Layout and Language: A Corpus of Documents Containing Tables

Matthew Hurst
Human Communication Research Center, University of Edinburgh
matth@cogsci.ed.ac.uk

January 14, 2000

Abstract

This paper describes the collection of a corpus of documents that contain one or more tables. Some results are then presented which go some way to characterising the table in terms of its relationship to the content of the document it appears in.

1 Introduction

Though the field of information extraction (IE) has been one of the most successful to come from the NLP/CL community, it has thus far concentrated on documents with simple logical structure. It seems appropriate, however, to consider more complex documents — not solely due to the improvement in extraction technologies, but also due to the content and meta-information that complex structure can offer the IE task.

One often used but never exploited component of a more complex document model is the table. Its utility is compact information presentation, a definite boon for IE processes, however it can also offer more information for discourse and domain knowledge sub-processes.

A table processing system (TabPro) is being developed and part of that research requires the construction of a corpus of documents containing tables. This corpus is used for training classification processes and evaluating the performance of the system as a whole. A complete description of the model of tables mentioned in this paper can be found in [Hurst, 1999].

2 Quality and Corpus Creation

The construction of a corpus of documents requires the examination of a number of issues relating to the quality of the corpus. This examination has a strong relationship with the set of tasks for which the corpus will be a data resource. We must consider at least:

- *The classification of documents as being of the type required for the task.* Do we wish the document to be composed of only those documents or do we wish some examples from outside the class to be included for appropriate noise? and if so, how much? For example, in a corpus of document containing tables, do we wish to include some without tables, or even some with no content other than tables?
- *Consistency within the document.* With respect to certain document elements, do we require that all examples of those elements be within a certain class, or that some exceptions occur? For example, with tables, do we wish to include document elements marked as tables which we, or our processes, would actually classify as being lists or some other table-like element?
- *Reflection of the distribution of documents.* Do we wish to include certain documents due to their interesting qualities at the expense of producing a corpus which reflects the distribution of those qualities in the real world space of documents? For example, do we include a document due to its inclusion of a interesting table?
- *Variety of content domains.* Should a particular content domain be focused on? What are our expectations regarding the variety or consistency of phenomena across domains?
- *Variety of document genres.* Should the corpus be collected from one type of document, for example, journal papers?
- *Errorful source documents.* Should errors, such as spelling, grammar, consistency, be corrected in the source document?

- *Errors in particular document elements.* Should errors in particular document elements be corrected? Collecting documents which include tables from the web has demonstrated the the high error rate which humans can deal with. This is possibly a result of the procedural interpretation of html documents by browsing software.

- *Errors in transformation from source documents.* What threshold should be set for the transformation of source documents into the document description used by the corpus? How can this be judged?

- *Errors in corpus judgement.* If the corpus is to be used for training and testing purposes, what degree of accuracy should we expect for the judgement of the corpus? and how should this be tested?

All of the above issues have to be considered in conjunction with some idea of the type of processes which will use the corpus. In the case of the table processing system being currently designed for the information extraction task, robustness is an important issue. When considering the quality of the transformation of source documents for the corpus into documents marked up according to the document type definition of the corpus if the same automatic system is to be used in the system at run time, then the errors due to this process should be accepted. Or, put another way, the translation should be a constant factor and not one which changes between corpus collection and run-time processing.

Errors in the markup of the judgements of the document with respect to the system must be considered in light of the manner in which results are scored. It can not be expected that a system will achieve 100 % performance for a task if there are errors in the judgement of the corpus — and in fact, such a performance would be open to misinterpretation.

As for the content domains of the documents and the genre of the documents, the focus of the process must again be considered. As this is a corpus of documents containing tables, we are interested in the possibility of the variation of the relationship between the table content and the document content as well as the document structure. Consequently, by including a variety of documents both in terms of content and structure in the corpus we have the required data for both making assumptions about such relationships as well as training on particular subsets according to content and genre. It is generally said that you can't have too much data in a corpus, however you still have to delimit the dimensions over which you expect the interesting phenomena to range and generalise.

3 The Corpus

The corpus has been collected from html files (often the html version of papers found online, or resulting from other transformation processes such as `latex2html`), LaTeX files and a small number of `ascii` files. Two translators were written, one from html to the required format and the other from LaTeX to the required format. The content domains are described in Figure 1.

All errors found in the source document and which were translated in to the corpus format were preserved. For example, missing cells, incorrectly aligned table elements and so on. However, a number of errors resulting from the translation process were repaired. In particular, when html documents contained text in a tabular form — i.e. when the `<TABLE>` tag was used for its effect, not to indicate a table — such non-tables were removed, and the content preserved. Other errors occurred with the transformation of tags simply due to the universal poor quality of web pages (both human generated and automatically generated).

