南京航空航天大學
Nanjing University of Aeronautics and Astronautics

组会汇报

2025.10.21

TAP4LLM: Table Provider on Sampling, Augmenting, and Packing Semi-structured Data for Large Language Model Reasoning

Yuan Sui[1*†], Jiaru Zou[2*†], Mengyu Zhou[3‡], Xinyi He[4†],
Lun Du[5§], Shi Han[3], Dongmei Zhang[3]
[1] National University of Singapore [2] University of Illinois Urbana-Champaign
[3] Microsoft [4] Xi'an Jiaotong University [5] Ant Research
yuansui@comp.nus.edu.sg, jiaruz2@illinois.edu, hxyhxy@stu.xjtu.edu.cn
{mezho, shihan, dongmeiz}@microsoft.com, dulun.dl@antgroup.com

EMNLP 2024

背景

1）现有表格推理方案忽略大型表格，且其存在关键信息分散、信息缺失等问题；

2）LLMs上下文窗口有限，且当表格信息模糊时易产生误解或幻觉。

3）表格需转换为 LLMs 可理解格式，同时要在有限 token 预算内平衡表格内容与补充信息占比

优化思路

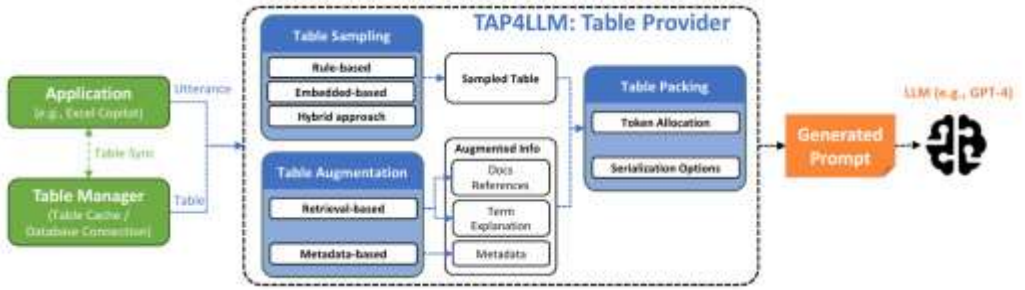
1.表格采样：将大表T分解为具有特定行和列的子表T'

