Semantic Structure Extraction for Spreadsheet Tables with a Multi-task Learning Architecture

Haoyu Dong[1], **Shijie Liu**[2], **Zhouyu Fu**[3], **Shi Han**[1], **Dongmei Zhang**[1]
[1]Microsoft Research, Beijing, China.
[2]Beihang University, Beijing, China
[3]Alibaba Local Service Company (ALSC) Lab, Beijing, China
{hadong, shihan, dongmeiz}@microsoft.com, shijie_liu@buaa.edu.cn, zhouyu.fz@alibaba-inc.com

Abstract

Semantic structure extraction for spreadsheets includes detecting table regions, recognizing structural components and classifying cell types. Automatic semantic structure extraction is key to automatic data transformation from various table structures into canonical schema so as to enable data analysis and knowledge discovery. However, they are challenged by the diverse table structures and the spatial-correlated semantics on cell grids. To learn spatial correlations and capture semantics on spreadsheets, we have developed a novel learning-based framework for spreadsheet semantic structure extraction. First, we propose a multi-task framework that learns table region, structural components and cell types jointly; second, we leverage the advances of the recent language model to capture semantics in each cell value; third, we build a large human-labeled dataset with broad coverage of table structures. Our evaluation shows that our proposed multi-task framework is highly effective that outperforms the results of training each task separately.

1 Introduction

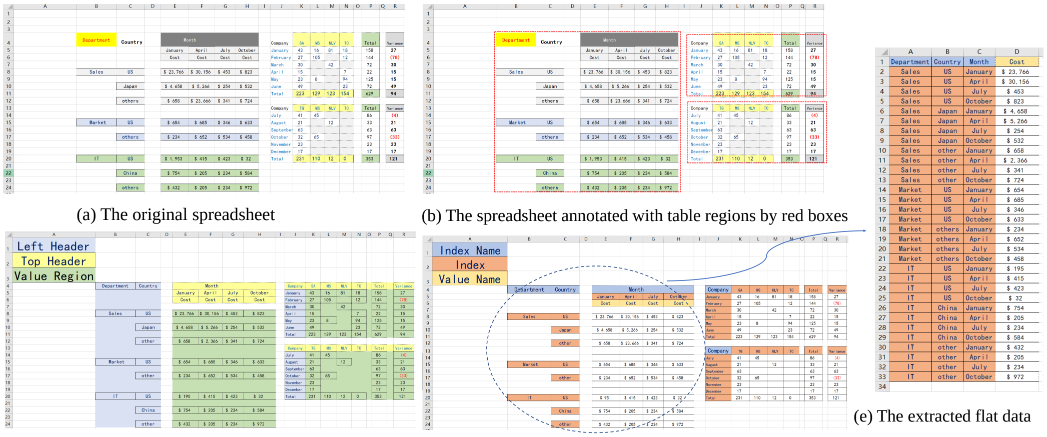
Spreadsheets are the most popular end-user development tool for data management and analysis. Unlike programming languages or databases, no syntax, data models or even vague standards are enforced for spreadsheets. Figure1(a) shows a real-world spreadsheet. To enable intelligent data analysis and knowledge discovery for the data in range B4:H24, one needs to manually transform the data to a standard form as shown in Figure1(e). It would be highly desirable to develop techniques to extract the semantic structure information for automated spreadsheet data transformation.

Semantic structure extraction entails three chained tasks to: (1) detect tables and locate their respective ranges, shown with red rectangular boxes in Figure1(b); (2) split each table into different components, highlighted with different colors in Figure1(c); (3) categorize individual cells into their corresponding cell types depending on their roles in the transformed table, encoded with different colors in Figure1(d). We also show the transformed data in Figure1(e), where different cell types are highlighted using the same coloring scheme as in Figure1(d).

Learning the semantic structure for spreadsheets is challenging. While table detection is confounded by the diverse multi-table layouts, component recognition is confounded by the various structures of table components, and cell type classification requires semantic-level understanding of cell values. Moreover, the tasks are chained in the sense that latter tasks need to leverage the outcomes of prior tasks. This poses challenges on preventing error propagation, but also provides opportunities for utilizing additional cues from other tasks to improve the current task. For example, header extraction may help table detection since headers need to be inside the table region and vice versa.

In this paper, we present a multi-task learning framework to solve spreadsheet table detection, component recognition, and cell type classification jointly. Our contributions are as follows:

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.



