2015 The 5th International Workshop on Computer Science and Engineering
doi: 10.18178/wcse.2015.04.014

From Unstructured to Structured Tabular Data Using a Rule Engine

Alexey O. Shigarov, Igor V. Bychkov

Institute for System Dynamics and Control Theory of SB RAS, 134 Lermontov st., 664033, Irkutsk, Russia

Keywords: unstructured tabular data integration, table understanding, information extraction from tables, table analysis and interpretation.

Abstract. Today, a huge amount of unstructured tabular data is contained in tables from different sources, e.g. image documents, web pages, and spreadsheets. Sometimes these tables are only available data source. To use that information in business intelligence we need to transform data from these tables to structured form like relational databases. We propose an approach to the tabular data transformation from unstructured (spreadsheets) to structured (relational databases) form using a rule engine. Our table interpretation rules can use spatial, style (typographical), and natural language information from tables. The experimental evaluation shows that the approach can be applied to a wide range of tables from statistical and financial reports.

Introduction

Nowadays, many researchers in data management, e.g. [1-4], note that issues on unstructured data integration become increasingly important. Unstructured data usually refers to any information that does not have a predefined formal data model or does not fit into a table of a relational database. The documents, web pages, and spreadsheets may contain tables, which do not have any formal data model. These tables are intended to be interpreted by humans but not designed for high-level machine processing like SQL queries.

In practice, the transformation of tabular data from unstructured to structured form is required in many cases. For example, tables presented in unstructured form are often the only available source of statistical or financial information. But only after transforming information from these tables to databases it is available for using in business intelligence, including online analytical processing, data mining, and knowledge discovery.

To transform tabular data we need to automate table understating, which is consists in recovering relationships among entries (data values), labels (attributes), and dimensions (categories) [5]. As Hurst [6] notes, the table understanding involves the following steps: (1) table location (to detect positions of a table inside a source); (2) table recognition (to recover individual cells); (3) functional analysis (to find attributes and data in cells, i.e. to recover cell roles); (4) structural analysis (to recover relationships between cells); and (5) interpretation (to extract facts from a table, e.g. relationships between labels and dimensions).

The present work is restricted to the issues: how to recover semantic relationships in a table (i.e. cell-role, label-value, label-label, and label-dimension pairs). In terms of Hurst [6], we suggest to automate the following steps of table understanding: functional analysis, structural analysis, and interpretation.

There are several challenges in the table understanding. A table can be produced or generated by a huge amount of ways. Table features originate from typographical standards, corporative practice, ad hoc software, data formats, and human inventiveness. To reduce the complexity of table understanding the existing methods use various assumptions (heuristics) about tables. Usually those assumptions are entirely embedded in their algorithms. It constrains a range of tables, which can successfully be understood by them.

We assume that tables produced by the same vendor often have similar structures, styles, and content. It allows defining a set of production rules for describing how these tables can be analyzed and interpreted. We propose to develop separate sets of table interpretation rules (knowledge bases) for different sets of similar tables. In that case, the process of the table understanding is performed

as rule firing. It provides processing of a wide range of tables having various complex structures and features.

We develop the CELLS system based on the proposed approach. The system is designed for integrating unstructured tabular data. It allows extracting data from tables presented in Excel spreadsheet files and generating tables in structured (canonical) form. The system use Drools Expert (drools.org) as a rule engine. Table interpretation rules are expressed in MVEL (mvel.codehaus.org) expression language. The experimental results demonstrate that the system can be applied for populating a database from spreadsheets with unstructured tabular information.

Tabular Data

Unstructured tabular data, including spatial, style, and natural language content of cells, can be obtained from Excel spreadsheets by existing tools (e.g. Apache POI). This information can be used to generate facts, which are asserted into a working memory for logical inference. To represent those facts about cells we propose a table model. Our table model is based on the set of general assumptions about cells, which are described the class of processing tables.

1.1 General assumptions for tables

1. A cell is characterized by the positions (coordinates) in the column and row space, style, and content.
2. A cell can be located on several consecutive rows and columns, i.e. it can cover a few grid tiles, which always form a rectangle.
3. A cell can contain only text.
4. A cell can serve as either entry (data value) or label (attribute).
5. An entry represents a data value and a label describes entries.
6. A label can address entries and other labels either in rows or columns only thus labels can form hierarchical relationships among themselves.
7. A label can be a value of a dimension.

An example of a table with those relationships is shown in Fig. 1.

