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
Riddarhuset	Sweden	country
Royal Court Orchestra	Royal Opera	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.:**

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
Riddarhuset	Sweden	country
Royal Court Orchestra	Royal Opera	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.:**

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
Riddarhuset	Sweden	country
Royal Court Orchestra	Royal Opera	part of

Entities and relations / knowledge base records

Complete Example

Input

Auguste and Luis Lumière were born in Besançon, France, to Charles and Jeanne.

Output

Name	Surname	Place of birth
Auguste	Lumière	Besançon
Luis	Lumière	Besançon
Charles	Lumière	*NULL*
Jeanne	Lumière	*NULL*

People

Key Observations

Context matters

Order matters

HOW DOES IT WORK?

Training

“

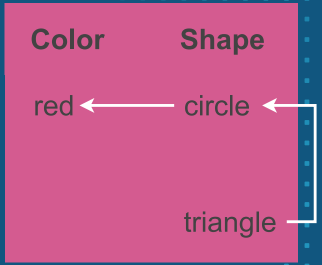
There are toys colored red, green, and blue on the table. The square is green, the triangle is blue, and the circle is in the remaining color.

```
Figures

<Column>
    Color
    <Cell>
    <Cell />
    <Cell />
</Column>

<Column>
    Shape
    <Cell> circle </Cell>
    <Cell />
    <Cell> triangle </Cell>
</Column>
```

(A) Decoder prompt

Color	Shape
red	circle
green	square
blue	triangle

(B) Gold standard



(C) Output after current step

red </Cell>

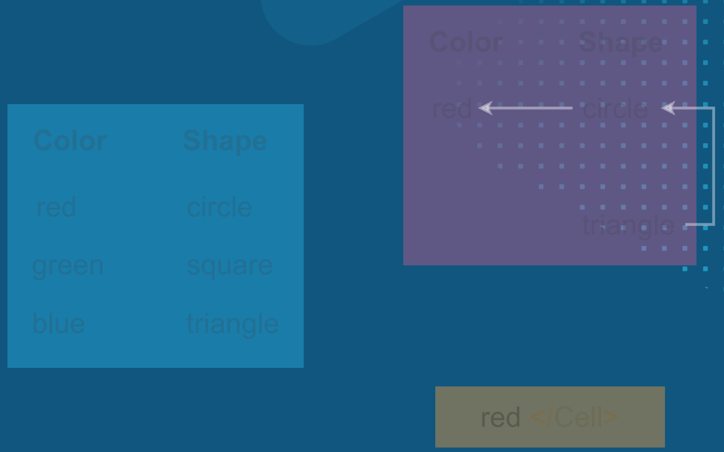
(D) Expected output

Training

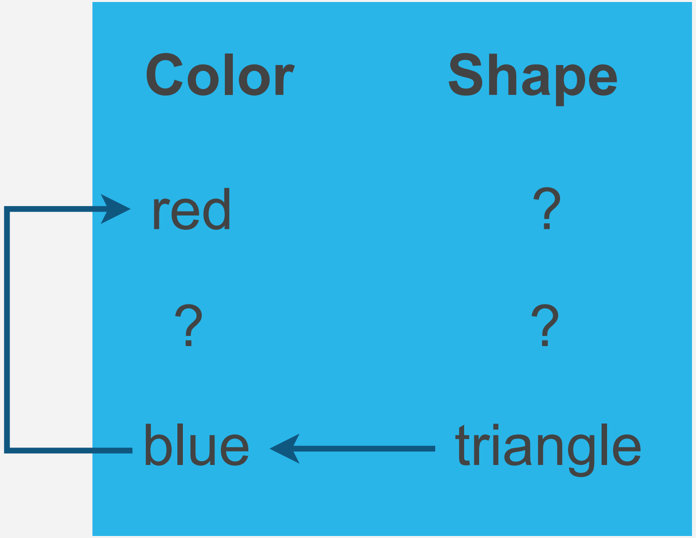
“

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.

