ACL 2025 VIENNA JULY 27 - AUGUST 1

Tree-of-Report: Table-to-Text Generation for Sports Game Reports with Tree-Structured Prompting

Shang-Hsuan Chiang, Tsan-Tsung Yang, Kuang-Da Wang, Wei-Yao Wang, An-Zi Yen, Wen-Chih Peng

National Yang Ming Chiao Tung University

ACL SRW 2025 (long paper)

Speaker: Shang-Hsuan Chiang
Date: 2025/07/28

Outline

1. Introduction
2. Method
3. Experiment
4. Conclusion
5. Q & A

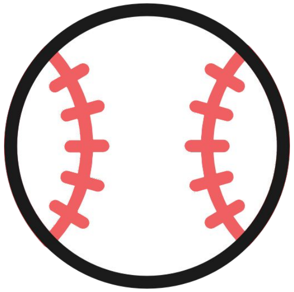
1. Introduction

Introduction

There are many kinds of sports games.

Badminton

Basketball



Baseball

Even more!

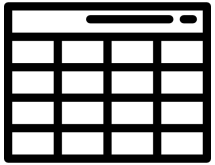
Introduction

How to watch a sports game?

Video

Easy to understand 👍

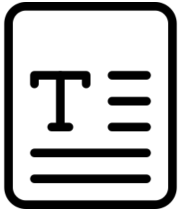
Time-consuming 👎



Table

Time-saving 👍

Hard to understand 👎



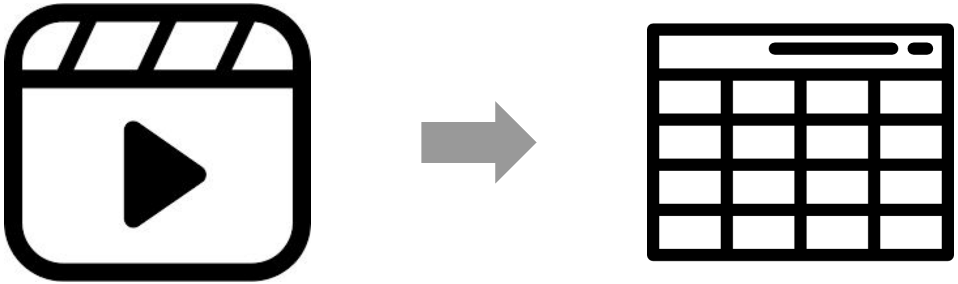
Text

Easy to understand 👍

Time-saving 👍

Introduction

How to convert a video into a table?

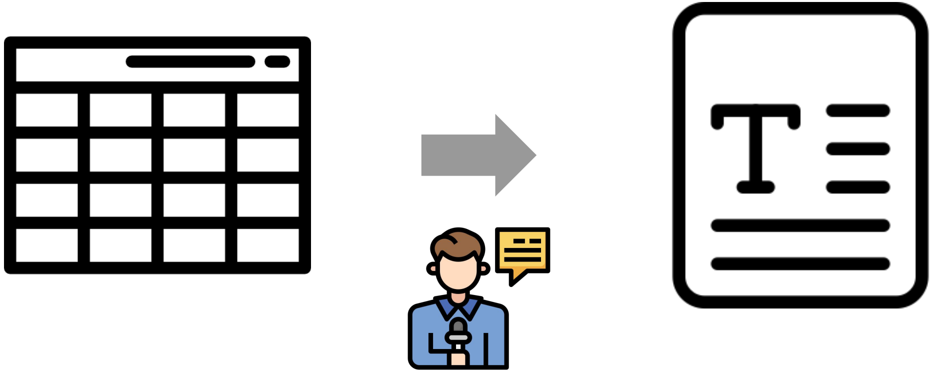


ShuttleSet [1] has already completed this task.