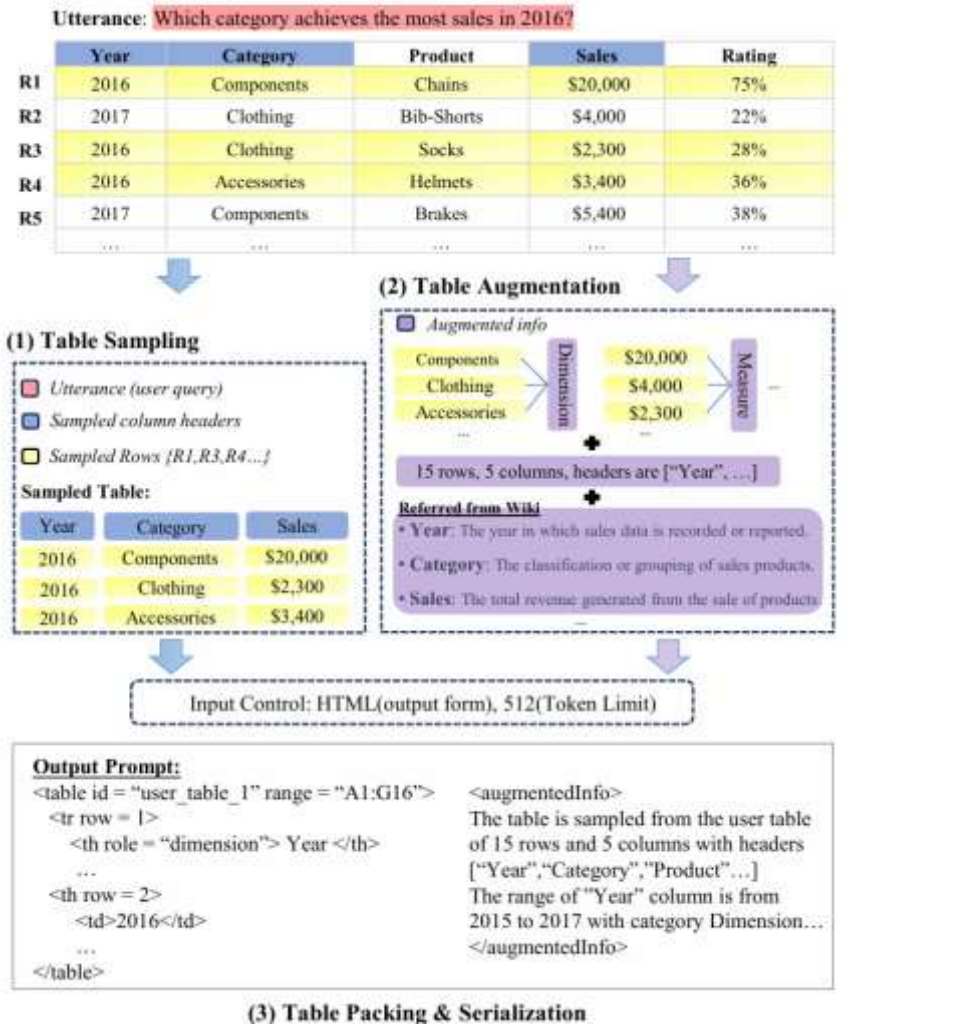
- 基于规则
- 基于embedding
- 基于LLMs

2.表格增强：显式地融入与原始表T相关的外部知识、元数据和属性

- 基于元数据
- 基于检索
- 基于一致性

3.数据打包：HTML,XML,JSON,CSV,NL+Sep,...





Methods — 表格采样　　将大表格分解为可处理的子表格

南京航空航天大學 Nanjing University of Aeronautics and Astronautics

1.基于规则的采样 —— 随机采样

① 每轮采样完全随机取k个样本

② 设定评估指标，计算每个子表与查询的相关性得分，得分最高者为最终样本

2.基于规则的采样 —— 均匀采样

① 设定最大token限制

② 从表头及表尾交替取样直到达到最大token限制

3.基于规则的采样 —— 内容快照采样

① 依据 n - gram 重叠率来选取与查询话语相关的行

② 选择重叠率最高的若干行作为采样结果

结论：不同类型采样方法性能有差异，基于嵌入的采样表现突出，其中语义采样结合列采样效果佳，无采样处理大表格会使性能严重下降。

4.基于嵌入的采样

语义采样
质心采样
} 混合采样 $h(r,c,u)=\alpha\left(\frac{1}{1+D(r,c)}\right)+\beta S(r,u)$

5.基于LLMs的采样

Sampling Type	Table Sampling Methods	SQA	FEVEROUS	TabFact	HybridQA	ToTTo
Rule-based Sampling	Random Sampling	27.30%	60.30%	55.17%	23.60%	40.12%
	Evenly Sampling	26.72%	61.87%	54.63%	5.32%	29.41%
	Content Snapshot (Yin et al., 2020)	28.24%	63.10%	56.92%	23.40%	47.51%
Embedding-based Sampling	Centroid-based Sampling	28.10%	63.50%	55.40%	24.03%	48.30%
	Semantic-based Sampling	28.32%	63.32%	59.80%	24.32%	49.14%
	w/ Column Grounding	**29.12%**	64.74%	60.23%	**25.14%**	**53.42%**
	Hybrid Sampling	28.79%	**65.34%**	**61.37%**	24.71%	51.63%
LLM-based Sampling	LLM-Decomposer (Ye et al., 2023b)	27.98%	62.34%	58.74%	24.98%	48.13%
-	No sampling (GPT-3.5)	27.60%	60.12%	56.20%	14.10%	47.42%
	No sampling (GPT-3.5, truncated)	23.54%	43.54%	52.12%	23.12%	30.42%

