Numerical Tuple Extraction from Tables with Pre-training

Qingping Yang[1,2], Yixuan Cao[1,2], Ping Luo[1,2,3]

[1] Institute of Computing Technology, Chinese Academy of Sciences

[2] University of Chinese Academy of Sciences

[3] Peng Cheng Laboratory

Background

- A tremendous amount of important data is stored in tables from the Web or vertical domains.

- However, these data are difficult to understand and apply to downstream tasks.
  - Reason: tables project high-dimensional data to two-dimensional layouts by leveraging visual grammar, which brings substantial flexibility to the table layouts.

- Most tools or models for tables only handle relational tables.
  - Converting arbitrary tables into relational data requires a massive investment in table layouts and specific scripts.

	A	B	C
1		**2019**	
2		Assets	Changes from the Previous Year (%)
3	**Current**	21,614	12.4
4	Inventories	16,883	18.1
5	Cash and cash equivalents	4,731	-4.2
6	**Non-current**	2,341	5.0
7	Trade and other receivables	921	17.9
8	Inventories	1,420	1.2
9	**Total**	23,955	11.8

Background

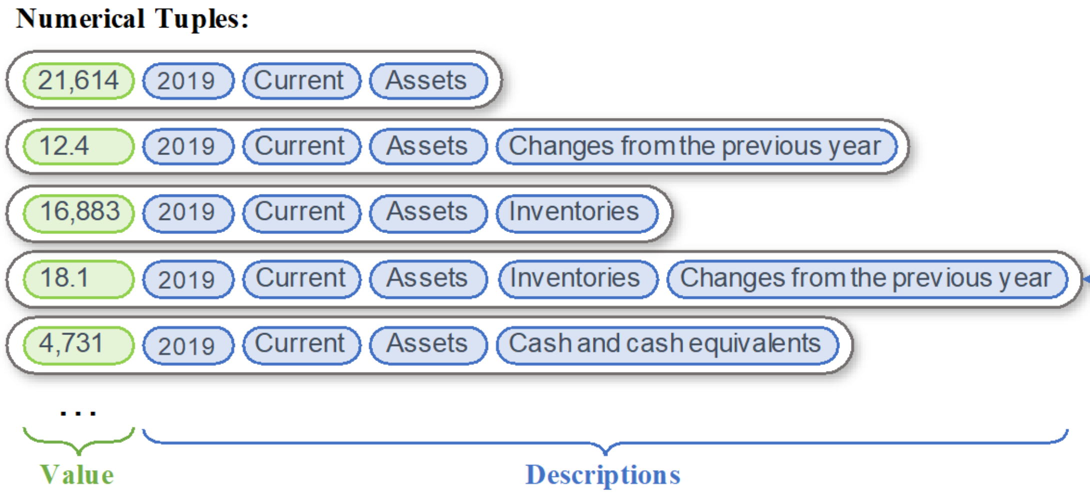
- A critical step to understanding data in tables is extracting numerical data.
  - Numerical tuple consists a value and several descriptions

- A table can be parsed into a set of numerical tuples with a relational format.

- For example, the meaning of cell `C4`: "*Compared from the previous year, the change of inventory in current assets for 2019 is 18.1%.*"