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

Introduction

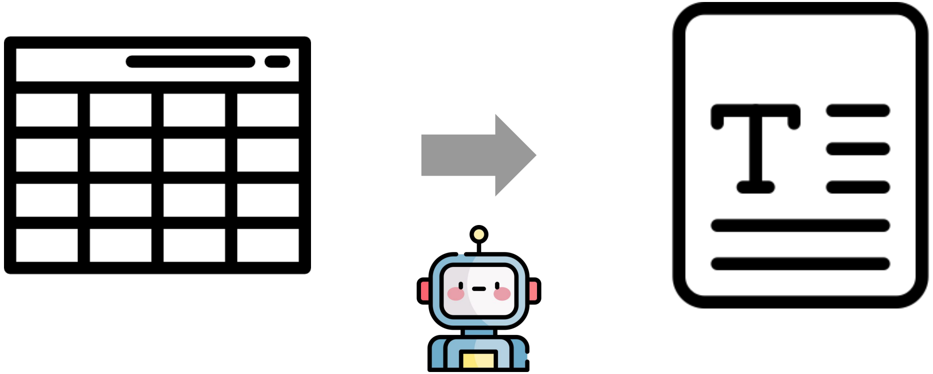
How to convert a table into a text?



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

Introduction

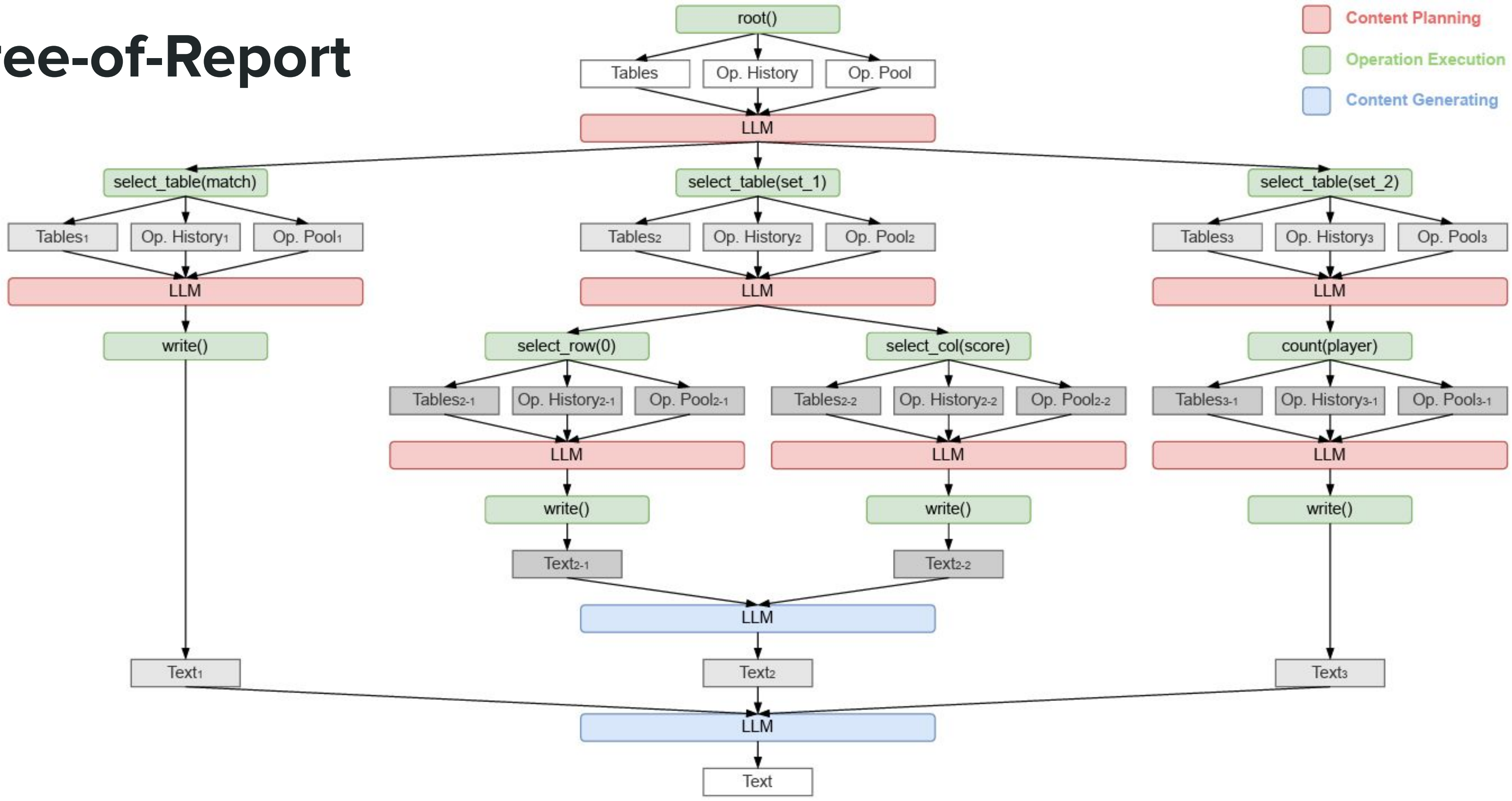
How to convert a table into a text?