In computational linguistic terms, this is a very small corpus. However, in this case, the number of documents (49) is not necessarily a good indication of the size of the corpus in terms of the features which the processes focus on. The corpus contains 163 tables. The total number of cells is 10, 696. The number of tokens in the corpus is approximately 173, 000. A sample of experiments which extracted the features describe elsewhere in this paper for corpora of varying size indicated that the proportion of feature values per classification became stable around 20 documents.

4 Corpus Markup

There are two stages to marking up a corpus. The first simply deals with the data as a store of a certain class of documents. It uses a markup scheme which reflects the structure of the document. For the corpus, the following document elements were tagged: title, author, section headings, tables, table headers and footers, cells and table captions.

The second stage of markup provides a description of the document with respect to certain processes. In this case, the tables in the document were

marked up with the correct results for certain table processing tasks. In other words, a description of the model of the table was included in the corpus document.

A java tool has been implemented to provide a fast way to include the judgement of the table in the corpus document. The tool loads in a document, displays the selected table and allows the user, though a point a click interface, to mark certain features for individual cells (for functional description) or sets of cells (for structural information). The GUI is shown in Figure 3.

5 Corpus Observations

Using the corpus, in conjunction with the modules in the TabPro system, we can gain some insight not only in to the nature of tables as a discrete document element, but also to the relationship between certain contextual document features and the content of the table. The reason why this is interesting and important is that, as some of these observations indicate, if a model of the table is to be constructed which will be useful for the task of information extraction, then the content of the table, the content of the document and the relationship between them is absolutely necessary. In other words, it is impossible to construct a model of the table looking only at structural features.

5.1 Physical Features

The physical features currently extracted and used by the TabPro system relate to the physical context of the cells on each of their four faces. Their distribution with respect to the task of classifying cells as being either data or access cells is shown in Figure 4.[1]

This data can be used by a simple naive bayes classifier to provide a baseline for the functional classification task. Experiments indicate that such an algorithm would give approximately 91 % precision and recall. This should be compared with the 70 % baseline which a process assigning the modal classification (data) to all cells would achieve.

A more complex analysis of the corpus of table and the characterisation of cells in terms of the function and physical context can be achieved by analysing the corpus for extended patterns of cells centred on a target cell. In this experiment, physical patterns were extracted from each table in turn and then applied using a classification module which matched the patterns to the physical layout of the table and then assigned the cell class according to the majority prediction. The precision and recall results for patterns of depths 1 to 9 are shown in Figure 2. What this tells us is how consistent a table's patterns are with the function of the cells and, at the same time, how deep, or context sensitive, the patterns are.

The results show that recall plateaus around 98 % indicating that tables contain reasonable regular structures with about 2 % variability per table.

5.2 Content Based Features

There are two general classes of content based features. Those which describe the content of the table in terms of the content of the document as a whole, and those which describe the content of the table in terms of the content of specific document elements. Generally, those dealing with specific elements, as intuition might suggest, have slightly more predictive power than the more general features though are sensitive to relative scarcity of those elements. The content based features looked at were tokens, digits, noun groups and noun group heads. And the areas in which they are computed were document paragraphs, section headings, table captions, sentences mentioning tables in the document, and sentences mentioning the specific table.

For the general case with document paragraphs, we can see that the results are as we might expect. Content generally discusses the indexing structure of the table.

Type	Value	DATA	ACCESS
Token	0	92.605	62.750
Token	1	7.395	37.250
Noun Group Head	0	95.348	65.769
Noun Group Head	1	4.652	34.231
Noun Group	0	94.207	59.331
Noun Group	1	5.793	40.669

As for the special document elements, if we look at the noun group heads found in the table captions, sentences referring to tables and section headings it was found that the presence of those terms in the table predicted 19.16 % of the access cells with 89.59 % precision. Adding content information to the physical features and applying the naive bayes classification algorithm resulted in 91 % precision and recall.

[1]Data and access cells are fully described in [Hurst, 1999]. In brief, a data cell is one whose content is the information sought by the reader. An access cell is one used to index the data and which, in part, contributes to the interpretation of the data. In the table displayed in Figure 3, the darker cells are the data cells and the lighter cells are the access cells.

