KSU at the NTCIR-17 UFO Task

Tomokazu Hayashi, Hisashi Miyamori

Kyoto Sangyo University

Abstract

The TDE subtask aims to classify each cell of a table in a given annual securities report into four classes, while the TTRE subtask aims to select the corresponding cells in the table associated with a given text in the securities report.
In the TDE subtask, we designed methods for cell type classification using exhaustive tree structures based on the spanning sizes of the merged cells in the table.
In the TTRE subtask, we designed methods for cell retrieval based on the cell class.

Conclusion

In the TDE subtask, the conventional method showed higher performance than the proposed method.
In the TTRE subtask, the method that does not consider the tree structure of the table shows higher performance in Value and Total than the proposed method.
There are three expected reasons for the low performance of the methods considering the tree structure of the table: "The tree structure proposed by TUTA was built with information on joined cells, indentation, and formulas, but in this experiment, the tree structure was built only with information on joined cells.", "The cell unit assumed by TUTA was different from that of the TDE subtasks.", and "The fact that there were few complex tables in the TDE subtask data set that required a tree structure.".
Additional experiments on the DeEx dataset show that the proposed method performs best on macro-F1.

Introduction

We focused on the fact that tables in securities reports have a complex structure created by merged cells in the tables, etc., and developed a method to incorporate this structure into the representation of tables in several ways.

TUTA

- Hierarchical information contained in the table is represented by a tree-based structure called a bi-dimensional coordinate tree
- State-of-the-art results were achieved on five datasets

TUTA's assumption

When there is a hierarchical structure in a table, the size of the merged cells decreases gradually from the top (or leftmost) to the bottom (or rightmost) of the table.

Proposed method

We propose a method to construct a bi-dimensional coordinate tree in descending order of the size of the joint cells in the table, obtained by exhaustively checking the sizes of the merged cells in a table.

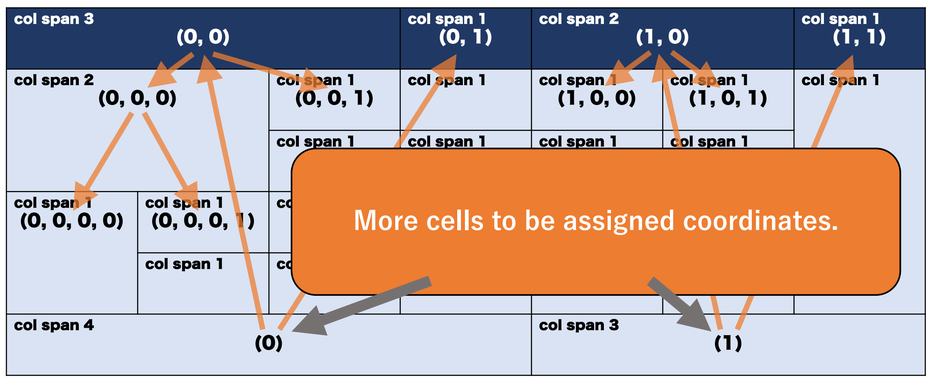

Figure 2: exhaustive vertical tree

Conventional method

Coordinates can only be defined for cells that follow the assumptions of TUTA, and large merged cells that appear in the middle of a table are not assigned coordinates.

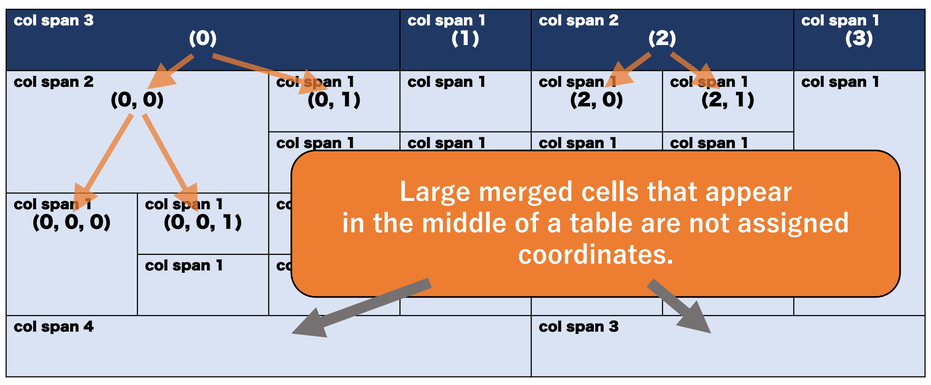

Figure 1: default vertical tree

Overview of the proposed method

1. **Coordinates can only be defined for cells that follow the assumptions of TUTA, and large merged cells that appear in the middle of a table are not assigned coordinates. The text of a given phrase and the text of each cell of a table in the same document are input to Text Encoder to obtain their respective embedded representations.**
2. **The similarity between the phrase and the cell text is calculated, and the Name is determined.**
3. **The features of the table are input into the model for Cell Type Classifier to obtain the class of each cell in the table.**
4. **The Value is determined based on the information in Name and the class of each cell in the table.**

