A Grammatical Approach to Understanding Textual Tables using Two-Dimensional SCFGs

Dekai Wu[1] **Ken Wing Kuen Lee**
Human Language Technology Center
HKUST
Department of Computer Science and Engineering
University of Science and Technology
Clear Water Bay, Hong Kong
{dekai,cswkl}@cs.ust.hk

Abstract

We present an elegant and extensible model that is capable of providing semantic interpretations for an unusually wide range of textual tables in documents. Unlike the few existing table analysis models, which largely rely on relatively *ad hoc* heuristics, our linguistically-oriented approach is systematic and grammar based, which allows our model (1) to be concise and yet (2) recognize a wider range of data models than others, and (3) disambiguate to a significantly finer extent the underlying semantic interpretation of the table in terms of data models drawn from relation database theory. To accomplish this, the model introduces Viterbi parsing under two-dimensional stochastic CFGs. The cleaner grammatical approach facilitates not only greater coverage, but also grammar extension and maintenance, as well as a more direct and declarative link to semantic interpretation, for which we also introduce a new, cleaner data model. In disambiguation experiments on recognizing relevant data models of unseen web tables from different domains, a blind evaluation of the model showed 60% precision and 80% recall.

1 Introduction

Natural language processing has historically tended to emphasize understanding of linear strings—sentences, paragraphs, discourse structure. The vast body of work that focuses on text understanding is often seen as an approximation of spoken language understanding. Yet real-life text is actually heavily dependent on visual layout and formatting, which compensate for cues normally found in spoken language but are absent in text. As Scott (2003) reiterated in the opening ACL'03 invited talk: "The overlay of graphics on text is in many ways equivalent to the overlay of prosody on speech... Just as prosody undoubtedly contributes to the meaning of utterances, so too does a text's graphical presentation contribute to its meaning. However... few natural language understanding systems use graphical presentational features to aid interpretation..." (Power *et al.*, 2003).

Nowhere is this more evident than in the widespread use of tables in real-world, unsimplified text documents. Tables have a comparable or greater complexity as other elements of text. Unfortunately, in mainstream NLP it is not uncommon for tables to be regarded as a somehow "degenerate" form of text, unworthy of the same degree of attention as the rest of the text. But as we will discuss, the degree of ambiguity in table understanding is at least as great as for many sense and attachment problems. Many of the same mechanisms used for understanding linear text are also required for table understanding. The same division of surface syntax and underlying semantics is found.

Indeed, to perceive the limitations of existing table understanding models, we may distinguish several very different levels of table analysis tasks. In **table classification**, the table is classified into one of several coarse categories (in the extreme case, some models simply predict whether the purpose of the table is for page layout versus tabular data). In **table synactic recognition**, the surface types of individual cells or block regions are labeled (e.g., as heading or data) but the underlying semantic relationships between the table elements remain unrecognized and usually highly ambiguous (i.e., no logical relations between the elements

[1] The authors would like to thank the Hong Kong Research Grants Council (RGC) for supporting this research in part through grants RGC6083/99E, RGC6256/00E, and DAG03/04.EG09.

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 905–912,
Sydney, July 2006. ©2006 Association for Computational Linguistics

in the table are assigned). In contrast, in **table semantic interpretation**, the exact logical relations between the elements in the table must be recognized (e.g., by associating the table and/or subregions thereof with precise table schemas in relational database style).

Existing table understanding work largely lies at the level of superficial table classification or syntactic recognition. Rarely, if ever, are precise logical relations assigned between the elements in the table. *Ad hoc* heuristic approaches tend to rule, rather than linguistic approaches.

On the other hand, in the linguistic approach advocated by Scott (2003) and (Power *et al.*, 2003), tables were not considered. The various physical presentation elements discussed included headings, captions, and bulleted lists—all of which exhibit numerous similarities to tabular elements. Possibly, tables were not considered because they are difficult to describe adequately within the expressiveness of common linguistic formalisms like CFGs.

The work presented here aims to address this problem. Our model provides an enabling foundation toward a linguistic approach by first shifting to a two-dimensional CFG framework. This permits us to construct a grammar where all the rules are meaningfully discriminative, such that—unlike existing table understanding models—any analysis of a table includes a full parse tree that assigns precise data model labels to all its regions (including nested subregions) thereby specifying the logical relations between the table's elements. Additionally, probabilities on the production rules support thresholding (or ranking) of the alternative candidate table interpretation hypotheses.

As with many natural language phenomena, a full model of disambiguation must ultimately integrate lexical semantics. However, in this research step we focus on the question of how much semantic interpretation can be performed on the basis of other features, in the absence of a lexical or ontological model. Just as syntax and morphology and prosody alone already permit much recognition and disambiguation of semantic roles and argument structure to be done for sentence, the same can be done for tables. At the same time, we believe future integration of lexical semantics will be facilitated by the grammatical framework of our model.

