link https://www.overleaf.com/project/660283be0dacb760e1c19ff7

补充实验：

Few Shot Result

Model	Setting	Easy		Medium		Hard		Average	
		RMSE	ER	RMSE	ER	RMSE	ER	RMSE	ER
gpt-3.5-turbo-16k	0-Shot	0.222	12.00	1.757	65.09	3.216	87.20	2.082	57.34
	0-Shot-CoT	0.234	12.73	1.747	64.36	3.136	87.40	2.045	57.21
	1-Shot	0.251	13.53	2.290	71.85	4.330	90.39	2.759	61.90
	1-Shot-CoT	0.176	8.82	3.184	73.81	4.524	91.45	3.239	61.97
gpt-4o	0-Shot	0.002	0.13	1.505	50.40	1.818	78.45	1.432	44.84
	0-Shot-CoT	0.003	0.20	1.311	41.65	1.998	78.12	1.410	40.40
	1-Shot	0.003	0.19	1.236	49.01	2.188	78.78	1.435	44.25
	1-Shot-CoT	0.003	0.13	1.193	44.96	2.722	84.28	1.628	43.58
	5-Shot	0.003	0.20	1.710	57.24	2.147	80.00	1.654	48.62
	5-Shot-CoT	0.002	0.13	1.552	56.32	2.024	72.90	1.526	46.38
LLaMA2-7B	QA-Fine-Tune	0.759	29.18	3.485	77.95	7.187	95.82	4.441	70.23
Mistral-7B		0.631	25.74	3.151	74.87	7.209	96.86	4.325	68.08
LLaMA3-8B		0.561	23.15	3.272	71.08	8.553	95.73	4.971	65.26

Case Study

LiveSum的Case Study见Appendix G

以Wiki40B为例，case study了50篇文章：

- Text-to-Tuple：平均正确率 81.6%
  - 抽出原句子
  - 不完整的三元组
  - 重复
- Tuple Integration: 有24%发生了同类项的合并，其余只是简单的排序或者重复输出
- Tuple-to-Table
  - 14% 把长句子粉碎了，只留下了不完整的三元组
  - 6% 在tuple的基础上继续合并同类项，把相关句子合并成了长句子

我们总结出现performance gap的原因：

- 长句子 -> 在 tuple-to-table 阶段出现错误

- tuple合并同类项后，句子长度增加，增加推理难度
-

General Response

(writing…)

We would like to express our sincere appreciation to all the reviewers for their insightful comments and valuable feedback.

We are excited that you recognize that:

* Our proposed LiveSum dataset, aiming at information aggregation, is **highly innovative and forward-thinking** (HhJM, Qr8Y), and **allows for broader use and more granular analysis** because of its division into three difficulty categories (Xtzn).
* Our proposed method, T3 (Text-Tuple-Table), **clearly details the prompting process** (Qr8Y) and **demonstrates significant performance improvements** (HhJM).
* Our paper **conducts thorough experiments** with a variety of LLMs and **provides comprehensive performance comparisons**. (HhJM, Xtzn, Qr8Y)
* Our paper is **well-structured with notable contributions** while stating the shortcomings of existing models (HhJM).

Moreover, we deeply appreciate the valuable suggestions and will incorporate them to improve our manuscript. These enhancements will include:

* We use a QA-based fine-tuning method to enhance model performance on the LiveSum dataset, analyze the reasons for the poor performance of current fine-tuning methods, and demonstrate the necessity of a well-designed fine-tuning approach for this task.
* We conduct an error analysis on the application of T3 to the Wiki40B dataset (Figure 5), identifying the sources and proportions of errors in the T3 pipeline.
* We test LLMs supporting large maximum tokens (e.g. gpt-3.5-turbo-16k, gpt-4o) in a few-shot setting on the LiveSum dataset, enhancing the comprehensiveness of the evaluation.

Thank you and please feel free to ask if you have any further questions.