Methods — 表格增强

1.基于元数据的增强

- 维度/度量 √
- 字段类型 √
- 表格大小 ×
- 统计特征 √
- 标题层次 ×

2.基于检索的增强

- 文档引用
- 术语解释

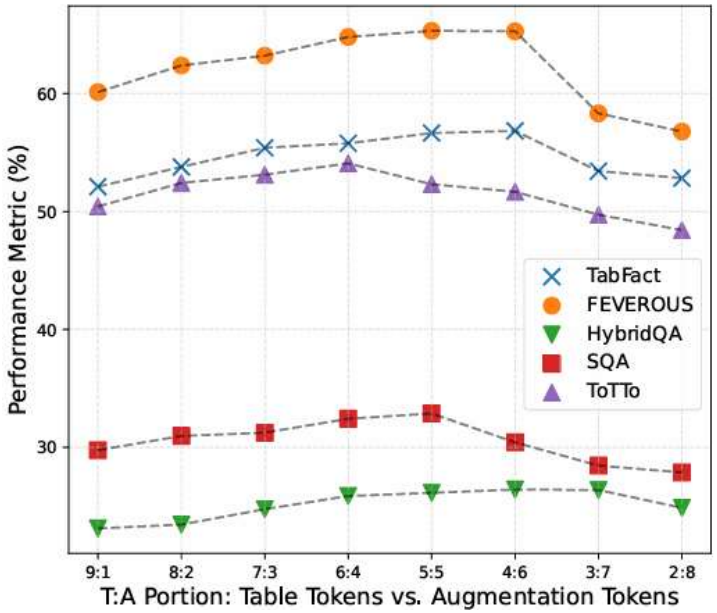
3.基于一致性的增强

① 向初始提示中附加提示“识别表格中与查询相关的关键值和范围”，获得模型见解

② 将见解融入初始提示中得到更详细的结果

Augmentation Aspect	SQA		FEVEROUS		TabFact		HybridQA		ToTTo	
	Acc	Delta	Acc	Delta	Acc	Delta	Acc	Delta	BLEU-4	Delta
baseline	28.32%	0.00%	63.32%	0.00%	59.80%	0.00%	24.32%	0.00%	49.14%	0.00%
D/M + SF	30.12%	1.80%	65.72%	2.40%	**62.67%**	2.87%	26.12%	1.80%	51.25%	2.11%
Table Size	28.85%	0.53%	63.40%	0.08%	60.30%	0.50%	24.94%	0.62%	49.03%	-0.11%
Statistics Feature	31.22%	2.90%	**66.51%**	3.19%	62.33%	2.53%	26.13%	1.81%	50.57%	1.43%
Header Hierarchy	-	-	-	-	-	-	-	-	48.64%	-0.50%
Docs References	**33.45%**	5.13%	63.13%	-0.19%	61.32%	1.52%	25.12%	0.80%	52.74%	3.60%
Term Explanations										
- LLM-based	31.59%	3.27%	64.12%	0.80%	62.32%	2.52%	26.24%	1.92%	**53.21%**	4.07%
- Heuristics-based	29.59%	1.27%	63.72%	0.40%	61.58%	1.78%	25.24%	0.92%	51.21%	2.07%
Self Prompting	30.45%	2.13%	65.24%	1.92%	62.32%	2.52%	**26.64%**	2.32%	52.36%	3.22%

其他实验



表格内容与增强信息的 token 占比为 5:5 或 4:6 时，性能最佳；过度倾斜增强信息（如 3:7）会导致核心表格信息被稀释，反而降低推理质量

Components of TAP4LLM	SQA		FEVEROUS		TabFact		HybridQA		ToTTo	
	Acc	Delta	Acc	Delta	Acc	Delta	Acc	Delta	BLEU-4	Delta
All	34.12%	0.00%	68.32%	0.00%	64.78%	0.00%	27.87%	0.00%	54.93%	0.00%
w/o table sampling	**26.54%**	-7.58%	**61.54%**	-6.78%	**58.12%**	-6.66%	**24.12%**	-3.75%	**48.47%**	-6.46%
w/o table augmentation - all	29.12%	-5.00%	63.74%	-4.58%	60.23%	-4.55%	25.14%	-2.73%	53.42%	-1.51%
w/o table augmentation - metadata-based	33.87%	-0.25%	64.38%	-3.94%	62.78%	-2.00%	26.98%	-0.89%	53.42%	-1.51%
w/o table augmentation - retrieval-based	31.42%	-2.7%	66.23%	-2.09%	62.97%	-1.81%	26.33%	-1.54%	52.67%	-2.26%
w/o table packing	31.87%	-2.25%	67.42%	-0.90%	63.28%	-1.50%	26.32%	-1.55%	52.87%	-2.06%

Tabular Representation, Noisy Operators, and Impacts on Table Structure Understanding Tasks in LLMs

