Layout & Language: Preliminary experiments in assigning logical structure to table cells

Matthew Hurst and **Shona Douglas**
Language Technology Group, Human Communication Research Centre,
University of Edinburgh, Edinburgh EH8 9LW UK
{Matthew.Hurst,S.Douglas}@edinburgh.ac.uk

Abstract

We describe a prototype system for assigning table cells to their proper place in the table's logical (relational) structure, based on a simple model of table structure combined with a number of measures of cohesion between cell contents. Preliminary results suggest that very simple string-based cohesion measures are not sufficient for the extraction of relational information, and that future work should pursue the aim of more knowledge/data-intensive approximations to a notional subtype/supertype definition of the relationships between value and label cells.

1 Introduction

Real technical documents are full of text in tabular and other complex layout formats. Most representations of tabular data are layout or geometry-based: in SGML, in particular, Marcy Thompson notes "table markup contains a great deal of information about what a table looks like... but very little about how the table relates the entries. ... [This] prevents me from doing automated context-based data retrieval or extraction."[1]

1.1 Views of tables

In (Douglas, Hurst, and Quinn, 1995) an analysis of table layout and linguistic characteristics was offered which emphasised the potential importance of linguistic information about the contents of cells to the task of assigning a layout-oriented table representation to the logical relational structure it embodies. Two views of tables were distinguished: a **denotational** and a **functional** view.

The denotation is the table viewed as a set of n-**tuples**, forming a **relation** between values drawn from n value-sets or **domains**. Domains typically consist of a set of values with a common supertype in some actual or notional Knowledge Representation scheme. The actual table may also include **label** cells which typically can be interpreted as a lexicalisation of the common supertype. We hypothesize that the contents of value cells and corresponding label cells for a given domain are significantly related in respect of some measures of **cohesion** that we can identify.

The **functional** view is a description of how the information presentation aspects of tables embody a **decision structure** (Wright, 1982) or reading path, which determines the order in which domains are accessed in building or looking up a tuple.

To express a given table denotation according to a given functional view, there is a repertoire of **layout patterns** that express how domains can be grouped and ordered for reading in two dimensions. These layout patterns constitute a syntax of table structure, defining the basic geometric configurations that domain values and labels can appear in.

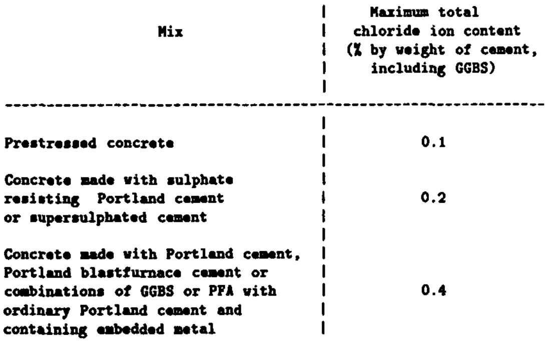
1.2 An information extraction task

Our application task is shallow information extraction in construction industry specification documents, containing many tables, which come to us via the miracles of OCR as formatted ASCII, e.g., in Figure 1.

The predominant argument type of this genre of specification documents can be thought of as a form of 'assignment', similar to that in programming languages. Our aim is to fit each assignment into a **frame** that contains various elements represented in terms of the sublanguage world model, a simple part-of/type-of knowledge representation.

The elements we are looking for are **entities**, **attributes** which the KR accepts as appropriate for

[1] (Thompson, 1996), p151.



Mix	Maximum total chloride ion content (% by weight of cement, including GGBS)
Prestressed concrete	0.1
Concrete made with sulphate resisting Portland cement or supersulphated cement	0.2
Concrete made with Portland cement, Portland blastfurnace cement or combinations of GGBS or PFA with ordinary Portland cement and containing embedded metal	0.4

Figure 1: Example from the application domain

those entities, a **unit** or type for each attribute, a **value** which the assignment gives to each attribute, and a **relationship** expressing the semantic content of the assignment. To extract these components, we would like to have a basic representation of the tuple structure of the table, plus information about any labels and how they relate to the values, in order to specify fully the relationship and its arguments.

1.3 Aims of the current work

Without some way of identifying domains we cannot extract the table relation we require. Our aim is to investigate the usefulness of a range of cohesion measures, from knowledge-independent to knowledge-intensive, in allowing us to select, from among those areas of table cells which are syntactically capable of being domains, those which in fact form the domains of the table. This paper reports the very beginning of the process.

2 The current prototype system

The system operates in two phases. In the first, a set of areas that might constitute domains is identified, using the constraints of table structure (geometric configuration) and cell cohesion. In the second, this candidate set is filtered to produce a consistent tiling over the table.

2.1 A simplified table structure model

The potential geometric configurations that we allow for a set of domain values (plus optional label) are called **templates**. A notation for specifying simple domain templates is defined as follows.