Best Regards,

Submission 388 Authors

Review 1

Paper Summary:

The paper proposes both a new dataset, LiveSum, and a new prompting method, Text-Tuple-Table. LiveSum consists of football commentary data, with corresponding tables created by human workers– while previous text-to-table datasets consist of text which is already highly similar to table format, LiveSum requires models to convert complex streams of text into tables of information, which is more applicable. Text-Tuple-Table involves first extracting tuples (subject, object, verb/value) from the input text, and then converting them into table format. This effectively organizes the complex input text into a “nice” format, which improves performance on text-to-table generation.

Summary Of Strengths:

The paper demonstrates massive increases in performance when using the Text-Tuple-Table method, which validates its benefits. In addition, the paper highlights an important advancement towards more nuanced forms of table generation, and paves the way for future work through the LiveSum dataset. The paper is also well-organized and clearly highlights its contributions while stating the shortcomings of existing methods. The paper provides comprehensive performance comparisons and intricate details about its dataset generation and testing settings.

Summary Of Weaknesses:

As the LiveSum dataset was hand-generated, it is difficult to scale this dataset creation method for larger datasets. It also lacks thorough explanation on the poor performance of existing models on this benchmark, especially after fine-tuning. A few example generations for some of the models could be useful to understand what kinds of mistakes these models are making.

Comments Suggestions And Typos:

Page 4, lines 265, 274, typo for “Instruction”

Author Response:

Thanks for the review and your recognition in [make a list to cover strengths again]

1. Current benchmark generation paradigms mainly cover (i) human annotation (ii) machine generated ones with human verification (iii) reformation of existing resources.

   For (i), it’s very expensive and time consuming to ask human annotators to curate tables, which makes it impractical to construct large benchmark for our paper.

For (ii), our experiments show that LLMs cannot perform well, thus it's impossible to incorporate it into benchmark curation pipeline.
We adopt (iii), reformation of existing resources, [Explain our data curation novelty here]. It is the only available way to curate high-quality and scalable benchmark. [Explain we only use ChatGPT to transform writing (not sure about this), this can be done automatically, so we can scale up]

2. add some explanations for poor performances, maybe add some guesses, such as it requires counting and numerical reasoning from text, which are shown to be difficult by [some previous works].

3. Add a small error analysis table presenting some errors. Or, alternatively, we can do this:

> Among 60 annotated error samples from INTENTUTILIZE, we found:
>
> - 40% errors are due to inaccurate understanding of the given intention. For example, the model chooses "iPod" when the given intention is "because the customer wanted to use them *with* his/her iPod".
> - 38.3% errors are due to inaccurate understanding of the given products. The reasoning in their response demonstrates inaccurate understanding of the purchased products or those in the options. Or, when the intention is not typical enough to filter out distractors, they fail to rely more on the purchased product to select the best option.
> - 21.7% errors are due to false-negative distractors or incorrect ground truth answers.

But the reviewer actually requires examples…So maybe do case studies with error analysis (guess the reason why LLM makes error)

Actually I did case studies in Appendix G. Maybe because it's at Page 17, he missed it LOL. Maybe it's enough 🤔

G Case Studies

Figure 8 lists the outputs of four LLMs with and without the application of the T^3 method on the data shown in Figure 6. For the results not utilizing T^3, we can not easily analyze why it generates a range of large or small values. However, for the results using the T^3 method, we perform a detailed examination. We randomly sample 100 results generated by GPT-4 applying T^3 and conduct a spot check, finding that all errors originated from the first stage. Among these errors, 78% are due to missing event tuples, and 21% are due to wrong event tuples. Here, we present two representative examples.