Ananya Singha
Microsoft, India
t-asingha@microsoft.com

José Cambronero
Microsoft, USA
jcambronero@microsoft.com

Sumit Gulwani
Microsoft, USA
sumitg@microsoft.com

Vu Le
Microsoft, USA
levu@microsoft.com

Chris Parnin
Microsoft,USA
chrisparnin@microsoft.com

NeurIPS 2023

8种输入格式+8种噪声操作+7种表格理解任务

输入格式

DFLoader Format

```
pd.DataFrame({
    Name : ['Alice', 'Bob', 'Charlie'],
    Age : [25, 30, 22],
    City : ['New York', 'Los Angeles', 'Chicago']
    },
    index=[0, 1, 2])
```

Json Format

```
{
  "0": {"Name":"Alice","Age":25,"City":"NewYork"},
  "1": {"Name":"Bob","Age":30,"City":"Los Angeles"},
  "2":{"Name":"Charlie","Age":22,"City":"Chicago"}
}
```

Data-Matrix Format

```
[
  ['', 'Name', 'Age', 'City'],
  [0, 'Alice', 25, 'New York'],
  [1, 'Bob', 30, 'Los Angeles'],
  [2, 'Charlie', 22, 'Chicago']
]
```

HTML Format

```
<table>
 <thead>
  <tr>
   <th></th>
   <th>Name</th>
   .....
   <th>Sex</th>
  </tr>
 </thead>
 <tbody>
  <tr>
   <th>0</th>
   <td>Alice</td>
   .....
   <td>F</td>
  </tr>
  .....
 </tbody>
</table>
```

Comma Separated Format

```
, Name, Age, City
0, Alice, 25, New York
1, Bob, 30, Los Angeles
2, Charlie, 22, Chicago
```

Tab Separated Format

	Name	Age	City
0	Alice	25	New York
1	Bob	30	Los Angeles
2	Charlie	22	Chicago

Markdown Format

	Name	Age	City
0	Alice	25	New York
1	Bob	30	Los Angeles
2	Charlie	22	Chicago

HTML No Space Format

```
<table><thead><tr><th></th><th>Name</th><th>Age</th><th>City</th><th>Sex</th></tr></thead><tbody><tr><th>0</th><td>Alice</td><td>25</td><td>New York</td><td>F</td></tr><tr><th>1</th><td>Bob</td><td>30</td><td>Los Angeles</td><td>M</td></tr><tr><th>2</th><td>Charlie</td><td>22</td><td>Chicago</td><td>M</td></tr></tbody></table>
```

噪声操作

SequentialColumnNames

	col_0	col_1	col_2	col_3
0	Alice	25	New York	F
1	Bob	30	Los Angeles	M
2	Charlie	22	Chicago	M

ShuffleRows

	Name	Age	City	Sex
1	Bob	30	Los Angeles	M
0	Alice	25	New York	F
2	Charlie	22	Chicago	M

SerializeRow

0	Name:Alice,Age:25,City:New York,Sex:F
1	Name:Bob,Age:30,City:Los Angeles,Sex:M
2	Name:Charlie,Age:22,City:Chicago,Sex:M

ShuffleColumnNames

	Age	Sex	Name	City
0	Alice	25	New York	F
1	Bob	30	Los Angeles	M
2	Charlie	22	Chicago	M

ShuffleColumns

	Age	Sex	Name	City
0	25	F	Alice	New York
1	30	M	Bob	Los Angeles
2	22	M	Charlie	Chicago

ColumnMerger

	Name-----Age-----City	Sex
0	Alice-----25-----New York	F
1	Bob-----30-----Los Angeles	M
2	Charlie-----22-----Chicago	M

ArbitraryColumnNames

	0MSTtV	Vi37d2Em	mXjqq0	xqFNEY1YnB
0	Alice	25	New York	F
1	Bob	30	Los Angeles	M
2	Charlie	22	Chicago	M

TransposeTable

	0	1	2
Name	Alice	Bob	Charlie
Age	25	30	22
City	New York	Los Angeles	Chicago
Sex	F	M	M

表格理解任务

	Name	Age	City	Sex
0	Alice	25	New York	F
1	Bob	30	Los Angeles	M
2	Charlie	22	Chicago	M
3	David	28	Boston	M
4	Emily	35	San Francisco	F
5	Frank	29	Dallas	M
6	Grace	27	Miami	F
7	Henry	32	Seattle	M
8	Ivy	24	Denver	F
9	Jack	33	Houston	M
10	Katherine	26	Atlanta	F
11	Liam	31	Phoenix	M
12	Mia	36	Philadelphia	F
13	Noah	23	San Diego	M
14	Olivia	29	Austin	F