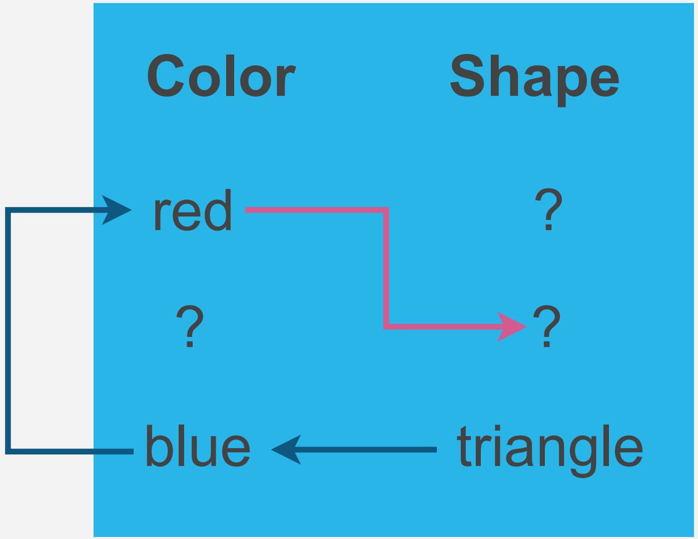
```
Figures

<Column>
    Color
    <Cell> red </Cell>
    <Cell />
    <Cell />
</Column>

<Column>
    Shape
    <Cell> circle </Cell>
    <Cell />
    <Cell> triangle </Cell>
</Column>
```