Table 1: Example "Martian" table (see text).

Pbje	Kwe	Zxc	Amc
Hoer	15 - 18	17 - 20	19 - 23
NQ	85 - 95%	70 - 90%	75 - 95%
Ncowifl	Djhi	Djhi	Rubzlx

One way to think about this is that we wish to model what you might be able to recognize from a "Martian" table such as that in Table 1. The non-Martian reader relies solely on knowledge of alphabets and numbers, can spot font and formatting clues, and is familiar with the conventions (i.e., grammars) of tables in general.

You might reasonably interpret this table as a collection of vertical records with an attributes header column *(Pbje, Hoer, NQ, Ncowifl)* on the left. You might additionally interpret it as a table that contains an record key header row *(Kwe, Zxc, Amc)* along with the attributes header column *(Pbje, Hoer, NQ, Ncowifl)*. You might assign the latter interpretation a slightly higher probability, noticing the slightly longer form of *Pbje* compared to *Kwe*, *Zxc*, and *Amc*. On the other hand, even without reading English, you could reject the interpretation as a collection of horizontal records under the header attributes row *(Pbje, Kwe, Zxc, Amc)*, since each row contains different forms and types, in a pattern that is consistent across columns. Other interpretations are also possible, but unlikely given the regularity of the patterns.

Thus by analyzing the structure of a table, the reader would form a hypothesis about its data model, providing a semantic interpretation that allows the reader to extract information from the table. As can be seen from the restored original English version of the same example in Table 2, the most likely interpretation was predicted even without access to specific lexical knowledge. We aim to show that a fairly useful baseline level of semantic interpretation accuracy can already be achieved, even with relatively little lexical and ontological knowledge.

We model these alternative hypotheses for the interpretation of ambiguous tables as competing parses. Just as with ordinary parsing and semantic interpretation, the reader often builds multiple competing interpretations of the same table.

Note that many previous models do not even distinguish between the alternative possible interpretations in the Martian example. Existing models such as Hurst (2000) and Yang (2002) interpret tables with the same structural layout simply by assigning them same data model, which stops short of recognizing that it is necessary to rank multiple competing interpretations that entail different sets of logical relations.

In contrast, our proposed model is capable of producing multiple competing parses indicating different semantic interpretations of tables having the same structural layout, by selecting specific data models for the table and its subregions.

2 Data Models for Specifying Semantic Interpretations

To begin, some formal basis is needed to facilitate precise specification of the alternative semantic interpretations of a table, such that the exact logical relations between its elements are unambiguously specified. This will enable us to then design a table understanding model that attempts to map any given table (and recursively, its subregions) to alternative data models depending on which is most appropriate.

The set of data models we define below is a more comprehensive and precise inventory than found in the previous table analysis models discussed in this paper. It describes all the common conventional patterns of logical relations we have found in the course of empirically analyzing tables from corpora. One advantage of this inventory of data models arises from our appropriation of relational database theory wherever possible to help describe the form of the data models (Silberschatz *et al.*, 2002), allowing broad coverage of different table types without sacrificing precision as to the logical relations between entities.

Each data model assigns a clear semantics in terms of logical relations between the table elements, thereby allowing extraction of relational facts. In contrast, previous work on table analysis tends to either classify a table using only one single limited data model (e.g., Hurst (2000)), or using data models which essentially are merely surface layout types whose semantics are vague and ambiguous (e.g., Yang (2002), Yang and Luk (2002), Wang *et al.* (2000), Yoshida *et al.* (2001)).

A table is a logical view of a collection of interrelated items usually presented as a row-column structure such that the reader's ability to access and compare information can be enhanced, as also noted by Wang (1996). From a database management system perspective, each table can be considered as a (tiny) database. Like a program, the reader accesses the data. As a result, we consider that every table must correspond to a data model, and this model determines how the reader extracts information from the table.

Table 2: Example from Table 1 in its original version, with the English words restored.

Date	Thu	Fri	Sat
Temp	15 - 18	17 - 20	19 - 23
RH	85 - 95%	70 - 90%	75 - 95%
Weather	Cool	Cool	Cloudy

Each data model has a schema which, as we shall see below, may or may not surface (partially or completely) as a subset of cells in the table that describe attributes. Recognizing the data models of a table correctly therefore also implies that both attribute-value pairings and table structures have been recognized.

At the top level, we categorize the data models into three broad types:

- Flat model: A table is interpreted as a database table in non-1NF normal relational model.
- Nested model: A table is interpreted as a database table in an object-relational model, which allow complex types such as nested relations and concept hierarchy.
- Dimensional data model: A table (usually cross-tabular) is interpreted as a data cube (multidimensional table) in a multidimensional data model.

We now consider each of these types of data models in turn.