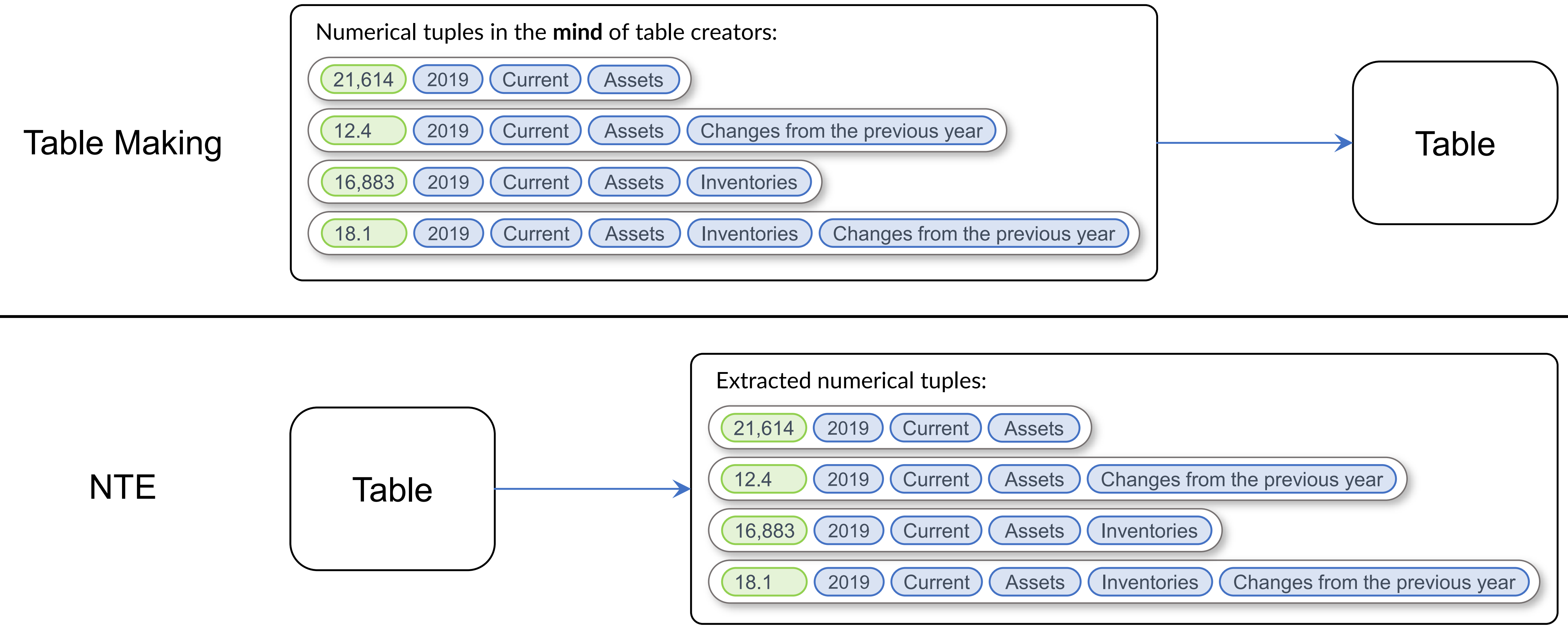
- TASK: **Numerical Tuple Extraction (NTE) from Tables.**

	A	B	C
1			2019
2		Assets	Changes from the Previous Year (%)
3	**Current**	21,614	12.4
4	Inventories	16,883	18.1
5	Cash and cash equivalents	4,731	-4.2
6	**Non-current**	2,341	5.0
7	Trade and other receivables	921	17.9
8	Inventories	1,420	1.2
9	**Total**	23,955	11.8