A template is delimited by a pair of brackets [...]. Within the brackets is a list of sub-templates, currently recursive only to depth 1 and taken to be stacked vertically in the physical table. If a template has no sub-templates, it consists of a triple (ww, dd, t). w and d are either integers or the wild card ?, and specify respectively the x-extent and y-extent of an area of cells that can match the template; the wild card matches any width, or depth, as appropriate. t specifies whether the (sub)template is to be counted as a value (tv) or a label area (tl).

A selection from a set of four possible **restrictions** on a template can be defined:

RESTRICTION	AREA MUST
-top	not contain top row
-left	not contain leftmost column
+right	contain rightmost column
+bottom	contain bottom row

The following templates are used currently:

lc: [[w1 d1 tl][w1 d? tv]] A label above a single column of values, of any height.

lr: [[w1 d1 tl][w? d1 tv]] A label above a single row of values, of any width.

v: [w? d? tv]{-top -left +right +bottom} A rectangular area consisting of only values, restricted to domains at the bottom right margin, typically accessed using both x and y keys.

c: [w1 d? tv] A single column of values.

2.2 A simplified cohesion model

The 'goodness' of a rectangular area of the table, viewed as a possible instantiation of a given template, is given by its score on the various cohesion attributes. Values assigned for each of the chosen attributes are combined in a weighted sum to yield two overall cohesion scores for each MatchedArea, the **value-value cohesion (v-v)** and the **label-value cohesion (l-v)** as follows.

We have a set of n **v-v** cohesion functions $\{f_0^{\text{v-v}}, f_1^{\text{v-v}} \ldots f_n^{\text{v-v}}\}$ which each take two cells and return a value between 0 and 1 which reflects how similar the two cells are on that function, and a set of n weights $\{w_0^{\text{v-v}}, w_1^{\text{v-v}} \ldots w_n^{\text{v-v}}\}$ which determine the relative importance of each function's result. Then for any area A composed of a set of cells we can calculate a measure of the area's cohesion as a set of domain values:

$$\text{VS} = \sum_{(c_i,c_j)\in A} \text{v-vScore}(c_i, c_j)$$

(where (c_i, c_j) is an ordered pair of cells)

$$\text{v-vScore} = \sum_{i=0}^{n} w_i^{\text{v-v}} f_i^{\text{v-v}} / \sum_{i=0}^{n} w_i^{\text{v-v}}$$

We have a set of m l-v cohesion functions $\{f_0^{\text{l-v}}, f_1^{\text{l-v}} \dots f_n^{\text{l-v}}\}$ which each take two cells and return a value between 0 and 1 which reflects how likely one of the cells is to be a label for the other, and a set of m weights $\{w_0^{\text{l-v}}, w_1^{\text{l-v}} \dots w_m^{\text{l-v}}\}$ which determine the relative importance of each function's result. Then for an area A composed of a set of cells and a label cell c_l we calculate a measure of the area's cohesion as a label plus set of domain values:

$$\text{LS} = \sum_{(c_l, c_v): c_v \in A} \text{l-vScore}$$

$$\text{l-vScore} = \sum_{i=0}^{m} w_i^{\text{l-v}} f_i^{\text{l-v}} / \sum_{i=0}^{m} w_i^{\text{l-v}}$$

A final score for the area is calculated as follows, depending on the type of template:

If the area's template contains values and a label:

$$\text{finalScore} = \frac{w_{\text{v-v}}\text{VS} + w_{\text{l-v}}\text{LS}}{w_{\text{v-v}} + w_{\text{l-v}}}$$

where $w_{\text{v-v}}$ and $w_{\text{l-v}}$ are weights reflecting the relative importance given to the VS and LS respectively.

If the area's template contains only values:

$$\text{finalScore} = \text{VS}$$

The cohesion attributes reported here have values between 0 and 1, where 0 corresponds to high and 1 to low similiarity:

ALPHA-NUMERIC RATIO: Given by

$$((\frac{|\alpha_a| - |N_a|}{|\alpha_a| + |N_a|} - \frac{|\alpha_b| - |N_b|}{|\alpha_b| + |N_b|})/2) + 0.5$$

where $|\alpha_a|$ is the number of alphabetic characters in string a and $|N_a|$ is the number of numeric characters in string a.

STRING LENGTH RATIO: A nondirectional comparison of string length.

2.3 Selecting a set of MatchedAreas

Given a set of templates, we find a set of MatchedAreas, rectangular areas of cells which satisfy a template definition and which reach a given cohesion threshold. The set of MatchedAreas contains no areas that are wholly contained in other matched areas for the same template.

From the set of MatchedAreas we select the areas we believe to be the domains of the table using a greedy algorithm which selects a set of cells that form a complete, non-overlapping tiling over the table.

3 Experiments

To test our system, we created a corpus of tables marked up in SGML with basic cell boundaries, allowing the template mechanism to determine the x and y position of cells. This representation is similar in relevant information content to many SGML table DTDs, and is also a plausible output from completely domain-independent techniques for table recognition in ASCII text or images, e.g., (Green and Krishnamoorthy, 1995). To this basic representation we added human-judgment information about the domains in each table (using an interface written in XEmacs lisp), specifying cell areas of values and labels for each domain.