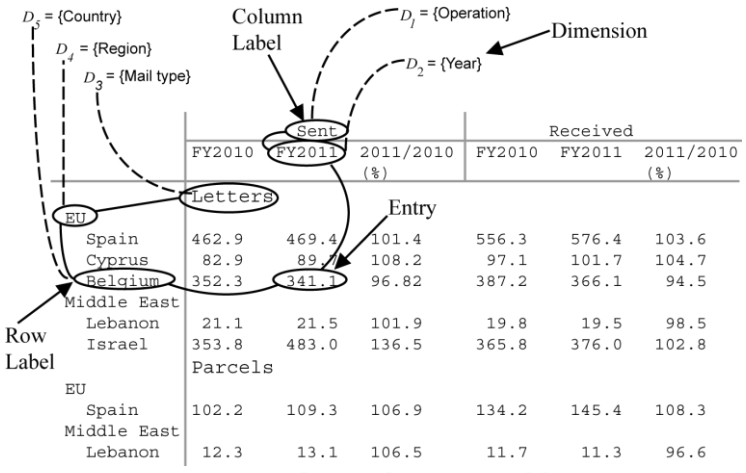

Fig.1: Relationships in a table

1.2 Table model

The model is designed to present facts about a table in process of logical inference. It consists of two levels: physical and logical. The first of them presents the visual composition of a table. The second level is intended for presenting the semantic composition of a table.

The physical level describes geometric positions, styles (settings of the graphical formatting) and textual content of cells. This level $T_p = (S_r, S_c, C)$ consists of the following sets.

1. S_r is a set of rows and S_c is a set of columns.
2. C is a set of cells where each cell $c = (c', p, G)$ includes: content c'; coordinates $p = (c_l, r_t, c_r, r_b)$ in the rows S_r and columns S_c (c_l is a left column, r_t is a top row, c_r is a right column, and r_b is a bottom row); a set of style settings (font metrics, colors, text alignment, borders, etc.) G.

The logical level presents semantic relationships (i.e. cell-role, label-entry, label-label, and label-dimension pairs).

This level $T_l = (D, L_r, L_c, E)$ consists of the following sets.

1. $D = \{D_i\}$ is a set of dimensions presented in the processed table. Each of them is a set of dimension values $D_i = \{d_j\}$.
2. L_r is a tree of row labels and L_c is a tree of column labels. These trees present relationships between their labels. Each label $l = (l')$ has content l', which is not a value of any dimension D_i.
3. E is a set of entries where each entry $e = (e', D', L')$ includes: content e'; a set of values from dimensions D_i related with this entry D', a set of labels from trees L_r and L_c related with this entry L'.

Table Interpretation Rules

a	**when** $c : CCell(rt == 1, cl > 1) **then** modify ($c) { setRole(Role.COLLABEL) }
b	**when** $c : CCell(style.getFont().getColor() == "#0000ff") **then** modify ($c) { setRole(Role.ENTRY) }
c	**when** $c1 : CCell() $c2 : CCell(rt == $c1.rb + 1, $c1.cl <= cl && cr <= $c1.cr) **then** $c1.addConnectedCell($c2)
d	**when** $d : CDimension(name == "YEAR") $c : CCell (cl == 1, text matches "[2][0][0-1][0-4]") **then** $c.setDimension($d)

Fig.2: Samples of table interpretation rules

Table interpretation rules define how we can interpret what we know (i.e. positions, style settings, and content of cells in a table) to recover what we do not know (i.e. semantic relationships in the table). The left hand side (when) of a rule defines conditions using known facts about cells and dimensions. The right hand side (then) of a rule recovers unknown facts about a table, including assignment cell roles (label or entry), binding cells (i.e. creating label-entry, label-label, and label-dimension pairs).

The Fig 2 shows the following samples of table interpretation rules: if a cell is located in the 1st row and not 1st column then it serves as a column label — (*a*); if the background of a cell is blue (#0000ff) color then it serves as an entry — (*b*); if a cell is directly located above another cell spanning it in columns completely then they are connected — (*c*); if a cell is located in the 1st

column and contains a text matching the regular expression "[2][0][0-1][0-4]" then it is a value of the dimension "YEAR" — (*d*).