Domain	Number of Documents	Number of Tables
Administration	9	14
Biochemistry	2	2
Computational Linguistics	6	14
Construction Industry	1	13
Finance	1	3
Information Technology	8	41
Internet	1	1
News	2	2
General Science	18	72
Sport	1	1
total	49	163

Figure 1: Content Domains

In this case, a gain of 2 % recall for access cells is offset by a drop of less than 1 % recall for data cells .

6 Conclusions

This paper has given a very general outline of the motivation for and the process of collecting a corpus of documents containing tables. It has also suggested how inspection of the corpus through the application of some simple shallow linguistic analysis reflects to some degree our intuitions regarding the relationships between the table and the content of the document, vis, the document discusses the indexing structure of the table. This result is combined with the fact that it is impossible to predict the functional model of a table purely through observing the physical pattern of cells and the content must be taken in to account.

Recent developments in the TabPro system have resulted in precision and recall scores for the functional classification of cells of between 94 and 95 %. This analysis, which combines physical and content based features, beats the best physical analysis which currently stands at around 91 to 92 % recall for the same precision as above. Although the difference is only a couple of percentage points, the real result lies in the correct classification of access cells (which account for less than 30 % of cells though are of greater importance). Here, physical analysis gives 80 % recall at around 94 % precision, where as the analysis including content information gets 90 % recall at around 92 % precision.

The task of classifying table cells as data or access has all the characteristics of a typical AI problem. As results become asymptotic, the effort becomes exponential. In this case, the last few percent will be won through increasingly deep analyses of the content based relationships between access cells and the document in general and specific document elements.

References

[Hurst, 1999] Hurst, M. (1999). Layout and language: Beyond simple text for information interaction — modelling the table. In *2nd International Conference on Multimodal Interfaces*, Hong Kong.

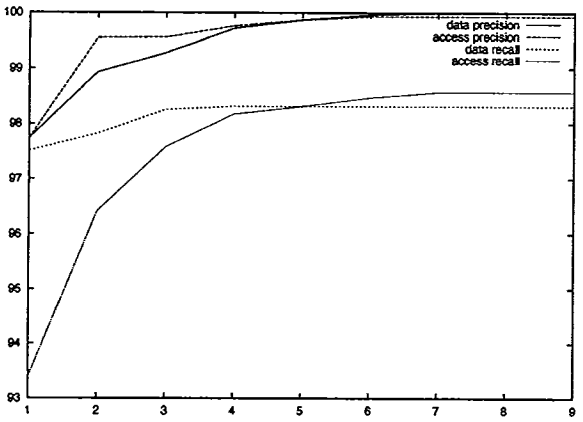

Figure 2: Corpus Predictability for Pattern Depth

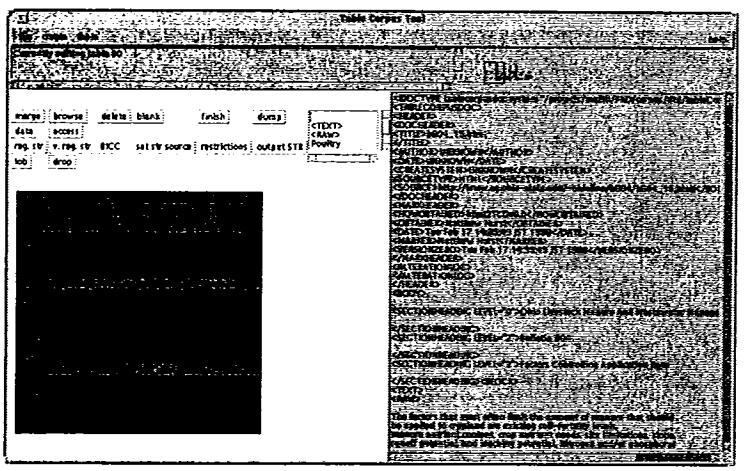

Figure 3: The Table Corpus Tool.

	MARGIN	ONE_TO_ONE	ONE_TO_MANY	MANY_TO_ONE	INTERNAL_SPACE
Top	6.2	86.2	0.5	6.8	0.3
data/access	6.0/94.0	79.0/21.0	43.3/56.7	58.6/41.4	50/50
Bottom	8.3	86.1	1.6	3.6	0.3
data/access	76.4/23.6	72.7/27.3	8.8/91.2	66.9/33.1	44.4/55.6
Left	17.0	80.8	0.0	1.9	0.0
data/access	3.4/96.6	87.6/12.4	50.0/50.0	10.6/89.4	0.0/0.0
Right	17.9	80.8	0.6	0.3	0.3
data/access	95.3/14.7	69.3/30.7	1.5/98.5	50.0/50.0	73.3/26.7

Figure 4: Ratio of physical characteristics.