A Common Framework for Developing Table Understanding Models

Jay Pujara, Arunkumar Rajendran, Majid Ghasemi-Gol, and Pedro Szekely

Information Sciences Institute, University of Southern California,
jpujara@isi.edu, arunk26jan@gmail.com, ghasemig@usc.edu, pszekely@isi.edu
https://isi.edu

Abstract. A wealth of knowledge is contained in tabular data, and there are a vast number of efforts to model and capture this knowledge. Unfortunately, these efforts have disparate inputs, outputs, and goals hindering research progress and making table understanding tools difficult to use in practice. In this paper, we propose a table understanding framework that formalizes the problem of understanding tabular data into three distinct subtasks: cell classification, block detection, and relation prediction. We introduce a common API for table understanding systems that supports a host of existing approaches and allows easy development of new approaches. Our framework supports approaches that range from heuristic rules to probabilistic models, allows outputs that span simple, correlational tuples to sophisticated, semantic knowledge graphs, and provides tools for visualizing model outputs and transforming complex tabular data into flattened relational dataframes.
.

1 Introduction

Tables are a natural way for humans to express complex relational and quantitative data, resulting in a proliferation of valuable, structured data on Web pages [1], spreadsheets, and databases. When producing and using tabular data, humans often follow well-documented principles for organization and layout [6]. The conceptual underpinnings of the representation of tables have been thoroughly investigated from many perspectives including presentational, functional, structural, and semantic [4]. Unfortunately, the theoretical depth of table models has not been fully reflected in implementations of table understanding systems, and no overarching framework exists for integrating various efforts on table understanding. In this paper, we propose a practical table understanding framework that decomposes this complex problem into a set of modular, well-specified subtasks and provides associated APIs to encourage a common research paradigm for table understanding.

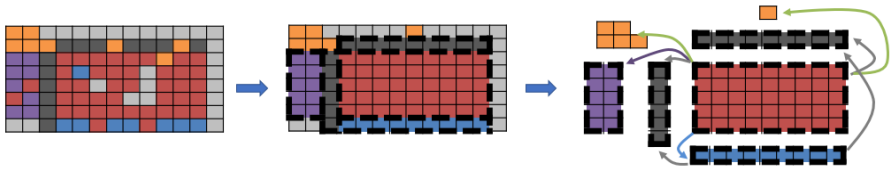

Fig. 1: An illustration of the three subtasks in the table understanding problem: cell classification, block detection, and layout relation prediction

2 Problem Definition

We formulate the table understanding problem based on the exhaustive efforts on building theoretical models of tabular data [6,3]. We define a table T as a JxK matrix structure composed of cells $(c_{j,k})$, $T \triangleq [c_{1,1} \dots c_{1,K} \dots c_{J,1} \dots c_{J,K}]$. We assume that the layout of a table implicitly expresses relationships between a set of cells. The goal of a table understanding system is to recognize and represent these relationship between cells.

Relationships between cells can be expressed at a simple, correlational level (identifying a tuple of associated values) or at a deep, semantic level (producing a knowledge graph with ontological mappings of entities, types and properties). We believe there is a need to support the full spectrum of table understanding methods. For tables with complex layouts and specialized domains, correlational associations may be immensely valuable. For conventional layouts with mainstream entities, a more knowledge-driven output may be desired.

Our table understanding framework provides the overall architecture and interface to support the full spectrum of table understanding approaches, from structural and syntactic to fully semantic systems. We organize this framework three primitive operations, illustrated in Figure 1: cell classification, block detection, and relation prediction. These three tasks can be viewed as answering three basic questions:

Cell Classification: What type of data does this cell contain?
Block Detection: What logical groupings of cells are present?
Layout Relationships: How are groups of cells related to each other?

2.1 Cell Classification

The goal of cell classification is to assign a label, $(l^c_(j, k))$, to each cell, $\mathbf{CC}(T) \rightarrow \left[l^c_{1,1} \dots l^c_{1,K} \dots l^c_{J,1} \dots l^c_{J,K}\right]$. The domain of cell labels can be customized based on the complexity of the table understanding system, for example producing simple datatype labels (string, integer, float, datetime) for a syntactic system, more functional labels (metadata, header, attribute, value) based on table structure [5], or perform semantic typing [2] of ontological classes (Person, Place, Organization) for a semantic table understanding system.

