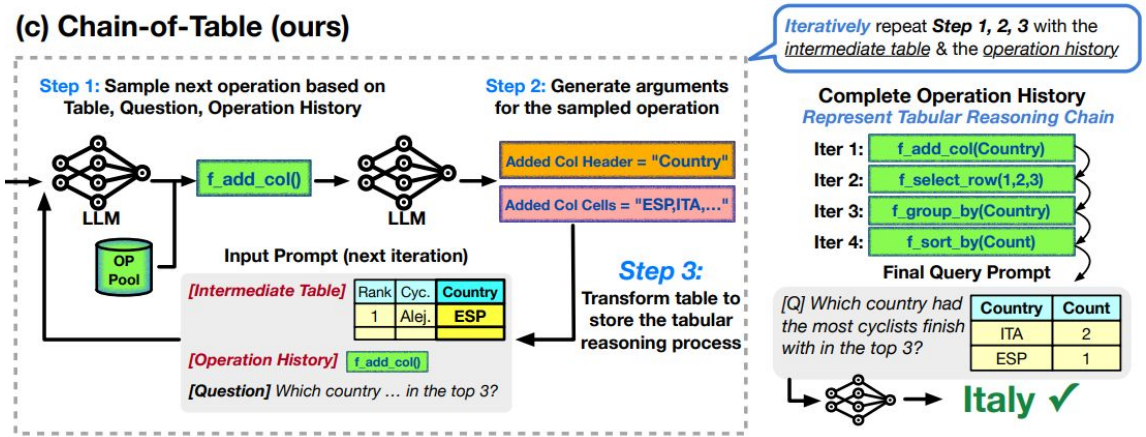
- Exploring multiple reasoning paths
- Self-evaluating choices
- Backtracking



[10] Tree of Thoughts: Deliberate Problem Solving with Large Language Models [NeurIPS 2023]

Chain-of-Table [11]

- Dynamic planning
- Argument generation
- Operation history



[11] Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding [ICLR 2024]

Comparison

Method	Prompting	Chain	Tree	Table	Table-to-Text Generation
N-shot [8]	✔	✘	✘	✘	✘
Chain-of-Thought [9]	✔	✔	✘	✘	✘
Tree-of-Thought [10]	✔	✔	✔	✘	✘
Chain-of-Table [11]	✔	✔	✘	✔	✘
Tree-of-Text (Ours)	✔	✔	✔	✔	✔

3. Problem

Problem

What are the problems with model-based methods?

1. The dataset size is **limited** (e.g., ShuttleSet+ contains only 58 instances)
2. Making model training highly **challenging**

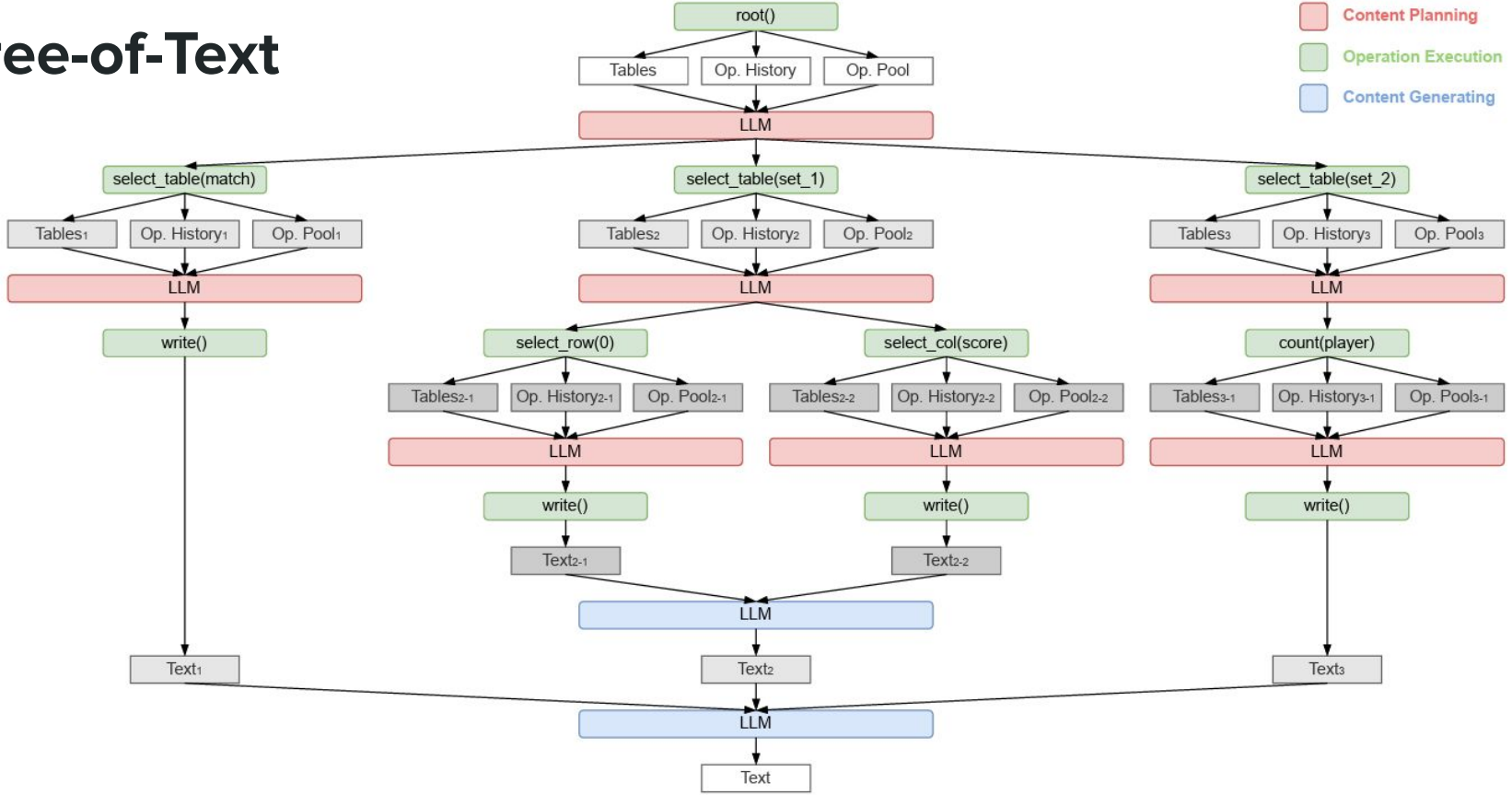
Problem

What are the problems with prompt-based methods?

