STABLE. TABLE GENERATION FRAMEWORK FOR ENCODER-DECODER MODELS

Michał Pietruszka, Michał Turski, Łukasz Borchmann, Tomasz Dwojak, Gabriela Pałka, Karolina Szyndler, Dawid Jurkiewicz, Łukasz Garncarek
Find us at NeurIPS

Michał Pietruszka
Snowflake

Michał Turski
Snowflake

Łukasz Borchmann
Snowflake
WHAT IS THIS ALL ABOUT?
Unification under table generation framework

Input
Document, e.g.:
- Invoice
- Wikipedia article
- Plain text news

Encoder-decoder model

Output
Task-dependent data structure, e.g.:

- Description
- Quantity
- Unit price
- Total

Ice cream 2 5 10
Bread 1 2 2
Soda 1 3 3

Extracted line items

- Property
- Value
- Citizenship
- Russian Empire
- Date of birth
- 1915-01-15
- Place of birth
- Saint Petersburg

Key information / property-value pairs

- Subject
- Object
- Relation
- Subject: Stockholm
- Object: Sweden
- Relation: country
- Subject: Royal Court Orchestra
- Object: Royal Opera
- Relation: part of

Entities and relations / knowledge base records
Unification under table generation framework

**Input**
- Document, e.g.: Invoice, Wikipedia article, Plain text news

**Encoder-decoder model**

**Output**
- Task-dependent data structure, e.g.: Table with extracted line items and key information/property-value pairs

### Extracted line items
- **Description**: Ice cream, Bread, Soda
- **Quantity**: 2, 1, 1
- **Unit price**: 5, 2, 3
- **Total**: 10, 2, 3

### Key information/property-value pairs
- **Property**
  - Citizenship
  - Date of birth
  - Place of birth
- **Value**
  - Russian Empire
  - 1915-01-15
  - Saint Petersburg

### Entities and relations / knowledge base records
- Subject: Riddarhuset, Royal Court Orchestra
- Object: Sweden, Royal Opera
- Relation: country, part of
Unification under table generation framework

Input

Document, e.g.:

- Invoice
- Wikipedia article
- Plain text news

Encoder-decoder model

Output

Task-dependent data structure, e.g.:

Extracted line items

- **Description**
  - Ice cream
  - Bread
  - Soda

- **Quantity**
  - 2
  - 1
  - 1

- **Unit price**
  - 5
  - 2
  - 3

- **Total**
  - 10
  - 2
  - 3

- **Total**

Key information / property-value pairs

- **Property**
  - Citizenship
  - Date of birth
  - Place of birth

- **Value**
  - Russian Empire
  - 1915-01-15
  - Saint Petersburg

Entities and relations / knowledge base records

- **Subject**
  - Riddarhuset
  - Royal Court Orchestra

- **Object**
  - Sweden
  - Royal Opera

- **Relation**
  - country
  - part of
Auguste and Luis Lumière were born in Besançon, France, to Charles and Jeanne.

<table>
<thead>
<tr>
<th>Name</th>
<th>Surname</th>
<th>Place of birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auguste</td>
<td>Lumière</td>
<td>Besançon</td>
</tr>
<tr>
<td>Luis</td>
<td>Lumière</td>
<td>Besançon</td>
</tr>
<tr>
<td>Charles</td>
<td>Lumière</td>
<td>NULL</td>
</tr>
<tr>
<td>Jeanne</td>
<td>Lumière</td>
<td>NULL</td>
</tr>
</tbody>
</table>
Key Observations

Context matters

Order matters
HOW DOES IT WORK?
There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.
There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.
Training

<table>
<thead>
<tr>
<th>Color</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>blue</td>
<td>triangle</td>
</tr>
</tbody>
</table>
Training

Color | Shape
------|------
red   | ?
?     | ?
blue  | triangle
Training

<table>
<thead>
<tr>
<th>Color</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>blue</td>
<td>triangle</td>
</tr>
</tbody>
</table>
Training

<table>
<thead>
<tr>
<th>Color</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>blue</td>
<td>triangle</td>
</tr>
</tbody>
</table>
Cell dependencies

**TABULAR BIAS**

Encodes the relative position of table cells in which the tokens lie.

\[ \tau_{ij} = \begin{cases} R(r_i - r_j) + C(c_i - c_j) & \text{if } r_j > 0 \\ R_0 + C(c_i - c_j) & \text{if } r_j = 0 \end{cases} \]

**LOCAL SEQUENTIAL BIAS**

Corresponds to the relative sequential position of tokens belonging to the same cell.

\[ \lambda_{ij} = \begin{cases} L(i - j) & \text{if } (c_i, r_i) = (c_j, r_j) \\ 0 & \text{otherwise} \end{cases} \]

<table>
<thead>
<tr>
<th>Color</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>circle</td>
</tr>
<tr>
<td>green</td>
<td>square</td>
</tr>
<tr>
<td>blue</td>
<td>triangle</td>
</tr>
</tbody>
</table>
Recall the Key Observations

- Context matters
- Order matters
There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.