2.1 Flat model

A flat model is used for the semantic interpretation of any table as a relational database table in non-1NF. For example, tables such as Tables 2 and 3 are often interpreted by humans in terms of flat models. It is obvious that Table 3 can be viewed as a relational database table with a schema (Pos, Teams, Pld, Pts) and three records, because the table's surface form resembles how records are stored in a relational database tables. Similarly, Table 2 resembles a relational database table, but transposed to a vertical orientation, with the first

Table 3: Example of a ranking table, which is typically laid out in a flat relational model.

Pos	Teams	Pld	Pts
1.	Chelsea	38	95
2.	Arsenal	38	83
3.	Man United	38	77

column as the schema (Date, Temp, RH, Weather) and other columns as data records.

The flat model is closest to the 1-dimensional table approach used by the majority of previous models, but our approach designates the flat model as a semantic representation, in contrast to the previous models which see 1-dimensional tables merely as a syntactic surface form (e.g., Yang (2002), Yang and Luk (2002), Wang *et al.* (2000), Yoshida *et al.* (2001)). While such previous models only recognize tables that are physically laid out in this form, our approach clearly delineates an explicit separation of syntax and semantics, which provides greater flexibility allowing any table to be interpreted as a flat model, regardless of its surface form (though the flat model interpretation is more common for some surface forms than others).

As an example showing that any kind of table can be categorized as flat model, consider Table 6. Even such a table can be semantically interpreted as a flat model because related attributes can join together to form a composite attribute, though humans would less naturally choose this semantic interpretation. Certainly there are hierarchical relationship between attributes; for example, Ass1 is a subtype of Assignments. However, it is also valid to consider the attributes along a hierarchical path as one composite attribute. For example, "Mark -> Assignments -> Ass1" becomes the single attribute "Mark-Assignments-Ass1". Then the complete flat model schema is (Year, Team, Mark-Assignments-Ass1, Mark-Assignments-Ass2, Mark-Assignments-Ass3, Mark-Examinations-Midterm, Mark-Examinations-Final), and the first record is (1991, Winter, 85, 80, 75, 60, 75, 75).

2.2 Nested model

With the exception of Hurst (2000), previous work has not generally considered nested models in explicit fashion. Hurst (2000)'s model is based on Wang (1996)'s abstract table model, in which attributes may be related in a hierarchical way. On the other hand, Wang *et al.* (2000) oversimplistically considers nested models as 1-dimensional, thus missing the correct relationships between attributes and values.

A nested model can be seen as a generalization of the flat model, in which attributes may be related through composition or inheritance. Table 6 is naturally interpreted as a nested data model because the attributes have an inheritance relationship. The corresponding schema is (Year, Team, Mark (Assignments (Ass1, Ass2, Ass3), Examinations (Midterm, Final, Grade)).

A nested model is not appropriate for tables without hierarchical structure, such as Table 2 and Table 3.

2.3 Dimensional model

Our approach also nicely handles dimensional models, which are generally handled quite weakly in previous models. A dimensional model refers to a table, such as the table in Table 4, that resembles a view of collection of data stored in multidimensional data model. A multidimensional data cube, as described in the database literature (e.g., Han and Kamber (2000), Chaudhuri and Dayal (1997)), consists of a set of numeric measures (though in fact the data need not be numeric), each of which is determined by a set of dimensions. Each dimension is described by a set of attributes. For example, Table 5 can be semantically interpreted using the multidimensional data model depicted in Figure 1. Likewise, the cross-tabular table in Table 4 can also be semantically interpreted using the same multidimensional data model in Figure 1. The value of the first three columns in Table 5 are the dimension attributes and the revenue values are the measures.

In contrast, among previous models, Yang (2002) produces a semantically incorrect recognition of a multidimensional table that inappropriately presents the attributes in hierarchical structure. Yang and Luk (2002) and Wang *et al.* (2000) only recognize the simplest 2-dimensional case and apparently cannot handle 3 or more dimensions. Yoshida *et al.* (2001) only handle 1-dimensional cases.

A dimensional model is an inappropriate interpretation for non-cross-tabular tables, such as Table 2 and Table 3. A dimensional model is also not valid for tables such as Table 6. Semantically, it is not possible for "Assignments" and "Midterm"

Table 4: Example table showing revenue according to Location = {Vancouver, Victoria}, Type = {Phone, Computer} and Time = {2001, 2002}, using a tabular view of a 3-dimensional data cube.

	Vancouver		Victoria	
	Phone	Computer	Phone	Computer
2001	845	1078	818	968
2002	943	1130	894	1024

Table 5: Example relational database table containing the same logical information as Table 4.