1. Previous methods directly input the entire table into the LLM, making it difficult for the model to fully **understand the table structure**
2. These methods directly output the final text in a single step, limiting the model's ability to **process detailed information**

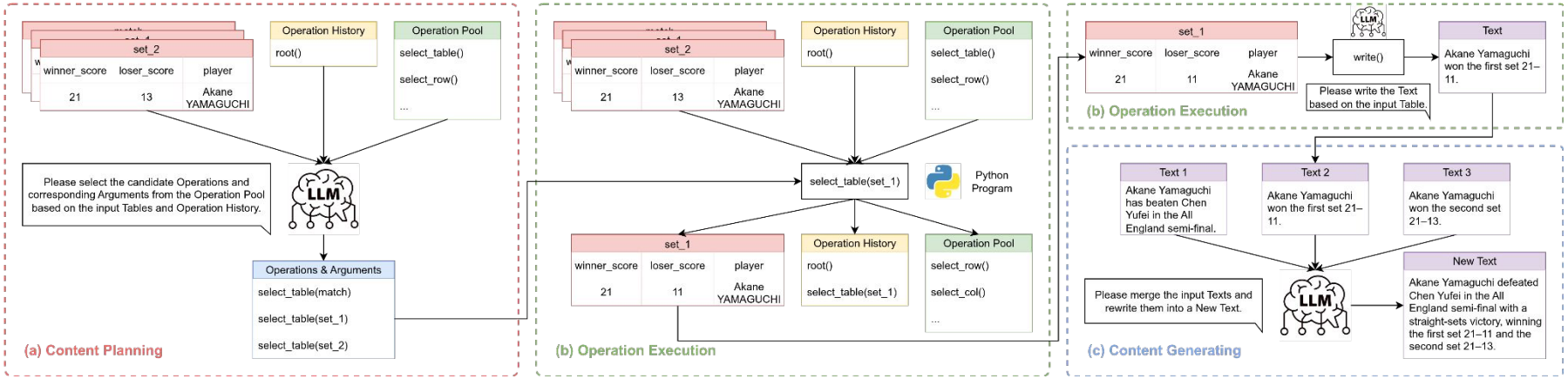
4. Solution

Tree-of-Text

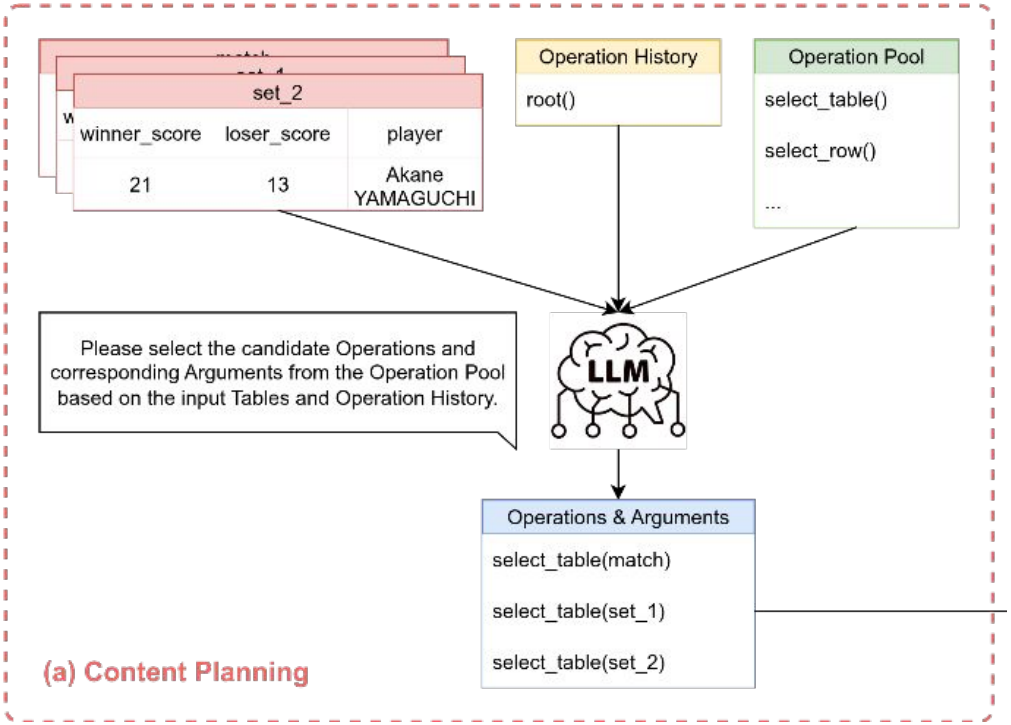


[12] Algorithm 426: Merge sort algorithm [M1]