- DataType Lookup Test: What type (using Pandas datatype notation) is column Age?
- Column Lookup Test: What column is the value Emily in?
- Row Lookup Test: What row is the value Seattle in?
- Navigation Test: What value is at row 7 and column City?
- Table Transpose Test: Can you transpose the table?
- Table Reconstruction Test: Can you reconstruct the table by deserializing the table above?
- Table Column Reorder Test: Can you reorder the table such that the column are in this new order ['Sex', 'Name', 'Age', 'City']?

DFLoader Format

```
pd.DataFrame({
    Name : ['Alice', 'Bob', 'Charlie'],
    Age : [25, 30, 22],
    City : ['New York', 'Los Angeles', 'Chicago']
    },
    index=[0, 1, 2])
```

Json Format

```
{
  "0": {"Name":"Alice","Age":25,"City":"NewYork"},
  "1": {"Name":"Bob","Age":30,"City":"Los Angeles"},
  "2":{"Name":"Charlie","Age":22,"City":"Chicago"}
}
```

Data-Matrix Format

```
[
  ['', 'Name', 'Age', 'City'],
  [0, 'Alice', 25, 'New York'],
  [1, 'Bob', 30, 'Los Angeles'],
  [2, 'Charlie', 22, 'Chicago']
]
```

HTML Format

```
<table>
 <thead>
  <tr>
   <th></th>
   <th>Name</th>
   .....
   <th>Sex</th>
  </tr>
 </thead>
 <tbody>
  <tr>
   <th>0</th>
   <td>Alice</td>
   .....
   <td>F</td>
  </tr>
  .....
 </tbody>
</table>
```

Comma Separated Format

```
, Name, Age, City
0, Alice, 25, New York
1, Bob, 30, Los Angeles
2, Charlie, 22, Chicago
```

Tab Separated Format

	Name	Age	City
0	Alice	25	New York
1	Bob	30	Los Angeles
2	Charlie	22	Chicago

Markdown Format

	Name	Age	City
0	Alice	25	New York
1	Bob	30	Los Angeles
2	Charlie	22	Chicago

HTML No Space Format

```
<table><thead><tr><th></th><th>Name</th><th>Age</th><th>City</th><th>Sex</th></tr></thead><tbody><tr><th>0</th><td>Alice</td><td>25</td><td>New York</td><td>F</td></tr><tr><th>1</th><td>Bob</td><td>30</td><td>Los Angeles</td><td>M</td></tr><tr><th>2</th><td>Charlie</td><td>22</td><td>Chicago</td><td>M</td></tr></tbody></table>
```

Table 1: Average Pass@1 for fact-finding tasks. DFLoader provides overall high pass@1 performance.

Table Formats	ColumnLookupTests	DataTypeLookupTests	NavigationTests	RowLookupTests	Overall
CommaSeparated	64.43	95.00	65.57	78.14	75.78
DFLoader	72.71	95.29	68.29	82.86	**79.79**
DataMatrix	62.57	84.00	56.57	**87.43**	72.64
Json	65.00	**96.43**	**71.43**	78.86	77.93
Markdown	61.43	85.86	48.71	73.29	67.32
TabSeparated	67.00	94.00	64.43	78.14	75.8
HTML	**79.83**	94.67	58.83	52.33	71.4
HTMLNoSpace	73.00	93.50	62.00	59.50	72.00

Table 2: F1 scores for transformation tasks. DFLoader and JSON format, with structural element isolation and repetition, enable high performance on average across transformation tasks.