Legend

<table>
<thead>
<tr>
<th>Probability</th>
<th>Candidate value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>High-score candidate</td>
</tr>
<tr>
<td>Value kept from the previous step</td>
<td></td>
</tr>
</tbody>
</table>
Inference

**Input**

There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.

**Legend**

<table>
<thead>
<tr>
<th>Probability</th>
<th>Candidate value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value kept from the previous step</td>
<td></td>
</tr>
</tbody>
</table>

**Probability**

<table>
<thead>
<tr>
<th></th>
<th>Colors</th>
<th>Shapes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>red</td>
<td>0.4</td>
</tr>
<tr>
<td>0.9</td>
<td>green</td>
<td>0.8</td>
</tr>
<tr>
<td>0.8</td>
<td>blue</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Step 2/5
**Inference**

**Input**

There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.

**Legend**

<table>
<thead>
<tr>
<th>Probability</th>
<th>Candidate value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>High-score candidate</td>
</tr>
<tr>
<td>Value</td>
<td>kept from the previous step</td>
</tr>
</tbody>
</table>

**Step 2/5**

<table>
<thead>
<tr>
<th>Colors</th>
<th>Shapes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9 red</td>
<td>0.4 square</td>
</tr>
<tr>
<td>0.9 green</td>
<td>0.8 square</td>
</tr>
<tr>
<td>0.8 blue</td>
<td>0.5 cross</td>
</tr>
</tbody>
</table>

Note that these are generated in parallel!
### Input

There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.

### Legend

<table>
<thead>
<tr>
<th>Probability</th>
<th>Candidate value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>High-score candidate</td>
</tr>
<tr>
<td>Value kept from the previous step</td>
<td></td>
</tr>
</tbody>
</table>

### Step 3/5

<table>
<thead>
<tr>
<th>Colors</th>
<th>Shapes</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>0.3 hexagon</td>
</tr>
<tr>
<td>green</td>
<td>0.9 square</td>
</tr>
<tr>
<td>1.0 blue</td>
<td>0.8 triangle</td>
</tr>
</tbody>
</table>
Inference

Input

There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.

Legend

<table>
<thead>
<tr>
<th>Probability</th>
<th>Candidate value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>High-score candidate</td>
</tr>
<tr>
<td>Value kept</td>
<td>Value kept from the previous step</td>
</tr>
</tbody>
</table>

Step 4/5

Colors

- red
- green
- blue

Shapes

- 0.6 circle
- square
- 0.8 triangle
Inference

Input

There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.

Legend

<table>
<thead>
<tr>
<th>Probability</th>
<th>Candidate value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>High-score candidate</td>
</tr>
<tr>
<td>Value kept</td>
<td>from the previous step</td>
</tr>
</tbody>
</table>

Step 5/5

Colors | Shapes
--- | ---
red  | circle
green  | square
blue  | triangle
WHAT ARE THE RESULTS?
### Results on public and private datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SOTA reference</th>
<th>Linearized</th>
<th>Our Model</th>
<th>Backbone Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWC</td>
<td>26.8</td>
<td>27.8</td>
<td><strong>30.8</strong></td>
<td>T5 2D + STable</td>
</tr>
<tr>
<td>CORD</td>
<td>96.3</td>
<td>92.4</td>
<td><strong>95.6</strong></td>
<td>TILT + STable</td>
</tr>
<tr>
<td>Rotowire Player</td>
<td>86.8</td>
<td>84.5</td>
<td><strong>84.5</strong></td>
<td>T5 + STable</td>
</tr>
<tr>
<td>Rotowire Team</td>
<td>86.3</td>
<td>83.8</td>
<td><strong>84.7</strong></td>
<td>TILT + STable</td>
</tr>
<tr>
<td>DWIE</td>
<td>62.9</td>
<td>60.2</td>
<td><strong>59.2</strong></td>
<td></td>
</tr>
<tr>
<td>Recipe Composition</td>
<td>71.9</td>
<td>60.1</td>
<td><strong>75.5</strong></td>
<td>TILT + STable</td>
</tr>
<tr>
<td>Payment Stubs</td>
<td>77.0</td>
<td>72.0</td>
<td><strong>79.1</strong></td>
<td></td>
</tr>
<tr>
<td>Bank Statements</td>
<td>61.1</td>
<td>58.7</td>
<td><strong>69.9</strong></td>
<td></td>
</tr>
</tbody>
</table>

Across different backbone models
TL;DR

**TRAINING**
Permutation-based decoder training

**DECODING**
Decoding mechanism that is data-dependent

**FRAMEWORK**
Document-to-table framework that works with any backbone
THANK YOU

michal.pietruszka@snowflake.com
https://arxiv.org/abs/2206.04045