Tree-of-Text


set_2
winner_score
loser_score
player
21
13
Akane YAMAGUCHI
Operation History
root()
Operation Pool
select_table()
select_row()
...
Please select the candidate Operations and corresponding Arguments from the Operation Pool based on the input Tables and Operation History.
LLM
Operations & Arguments
select_table(match)
select_table(set_1)
select_table(set_2)
(a) Content Planning
select_table(set_1)
Python Program
set_1
11
select_row()
select_col()
(b) Operation Execution
write()
Text
Akane Yamaguchi won the first set 21–11.
Please write the Text based on the input Table.
Text 1
Akane Yamaguchi has beaten Chen Yufei in the All England semi-final.
Text 2
Akane Yamaguchi won the first set 21–11.
Text 3
Akane Yamaguchi won the second set 21–13.
Please merge the input Texts and rewrite them into a New Text.
New Text
Akane Yamaguchi defeated Chen Yufei in the All England semi-final with a straight-sets victory, winning the first set 21–11 and the second set 21–13.
(c) Content Generating

(1) Content Planning



(1) Content Planning

- Starting from the **root node**
- **LLM determines the operations and arguments for the child nodes**
- Input
  - **Tables** $T \leftarrow (Tj \mid j = 1, 2, \ldots, n)$
  - **Operation History** $OH \leftarrow (op \mid op = root())$
  - **Operation Pool** $OP \leftarrow (op \mid op \in operations, op \mathrel{!}= root())$
  - **Depth** $D \leftarrow 0$
- Output
  - **Operations and Arguments** $OA \leftarrow (Oi(Ai) \mid Oi \in OP, i = 1, 2, \ldots, d)$
  - *d* represents the **degree** of this node and must not exceed the **maximum degree** *MAX_DEGREE*

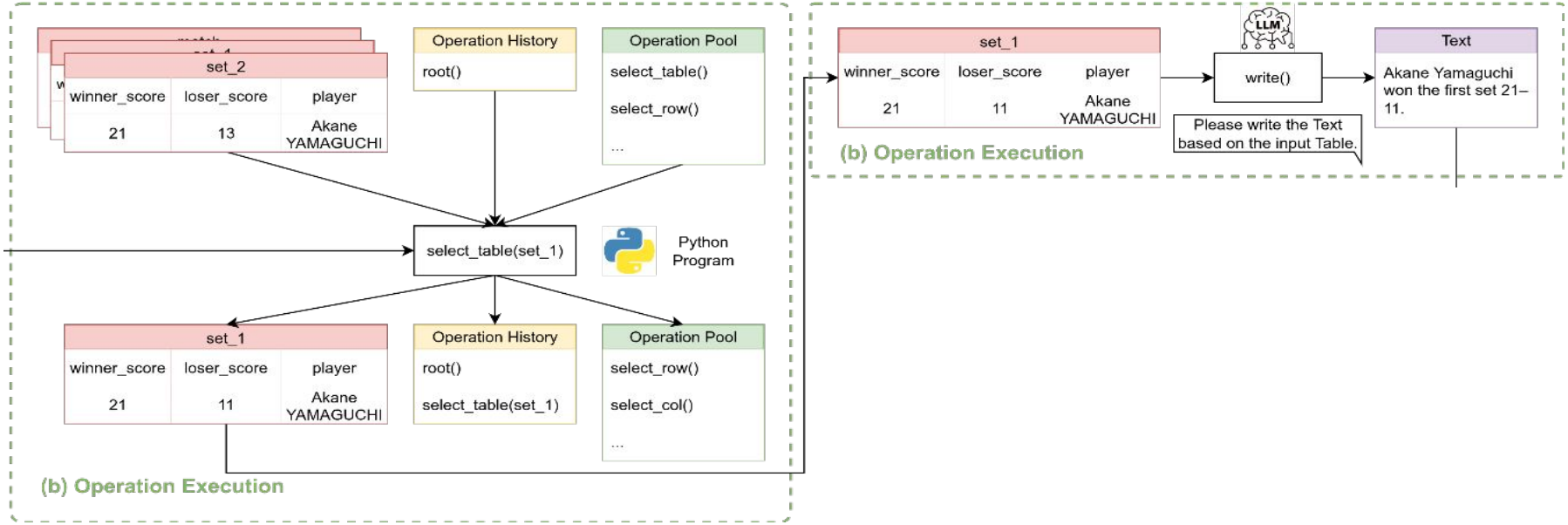
Prompt for Content Planning

```
System:

You are a content planner for the badminton game report.

Please select candidate Operations and corresponding Arguments from the Operation
    Pool based on the input Tables and Operation History. These candidate Operations
     will be the next Operation in the Operation History.

# Requirements

1. Strictly adhere to the requirements.
2. The output must be in English.
3. The output must be based on the input data; do not hallucinate.
4. The table format is {TABLE_FORMAT}.
5. The length of Operation History must be less than or equal to {MAX_DEPTH}.
6. The number of Operations must be less than or equal to {MAX_DEGREE}.
7. Only select Operations from the Operation Pool.
8. Arguments must match the format required by the corresponding Operations.
9. Operations & Arguments must follow this format: [operation_1(argument_1, ...),
    operation_2(argument_2, ...), operation_3(argument_3, ...), ...]
10. Only output Operations & Arguments!
11. The number of tokens in the Operations & Arguments must be within {
    PLANNING_TOKENS}.

# Table Description

{TABLE_DESCRIPTION}

# Operation Description

{OPERATION_DESCRIPTION}

User:

# Test

## Tables

{TABLES}

## Operation History

{OPERATION_HISTORY}

## Operation Pool

{OPERATION_POOL}

## Operations & Arguments
```

Operations

- **root():** Do nothing. Represent the root node of the tree.
- **select_table():** Select a table by the table name.
- **select_row():** Select the rows by the row indices.
- **select_col():** Select the columns by the column names.
- **count():** Count the number of unique values by the column names of tables.
- **sort():** Sort the rows by the column names in sorting orders.
- **filter():** Filter the rows by the column names, symbols, and values.
- **write():** Write the text based on the tables. Represent the leaf node of the tree.