2.2 Block Detection

Block detection identifies a region composed of individual cells that share a common functional role in a table, defined as a block (B_i). Blocks can be defined hierarchically, such that a single large block can be composed of several smaller blocks (and regions), some of which may be further subdivided. Thus a block can be defined as either a rectangular region of cells, or the union of a set of sub-blocks, $B_i \triangleq [B_{i1}, \ldots B_{ik} | \{c_{a,b} \ldots c_{x,y}\}]$. Block detection systems identify a set of blocks in a table and assign a label to each block, $\mathbf{BD}(T) \rightarrow \left[\left\langle B_1, l_1^b \right\rangle \ldots \left\langle B_s, l_s^b \right\rangle\right]$. Similar to cell labels, block labels can also be defined at several levels, ranging from syntactic (headers, notes, attributes, values) to semantic (entities from a particular domain).

2.3 Layout Relation Prediction

The final task in table understanding is determining the relational structure between blocks. Relationships can take many forms, but common relationships include subset relationships (e.g., a block of year attributes may be related to a block of month attributes because the months are temporal subsets of the years) and attribute-value relationships (e.g., a temperature measurement may have attributes of the year and month of measurement). Each relationship can be specified as a labeled, directed edge between blocks, $\mathbf{RP}(T) \rightarrow [\langle B_s, B_t, l_1^r \rangle \ldots \langle B_u, B_v, l_n^r \rangle]$. As with other subtasks, the label space can be defined at differing levels of granularity, from basic subset, indexing, and attribute relationships to ontologically meaningful properties (e.g., age, location, source).

3 Table Understanding Framework

Using the table understanding formalism introduced in the previous section, we have developed a table understanding framework that provides common abstractions and tools for all three table understanding subtasks. We summarize and illustrate the key features of our framework below.

Implementation: The table understanding framework is implemented as a set of Python APIs and accompanying documentation.

Common Representations: The framework defines a common representation for tabular data that can be loaded as CSV or Excel-style formats, and supports translation tools for Web tables.

APIs: In addition, the framework defines appropriate abstract classes for each table understanding subtask (as shown in Figure 2). We also provide an elegant method to supply custom labels for each task, so that structural and semantic modeling approaches can reuse the same abstract classes. Label outputs are specified as probability distributions over label classes to support machine learning models that produce scored outputs.

Reference Implementations: The table understanding framework supports several reference implementations, including baseline models that demonstrate simple, functional outputs and more sophisticated CRF-based cell classification

```python
import abc
import numpy as np
from reader.sheet import Sheet
from type.cell.cell_type_pmf import CellTypePMF
from typing import List


class CellClassifier(abc.ABC):
    @abc.abstractmethod
    def classify_cells(self, sheet: Sheet) -> 'np.ndarray[CellTypePMF]':
        pass
```

Fig. 2: Cell Classification API

		Health Outcomes		Health Factors	
State	County	Z-Score	Rank	Z-Score	Rank
Alabama					
Alabama	Autauga	-1.07	7	-0.49	11
Alabama	Baldwin	-1.54	2	-0.86	3
Alabama	Barbour	-0.17	33	0.56	58
Alabama	Bibb	-0.01	40	-0.12	29

Fig. 3: Colorized output from the table understanding framework

and relation prediction models and a decision-tree based block detection algorithm.

Tools: The framework supports several tools to allow developers to visualize and use the outputs of table understanding models. One such output, shown in Figure 3 is a colorized version of the table that shows cell types and block boundaries. Additional rows are added at the bottom of the sheet to summarize block relations. Another such output is a flattened dataframe representation (with a single record per row) generated by using relational mappings between blocks.

Open Source Release: Our table understanding framework API is available at https://github.com/usc-isi-i2/isi-table-understanding

Acknowledgements This material is based upon work supported by United States Air Force and the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8650-17-C-7715 and award W911NF-18-1-0027.

References

1. Crestan, E., Pantel, P.: Web-scale table census and classification. In: International Conference on Web Search and Data Mining. pp. 545–554 (2011)
2. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics **28**(3), 245–288 (2002)
3. Hurst, M.: Towards a theory of tables. International Journal of Document Analysis and Recognition **8**(2-3), 123–131 (2006)
4. Hurst, M.F.: The interpretation of tables in texts (2000)
5. Koci, E., Thiele, M., Romero, O., Lehner, W.: Cell classification for layout recognition in spreadsheets. In: International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management. pp. 78–100 (2016)
6. Wang, X.: Tabular extraction, editing, and formatting (1996)