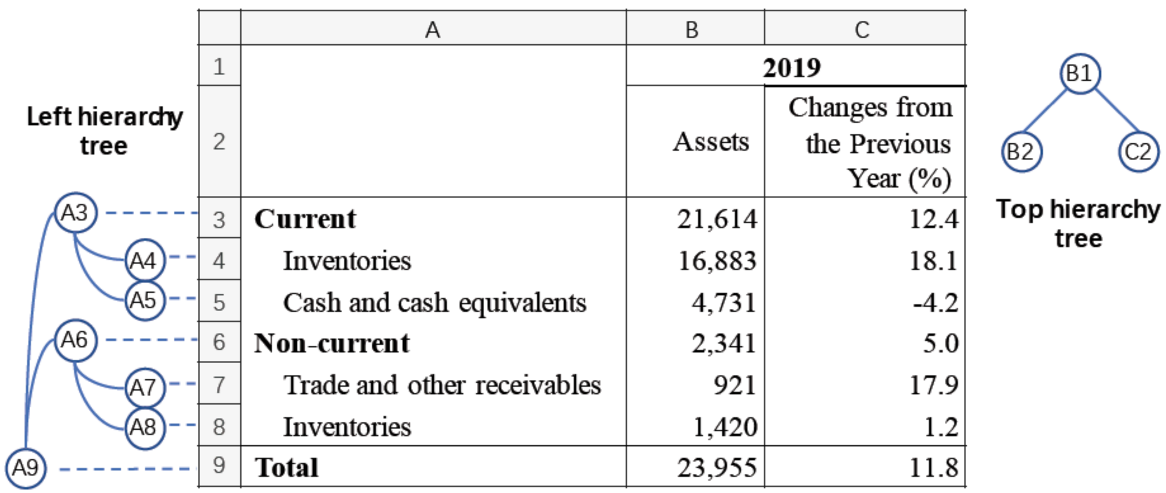
Background

The process of NTE can be imaged as an inverse process of table making.

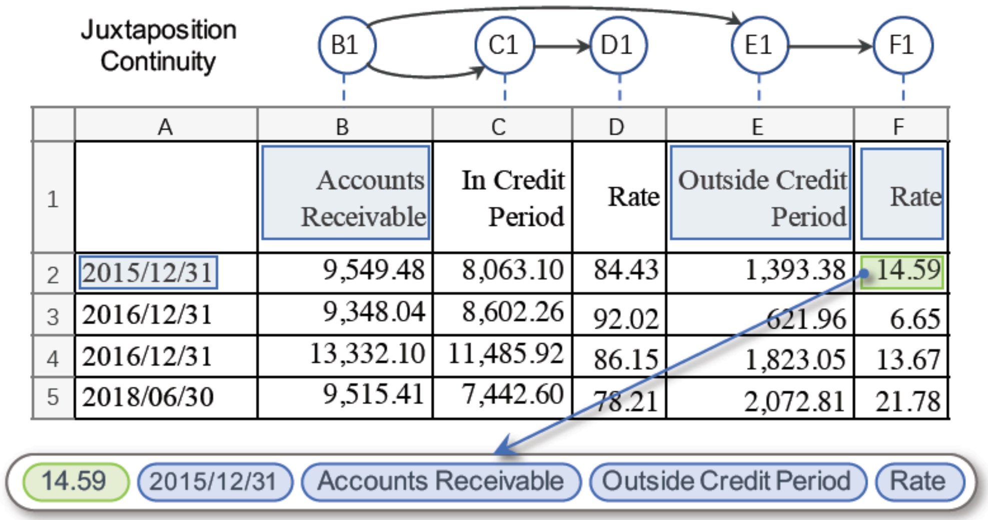


Background

- The relationships between cell
  - Hierarchy
  - Juxtaposition



(a) Top and Left Hierarchy Trees



(b) Juxtaposition of Cells

Related Work

- Previous methods for NTE:
  - First inferring the hierarchical tree of table headers and then constructing a numerical tuple from that tree [1, 2].
  - Transforming spreadsheet data using some examples provided by users [3, 4].

- There are three limitations:
  - Do not consider the *juxtaposition* between cells.
  - Require algorithm-human interaction or rule sets made by domain experts
  - Only evaluate their systems on small corpora that have up to 200 tables.

[1] Automatic web spreadsheet data extraction. *International Workshop on Semantic Search over the Web*. 2013.
[2] Rule-based spreadsheet data transformation from arbitrary to relational tables. 2017.
[3] FlashRelate: extracting relational data from semi-structured spreadsheets using examples. *ACM SIGPLAN Notices*. 2015.
[4] Foofah: Transforming Data By Example. SIGMOD. 2017.

Introduction

- We propose a new framework and evaluate it on a large test set.
  - Convert NTE task into a binary relation extraction task.
  - Encode each cell into a hidden vector by a table representation model.
  - Aggregate vectors in each candidate pair to obtain their predicting result.

- The crucial question is how to represent a cell.
  - **TableLM**, a BERT-based pre-trained language model.
    - Multi-modal.
    - Work on arbitrary types of tables.
    - Remove numerical values.
    - Pre-trained with contrastive learning.