(2) Operation Execution



(2) Operation Execution

- **Execute** *Oi(Ai)* **in Operations and Arguments** *OA* **respectively**
- **Update Tables** *Ti*, **Operation History** *OHi*, **Operation Pool** *OPi*, and **Depth** *Di*
  - *Ti ← Oi(T, Ai)*
  - *OHi ← OH + Oi(Ai)*
  - *OPi ← OP − Oi()*
  - *Di ← D + 1*
- Pass *Ti*, *OHi*, *OPi*, and *Di* to the **child nodes respectively**
  - **(1) Content Planning**
  - **(2) Operation Execution**

(2) Operation Execution

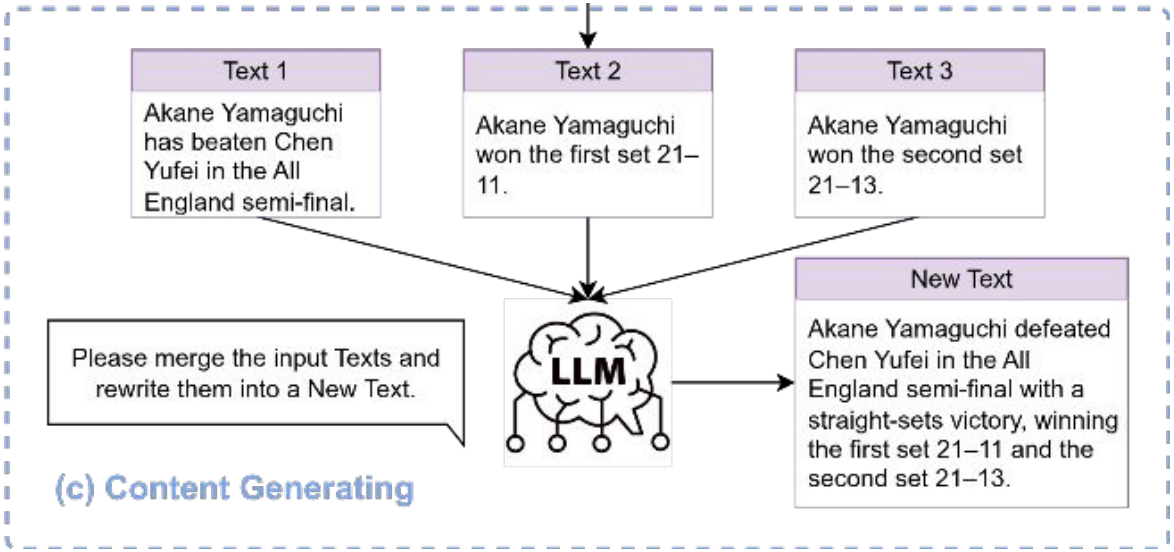
- **The process continues recursively, until...**
- **If the depth *D_i'* reaches the maximum depth *MAX_DEPTH***
  - The LLM is used to generate a textual description *t* of the input table *T*, which is then returned to the parent node
- **If a *write()* operation is encountered**
  - The LLM writes a short text *t_i'* based on the input table *T* as well
  - Since other child nodes also return texts *t_i'*, we collect them into a sequence

    $t' = (t_i' \mid i = 1, 2, ..., d)$

Prompt for write() operation

```
System:

You are a content writer for the badminton game report.

Please write the Report based on the input Table.

# Requirements

1. Strictly adhere to the requirements.
2. The output must be in English.
3. The output must be based on the input data; do not hallucinate.
4. The Table format is {TABLE_FORMAT}.
5. The Report can only describe the content included in the Tables and cannot
    describe anything not included in the Tables.
6. The Report must consist of only one paragraph.
7. The number of tokens in the Report must be within {WRITE_TOKENS}.

# Table Description

{TABLE_DESCRIPTION}

User:

# Test

## Tables

{TABLES}

## Report
```

(3) Content Generating



(3) Content Generating

- The LLM then **merges** these short texts *t'* and **rewrites** into a new text *t*
- Return *t* to the **parent node**
  - **(3) Content Generating**
- This recursive process continues until it returns to the **root node**
- The text *t* returned from the root node is the **final output**

Prompt for Content Generating

```
System:

You are a content generator for the badminton game report.

Please merge and rewrite a New Report based on the input Reports.

# Requirements

1. Strictly adhere to the requirements.
2. The output must be in English.
3. The output must be based on the input data; do not hallucinate.
4. The New Report must include all the content from the input Reports; do not omit
    any information.
5. The New Report must follow the order of the input Reports.
6. The number of tokens in the New Report must be within {GENERATING_TOKENS}.

User:

# Test

## Reports

{REPORTS}

## New Report
```

Optimizations

1. Unlike Chain-of-Table, which generates operations first and then arguments, our method generates operations and arguments **in one step**
2. If a node has **one child node**, there is no need to use the LLM for merging
3. LLM is used for **merging only at the root node**, while other nodes simply concatenate texts