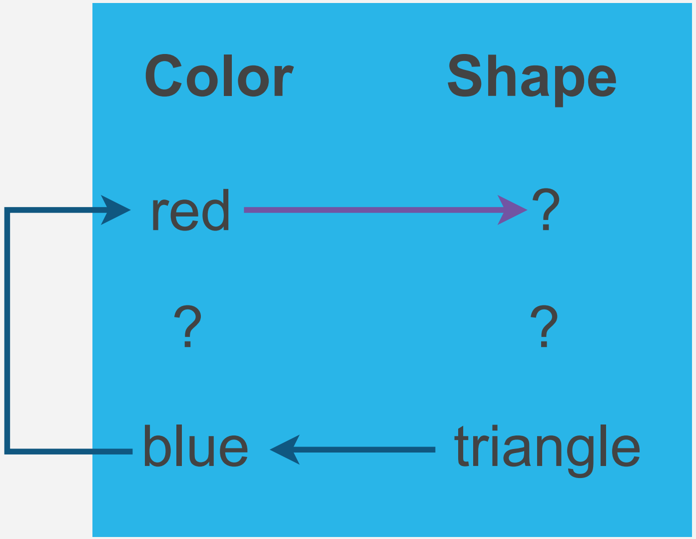
Training



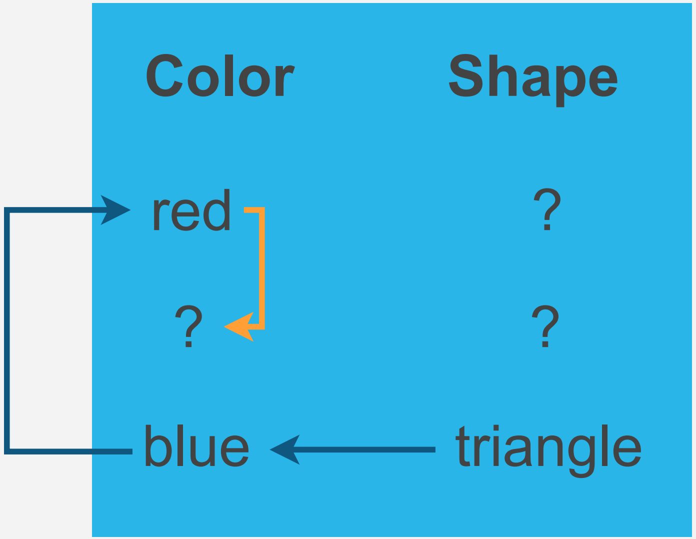
Training



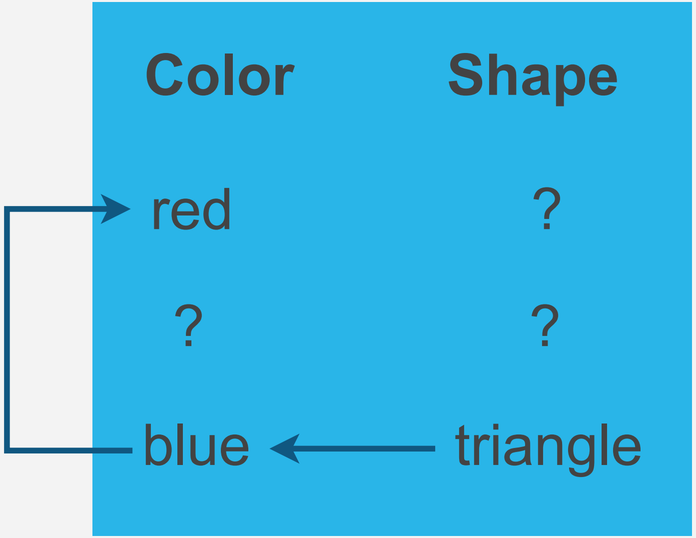
Training



Training



Training



Cell dependencies

TABULAR BIAS

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

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

Color	Shape
red	circle
green	square
blue	triangle

LOCAL SEQUENTIAL BIAS

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

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

Recall the Key Observations

Context matters

Order matters

Inference

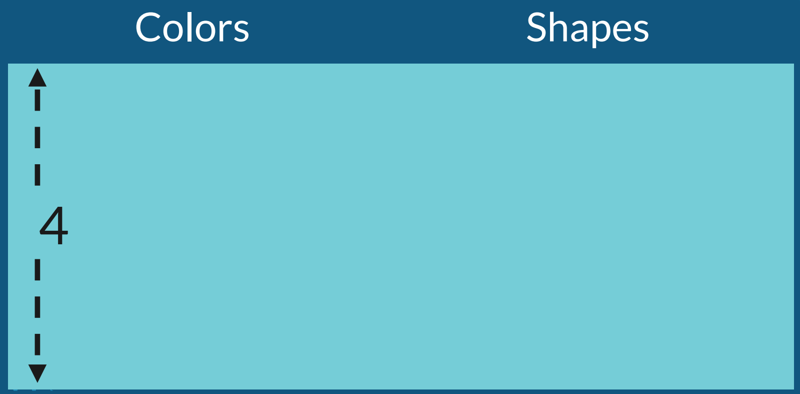
Input

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

Probability	Candidate value
Probability	High-score candidate
Value kept from the previous step	

Step 1/5



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

Probability	Candidate value
Probability	High-score candidate
Value kept from the previous step	

Step 2/5

Colors		Shapes	
0.9	red	**0.4**	square
0.9	green	**0.8**	square
0.8	blue	**0.5**	cross

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

Probability	Candidate value
Probability	High-score candidate
Value kept from the previous step	

Step 2/5

Colors	Shapes
0.9 red	**0.4** square
0.9 green	**0.8** square
0.8 blue	**0.5** cross

Note that these are generated in parallel!

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

Probability	Candidate value
Probability	High-score candidate
Value kept from the previous step	

Step 3/5

Colors	Shapes
red	**0.3** hexagon
green	**0.9** square
1.0 blue	**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

Probability	Candidate value
Probability	High-score candidate
Value kept from the previous step	

Step 4/5

Colors	Shapes
red	**0.6** circle
green	square
blue	**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

Probability	Candidate value
Probability	High-score candidate
Value kept from the previous step	

Step 5/5

Colors	Shapes
red	circle
green	square
blue	triangle

WHAT ARE THE RESULTS?

Results on public and private datasets

Across different backbone models

Dataset	SOTA reference	Linearized	Our Model	
PWC	26.8	27.8	**30.8**	**T5 2D** + STable
CORD	**96.3**	92.4	95.6	TILT + STable
Rotowire *Player*	**86.8**	84.5	84.5	
Rotowire *Team*	**86.3**	83.8	84.7	**T5** + STable
DWIE	**62.9**	60.2	59.2	
Recipe Composition	71.9	60.1	**75.5**	
Payment Stubs	77.0	72.0	**79.1**	**TILT** + STable
Bank Statements	61.1	58.7	**69.9**	

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

snowflake®