Missing Event Tuple Example

Player18(Home Team) earns a free kick on the left wing after being fouled by Player20(Away Team).

```
(Player18, Home Team, Free Kick)
(Player20, Away Team, Foul)(missing)
```

Wrong Event Tuple Example

Player17(Home Team) from the Home Team draws a foul in the penalty area, resulting in a penalty conceded by Player33(Away Team).

```
(Player17, Home Team, Foul)(wrong)
(Player33, Away Team, Foul)
(Player17, Home Team, Free Kick)(missing)
```

4. Thanks for your advice! We will fix the typo accordingly.

Reviewer 1

Thanks for your dedicated review and your recognition in the:

- forward-looking nature of our proposed LiveSum dataset.
- the well-organized structure of our paper.
- the clarity of our contributions.
- the comprehensiveness of our experiments.

We hope the following paragraphs can address your concerns one by one.

> As the LiveSum dataset was hand-generated, it is difficult to scale this dataset creation method for larger datasets.

（我因为确实是人工标注的label，所以感觉辩不了这句话了）

While we acknowledge that a large amount of manual annotation is required to ensure the quality of our dataset, it is worth noting that the size of the dataset we have annotated is comparable to that of current mainstream text-to-table datasets, such as Struct-Bench[1]. As our experiments have shown, LLMs perform poorly on such task, making them infeasible to be the curator of large-scale datasets. Automatic parsing methods cannot generalize between different corpora, which require human-defined rules for each dataset. Thus, human annotation is indeed the only reliable way for constructing high quality evaluation benchmarks. We do anticipate future advancements of LLMs to enable large-scale dataset curation.

> It also lacks thorough explanation on the poor performance of existing models on this benchmark, especially after fine-tuning.

Since fine-tuning is end-to-end, we are unable to explicitly analyze the reasons why the fine-tuned model performs poorly. However, we conjecture that this task requires the model to learn the ability to perform counting and numerical reasoning within the text, which is difficult. Carefully designed methods may be needed to fine-tune the model. In order to verify our hypothesis, we conduct a new set of experiments, fine-tuning according to the method described in [2], treating each cell as a Question-Answering task for fine-tuning. Although this approach is relatively inefficient, where time is traded for performance, the results can serve as a reference.

Model	Easy-RMSE	Easy-ER	Medium-RMSE	Medium-ER	Hard-RMSE	Hard-ER	Average-RMSE	Average-ER
LLaMA-2-7B	0.759	29.18	3.485	77.95	7.187	95.82	4.441	70.23
Mistral-7B-Instruct-v0.2	0.631	25.74	3.151	74.87	7.209	96.86	4.325	68.08
LLaMA-3-8B	0.561	23.15	3.272	71.08	8.553	95.73	4.971	65.26

It is evident that Mistral-7B-Instruct-v0.2 and LLaMA-2-7B, under the fine-tune settings of [2], show significant improvements in performance on the Easy, Medium, and Average section compared to the

fine-tuning effects of Struc-Bench (see Table 1), with basically no change in performance on the Hard section. The experimental results show that for complex tasks, directly fine-tuning the model on the original text and tables may yield very poor performance, some carefully constructed fine-tuning methods may be needed to further improve the poor performance.

> A few example generations for some of the models could be useful to understand what kinds of mistakes these models are making.

We have listed the results of the case study with error analysis in Section 6.4, which is linked to Appendix G.

In addition to conducting a case study on the LiveSum dataset, we also carry out a supplementary case study on the Wiki40B dataset. We sample 50 articles and observe the following situations:

* In the first phase, the average accuracy of the tuples obtained from the text-to-tuple process is 81.6%. We define erroneous tuples as those containing: (i) direct extractions of original sentences, (ii) incomplete triples, and (iii) duplicate triples.
* In the second phase, tuple integration, 24% of the articles undergo integration of similar items, while the remainder simply involves sorting and categorizing the tuples, or outputting them unchanged.
* In the third phase, tuple-to-table essentially involves treating each tuple as a row to be filled into a table, but two scenarios may introduce errors: (i) 14% of the articles involve the fragmentation of long sentences, leaving only incomplete tuples, and (ii) 6% of the articles continue to merge similar items based on the tuples, combining related tuples into long sentences.