Algorithm

Algorithm 1 Tree-of-Text

Require: Tables T, Operation History OH, Operation Pool OP, Depth D, Max Depth MAX_DEPTH, Max Degree MAX_DEGREE
Ensure: Text t

function TREE-OF-TEXT(T, OH, OP, D)
▷ Depth must not exceed Max Depth
  if $D \geq MAX\_DEPTH$ **then**
    $t \leftarrow$ WRITE(T)
    return t
  end if
▷ Content Planning
  $OA \leftarrow$ CONTENT_PLANNING(T, OH, OP)
▷ Operation Execution
  $t' \leftarrow ()$
  for each $O_i(A_i) \mid O_i \in$ OP, $i = 1, 2, \ldots, d$ in OA **do**
▷ Degree must not exceed Max Degree
    if $i \geq MAX\_DEGREE$ **then**
      break
    end if
    if $O_i = write()$ **then**
      $t'_i \leftarrow$ WRITE(T)
    else
      $T'_i \leftarrow O_i(T, A_i)$
      $OH'_i \leftarrow OH + O_i(A_i)$
      $OP'_i \leftarrow OP - O_i()$
      $D'_i \leftarrow D + 1$
      $t'_i \leftarrow$ TREE-OF-REPORT(T'_i, OH'_i, OP'_i, D'_i)
    end if
    $t' \leftarrow t' + t'_i$
  end for
▷ Content Generating
  $t \leftarrow$ CONTENT_GENERATING(t')
  return t
end function

Main Program
$T \leftarrow (T^j \mid j = 1, 2, \ldots, n)$
$OH \leftarrow (op \mid op = \text{root}())$
$OP \leftarrow (op \mid op \in \text{operations},\ op \neq \text{root}())$
$D \leftarrow 0$
$t \leftarrow$ TREE-OF-TEXT(T, OH, OP, D)

5. Experiment

5.1. Dataset

🏸 ShuttleSet+

- We introduce a new dataset, **ShuttleSet+**, derived from **ShuttleSet22 [13].**
- Since ShuttleSet22 does not include corresponding textual reports for each match, we collected **human-written reports** in English for each game from online sources such as the BWF and Olympics websites, and renamed the dataset as ShuttleSet+.
- **58 matches**
  - Train: 40
  - Valid: 9
  - Test: 9

[13] ShuttleSet22: Benchmarking Stroke Forecasting with Stroke-Level Badminton Dataset [CoRR 2023]

Data Preprocessing for ShuttleSet+

1. We retain only the final stroke of each rally.
2. We selected the nine most essential columns, renaming and reordering to improve clarity while removing unrelated fields.
3. We translated the values in the ball_type, win_reason, and lose_reason into English.
4. We reorder the table columns according to the order specified in the table description of ShuttleSet+.

🏀 RotoWire-FG [14]

- The **RotoWire-FG [14]** dataset is an extension of the original **RotoWire [15]** dataset.
- **Basketball** game summaries from RotoWire Game Recaps covering the years 2017–2019, and aligned with official NBA box score tables.
- **7,635 summaries**
  - Train: 5,340
  - Valid: 1,147
  - Test: 1,148

[14] Revisiting Challenges in Data-to-Text Generation with Fact Grounding [INLG 2019]
[15] Challenges in Data-to-Document Generation [EMNLP 2017]

Data Preprocessing for RotoWire

1. We convert the original data from JSON format into multiple CSV tables: game, home_line, vis_line, and box_score.
2. We reorder the table columns according to the sequence specified in the table description of RotoWire.

⚾ MLB [16]

- **Baseball** statistics paired with human-written summaries in English sourced from the ESPN website.
- Compared to RotoWire, it is approximately **five times larger**, featuring a broader vocabulary and longer summaries.
- **26,304 instances**
  - Train: 22,821
  - Valid: 1,739
  - Test: 1,744

	ROTOWIRE	MLB
Vocab Size	11.3K	38.9K
# Tokens	1.5M	14.3M
# Instances	4.9K	26.3K
Avg Length	337.1	542.05
# Record Types	39	53
Avg Records	628	565

Table 1: Vocabulary size, number of tokens, number of instances (i.e., record-summary pairs), average summary length, number of record types and average number of records in ROTOWIRE and MLB datasets.

[16] Data-to-text Generation with Entity Modeling [ACL 2019]

Data Preprocessing for MLB

1. We first convert the original data from JSON format into multiple CSV tables: game, home_line, vis_line, box_score, and play_by_play.
2. We remove these redundant rows to streamline the dataset.
3. We reorder the table columns according to the sequence specified in the table description of MLB.

5.2. Evaluation Metric

5.2.1. Automatic Evaluation

Information Extraction (IE) [17]

- **Information**
  - (table|column|value)
  - e.g. (match|winner|Akane YAMAGUCHI)
- **LLM-based IE model**
  - **Extract information from the text.**
  - To validate its reliability, we manually annotated a set of information and compared it with that extracted by the LLM.
  - It achieved over **70%** on all evaluation metrics with **few-shot prompting**.

[17] Challenges in Data-to-Document Generation [EMNLP 2017]

LLM-based IE model

- To validate its reliability, we manually annotated a set of information and compared it with that extracted by the LLM.
- It achieved over **70%** on all evaluation metrics with **few-shot prompting**.

Prompt	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%
Zero-shot	**14.0000**	**100.00**	70.56	**76.57**	71.51	26.80
One-shot	12.3333	**100.00**	75.35	70.46	70.71	38.24
Few-shot	10.3333	**100.00**	**93.89**	**76.57**	**83.86**	**71.01**

Prompt for the LLM-based IE model

```
System:

You are a relation extractor for the badminton game report.

Please extract the Report Relation contained in the Report from the Table Relation.

There is an Example that you can refer to.

# Requirements

1. Strictly adhere to the requirements.
2. The output must be in English.
3. The output must be based on the input data; do not hallucinate.
4. Please do not output any Report Relation that is not included in the Report.
5. Please do not output any Report Relation that is not included in the Table
    Relation.
6. The Report Relation must contain all the relations from the input Report; do not
    omit any relation.
7. The Report Relation must follow the order in the input Report.
8. The Report Relation must follow the format: [(table|column|value), (table|column|
    value), ...]

# Table Description

{TABLE_DESCRIPTION}

User:

# Test

## Report

{REPORT}

## Table Relation

{TABLE_RELATION}

## Report Relation
```

Automatic Evaluation [17]

- **Relation Generation (RG)** ⬆️
  - **Count (#)** and **Precision (P%)** of information extracted from the generated text and table.
- **Content Selection (CS)** ⬆️
  - **Precision (P%)**, **Recall (R%)**, and **F1 score (F%)** of information extracted from the generated text and referenced text.
- **Content Ordering (CO)** ⬆️
  - The complement of the **Damerau-Levenshtein Distance (DLD%)** between information extracted from the generated text and referenced text.