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

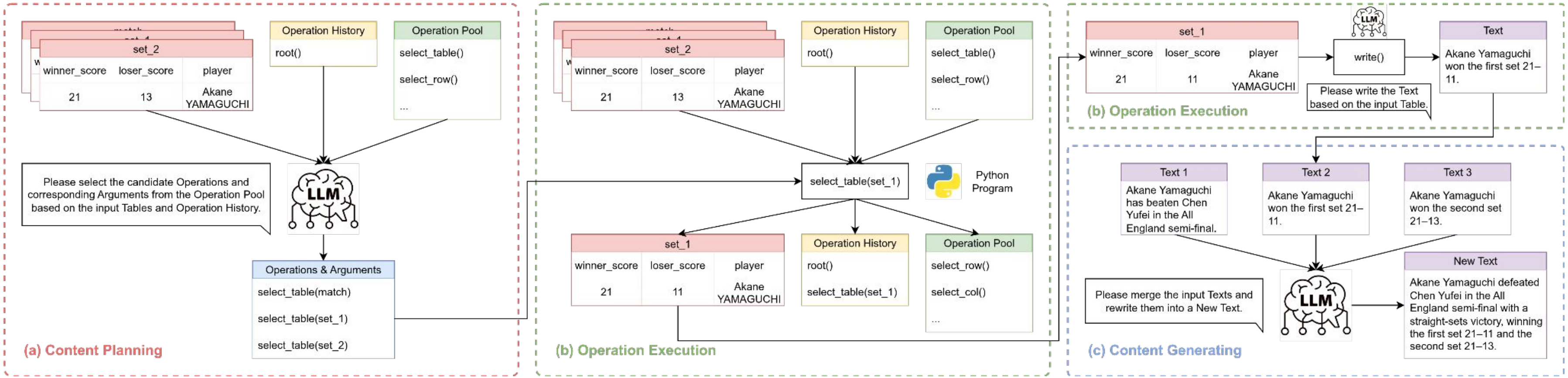
2. Method

Tree-of-Report

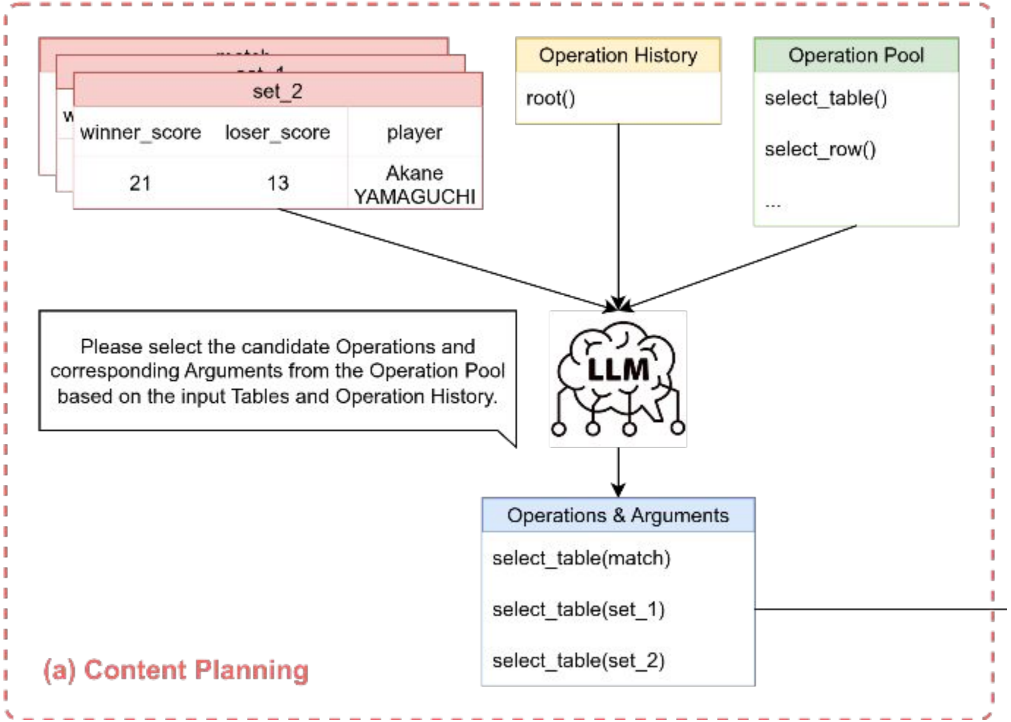


[12] Algorithm 426: Merge sort algorithm [M1]