In summary, we identify the main sources of errors in the T3 pipeline, which primarily stem from the text-to-tuple stage. Erroneous tuples, incomplete tuples, or tuples composed of long sentences can accumulate errors in subsequent stages.

> Page 4, lines 265, 274, typo for “Instruction”

Thanks for your advice! We will fix the typo accordingly.

[1] Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, Mark Gerstein. 2023. Struc-Bench: Are Large Language Models Really Good at Generating Complex Structured Data? Preprint, arXiv:2309.08963

[2] Anirudh Sundar, Christopher Richardson, Larry Heck. 2024. gTBLS: Generating Tables from Text by Conditional Question Answering. Preprint, arXiv:2403.14457

Soundness: 4.5
Overall Assessment: 4.5

Review 2

Paper Summary:

This paper introduces LIVESUM, a new benchmark dataset designed for generating summary tables of competitions based on real-time commentary texts. It evaluates the performance of state-of-the-art large language models (LLMs) on this task in both fine-tuning and zero-shot settings. Additionally, it proposes a novel pipeline called T3 (Text-Tuple-Table) to enhance their performance. Extensive experimental results demonstrate that LLMs continue to struggle with this task even after fine-tuning.

Summary Of Strengths:

1. This paper conducted thorough experiments to evaluate the performance of the dataset and the T3 method. It utilized approximately 20 models, ranging from fine-tuning to zero-shot settings, providing a comprehensive and valid assessment. The study included an ablation analysis for different prompting methods and tested various metrics to ensure robust evaluation.
2. The dataset was categorized into easy, medium, and hard sections, facilitating its use by others and enabling more granular analysis and benchmarking.

Summary Of Weaknesses:

1. The T3 method proposed in this paper follows and combines previous works, including InstructUIE[1] and Structsum generation[2] without introducing novel designs in the prompting or methodology.
2. The experiments did not include a few-shot setting, which limits the comprehensiveness of the evaluation.

Comments Suggestions And Typos:

N/A

Author Response:

(Weiqi)

Thanks for the review and your recognition in [make a list to cover strengths again]

1. introduce why not simply combine existing works. Maybe point out the difference and why previous methods won't fit into our proposed task.

2. Add some few-shot experiments, (and also my fine-tuning QA experiments) to show that all methods fail.

We sincerely appreciate your valuable advice and hope that our response will assist you in raising your score. Thank you once again!

Final Version

```
Thanks for your kind comments and your recognition in:

- our detailed experiments,

- comprehensive analysis,

- useful categorization of dataset difficulties.



In the following paragraph, we hope to address your concerns one by one.
```

> The T3 method proposed in this paper follows and combines previous works, including InstructUIE[1] and Structsum generation[2] without introducing novel designs in the prompting or methodology.

We would like to clarify that this paper is not a combination of InstructUIE[1] and StructSum[2]. Instead, this paper is the first to propose a pipeline method in the text-to-table task that involves initially extracting tuples and then integrating these tuples to form a table, which significantly outperforms current approaches. InstructUIE[1] is merely one implementation method used in the first phase of our approach. As for the implementation of the third phase, we follow mainstream methods, i.e. the prompt used in StructSum[2] and Struct-Bench[3].

It is worth noting that the main contribution of StructSum[2] lies in their initial use of prompts to segment the input text. The subsequent step of generating a table from text is the same as Struct-Bench[3], which also use prompts. We also follow this common practice. Therefore, our approach is orthogonal to StructSum[2], making them compatible for combined use.

> The experiments did not include a few-shot setting, which limits the comprehensiveness of the evaluation.