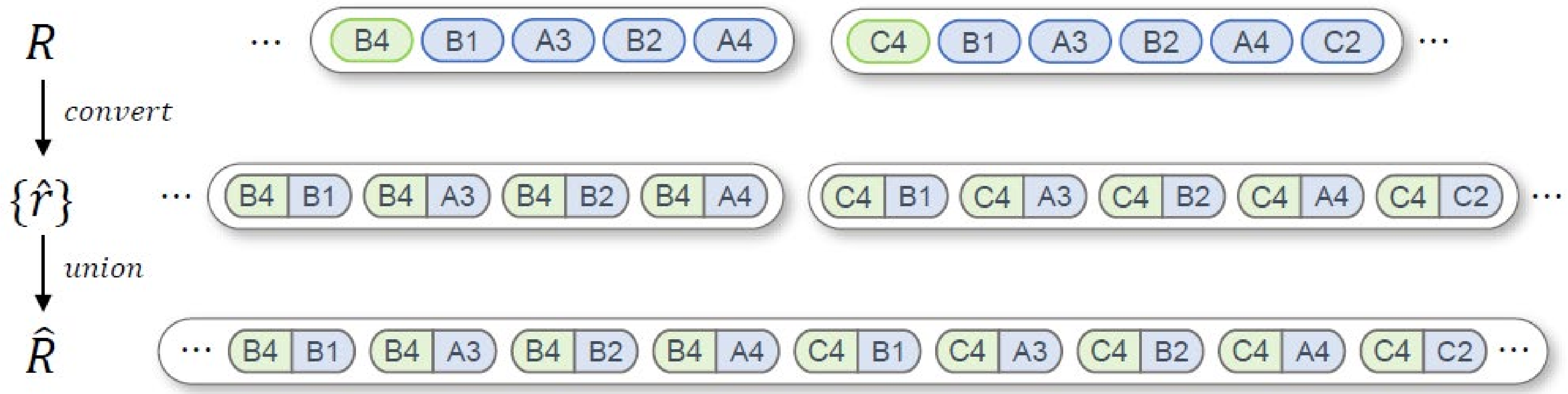
Definition

- Numerical cell set T_v
- Non-numerical cell set T_s
- Numerical Tuple

$$r = (v, D)$$
$$v \in T_v, D = \{d_i | 1 \le i \le K, d_i \in T_s\}$$

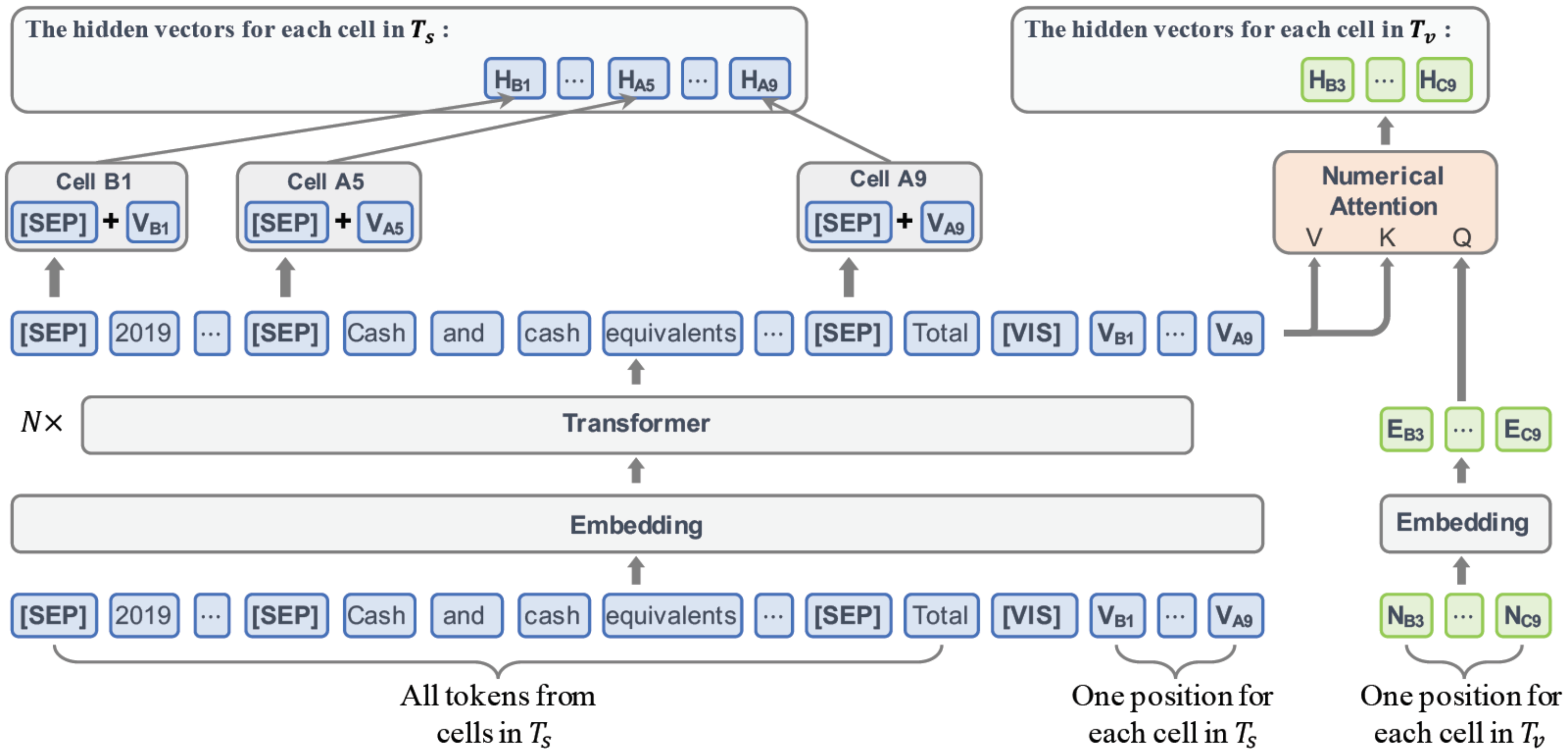
Framework

- Problem Conversion
  - A tuple are converted into several pairs.
  - The task is converted into a problem of relation extraction between cells.



