NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001282

OUC at NTCIR-17 UFO TDE and TTRE

Eisaku Sato
Otaru University of Commerce
Japan
g2020149@edu.otaru-uc.ac.jp

Keiyu Nagafuchi
Hokkaido University
Japan
nagafuchi@ist.hokudai.ac.jp

Yuma Kasahara
Hokkaido University
Japan
kasahara.yuma.l4@elms.hokudai.ac.jp

Kazuma Kadowaki
The Japan Research Institute, Limited
Japan
kadowaki.kazuma@jri.co.jp

Yasutomo Kimura
Otaru University of Commerce
Japan
kimura@res.otaru-uc.ac.jp

ABSTRACT

The OUC team participated in the Table Data Extraction (TDE) subtask and Text-to-Table Relationship Extraction (TTRE) of NTCIR-17 Understanding of Non-Financial Objects in Financial Reports (UFO). We report on our methodology in these subtasks and discuss the official results.

TEAM NAME

OUC

SUBTASKS

Table Data Extraction, Text-to-Table Relationship Extraction

1 INTRODUCTION

The Objects in Financial Reports (UFO) task consists of two subtasks: table data extraction (TDE) and text-to-table relationship extraction (TTRE) [1]. We describe our methodology in these subtasks and discuss the official results.

2 TDE

The TDE is a subtask that classifies the table cells in security reports into four classes: Metadata, Header, Attribute, and Data.

2.1 Methods

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model based on Transformers[2][3]. It is pre-trained by simultaneously executing two unsupervised learning tasks, masked language modeling (MLM) and next sentence prediction (NSP), on a large corpus. MLM is a task that randomly masks certain words in an input sentence and predicts the words that were masked. NSP is a task that identifies whether two input sentences are adjacent. These two tasks allow BERT to learn sentences in both directions. BERT can also be applied to various downstream tasks by fine-tuning.

We used the pre-trained BERT published by Tohoku University[1]. We also constructed FP_yuho, a BERT model that was pre-trained using the Wikipedia corpus then pre-trained on a corpus of annual security reports. The four models used for TDE are listed below.

(1) cl-tohoku/bert-base-japanese-v2 (base-v2)
(2) cl-tohoku/bert-base-japanese-v3 (base-v3)
(3) cl-tohoku/bert-large-japanese-v2 (large-v2)
(4) FP_yuho

[1]https://huggingface.co/cl-tohoku/

Table 1: Example of training dataset for TDE

text	class	label
関係会社の状況 (Status of Affiliates)	metadata	0
資本金 (Capital Stock)	header	1
ABC 株式会社 (ABC Corporation)	attribute	2
1 億円 (100 million yen)	data	3

Table 2: Hyperparameter settings for fine-tuning BERT in TDE

Hyperparameter	Value
Batch size	32
Learning rate	2e-5
Epoch	{20, 60}
Warmup ratio	0.1
Max seq length	128

The dataset for training is shown in Table 1 and created by extracting the text in the cells and combining it with the labels corresponding to the annotated classes. Using this dataset, each of the aforementioned models was fine-tuned with text as input and label as output.

Hyperparameters for fine-tuning were set as shown in Table 2, referring to study by Mosbach et al[4].

2.2 Results

As shown in Table 3, the ID88 method recorded the best score. This method uses the cl-tohoku/bert-large-japanese-v2 model.

Table 3: Scores of TDE subtask in formal run

ID	Team	F-Measure
88	OUC	0.9217
64	OUC	0.9195
56	OUC	0.9145
67	OUC	0.9129
76	OUC	0.9117
42	OUC	0.9113
41	OUC	0.9088
82	OUC	0.9066
96	OUC	0.9036
68	OUC	0.9004
62	OUC	0.8798
40	OUC	0.4657
39	OUC	0.4657

Table 4: Best performing method per pre-trained model

Rank	ID	Model	F-Measure
1	88	cl-tohoku/bert-large-japanese-v2	0.9217
2	56	FP_yuho	0.9145
3	42	cl-tohoku/bert-base-japanese-v2	0.9113
4	41	cl-tohoku/bert-base-japanese-v3	0.9088