Thank you for your suggestion. Because the input lengths of our dataset generally exceed 2000 tokens, even testing with 1-shot would surpass the maximum token limit (4096) of most LLMs. Therefore, we do not test the few-shot performance in the original paper. However, we select two models, gpt-3.5-turbo-16k and gpt-4o, for supplementary experiments, and the results are as follows. Regarding the selection of few-shot examples, we randomly choose samples from the

training set where all events occur. In the Chain of Thought (CoT) setting, the reasoning process is included in the examples.

gpt-3.5-turbo-16k

Model	Easy-RMSE	Easy-ER	Medium-RMSE	Medium-ER	Hard-RMSE	Hard-ER	Average-RMSE	Average-ER
0-Shot	0.222	12.00	1.757	65.09	3.216	87.20	2.082	57.34
0-Shot (CoT)	0.234	12.73	1.747	64.36	3.136	87.40	2.045	57.21
1-Shot	0.251	13.53	2.290	71.85	4.330	90.39	2.759	61.90
1-Shot (CoT)	0.176	8.82	3.184	73.81	4.524	91.45	3.239	61.97

gpt-4o

Model	Easy-RMSE	Easy-ER	Medium-RMSE	Medium-ER	Hard-RMSE	Hard-ER	Average-RMSE	Average-ER
0-Shot	0.002	0.13	1.505	50.40	1.818	78.45	1.432	44.84
0-Shot (CoT)	0.003	0.20	1.311	41.65	1.998	78.12	1.410	40.40
1-Shot	0.003	0.19	1.236	49.01	2.188	78.78	1.435	44.25
1-Shot (CoT)	0.003	0.13	1.193	44.96	2.722	84.28	1.628	43.58

| 5-Shots | 0.003 | 0.20 | 1.710 | 57.24 | 2.147 | 80.00 | 1.654 | 48.62 |

| 5-Shots (CoT) | 0.002 | 0.13 | 1.552 | 56.32 | 2.024 | 72.90 | 1.526 | 46.38 |

The results indicate that few-shot learning does not consistently improve model performance on this task, although there is a significant reduction in the error rate for the hard portion under the 5-shots-CoT setting. We speculate that in such complex tasks, the model may not effectively learn and understand the examples, and instead, be influenced by the examples, leading to incorrect answers.

We sincerely appreciate your valuable advice and hope that our response will assist you in raising your score. Thank you once again!

[1] Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. Preprint, arXiv:2304.08085.

[2] Parag Jain, Andreea Marzoca, and Francesco Piccinno. 2024. Structsum generation for faster text comprehension. Preprint, arXiv:2401.06837

[3] Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, Mark Gerstein. 2023. Struc-Bench: Are Large Language Models Really Good at Generating Complex Structured Data? Preprint, arXiv:2309.08963

Soundness: 3 = Acceptable: This study provides sufficient support for its major claims/arguments. Some minor points may need extra support or details.
Overall Assessment: 3 = Good: This paper makes a reasonable contribution, and might be of interest for some (broad or narrow) sub-communities, possibly with minor revisions.

Review 3

Paper Summary:

This paper introduces LIVESUM, a dataset for generating summaries based on commentary texts of soccer games. Additionally, the paper proposes Text-Tuple-Table (T3), a pipeline to summarize these commentaries into a tabular form. Results from the paper indicate that T3 achieves the best RMSE on LIVESUM under a zero-shot setting.

Summary Of Strengths:

3. In contrast to prior text-to-table datasets that invert existing tables to generate paragraphs, LIVESUM aggregates information from match summaries.
4. A clear description of the prompting process used in T3, the proposed approach to LIVESUM
5. By implementing code generation to aggregate events, the approach is able to count said events for accurate summarization
6. Extensive experiments with a variety of LLMs

Summary Of Weaknesses:

1. 3.2 line 209-211, some additional details on how the workers were compensated is necessary as per the responsible NLP checklist item D2 (https://aclrollingreview.org/responsibleNLPresearch/)
2. How does the tuple-to-table framework in 4.3 handle the situation where some tuples are missing? Is the model instructed to pad the missing cells with zeros/unks?
3. 6.3.1, line 452-454: it is unclear on what this modification to the model's outputs are
4. Additional details on the fine-tuning experiments are required. Is this fine-tuning performed end-to-end (text-to-table) or step-by-step (text-to-tuple, information integration, tuple-to-table)? This would help provide some clarity to the statement in lines 339-340 about the zero-shot experiments outperforming fine-tuning.
5. Furthermore, it is concerning that fine-tuning is outperformed by T3 on LIVESUM but the trend is the other way round on Struc-Bench. Some clarity on this difference is required.
6. Lines 503-505 suggests that an error-propagation problem exists, the authors could provide some intermediate results on the accuracy of the text-to-tuple, information integration stages, and tuple-to-table independently
7. The evaluation metrics are unclear
    a. Is there a check to ensure the output table is the same size as the ground truth?
    b. Does the RMSE consider the difference in the magnitude of outputs (that is, does the metric penalize models for generating values that significantly differ from the ground truth)? Some equations detailing the evaluation metrics would alleviate these concerns

Comments Suggestions And Typos:

1. line 496: typo in Divide

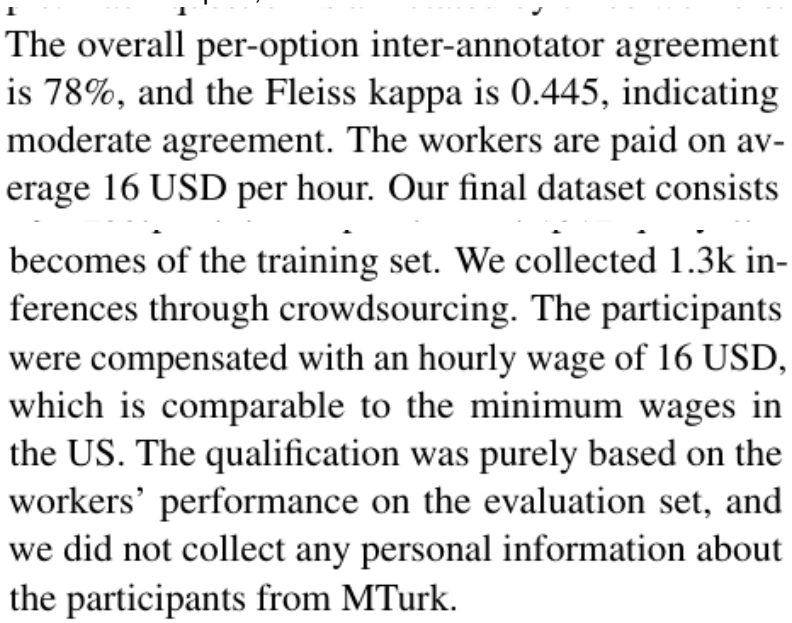
2. Section 3.1, since ChatGPT is used to generate the dataset, it would be helpful to have the prompt template from Appendix A.1 in the main paper
3. Some additional work on table generation and additional datasets could be mentioned in Related Work (https://arxiv.org/pdf/2206.04045, https://openreview.net/pdf?id=gs4swxtIAQ, https://arxiv.org/pdf/2404.12580)

Author Response:

(Weiqi)

Thanks for the review and your recognition in [make a list to cover strengths again]

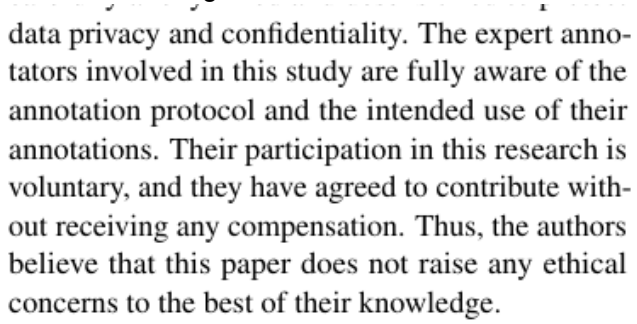
1. Shit… This is severe 😕. It could actually cause a desk rejection LOL.
   If the workers are paid, we need to add