[17] Challenges in Data-to-Document Generation [EMNLP 2017]

Automatic Evaluation [17]

- **Time (in seconds) ⬇️**
  - Average time required to generate a text.
- **Cost (in $0.001 USD) ⬇️**
  - Average cost required to generate a text.

[17] Challenges in Data-to-Document Generation [EMNLP 2017]

5.2.2 Human Evaluation

Human Evaluation [18]

1. Analyze each summary against the corresponding tables and count the number of
    a. **Supported Facts:** Statements consistent with the table.
    b. **Contradicted Facts:** Statements inconsistent with the table.
2. Select the best and worst summary from the five options based on three criteria
    a. **Coherence:** How logically and smoothly the ideas and events are connected throughout the report.
    b. **Conciseness:** How effectively a report conveys information using as few words as necessary, without unnecessary repetition or irrelevant details.
    c. **Grammaticality:** Whether the text follows the rules of standard English grammar.

[18] Data-to-text Generation with Variational Sequential Planning [TACL 2022]

5.3. Implementation Detail

Implementation Detail

- **LLM:** gpt-4o-mini
- **Max depth:** 5
- **Max degree:** 5
- **Operation pool:** All operations
- **Table format:** CSV

5.4. Quantitative Result

5.4.1. Automatic Evaluation

🏸 ShuttleSet+

ShuttleSet+	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Time	Cost
Zero-shot	14.00	82.69	65.81	75.90	69.85	48.25	7.53	0.86
One-shot	13.56	81.61	65.57	72.75	68.36	49.40	6.59	1.12
Few-shot	13.78	82.81	66.57	75.33	70.20	53.00	**6.00**	2.20
Chain-of-Thought	12.33	81.35	71.33	73.58	71.41	52.27	6.68	**0.81**
Tree-of-Thought	13.67	79.40	61.98	69.31	63.91	46.50	63.11	9.62
Chain-of-Table	15.89	94.31	69.22	79.74	73.14	40.42	73.67	14.44
Tree-of-Text	**16.78**	**95.79**	**73.74**	**85.16**	**78.03**	**69.30**	29.04	5.71
	5.6%	1.5%	6.5%	6.8%	6.7%	30.8%	40%	40%

RotoWire-FG

RotoWire-FG	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Time	Cost
Zero-shot	47.13	94.13	63.44	**63.02**	63.23	27.74	5.59	0.63
One-shot	42.90	94.24	65.78	62.73	64.22	28.48	**5.07**	0.90
Few-shot	42.94	94.48	69.22	61.12	**64.92**	29.71	5.38	1.48
Chain-of-Thought	**49.29**	93.54	64.06	62.98	63.51	28.51	7.76	**0.61**
Tree-of-Thought	45.73	94.55	65.82	62.80	64.28	28.55	54.62	8.18
Chain-of-Table	36.94	94.95	70.93	56.26	62.75	30.19	63.75	12.54
Tree-of-Text	34.68	**95.11**	**74.65**	55.88	63.92	**31.90**	33.56	7.69
		0.2%	5.2%			5.7%	53%	61%

⚾ MLB

MLB	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Time	Cost
Zero-shot	84.21	87.68	60.92	61.95	61.29	58.07	12.13	**1.46**
One-shot	82.80	87.25	60.68	61.04	60.86	58.28	9.98	3.05
Few-shot	84.54	88.18	60.43	61.36	60.75	57.25	**8.47**	4.85
Chain-of-Thought	**84.67**	88.61	61.93	62.57	61.98	59.21	9.34	1.73
Tree-of-Thought	80.38	86.25	64.33	63.29	63.37	59.64	47.63	7.25
Chain-of-Table	80.99	**88.67**	64.02	60.81	61.36	57.55	55.60	10.80
Tree-of-Text	80.11	87.04	**65.84**	**63.42**	**63.69**	**59.83**	29.18	6.77
			2.3%	0.2%	0.5%	0.3%	53%	67%

5.4.2. Human Evaluation

🏸 ShuttleSet+

ShuttleSet+	#Supp.	#Cont.	Cohe.	Conc.	Gram.
Gold	3.78	**0.78**	**100.00**	**100.00**	**100.00**
Chain-of-Thought	3.67	2.33	-100.00	**100.00**	-100.00
Tree-of-Thought	6.56	2.33	-66.67	-44.44	0.00
Chain-of-Table	7.00	2.11	77.78	-100.00	50.00
Tree-of-Text	**8.22**	1.00	**100.00**	-100.00	55.55

RotoWire-FG

RotoWire-FG	#Supp.	#Cont.	Cohe.	Conc.	Gram.
Gold	9.00	**0.44**	**100.00**	**100.00**	**100.00**
Chain-of-Thought	10.22	2.11	-100.00	66.67	-66.67
Tree-of-Thought	**14.67**	1.44	-50.00	50.00	-50.00
Chain-of-Table	11.11	0.56	66.67	-50.00	0.00
Tree-of-Text	13.33	**0.44**	**100.00**	-77.78	55.56