Table 5: Portion of table for classification target

	連結貸借対照表計上額 The amounts reported on the consolidated balance sheet
(8) コマーシャル・ペーパー (8)Commercial paper	-
(9) 未払金 (9)Accrued expenses	(126,024)
(10) 未払法人税等 (10)Income taxes payable	(28,068)
(11) 預り連絡運賃 (11)Deferred Contact Fare	(2,686)

2.3 Discussion

The models, the model with the best scores after fine-tuning are listed in Table 4. Table 5 is an excerpt of the table for classification. For the cells included in this table, the outputs from the above four models were compared with the gold-standard data, and the relationship is shown in Table 6. However, Data was omitted because it was correct for all four models. The large-v2 model with the highest total score was able to correctly classify cells that the other three models were unable to answer correctly, suggesting the usefulness of the large model in this task. The FP_yuho model, which was originally trained, had a total score similar to the base-v2 model, but there were differences in the cells that were correctly or incorrectly classified.

3 TTRE

TTRE is a subtask that maps text in the text within an annual report to cells in tables within the same document. To map text to cells, it is necessary to map multiple tables in a document to text first. Therefore, we divided this task into two steps and attempted the final task by using different methods in each step. The details of each step are given below.

Step 1: Correspondence between text and table

Match the text in the marked text with the table containing the corresponding cells. Two methods were used: a rule-based method and method using Sentence-BERT.

Step 2: Mapping text to cells

Based on the combination of text and table mapped in Step 1, text-to-cell mapping is executed. A method using ChatGPT (GPT-4) was used.

3.1 Step 1: Correspondence Between Text and Table

3.1.1 Rule-based Method. For any given marked text, the table immediately preceding it was mapped to the pair. However, if there was no table immediately preceding the text, as in the case where the text was at the beginning of the document, the text was paired with the table immediately following it.

3.1.2 Method Using Sentence-BERT. Sentence-BERT is a model that specializes BERT to compute document vectors. Like BERT, it can be fine-tuned for a specific task. Therefore, we constructed a dataset from the training data and carried out fine-tuning of Sentence-BERT.

Fine-tuning was performed on the pre-trained BERT model cl-tohoku/bert-base-japanese-v2 published by Tohoku University with the following settings.

To use multiple negatives ranking loss as the loss function, a triplet of Anchor (reference sentence), Positive (sentences similar to the reference sentence), and Negative (sentences not similar to the reference sentence) had to be created as the training data set. Details on the creation of the triplet are given below.

Anchor

The marked sentences were used as the Anchor. However, since most of the marked sentences were words, the sentence in the p tag containing the marked sentence was used as the Anchor.

Positive, Negative

Sentences texting the table corresponding to the marked sentence were designated as Positive, and sentences texting one table randomly selected from the rest were designated as Negative. The following four steps were used to process the table into a single sentence.

Step 1: Obtain any table row by row.
Step 2: For each row of cells, remove duplicates.
Step 3: Add a Tab at the end of each cell.
Step 4: Add a line break at the end of each row to make it a single sentence.

Using the fine-tuned model we constructed, we calculated the document similarity between the text and textualized table and mapped the text and table with the highest values as pairs.

Table 6: Example comparison of output of gold standard and each model for cells included in Table 5

cell-text	Gold standard	base-v2	base-v3	large-v2	FP_yuho
連結貸借対照表計上額 The amounts reported on the consolidated balance sheet	**header**	**header**	**header**	**header**	**header**
(8) コマーシャル・ペーパー (8)Commercial paper	**attribute**	**attribute**	**attribute**	**attribute**	**attribute**
(9) 未払金 (9)Accrued expenses	**attribute**	**attribute**	header	**attribute**	**attribute**
(10) 未払法人税等 (10)Income taxes payable	**attribute**	header	header	**attribute**	**attribute**
(11) 預り連絡運賃 (11)Deferred contact fare	**attribute**	**attribute**	header	**attribute**	header

Table 7: Hyperparameter settings for fine-tuning Sentence-BERT in TTRE

Hyperparameters	Value
Batch size	16
Learning rate	2e-5
Epoch	4
Warmup ratio	0.1
Max seq length	128

Table 8: Hyperparameter settings for fine-tuning Sentence-BERT in TTRE

Model	GPT-4 (June 10, 2023)
Temperature	0.0
Prompt	source code1

3.2 Step 2: Mapping text to cells

In this step, ChatGPT was used to extract the cells corresponding to the text; as input to ChatGPT, the marked text and textualized table are given. The output is the text of the cells. For the textualization of the tables, we used the same method as for the textualization of tables in Sentence-BERT's fine-tuning.