The rule engine matches known facts about cells and dimensions (their value ranges) against those production rules. In the result of firing the rules we recover table semantic relationships (the logical level of our model). Additionally, after rule firing, we try to harmonize extracted labels and data, using dictionaries with reference values. Also, we try to detect dimension values among labels, using dictionaries with dimension values. Recovered semantic relationships are used to generate the table in the structured (canonical) form. It includes the following fields: DATA contains data (entries); RLABEL contains label paths from leaves to roots in the non-degenerate tree L_r; CLABEL contains label paths from leaves to roots in the non-degenerate tree L_c; the set of fields D1,..., DN contain values of the corresponding dimensions D_i.

DATA	OPERATION	YEAR	MAIL TYPE	REGION	COUNTRY
462.9	Sent	2010	Letters	EU	Spain
82.9	Sent	2010	Letters	EU	Cyprus
...	...	...	...	...	...
12.3	Sent	2010	Parcels	Middle East	Lebanon
469.4	Sent	2011	Letters	EU	Spain
89.7	Sent	2011	Letters	EU	Cyprus
341.1	Sent	2011	Letters	EU	Belgium
21.5	Sent	2011	Letters	Middle East	Lebanon
...	...	...	...	...	...
13.1	Sent	2011	Parcels	Middle East	Lebanon
556.3	Received	2010	Letters	EU	Spain
...	...	...	...	...	...
11.3	Received	2011	Parcels	Middle East	Lebanon

Fig.3: The table in the structured (canonical) form

Each tuple in the canonical form presents the relationships between the entry, the label path in the tree L_r, the label path in the tree L_c, and values of the recovered dimensions D_i. In the ideal case where each label is assigned to a dimension the label trees become degenerate and the canonical form does not include RLABEL and CLABEL fields. Generated canonical forms can be exported into a relational database using standard tools of database management systems. The example of canonical form is shown in Fig.3.

Experimental Evaluation

For experimental evaluation we formed the collection of test data that includes 97 tables in Excel spreadsheets collected from 7 different sources (public governmental statistical and financial reports). They contain 17716 cells (including 10243 entries and 2872 labels) and 1614 internal relationships only between labels (excluding relationships from roots of label trees). We developed 7 knowledge bases separately for every source. They contain from 10 to 16 rules. We evaluated only the recovering of entries, labels, and internal relationships only between labels without external relationships between labels and dimensions. All entries, labels, and internal relationships were recovered with absolute accuracy. Rule firing took about 4 seconds on the processor Intel Core 2 Quad, 2,66 GHz. The test collection and experimental results are available at address http://cells.icc.ru/test.

Related Work

Existing methods for table understanding can be divided into two groups: (1) domain-specific [7-9] and (2) domain-independent [10-15].

The domain-specific methods are based on using ontologies or knowledge bases describing a particular domain. These methods allow binding natural language content of a table with concepts

of the particular domain. For instance, the method from the TANGO project [7] is based on a library of frames containing knowledge about lexical content of tables. Each frame describes a data type using regular expressions, dictionaries, and open resources like the lexical database WordNet. Embley et al. [8] use ontologies developed specially for information extraction. In addition to objects, relationships and constraints the extraction ontology includes a set of data frames, which are associated with sets of objects. Those data frames allow binding table content with objects of the ontology using regular expressions. Wang et al. [9] consider the problem of understanding a web table as associating the table with semantic concepts presented in the PROBASE knowledge base.

The listed above methods [7-9] principally use an analysis of natural language content from tables. It is not always enough in practice. Information extraction from tables often requires an analysis of spatial and style information for high accuracy.

The domain-independent methods [10-15] are based on an analysis and interpretation of spatial, style and text information from tables instead of using external knowledge on a specific domain. For instance, Gatterbauer et al. [10] propose to use only an analysis of spatial and style information in CSS2 format. Their method is based on assumptions about style information designed for several common types of web-tables. Also Pivk et al. [11, 12] suggest a methodology and TARTAR system for automatic transforming HTML tables into logical structured form (semantic frames). The TARTAR system is based on heuristics on structure and text content of a table, which are designed for three typical table types. Kim et al. [13] use an analysis of spatial, style, and natural language information from web tables based on embedded rules and regular expressions for five table types. The recent papers [14, 15] discuss the method for transforming data from web tables to a relational database. The method provides grouping attributes into categories, using only an analysis of table structure. It is based on several embedded in algorithms assumptions on regular structure of pivot tables.

The mentioned domain-independent methods [10-15] are based on using a limited set of assumptions on table structures, styles, and content which originate from a few common types of tables. These assumptions are embedded in the proposed algorithms. They limit classes of tables, which can be analyzed and interpreted by those methods with high accuracy.