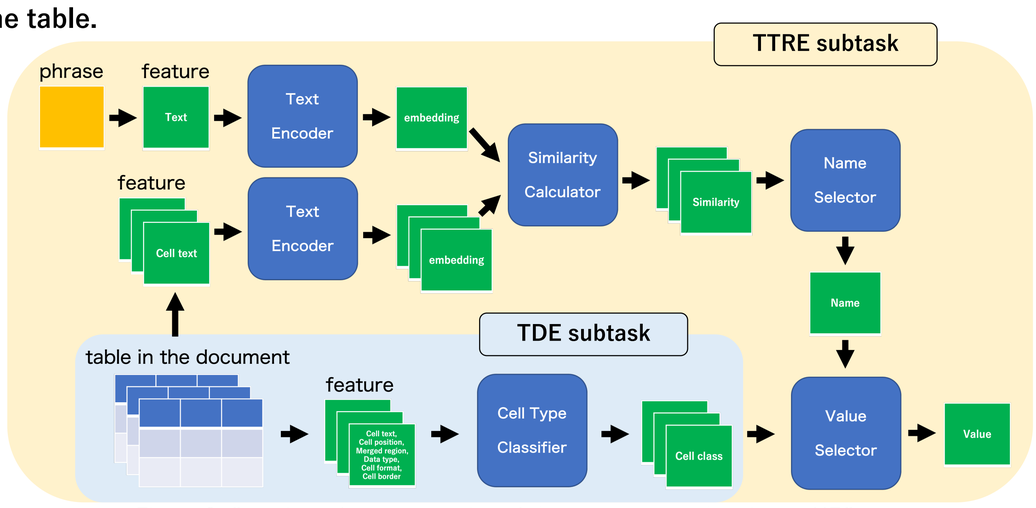

Figure 3: Overview of the processing of the proposed method in the UFO task

Results and discussion

The Macro-F1 score showed that the TUTA default tree performed the best. The Precision and Recall scores also showed that the TUTA default tree performed the best. For "Total," the method using multilingual-e5 as the Text Encoder and TUTA no tree as the Cell Type Classifier showed the best performance.

Table 1: Scores of TDE subtask in formal run and late submission

ID	Method	Precision	Recall	Macro-F1
81	**TUTA** no tree	0.7940	**0.8216**	0.8058
140	**TUTA** default tree	**0.8815**	**0.8267**	**0.8496**
150	**TUTA** exhaustive tree	**0.8533**	0.8196	**0.8352**

Table 2: Scores of TTRE subtask in formal run and late submission

ID	Method	Name			Value			Total
		Precision	Recall	F1	Precision	Recall	F1	F1
122	**multilingual-e5**	**0.3556**	**0.4574**	**0.3221**	0.0857	**0.5069**	0.1186	0.2204
127	**multilingual-e5 + TUTA** no tree	**0.3556**	**0.4574**	**0.3221**	**0.2682**	0.5062	**0.2719**	**0.2970**
146	**multilingual-e5 + TUTA** default tree	**0.3556**	**0.4574**	**0.3221**	**0.2658**	0.5062	**0.2704**	**0.2962**
156	**multilingual-e5 + TUTA** exhaustive tree	**0.3556**	**0.4574**	**0.3221**	0.2646	**0.5065**	0.2659	0.2940

As an additional experiment, we also validated using a dataset for cell type classification. The results show that the proposed method has the best performance.

Table 3: Scores of experiments with DeEx dataset

Method	metadata	notes	data	attributes	header	derived	macro-F1
TUTA no tree	0.8574	**0.4825**	**0.9937**	**0.8114**	**0.8701**	**0.8092**	**0.8041**
TUTA default tree	**0.8585**	0.4419	0.9930	**0.8382**	0.8547	0.7391	0.7876
TUTA exhaustive tree	**0.8671**	**0.5589**	**0.9931**	0.8045	**0.8638**	0.7491	**0.8061**
TUTA raw tree	0.8516	0.4271	0.9911	0.7888	0.8070	**0.7767**	0.7737

There are three expected reasons for the low performance of the methods considering the tree structure of the table: "The tree structure proposed by TUTA was built with information on joined cells, indentation, and formulas, but in this experiment, the tree structure was built only with information on joined cells.", "The cell unit assumed by TUTA was different from that of the TDE subtasks.", and "The fact that there were few complex tables in the TDE subtask data set that required a tree structure.".

KYOTO SANGYO UNIVERSITY

NTCIR-17 UFO