Table	TableColumnReorderTests	TableReconstructionTests	TableTransposeTests	Overall
CommaSeparated	95.33	74.33	99.00	89.55
DFLoader	99.33	**98.00**	98.33	**98.55**
DataMatrix	92.67	90.67	0.00	61.11
Json	**99.67**	85.00	**100.00**	94.89
Markdown	50.00	24.33	34.00	36.11
TabSeparated	93.33	92.33	50.00	78.55
HTML	50.00	86.00	83.33	73.11
HTMLNoSpace	83.33	84.00	83.33	83.55

- DFLoader整体最优，MarkDown性能低下
- LLMs 对结构表示的微小变化异常脆弱
- HTML格式过于冗长（即使不换行），尽量避免

SequentialColumnNames

	col_0	col_1	col_2	col_3
0	Alice	25	New York	F
1	Bob	30	Los Angeles	M
2	Charlie	22	Chicago	M

ShuffleRows

	Name	Age	City	Sex
1	Bob	30	Los Angeles	M
0	Alice	25	New York	F
2	Charlie	22	Chicago	M

SerializeRow

0	Name:Alice,Age:25,City:New York,Sex:F
1	Name:Bob,Age:30,City:Los Angeles,Sex:M
2	Name:Charlie,Age:22,City:Chicago,Sex:M

ShuffleColumnNames

	Age	Sex	Name	City
0	Alice	25	New York	F
1	Bob	30	Los Angeles	M
2	Charlie	22	Chicago	M

ShuffleColumns

	Age	Sex	Name	City
0	25	F	Alice	New York
1	30	M	Bob	Los Angeles
2	22	M	Charlie	Chicago

ColumnMerger

	Name-----Age-----City	Sex
0	Alice-----25-----New York	F
1	Bob-----30-----Los Angeles	M
2	Charlie-----22-----Chicago	M

ArbitraryColumnNames

	0MSTtV	Vi37d2Em	mXjqq0	xqFNEY1YnB
0	Alice	25	New York	F
1	Bob	30	Los Angeles	M
2	Charlie	22	Chicago	M

TransposeTable

	0	1	2
Name	Alice	Bob	Charlie
Age	25	30	22
City	New York	Los Angeles	Chicago
Sex	F	M	M

Table 4: Average F1 score delta from original to noisy for transformation tasks. Statistically significant values (p-value $< \frac{0.01}{8}$) are marked with "**".

Table Formats	Table Manipulation	TableColumnReorderTests	TableReconstructionTests	TableTransposeTests
Json	OriginalData	99.67	85.00	100.00
	ShuffleRows	-1.00	-45.00**	-13.33**
	ShuffleColumns	+0.33	-19.00	-40.67**
	ShuffleColumnNames	-0.34	-13.67	-29.33**
	SequentialColumnNames	+0.33	-9.00	-2.00
	ArbitraryColumnNames	+0.33	-4.33	-0.67
	TransposeTable	-89.00**	-78.33**	-42.00**
	ColumnMerger	+0.33	-85.00**	-75.33**
	SerializeRow	-59.00**	-46.33**	-100.00**
DFLoader	OriginalData	99.33	98.00	98.33
	ShuffleRows	+0.67	-78.67**	-16.33**
	ShuffleColumns	+0.67	-34.00**	-34.33**
	ShuffleColumnNames	+0.67	-31.33**	-26.33**
	SequentialColumnNames	+0.67	-54.67**	-1.00
	ArbitraryColumnNames	+0.67	-18.00**	-0.33
	TransposeTable	-43.33**	-98.00**	-83.00**
	ColumnMerger	+0.67	-98.00**	-81.00**
	SerializeRow	-33.33**	-73.33**	-60.33**
CommaSeparated	OriginalData	95.33	74.33	99.00
	ShuffleRows	-7.33	-41.66**	-70.33**
	ShuffleColumns	-4.66	-19.00	-33.00**
	ShuffleColumnNames	-32.00**	-9.66	-47.00**
	SequentialColumnNames	-67.33**	-13.00	-24.33**
	ArbitraryColumnNames	-28.66**	+4.34	-21.67**
	TransposeTable	+2.00	-65.00**	-98.33**
	ColumnMerger	-4.66	-74.33**	-80.33**
	SerializeRow	-95.33**	-57.66**	-99.00**
TabSeparated	OriginalData	93.33	92.33	50.00
	ShuffleRows	-4.00	-57.00**	-34.67**
	ShuffleColumns	-6.00	-31.00**	-6.00**
	ShuffleColumnNames	-59.33**	-29.66**	0.00
	SequentialColumnNames	-68.00**	-27.00**	-7.33**
	ArbitraryColumnNames	-45.33**	-13.00**	-2.00**
	TransposeTable	-44.66**	-83.00**	-50.00**
	ColumnMerger	-41.33**	-92.33**	-48.00**
	SerializeRow	-93.33**	-91.66**	-50.00**

Thanks