Conclusions

We propose the rule-based system to table understanding, using both domain independent (spatial and style) information and domain-specific (natural-language) information. The system provides analysis and interpretation for tables with complex structures. In particularly, it enable processing table features like cut-ins (headers in a table body), non-numerical data values, the duplication of multilingual labels, label columns, which are alternated by data columns.

We also use assumptions about table structures, styles and content. But, in contrast to the existing methods for table understanding, we divide assumptions into two parts: general and special. The first constant part is the general assumptions. They describe a wide class of tables. Our model is based on them. The second variable part is special assumptions about spatial, style and natural language features of tables. They are expressed as sets of table interpretation rules. These special assumptions are combined into sets (knowledge bases), which are designed for different subclasses of tables. That approach allows reaching the table understanding with high or even absolute accuracy for particular subclasses of tables within the class limited by the general assumptions.

Perhaps, the main application of our approach is the unstructured tabular data integration. The described principles of table understanding can be used in developing software for unstructured tabular data integration in business intelligence and information extraction from financial reports.

We discuss our approach in more detail in the paper [16].

Acknowledgements

The research work was financially supported by the Russian Foundation for Basic Research (grant no 15-37-20042 and 14-07-00166) and the Council for grants of the President of the Russian Federation (grant no SP-3387.2013.5).

References

[1] Ferrucci, D. & Lally, A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4), pp. 327-348, 2004.

[2] Feldman, R. & Sanger, J. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, Cambridge University Press, 2006.

[3] Inmon, W.H. & Nesavich, A. Tapping into *Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence*, *1st edition*. Prentice Hall PTR: NJ, USA, 2007.

[4] Doan, A., Naughton, J.F., Ramakrishnan, R., Baid, A., Chai, X., Chen, F., Chen, T., Chu, E. & Derose, P. Information extraction challenges in managing unstructured data. *SIGMOD Rec.*, 37(4), pp. 14-20, 2009.

[5] Embley, D.W., Hurst, M., Lopresti, D. & Nagy, G. Table-processing paradigms: a research survey. *Int. J. on Document Analysis and Recognition*, 8(2), pp. 66-86, 2006.

[6] Hurst, M. Layout and language: challenges for table understanding on the web. *Proc. of the 1st Int. Workshop on Web Document Analysis*, pp. 27-30, 2001.

[7] Tijerino, Y.A., Embley, D.W., Lonsdale, D.W., Ding, Y. & Nagy, G. Towards ontology generation from tables. *World Wide Web: Internet and Web Information Systems*, 8(3), pp. 261-285, 2005.

[8] Embley, D.W., Tao, C. & Liddle, S.W. Automating the extraction of data from HTML tables with unknown structure. *Data & Knowledge Engineering*, 54(1), pp. 3-28, 2005.

[9] Wang, J., Wang, H., Wang, Z. & Zhu, K.Q. Understanding tables on the web. *Proc. of the 31st Int. Conf. on Conceptual Modeling*, Springer-Verlag, pp. 141-155, 2012.

[10] Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B. & Pollak, B. Towards domain-independent information extraction from web tables. *Proc. of the 16th Int. Conf. on World Wide Web*, pp. 71-80, 2007.

[11] Pivk, A., Cimianob, P. & Sure, Y. From tables to frames. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3), pp. 132-146, 2005.

[12] Pivk, A., Cimiano, P., Sure, Y., Gams, M., Rajkovic, V. & Studer, R. Transforming arbitrary tables into logical form with TARTAR. *Data & Knowledge Engineering*, 60(3), pp. 567-595, 2007.

[13] Kim, Y.-S. & Lee, K.-H. Extracting logical structures from HTML tables. *Computer Standards & Interfaces*, 30(5), pp. 296-308, 2008.

[14] Embley, D.W., Nagy, G. & Seth, S. Transforming web tables to a relational database. *Proc. of the 22nd Int. Conf. on Pattern Recognition*, 2014.

[15] Nagy, G., Embley, D.W. & Seth, S. End-to-end conversion of HTML tables for populating a relational database. *Proc. of the 11th IAPR Int. Workshop on Document Analysis Systems*, IEEE Comp. Soc., pp. 222-226, 2014.

[16] Shigarov, A.O. Table understanding using a rule engine. *Expert Systems with Applications*, 42(2), pp. 929-937, 2015.