MLB

MLB	#Supp.	#Cont.	Cohe.	Conc.	Gram.
Gold	6.67	**0.33**	**100.00**	**100.00**	**100.00**
Chain-of-Thought	**13.44**	1.67	-100.00	50.00	-100.00
Tree-of-Thought	11.00	1.44	-66.67	0.00	0.00
Chain-of-Table	7.89	0.89	50.00	-33.33	44.44
Tree-of-Text	7.11	0.89	**100.00**	-66.67	**100.00**

5.5. Qualitative Result

Human

Short and correct

Human
Yamaguchi Akane defeats **Chen Yufei** in the women's singles **semi-final** **Yamaguchi Akane** has beaten **Chen Yufei 21-11**, **21-13** in the women's **All England semi-final**, setting up a final with An Seyoung tomorrow, Sunday 20 March. Billed as a battle between the world champ and the Olympic champ, **Yamaguchi** came out on top and put on a clinic after a slow start. She came from **1-5** down to clinch the **first game 21-11** and never looked back, **Chen** simply had no answer to **Yamaguchi**'s all-action style as she returned absolutely everything and took her chances clinically. **11-4** ahead at the interval of **game two** there was no coming back for **Chen** and **Yamaguchi** put it away with some breathtaking badminton. She'll face South Korean An tomorrow who also had a straight games victory over Tai Tzu Ying in her semi-final.

Chain-of-Table

Long but not correct

Chain-of-Table

In an exhilarating badminton match, **Akane Yamaguchi** faced off against **Chen Yufei** in a thrilling contest that showcased remarkable skill and determination from both players. The match unfolded over **two sets**, with each player demonstrating their prowess on the court.

In the **first set**, **Yamaguchi** started strong, winning the first rally with a **lob** that **went out of bounds for her opponent**. However, **Chen** quickly responded, taking the next rally with a **net shot** that also **went out of bounds**. The set saw a series of intense rallies, with **Yamaguchi** employing a mix of **net shots** and **smashes**, while **Chen** countered with strategic **clears** and **lobs**. The **first set** concluded with **Yamaguchi** edging out **Chen**, winning **12** points to Chen's **8**.

The **second set** continued the fierce competition, with both players exchanging points in rapid succession. **Yamaguchi**'s agility and tactical play were on full display as she executed a series of successful **smashes** and **defensive returns**. **Chen**, not to be outdone, showcased her own skills with powerful **wrist smashes** and precise **clears**. The set was marked by several **long rallies**, with both players refusing to give an inch.

Ultimately, **Yamaguchi** emerged victorious, winning the match with a final score of **21-17** in the **second set**. The match highlighted the exceptional talent of both players, with **Yamaguchi**'s ability to adapt her strategy and **Chen**'s relentless fighting spirit making for an unforgettable encounter. Fans were treated to a display of top-tier badminton, leaving them eagerly anticipating their next showdown.

Tree-of-Text

Longer, detailed, and correct

Tree-of-Text

In the **semi-finals** of the **YONEX All England Open Badminton Championships 2022**, **Akane Yamaguchi** faced off against **CHEN Yufei** in a thrilling match that lasted **41 minutes**. **Yamaguchi** emerged victorious, winning in **two sets** with scores of **21-11** and **21-13**.

In the **first set**, both players showcased their skills, with **CHEN Yufei** initially taking the lead. **CHEN** displayed impressive shots, including a successful **lob** that forced **Akane out of bounds** and a decisive **smash**. However, **Akane Yamaguchi** demonstrated her dominance by winning a total of **16 rallies**, showcasing her exceptional skills and strategic play. She capitalized on **CHEN**'s errors, including **landing judgment mistakes** and **hitting the net**, effectively turning the tide in her favor. The set concluded with **Yamaguchi** scoring **21** points to **CHEN**'s **11**.

The **first set** featured a diverse range of shot types, with "**return net**" being the most frequent at **7** occurrences, followed by "**lob**" at **6** and "**clear**" at **4**. Other notable shots included "**drop**" with **3**, "**smash**" with **2**, and **single** instances of "**cross-court net shot**," "**net shot**," and "**rush**." This variety contributed to the dynamics of the set.

In the **second set**, **Akane Yamaguchi** continued her strong performance, winning **21 rallies** compared to **CHEN Yufei**'s **13**. **Yamaguchi** utilized a series of effective shots, including a **lob** and a **back-court drive**, while **CHEN** managed to respond with a **smash** and a **wrist smash**, winning some points. The **second set** was marked by strategic plays and errors from both players, but **Yamaguchi** maintained her dominance, ultimately winning the set **21-13**.

The **second set** showcased a different shot distribution, with the **smash** being the most frequent, occurring **7** times. The **wrist smash** followed closely with **4** instances, while both the **return net** and lob were executed **4** and **3** times, respectively. Other notable shots included the **lob** and **net shot**, each appearing **3** times, along with **2 defensive return lobs** and a **back-court drive**, highlighting the diverse range of techniques employed by both players.

Overall, **Akane Yamaguchi**'s performance in the **semi-finals** of the **YONEX All England Open Badminton Championships** was a testament to her skill and strategic gameplay, leading her to a well-deserved victory against **CHEN Yufei**.

5.6. Ablation Study

The Effects of Large Language Models

- The performance of **llama3.1-405b** is only slightly worse than that of **gpt-4o-mini**, validating the generalizability of our method on open-source LLMs.
- However, **gpt-4o** did not outperform **gpt-4o-mini**, suggesting that gpt-4o-mini already performs sufficiently well on this task.