3.2.1 Settings of ChatGPT. In using the OpenAI API, the settings were those shown in Table 8.

source code 1: Prompt

```
{
"role": "system",
"content": f"""
    ##Data Extraction
    Receive textualized tabular data and
    understand exactly what it contains.
    Next, receive the querys of the data
    to be extracted and answer
    all applicable data, header of the data and
    attribute of the data.
    If no applicable data is found,
    set Value to null.
    However, modification of the data is prohibited.
    Do not give unit either.
    The output should be JSON as shown below.

    ##Output Example(JSON)
    (
        query1: [
            [data1, header1, attribute1],
           [data2, header2, attribute2]
        ],
        query2: [
            [data1, header1, attribute1],
            [data2, header2, attribute2]
        ],
        query3: null
    ),"""
},
{
"role": "user",
"content": f"""
    ##Input Data: {html_text}

    ##The Data to Be Extracted: {query}"""
}
```

3.2.2 Conversion of ChatGPT output to cell IDs. The format for submitting the results of this task is a pair of text IDs and cell IDs. However, since the output of ChatGPT is the text of a cell, it is necessary to obtain the cell ID from the text of the cell. Therefore, the ID of the cell, the text of which is an exact match between the text of the cell output by ChatGPT and the text of each cell in the corresponding table was used as the final output.

3.3 Results

The results of the formal-run are shown in Table 9. The method ID37 uses the rule-based approach in mapping text to table in Step 1. The method ID98 uses Sentence-BERT in the mapping between text and table in Step 1. The rule-based method ID32 resulted in a Name score about 4.3 points higher, Value score about 6.8 points higher, and Total score about 5.5 points higher than the ID98 method that used Sentence-BERT. Comparing the formal-run overall and rule-based methods, the Total score was lower, but Value score was fourth.

Table 9: Scores of TTRE subtask in formal run

ID	Team	Name	Value	Total
37	OUC	0.1447	0.1823	0.1635
98	OUC	0.1018	0.1146	0.1082

3.4 Discussion

The rule-based method performed better than the method using Sentence-BERT in linking the text of the main text to the table, indicating that there are issues when using Sentence-BERT for the linking operation. It can be inferred that not only the similarity between the text of the main text and the text of the table, but also the distance between the text of main text and table should be considered as the basis for the linking. It is also necessary to find a way to convert the table to text in a manner that makes the text more natural, rather than simply joining words together.

A further action to address the failure to link the text of the main text to the table would be to restructure the prompt so that when ChatGPT is used to extract specific cells from the table, it also outputs a certainty indicator. If certainty falls below an arbitrary threshold, the accuracy of the links could be improved by linking to the next most similar table.

It should be noted that the extraction process by ChatGPT was able to handle not only text in which a single cell is the correct answer, but also text in which multiple cells are the correct answer.

4 CONCLUSIONS

We participated in the Tabular Data Extraction (TDE) subtask and Text-to-Table Relationship Extraction (TTRE) subtask of NTCIR-17's Understanding Non-financial Objects in Financial Reports (UFO). In the TDE task, we used BERT for label classification. In the TTRE subtask, we used either a rule-based method or SentenceBERT to identify the target tables, followed by using ChatGPT to locate the relevant cells.

REFERENCES

[1] Yasutomo Kimura, Hokuto Ototake, Kazuma Kadowaki, Takahito Kondo, and Makoto P. Kato. Overview of the ntcir-17 ufo task. *Proceedings of The 17th NTCIR Conference*, 12 2023.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008, 2017.

[4] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines. In *International Conference on Learning Representations*, 2021.