Tree-of-Report



(1) Content Planning



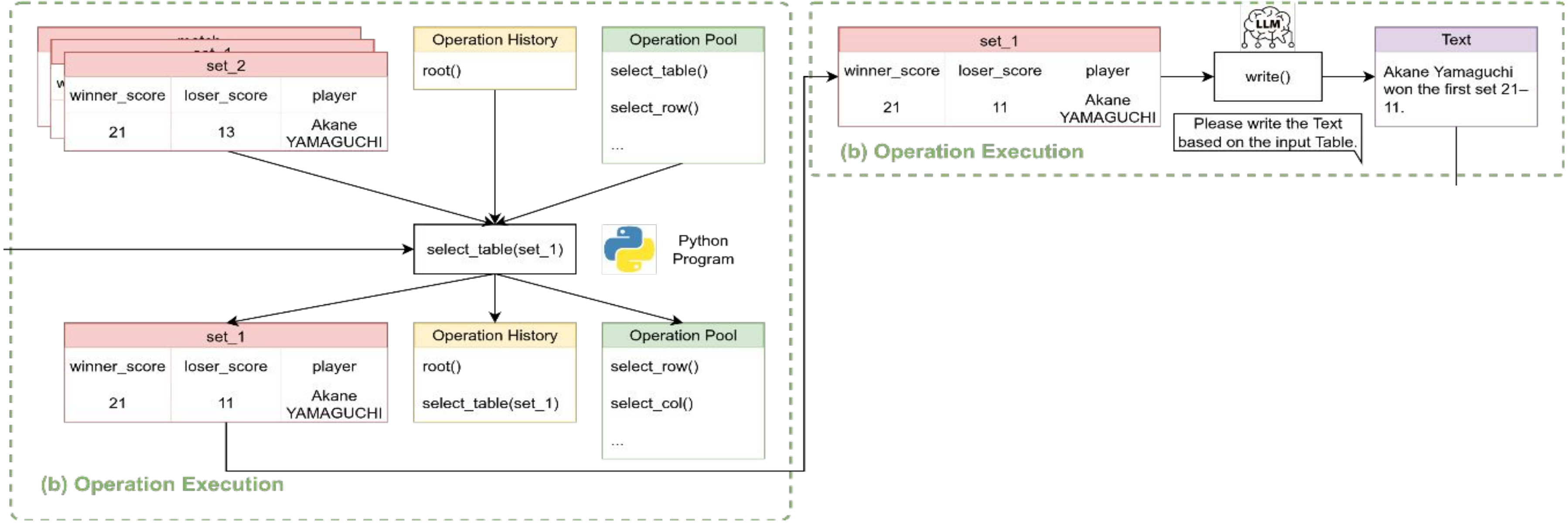
(1) Content Planning

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

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



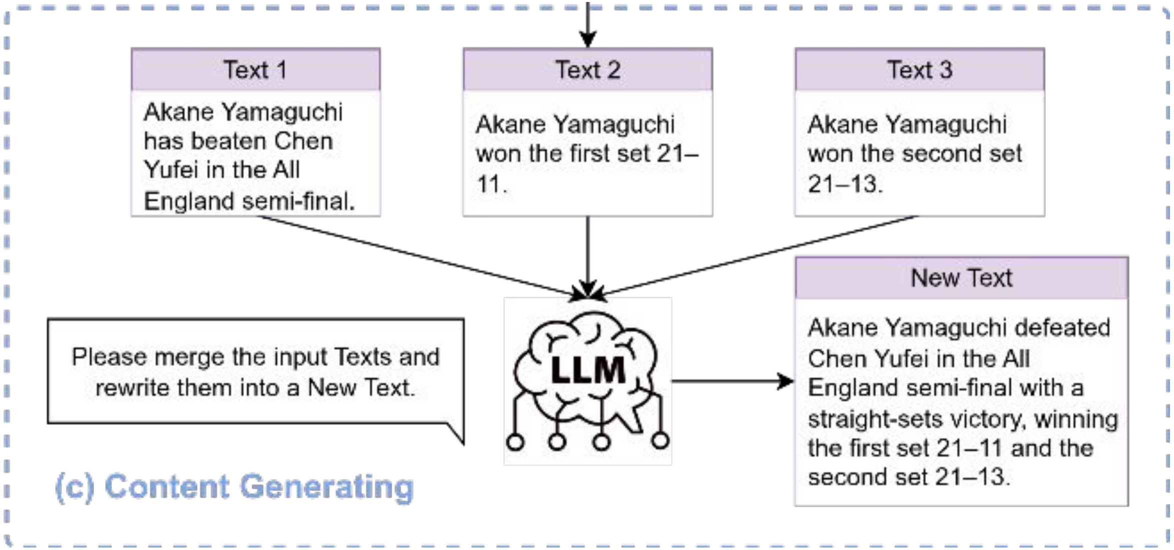
(2) Operation Execution

- **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' | i = 1, 2, ..., d)*

(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**

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 ≥ MAX_DEPTH then
        t ← WRITE(T)
        return t
    end if
                                          ▷ Content Planning
    OA ← CONTENT_PLANNING(T, OH, OP)
                                          ▷ Operation Execution
    t' ← ()
    for each O_i(A_i) | O_i ∈ OP, i = 1, 2, ..., d in OA do
                                          ▷ Degree must not exceed Max Degree
        if i ≥ MAX_DEGREE then
            break
        end if
        if O_i = write() then
            t'_i ← WRITE(T)
        else
            T'_i ← O_i(T, A_i)
            OH'_i ← OH + O_i(A_i)
            OP'_i ← OP − O_i()
            D'_i ← D + 1
            t'_i ← TREE-OF-REPORT(T'_i, OH'_i, OP'_i, D'_i)
        end if
        t' ← t' + t'_i
    end for
                                          ▷ Content Generating
    t ← CONTENT_GENERATING(t')
    return t
end function