The overall per-option inter-annotator agreement is 78%, and the Fleiss kappa is 0.445, indicating moderate agreement. The workers are paid on average 16 USD per hour. Our final dataset consists

becomes of the training set. We collected 1.3k inferences through crowdsourcing. The participants were compensated with an hourly wage of 16 USD, which is comparable to the minimum wages in the US. The qualification was purely based on the workers' performance on the evaluation set, and we did not collect any personal information about the participants from MTurk.

   https://arxiv.org/pdf/2403.07398
   If the workers are graduate students, we'd better add:



data privacy and confidentiality. The expert annotators involved in this study are fully aware of the annotation protocol and the intended use of their annotations. Their participation in this research is voluntary, and they have agreed to contribute without receiving any compensation. Thus, the authors believe that this paper does not raise any ethical concerns to the best of their knowledge.

😱 I forgot to mention I hire 5 phd students, directly use workers…

2. Seems too detailed, yours to answer 🙂
3. Tiny issue, explain a bit
4. I thought it's end2end, is it really possible to do step-by-step? Then we need one LLM per step, then it's not only one LLM but rather a mixture of expert LLMs…
5. This is indeed interesting… I think we can do the same as answering Reviewer 1.2 and 1.3
6. Same as 2.
7. Same as 2.

Thanks for your constructive and thoughtful comments on our paper and your recognition in

- the novelty of our LiveSum,

- the clarity of our method's description,

- the thoroughness of our experiments.

We hope the following paragraphs can address your concerns one by one.

> 3.2 line 209-211, some additional details on how the workers were compensated is necessary as per the responsible NLP checklist item D2 (https://aclrollingreview.org/responsibleNLPresearch/)

We would like to clarify that the five workers involved are postgraduate students who have voluntarily agreed to participate in this research without receiving any compensation.

> How does the tuple-to-table framework in 4.3 handle the situation where some tuples are missing? Is the model instructed to pad the missing cells with zeros/unks?

This step is performed by the LLMs based on the prompts. If not explicitly restricted in the instructions, the model may default to several possibilities: leaving it blank, filling in "unknown," filling in "not mentioned", etc. (see lines 449-452)

> 6.3.1, line 452-454: it is unclear on what this modification to the model’s outputs are

Here the modification involves removing words such as "unknown" and "not mentioned", and replacing them with empty spaces. This is done because, as mentioned in lines 449-452, in the ground truth, cells in the table that are not mentioned in the original text are left empty.

> Additional details on the fine-tuning experiments are required. Is this fine-tuning performed end-to-end (text-to-table) or step-by-step (text-to-tuple, information integration, tuple-to-table)? This would help provide some clarity to the statement in lines 339-340 about the zero-shot experiments outperforming fine-tuning.

It is end-to-end. In Table 1, both the Fine-Tune and Zero-Shot sections are end-to-end, while only T3 is step-to-step. Therefore, the conclusion in lines 339-340 that zero-shot experiments outperform fine-tuning is drawn under the same testing settings.

> Furthermore, it is concerning that fine-tuning is outperformed by T3 on LIVESUM but the trend is the other way round on Struc-Bench. Some clarity on this difference is required.

>

Firstly, we introduce the characteristics of Struc-Bench[1] in the introduction section (lines 46-64), specifically that it involves only the extraction of text without the need for integration, which allows fine-tuning to perform well. However, fine-tuning using the Struc-Bench approach does not yield high results on the LiveSum dataset. We also conduct additional experiments, fine-tuning according to the method described in [2], treating each cell as a Question-Answering task for fine-tuning. Although this approach is relatively inefficient, the results can serve as a reference.