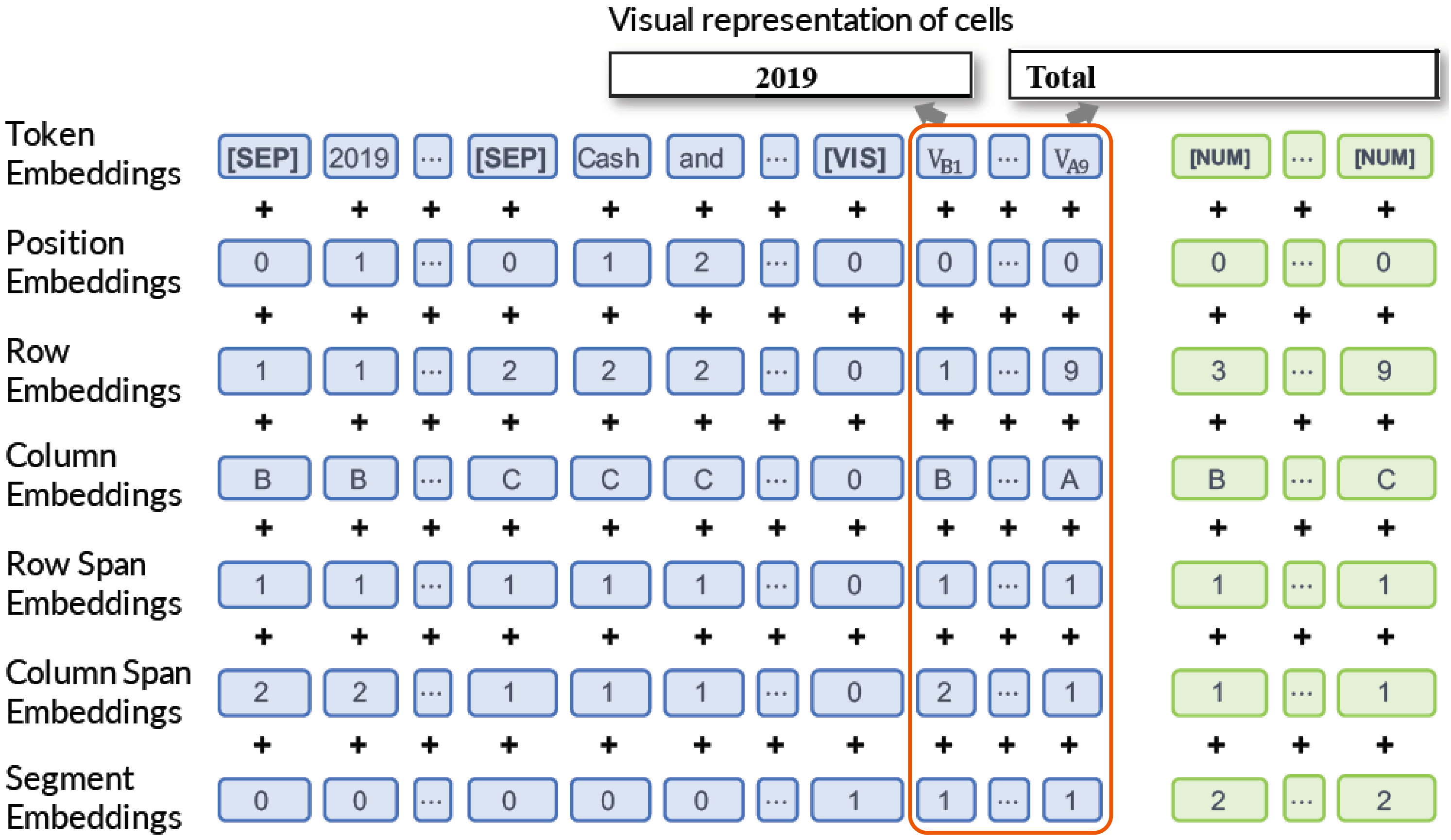
TableLM

- Overview



TableLM

- Embeddings



TableLM

- Transformer with Tabular Masked Attention

$$\text{attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}} \boxed{M})V$$