Main Program
T ← (T^j | j = 1, 2, ..., n)
OH ← (op | op = root())
OP ← (op | op ∈ operations, op ≠ root())
D ← 0
t ← TREE-OF-TEXT(T, OH, OP, D)
```

3. Experiment

3.1. Dataset

RotoWire 🏀

- Human-written NBA **basketball** game summaries in English paired with their corresponding box and line scores
- **4,853 summaries**
  - Train: 3,398
  - Valid: 727
  - Test: 728

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

MLB ⚾

- **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

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

ShuttleSet+ 🏸

- We introduce a new dataset, ShuttleSet+, derived from **ShuttleSet22.**
- 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

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

3.2. Evaluation Metric

Information Extraction (IE)

- **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 **60%** on all evaluation metrics with **one-shot prompting**.

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

Evaluation Metric

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

Evaluation Metric

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

3.3. Implementation Detail

Implementation Detail

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

3.4. Quantitative Result

RotoWire 🏀

RotoWire	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Avg.	Time	Cost
Zero-shot	29.69	**97.54**	56.70	55.77	50.85	30.77	58.33	7.93	0.63
One-shot	31.17	95.72	57.86	56.26	51.12	29.53	58.10	5.07	0.90
Few-shot	28.27	95.37	57.67	54.45	51.07	30.60	57.83	5.38	1.48
Chain-of-Thought	28.46	96.52	**58.33**	55.57	52.20	32.83	59.09	7.76	0.61
Tree-of-Thought	34.97	95.26	54.43	60.17	52.41	33.66	59.19	54.62	8.18
Chain-of-Table	**41.96**	92.47	53.47	61.53	50.70	32.63	58.16	**63.75**	**12.54**
Tree-of-Report	32.89	96.81	56.98	**63.33**	**54.92**	**36.37**	**61.68**	21.07	2.43
							+2.49%	53%	61%

MLB ⚾

MLB	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Avg.	Time	Cost
Zero-shot	**53.17**	84.22	60.45	60.65	49.03	39.25	58.72	7.08	1.62
One-shot	41.15	90.91	70.94	61.95	56.67	47.32	65.56	8.72	1.77
Few-shot	44.16	89.69	71.64	61.91	56.68	46.80	65.34	10.36	1.78
Chain-of-Thought	39.88	94.11	73.72	63.40	56.79	46.46	66.90	9.34	1.73
Tree-of-Thought	33.78	95.83	73.28	**64.00**	59.34	48.50	68.19	50.96	7.25
Chain-of-Table	28.13	95.37	80.20	60.33	59.60	50.21	69.15	**55.60**	**10.80**
Tree-of-Report	30.78	**97.54**	**84.19**	63.48	**62.99**	**53.78**	**72.40**	29.18	6.77

+3.25% 53% 67%

ShuttleSet+ 🏸

ShuttleSet+	RG #	RG P%	CS P%	CS R%	CS F%	CO DLD%	Avg.	Time	Cost
Zero-shot	13.67	85.19	86.01	86.01	86.01	86.01	85.85	7.53	0.86
One-shot	12.22	84.02	83.26	74.42	78.38	56.99	75.42	6.59	1.12
Few-shot	14.33	90.22	87.72	86.58	86.99	82.31	86.76	6.00	2.20
Chain-of-Thought	13.67	85.53	84.97	84.62	84.70	83.49	84.66	6.68	0.81
Tree-of-Thought	13.33	81.92	81.35	82.48	81.88	81.35	81.80	63.11	9.62
Chain-of-Table	15.00	93.46	89.37	89.37	89.37	89.37	90.19	**73.67**	**14.44**
Tree-of-Report	**15.78**	**98.04**	**93.94**	**93.94**	**93.94**	**93.94**	**94.76**	29.04	5.71

+4.57% 40% 40%

3.5. 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-Report

Longer, detailed, and correct

Tree-of-Report
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.

4. Conclusion

Conclusion

1. In the task of table-to-text generation for sports game reports, we introduce **Tree-of-Report,** a novel framework that recursively decomposes tables into smaller sub-tables, generates short textual descriptions for each sub-table, and merges these short texts into a complete report.
2. We introduce a new sports report dataset, **ShuttleSet+**, containing rally-level data from 58 badminton matches along with the corresponding human-written reports.
3. Tree-of-Report outperforms other prompt-based baselines on the RotoWire, MLB, and ShuttleSet+ datasets while maintaining relatively lower time and cost, demonstrating its superiority in both **effectiveness** and **efficiency**.

5. Q & A

Thank you for listening!

Shang-Hsuan Chiang

Department of Computer Science, National Yang Ming Chiao Tung University,

Hsinchu, Taiwan

andy10801@gmail.com

Applying for PhD!

Appendix

A. Example Prompt

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
```

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
```

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
```

B. Data Preprocessing

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+.

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.

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.

C. LLM-based IE Model

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 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
```

D. 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 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