Model	Easy-RMSE	Easy-ER	Medium-RMSE	Medium-ER	Hard-RMSE	Hard-ER	Average-RMSE	Average-ER
LLaMA-2-7B	0.759	29.18	3.485	77.95	7.187	95.82	4.441	70.23
Mistral-7B-Instruct-v0.2	0.631	25.74	3.151	74.87	7.209	96.86	4.325	68.08
LLaMA-3-8B	0.561	23.15	3.272	71.08	8.553	95.73	4.971	65.26

It is evident that Mistral-7B-Instruct-v0.2 and LLaMA-2-7B, under the fine-tune settings of [2], show significant improvements in performance on the Easy, Medium, and Average section compared to the fine-tuning effects of Struc-Bench (see Table 1), with basically no change in performance on the Hard section. This indicates that a carefully designed fine-tuning method, where time is traded for performance, can still achieve comparable performance to zero-shot on such a complex task, but there is still a significant gap compared to the performance of T3.

> Lines 503-505 suggests that an error-propagation problem exists, the authors could provide some intermediate results on the accuracy of the text-to-tuple, information integration stages, and tuple-to-table independently

Thanks for your suggestion. We carry out a supplementary case study on the Wiki40B dataset. We sample 50 articles and observe the following situations:

* In the first phase, the average accuracy of the tuples obtained from the text-to-tuple process is 81.6%. We define erroneous tuples as those containing: (i) direct extractions of original sentences, (ii) incomplete triples, and (iii) duplicate triples.

* In the second phase, tuple integration, 24% of the articles undergo integration of similar items, while the remainder simply involves sorting and categorizing the tuples, or outputting them unchanged.

* In the third phase, tuple-to-table essentially involves treating each tuple as a row to be filled into a table, but two scenarios may introduce errors: (i) 14% of the articles involve the fragmentation of long sentences, leaving only incomplete tuples, and (ii) 6% of the articles continue to merge similar items based on the tuples, combining related tuples into long sentences. (While this step may not necessarily introduce errors, it could increase the difficulty of reasoning, and the AutoQA metric may decrease.)

In summary, we identify the potential issues in the T3 pipeline as follows: (i) errors in tuple extraction, (ii) the extracted tuples are long sentences, which lead to errors in subsequent processes, and (iii) the extracted tuples are merged with similar items, resulting in increased sentence length and making the reasoning more challenging.

> The evaluation metrics are unclear

> 1. Is there a check to ensure the output table is the same size as the ground truth?

> 2. Does the RMSE consider the difference in the magnitude of outputs (that is, does the metric penalize models for generating values that significantly differ from the ground truth)? Some equations detailing the evaluation metrics would alleviate these concerns

1. It depends. For LiveSum, we check all the output tables and there are no errors as all output tables are consistently 3 rows by 9 columns. For Struc-Bench, the evaluation criteria include assessments of the size of the output table, specifically through the Format H-score and Format P-score. For Wiki40B, since there is no ground truth available, the size of the output tables is not checked.

2. No. We believe that the inherent nature of RMSE to amplify particularly severe deviations is precisely why we choose it. The equation is as follows$$RMSE=\sqrt{\frac{\sum_{i=1}^n(y_i-\tilde{y}_i)^2}{n}}$$

where y_i and $\tilde{y}_i$ represent the contents of the cell at index i (for 2-d tables, the index is calculated after reshaping to the 1-d table) in the ground truth table and the output table, respectively.

Regarding the "Comments Suggestions and Typo section", thank you for your suggestions. We will incorporate these changes in the camera-ready version accordingly.

[1] Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, Mark Gerstein. 2023. Struc-Bench: Are Large Language Models Really Good at Generating Complex Structured Data? Preprint, arXiv:2309.08963

[2] Anirudh Sundar, Christopher Richardson, Larry Heck. 2024. gTBLS: Generating Tables from Text by Conditional Question Answering. Preprint, arXiv:2403.14457

Confidence: 4 = Quite sure. I tried to check the important points carefully. It's unlikely, though conceivable, that I missed something that should affect my ratings.
Soundness: 4 = Strong: This study provides sufficient support for all of its claims/arguments. Some extra experiments could be nice, but not essential.