  - M is the tabular visibility matrix
    - $M_{i,j} = 1$, if token_i and token_j are in the same row or column.

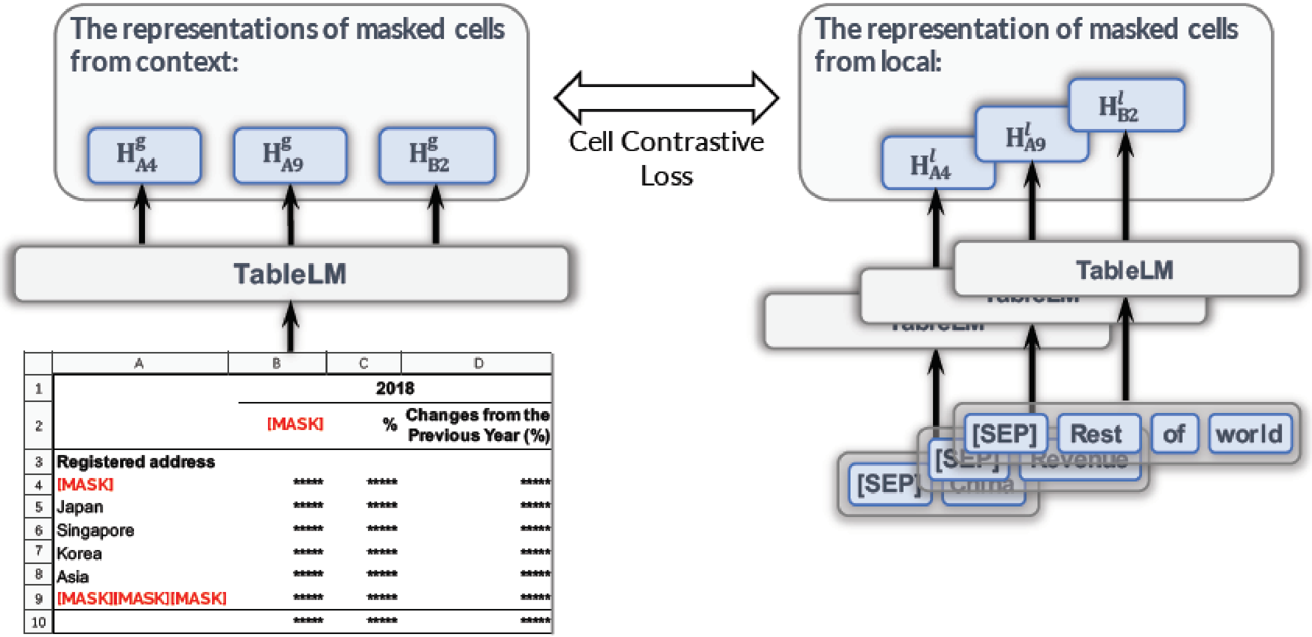
- Cell Representations
  - Non-numerical cells