LLMs	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Time	Cost
llama3.1-8b	7.44	49.21	45.94	43.14	43.95	42.48	**10.91**	**5.65**
llama3.1-70b	11.33	86.57	64.53	68.54	62.50	59.18	33.12	71.45
llama3.1-405b	15.56	96.17	93.89	91.98	92.77	91.98	57.16	129.17
gpt-4o-mini	**15.78**	**98.04**	**93.94**	**93.94**	**93.94**	**93.94**	29.04	5.71
gpt-4o	**15.78**	**98.04**	93.29	93.29	93.29	93.29	33.73	54.88

The Analysis of Max Depth & Max Degree

- If more **detailed text** is required, **increasing max depth and max degree** improves performance at the expense of higher computational cost.
- Conversely, for more **general text**, **reducing the max depth and max degree** lowers both the level of detail and the cost.

Max Depth	Max Degree	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Time	Cost
5	5	**15.78**	**98.04**	**93.94**	**93.94**	**93.94**	**93.94**	29.04	5.71
3	5	15.67	95.99	91.42	92.72	92.03	91.42	16.60	2.37
5	3	15.67	97.21	92.46	92.46	92.46	92.46	29.48	4.90
3	3	13.89	88.18	83.43	82.00	82.56	82.00	**11.79**	**1.82**

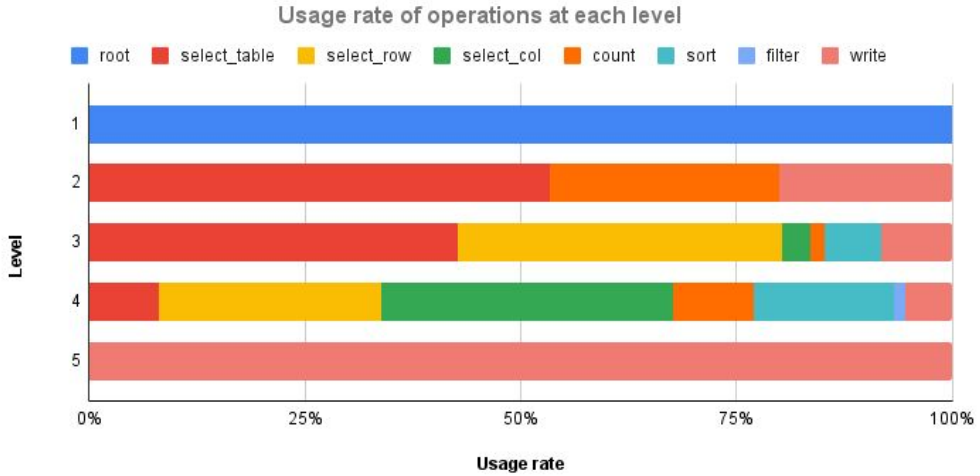
The Influences of Operation Pool

- Overall, maintaining **all operations** provides the most balanced performance, demonstrating greater robustness.

Operation Pool	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Time	Cost
All operations	**15.78**	98.04	**93.94**	93.94	**93.94**	**93.94**	29.04	5.71
w/o `select_table()`	15.44	**98.69**	82.57	92.94	85.57	82.57	44.45	6.11
w/o `select_row()`	15.33	98.04	84.53	94.90	87.53	84.53	48.00	6.80
w/o `select_col()`	15.11	**98.69**	85.19	93.33	86.95	82.96	49.80	7.49
w/o `count()`	15.44	**98.69**	82.57	92.94	85.57	82.57	**25.30**	**4.20**
w/o `sort()`	15.44	**98.69**	85.19	**95.56**	88.18	85.19	36.64	5.64
w/o `filter()`	15.44	**98.69**	82.57	92.94	85.57	82.57	33.34	5.53

The Influences of Operation Pool

- Overall, maintaining **all operations** provides the most balanced performance, demonstrating greater robustness.



The Impacts of Table Formats

- **CSV** achieves the best performance.
- While **PIPE** and **HTML** perform similarly, they have significantly higher time and cost due to requiring more symbols to represent the table.
- **Markdown** performs the worst, likely because LLMs have been pre-trained on fewer examples of this format.

Table Format	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Time	Cost
CSV	**15.78**	**98.04**	**93.94**	**93.94**	**93.94**	**93.94**	**29.04**	**5.71**
PIPE	**15.78**	**98.04**	93.29	93.29	93.29	93.29	78.53	9.63
HTML	15.67	97.39	92.64	92.64	92.64	92.64	104.99	19.26
Markdown	14.67	92.31	87.56	86.75	87.10	84.14	62.65	9.80

6. Conclusion

Conclusion

- **Problem**
  - **Model-based Methods:** The dataset size is limited (e.g., ShuttleSet+ contains only 58 matches)
  - **Prompt-based Methods:** Hallucination issues and shorter outputs.
- **Solution**
  - **Content Planning:** Select appropriate operations and arguments.
  - **Operation Execution:** Execute operations to decompose tables into smaller sub-tables.
  - **Content Generating:** Merge short texts and rewrite them into a long text.
- **Experiment**
  - **Effectiveness:** Experiments show that Tree-of-Text achieves the best performance on ShuttleSet+, leads in RG and CO on RotoWire-FG, and excels in CS and CO on MLB.
  - **Efficiency:** Our method achieves about 40% of Chain-of-Table's time and cost.

Conclusion

- **Limitations**
  - Our approach requires **manually tuning configurations and prompts** for the corresponding dataset.
  - Our proposed approach still requires **higher time and cost** than Few-shot and Chain-of-Thought.
- **Future Works**
  - One of the interesting research directions could be to explore **automatic selection for configurations and prompts**.
  - **Parallel processing** or **knowledge distillation** is a potential direction for further research.

7. Q&A

國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Thank you for listening!

Speaker: Shang-Hsuan Chiang

Advisor: Wen-Chih Peng

Date: 2025/07/21