The tables were taken from a corpus of formatted ASCII documents in the domain of construction industry specifications. 29 tables consisting of 91 domains were open to examination during the experimental development; 4 tables consisting of 13 domains were held back as a test set.

The tests we ran had different combinations of the cohesion measures **alphanum** and **string-length** with a factor **ignorelabel**, which corresponds to reducing the weighting w_{l-v} for the goodness of the label match to 0. The **unseen** condition is the last (best-performing) combination, run on the held back data.

4 Results and future work

The recall results are given in Table 1. The experiment column specifies the trial in terms of the factors defined above. The templates columns specify which templates are included in the trial. The recall score for each trial is the number of matched areas that perfectly agree with the boundary and type of a domain as marked by the human judge, as a percentage of the number of domains identified by the human judge. (Since the selection algorithm produces only a single tiling for each table, precision was not explicitly measured.)

4.1 Effect of templates

Increasing the number of templates available at one time reduces the recall performance because of confusion during the selection process; if we used only the **lc** template, for instance, we would get better performance overall per domain (in this application area). The true performance of the system has to be judged with respect to the complete set (the rightmost column in the results table), however, since all these templates are needed to match even quite simple tables.

Experiment	Templates available														
	$\{lc\}$	$\{lr\}$	$\{v\}$	$\{c\}$	$\{lc, lr\}$	$\{lc, v\}$	$\{lc, c\}$	$\{lr, v\}$	$\{lr, c\}$	$\{v, c\}$	$\{lc, lr, v\}$	$\{lc, lr, c\}$	$\{lc, v, c\}$	$\{lr, v, c\}$	$\{lc, lr, v, c\}$
alphanum	84	3	3	3	82	32	60	5	2	7	33	59	24	9	26
stringlength	41	1	0	0	42	30	35	1	1	0	31	36	26	1	27
alphanum, ignorelabel	84	3	3	3	84	34	84	5	2	7	36	84	34	9	36
stringlength, ignorelabel	41	1	0	0	42	34	41	1	1	0	35	42	34	1	35
alphanum, stringlength	75	2	3	3	75	45	68	4	1	7	45	67	42	8	42
alphanum, stringlength, ignorelabel	75	2	3	3	76	47	75	5	2	7	48	76	47	8	48
unseen	77	8	0	0	77	62	62	8	8	0	62	62	54	0	54

Table 1: Recall results for all experimental conditions: % of actual domains correctly identified

The simple templates used here are also not adequate for more complex tables with patterns of recapitulation and multiply layered spanning labels. We intend to take a more sophisticated view of possible geometric configurations, perhaps similar to the treatment in (Wang, 1996), and use the idea of reading paths to extract the tuples by relating the appropriate values from different domains.

4.2 Effect of cohesion measures

The alphanum and stringlength measures in combination do perform rather better than alone. However, ignoring **l-v** cohesion always improves recall; these cohesion measures do not help in distinguishing between labels and values, or in linking labels with value-sets.

This will be more of a problem when we deal with more complex tables with complex multi-cell labels. In future, we intend to investigate the effect of more sophisticated cohesion measures, including the use of thesaural information from domain-independent sources and corpus-based Knowlege Acquisition, e.g., (Mikheev and Finch, 1995), which should form better approximations to the supertype/subtype distinction.

Combining a number of measures, in the kind of framework we have presented here, should allow graceful performance over a wide range of domains using as much information as is available, from whatever source, as well as convenient evaluation of the relative contribution of different sources.

Acknowledgements

We acknowledge the support of BICC plc who supplied data and funded the first author during most of this work, and of the Engineering and Physical Sciences Research Council of the UK, who funded the second author under the *CISAU* project (IED4/1/5818).

References

Douglas, Shona, Matthew Hurst, and David Quinn. 1995. Using natural language processing for identifying and interpreting tables in plain text. In *Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 535–545.

Green, E. and M. Krishnamoorthy. 1995. Recognition of tables using tables grammars. In *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 261–278, University of Nevada, Las Vegas, April.

Mikheev, A. and S. Finch. 1995. A workbench for acquisition of ontological knowledge from natural text. In *Proceedings of the 7th conference of the European Chapter for Computational Linguistics*, pages 194–201, Dublin, Ireland.

Thompson, Marcy. 1996. A tables manifesto. In *Proceedings of SGML Europe*, pages 151 – 153, Munich, Germany.

Wang, Xinxin. 1996. *Tabular Abstraction, Editing, and Formatting*. Phd, University of Waterloo, Ontario, Canada.

Wright, Patricia. 1982. A user-oriented approach to the design of tables and flowcharts. In David H. Jonassen, editor, *The Technology of Text*. Educational Technology Publications, pages 317–340.