	A	B	C	D
1	Department	Country	Month	Cost
2	Sales	US	January	$ 23,766
3	Sales	US	April	$ 30,156
4	Sales	US	July	$ 453
5	Sales	US	October	$ 823
6	Sales	Japan	January	$ 4,658
7	Sales	Japan	April	$ 5,266
8	Sales	Japan	July	$ 254
9	Sales	Japan	October	$ 532
10	Sales	other	January	$ 658
11	Sales	other	April	$ 2,366
12	Sales	other	July	$ 341
13	Sales	other	October	$ 724
14	Market	US	January	$ 654
15	Market	US	April	$ 685
16	Market	US	July	$ 346
17	Market	US	October	$ 633
18	Market	others	January	$ 234
19	Market	others	April	$ 652
20	Market	others	July	$ 534
21	Market	others	October	$ 458
22	IT	US	January	$ 195
23	IT	US	April	$ 415
24	IT	US	July	$ 423
25	IT	US	October	$ 32
26	IT	China	January	$ 754
27	IT	China	April	$ 205
28	IT	China	July	$ 234
29	IT	China	October	$ 584
30	IT	other	January	$ 432
31	IT	other	April	$ 205
32	IT	other	July	$ 234
33	IT	other	October	$ 972
34				

(a) The original spreadsheet

(b) The spreadsheet annotated with table regions by red boxes

(e) The extracted flat data

(c) The spreadsheet annotated with component masks

(d) The spreadsheet annotated with cell type masks

Figure 1: An example that visually illustrates the table semantic structure extraction process.

1. We formulate spreadsheet table structure extraction as a coarse-to-fine process including table detection, component recognition, and cell type classification. We also build a large labeled dataset.

2. To capture the rich information in spreadsheet cells for model training, we devise a featurization scheme containing both hand-crafted features and model-based semantic representations.

3. We propose a multi-task framework that can be trained to simultaneously locate table ranges, recognize table components and extract cell types. Our evaluation shows that the proposed multi-task framework is highly effective that outperforms the results of training each task separately.

2 Problem Statement

Table detection is the task of detecting all tables on a sheet and locating their respective ranges.

Semantic table component recognition is the task of recognizing the top headers, the left headers and the value region in a table. As shown in Figure1(b), a top (left) header contains the labels of table columns (rows) and a value region contains table data (i.e., a body of the table).

Cell type classification is the task of classifying each cell into a certain type such as value, value name, index, and index name. A **value** is a basic unit in the value region. A **value name** is a summary term that describes values. As shown in Figure1(a), "Cost" at E6 is a value name to describe the values in E8:H24. After the data extraction, as shown in Figure1(e), "Cost" at D1 is the label of Column D. An **index** refers to individual values that can be used for indexing data records. In Figure1(a), "January" - "October" at E5:H5 are indexes of columns E - H respectively. A group of indexes is used to breakdown the dataset into subsets. After data transformation, it will form a single data field as Column C shows in Figure1(e). An **index name** is a summary term that describes the indexes. In the previous example, " Month" is the index name of indexes "January" - "October". After data transformation, the " Month" in Figure1(a) corresponds to the column label at C1 in Figure1(e).

3 Method

3.1 Datasets

The web-crawled WebSheet dataset contains 4,290,022 sheets with broad coverage of diverse table structures [1]. We propose a human-labeled dataset, SemanticSheet, for semantic table structure extraction. It includes: (1) 22,176 tables with annotated bounding boxes, which are sampled from WebSheet using an active learning method [1]; (2) 9,053 tables with our structural component annotations, which are randomly sampled from the annotated bounding boxes; (3) 3,503 tables with our cell type annotations, which are sampled using heuristics to balance different table structures based on the structural component annotations. To control labeling quality, all labeled tables have to be verified by an experienced human labeler and be revised until the overall agreement achieves 95%.

3.2 Cell Featurization

To capture effective cell features, we leverage both hand-crafted features and learning-based semantic representations. In general, there are four major information sources of a cell, i.e., value string, data

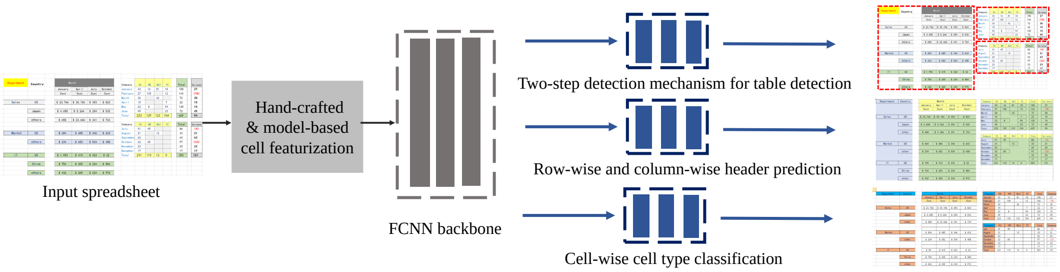

Input spreadsheet
Hand-crafted & model-based cell featurization
FCNN backbone
Two-step detection mechanism for table detection
Row-wise and column-wise header prediction
Cell-wise cell type classification

Figure 2: Framework for spreadsheet table detection, component recognition, and cell classification.

format, cell format, and formula. [1] proposed 20 hand-crafted features and proved its high effectiveness in table detection. But this featurization schema lacks high-level semantic features, which are important for table semantic structure extraction, especially the task of cell type classification.

The recently proposed language models in the natural language processing domain enable to learn embeddings of token sequences with remarkable generalizability. This motivates us to leverage the recent advances in language modeling to derive such high-level semantics in the spreadsheet domain. We incorporate BERT [2] to extract semantic embeddings for cell values. To control the complexity in our model, a point-wise CNN is adopted to reduce the embedding size for each cell value to 32. This CNN model can be jointly trained with the other modules in our framework.