$$H_{i,j} = \text{LayerNorm}([\text{SEP}]_{i,j} + V_{i,j})$$

  - Numerical cells
    - *Numerical Attention*

TableLM

- Loss functions:
  - Masked Language Model Loss
  - Cell Contrastive Loss
    - Mask whole cell randomly and get their own semantic representation.



Experiments

- Dataset
  - FinTab-Tuples,
    - 19,264 tables from Chinese financial documents crawled from CNINFO [5]
      - Tables in finance are data-intensive.
  - FinTab-Tuples-CT (Complex Table)
    - Complex tuple: contains a description that is not in the same row or column as the value of the tuple.
    - Complex Table: contains at least one complex tuple.
  - Dataset for pre-training
    - FinFormulas [6]
      - 190,179 tables from 4,746 Chinese financial documents.

Table 1: Statistic of FinTab-Tuples

# tables	19,264
# complex tables	8,906
# labeled tuples	604,111
# labeled complex tuples	191,344
Avg. % numerical cells per table	63.29%
Avg. % tuples in cells per table	58.19%
Avg. % complex tuples in tuples per table	27.22%
Avg. % tuples in cells per complex table	60.01%
Avg. % complex tuples in tuples per complex table	58.90%
Avg. # rows per table	9.32
Avg. # columns per table	5.88

[5] http://www.cninfo.com.cn/
[6] Numerical Formula Recognition from Tables. KDD. 2021.

Experiments

- Metric
  - F1-score at pair level.
  - F1-score at tuple level.
  - Table Level Accuracy.

- Baseline
  - TaFor [6]
    - Encodes a table and produces hidden representations of its cells.

[6] Numerical Formula Recognition from Tables. KDD. 2021.

- Performance

Table 2: Results (%) of methods on two test sets. Here, Acc. is an abbreviation for accuracy, F1-P is the F1-score at pair level, F1-T is the F1-score at tuple level.

	FinTab-Tuples-T			FinTab-Tuples-CT		
	Acc.	F1-P	F1-T	Acc.	F1-P	F1-T
TaFor	63.06	95.53	80.43	56.47	94.91	74.28
TableLM	**71.44**	**96.99**	**85.63**	**63.58**	**96.20**	**79.44**

Table 3: Ablation Results (%) on two test sets. Here Acc., F1-P, F1-T are the same as Table 2.

	FinTab-Tuples-T			FinTab-Tuples-CT		
	Acc.	F1-P	F1-T	Acc.	F1-P	F1-T
TableLM	**71.44**	**96.99**	**85.63**	63.58	**96.20**	**79.44**
w/o vision	54.34	95.25	77.17	55.53	94.30	71.52
w/o CCL	66.49	96.51	83.54	**64.58**	95.66	78.34
from scratch	63.47	96.58	83.66	58.66	94.84	74.98

THANK YOU