3.3 Multi-task Framework

Figure2 shows our framework, including a shared backbone and three branches for multi-task learning.

Convolutional neural network backbone: Since our featurization method only captures cell-level information, we adopt a Fully Convolutional Neural Network (FCNN) to learn spatial correlations between cells. The FCNN backbone enables our framework to learn shared feature representations, which are able to generalize across multiple different tasks.

Table detection: We adopt and implement the two-stage detection mechanism introduced in [1].

Structural component recognition: This module learns to predict the separation lines between the headers and value region. If there is no top or left header in a table, this module predicts the top and left boundary of the value region. A row-wise average pooling and a softmax function are used to predict the horizontal separation line across rows. And similar with column-wise predictions.

Semantic cell type classification: This module uses a FCNN for cell-level type classification.

Moreover, to leverage the relationships in our three coarse-to-fine tasks, the component recognition branch takes the results of the table detection as an additional feature channel, and the cell type classification branch takes the results of the component recognition as it's additional feature channel.

4 Experiments

4.1 Implementation and Experiment Setup

We take ResNets [3] as the backbone in our framework and exclude all pooling layers to avoid losing cell-level precision. The CNNs for structural component recognition, and cell type classification consist of five convolutional layers. The whole dataset is randomly divided into 80% training set and 20% test set. The loss functions for different tasks are added together for joint training.

4.2 Evaluation Metrics

For table detection, we adopt the Error-of-Boundary (EoB) metric [1] to measure how precisely the detection result is aligned to the ground truth bounding box. For structural component recognition, we calculated the accuracy for top and left header separation lines. For cell type classification, we report the average F1 for the index, index name, and value name predictions.

4.3 Multi-task Framework Evaluation

For the comparison study, we adapt Mask RCNN [4], the state-of-the-art multi-task method for image object detection and segmentation. To evaluate the effectiveness of multi-task joint training, we conduct a comparison between single-task and multi-task training of our proposed method.

Table 1: Comparison results of spreadsheet table semantic structure extraction.

%	Tabel detection				Component recognition		Cell type classification		
	EoB-0		EoB-2		Top header	Left header	Value name	Index	Index name
Model	Recall	Precision	Recall	Precision	Accuracy	Accuracy	F1	F1	F1
Mask R-CNN	48.4	40.2	75.5	64.2	56.5	59.1	44.3	65.0	50.2
Single-task with BERT	81.1	77.4	91.5	87.2	90.4	89.9	70.3	84.6	69.3
Multi-task with BERT	**81.8**	**78.3**	**92.1**	**89.1**	**91.2**	**92.3**	**71.9**	**85.3**	**70.8**
Multi-task w/o BERT	81.5	**78.3**	91.8	89.0	90.1	90.8	66.2	82.2	64.4

For table detection evaluation, as shown in the left part of Table 1, our method achieves improvements over all baselines. Moreover, the multi-task version of our method performs better than its single-task version in both EoB-0 and EoB-2, indicating that other tasks help with table detection. We attribute such improvements to the learning of intrinsic relationships between these tasks by training a joint model. For the component recognition evaluation, compared with single-task training, our multi-task framework also achieves 0.8% and 2.4% accuracy gains. And the right side of Table 1 shows the results of cell type classification, our method still achieves a large margin of improvement over Mask R-CNN. Compared with single-task training, the joint framework improves the value name, index, and index name predictions by 1.6%, 0.7%, and 1.5% respectively.

4.4 Effectiveness of Semantic Features Extracted by Language Models

We design a comparison experiment to evaluate the effectiveness of language models. For the baseline (multi-task w/o BERT), we only use the 20-dimensional hand-crafted features; while for the treatment (multi-task with BERT), we use the full feature set including the semantic features extracted by BERT. The comparison results are shown in Table 1, indicating that by incorporating the semantic features, our proposed model achieves higher (or on-par) accuracy in all these three tasks. Specifically, since cell type classification heavily depends on semantics, there is 6.4% F1 gain for the index name prediction and 5.7% F1 gain for the value name prediction.

5 Related Work

Some research has been done to extract table data from a spreadsheet [1, 5, 6, 7]. But they only focused on one of the subtasks such as spreadsheet table detection and hierarchical header extraction. And the semantics in spreadsheets have largely been overlooked. We first propose a multi-task framework to learn table detection, component recognition, and cell type recognition simultaneously on a large scale and incorporate language models for capturing semantics in spreadsheets.

References

[1] Haoyu Dong, Shijie Liu, Shi Han, Zhouyu Fu, and Dongmei Zhang. Tablesense: Spreadsheet table detection with convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[5] Elvis Koci, Maik Thiele, Óscar Romero Moral, and Wolfgang Lehner. A machine learning approach for layout inference in spreadsheets. In *IC3K 2016: Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management: volume 1: KDIR*, pages 77–88. SciTePress, 2016.

[6] Kerry Shih-Ping Chang and Brad A Myers. Using and exploring hierarchical data in spreadsheets. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2497–2507. ACM, 2016.

[7] Zhe Chen and Michael Cafarella. Integrating spreadsheet data via accurate and low-effort extraction. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1126–1135. ACM, 2014.