Location	*Type*	*Time*	*Revenue*
Vancouver	Phone	2001	845
Vancouver	Phone	2002	943
Vancouver	Computer	2001	1078
Vancouver	Computer	2002	1130
Victoria	Phone	2001	818
Victoria	Phone	2002	894
Victoria	Computer	2001	968
Victoria	Computer	2002	1024

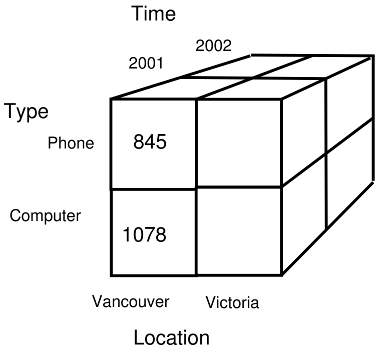

Figure 1: Multidimensional data cube corresponding to Tables 4 and 5.

Table 6: Example table of grades.

Year	Team	Mark					
		Assignments			Examinations		
		Ass1	Ass2	Ass3	Midterm	Final	Grade
1991	Winter	85	80	75	60	75	75
	Spring	80	65	75	60	70	70
	Fall	80	85	75	55	80	75
1992	Winter	85	80	70	70	75	75
	Spring	80	80	70	70	75	75
	Fall	75	70	65	60	80	70

to belong to different dimensions because it is incorrect to determine the score by both “Assignments” and “Midterm”. Syntactically, the texts in the last attribute row of Table 6 are all unique; however, the last attribute row of the table in Table 4 is a repeating sequence of (”Phone”, ”Computer”). Therefore, to a non-English reader, an English cross-tabular table which possess repeating sequences in the attribute rows is likely to be semantically interpreted as a dimensional model, while a cross-tabular table which does not have this property is likely to be interpreted as a nested model.

3 A 2D SCFG Model for Table Analysis

In this section, we will present our two-dimensional SCFG parsing model for table analysis which has several advantages over the ad hoc approaches. First, the probabilistic grammar approach permits a cleaner encapsulation and generalization of the kind of knowledge that previous models attempted to capture within their ad hoc heuristics. Most previous works (e.g. Yang (2002), Yang and Luk (2002), Hurst (2000), Hurst (2002)) gradually built up their ad hoc heuristics manually by inspecting some set of training samples. This approach may work if tables are from limited domains of similar nature. However, like text documents, the syntactic layout of textual tables may be determined by its context as well as its language. For instance, it is natural for an Arabic reader to read an Arabic table taking the rightmost column as the attribute column, instead of the leftmost column. Yoshida *et al.* (2001) use machine-learning techniques to analyze nine types of table structures, all 1-dimensional. Our grammar-based approach allows the model to be readily adapted to different situations by applying different sets of grammar rules.

Another advantage is that grammatical approach can make more accurate decisions while being simpler to implement, because it requires only a single integrated parsing process to complete the entire table analysis. This includes classifying the functions of each cell (as attribute or value), pairing attributes and values, and identifying the structure and the data model of a table. In contrast, previous works require several stages to complete the entire analysis, introducing complex problems that are difficult to resolve, such as premature commitment to incorrect early-stage decisions.

To our knowledge Wang *et al.* (2000) is the only textual table analysis model that uses a grammar to describe table structures. However, in that case, only a simple template matching analyzer is used. Their grammar notation is unable to show both physical structure and the semantics of a table at the same time in a hierarchical manner. In contrast, information such as "a data block contains three rows of data cell" can be stored in the parse tree constructed by our parsing model.

Outside of the table understanding literature, there exists a different 2D parsing technique called PLEX (Feder, 1971), (Costagliola *et al.*, 1994) which allows an object to have finite sets of attaching points. PLEX is used to generate 2D diagrams such as molecular structures, circuit diagrams and flow charts in a grammatical way. However, we consider it too complex and computationally expensive for our application because it does not exploit that fact that a textual table cell only has at most four attaching points in fixed directions.

Our parser is a two-dimensional extension of the conventional probabilistic chart parsing algorithm (Lari and Young, 1990), (Goodman, 1998). Intuitively, consider a sentence as a vector of tokens that will be parsed horizontally; then a table is a matrix of tokens (like a crossword puzzle) that will be parsed both horizontally and vertically. Because of this, our parser must run in both directions. We achieve this by employing a grammar notation that specifies the direction of parsing.

The two-dimensional grammar notation includes of a set of nonterminals, terminals, and two generation operators "–>" and "|->". Let X be a nonterminal and Y, Z, be two symbols which may be either nonterminals or terminals. Then:

- X –> Y Z denotes a horizontal production rule saying that the nonterminal X horizontally generates two symbols Y and Z.
- X |-> Y Z denotes a vertical production rule saying that the nonterminal X vertically generates two symbols Y and Z.
- X –> Y or X |-> Y equivalently denote a unary production rule saying that the nonterminal X generates a symbol Y.

We assume that all rules are binary without loss of generality, since any grammar can be mechanically binarized without materially changing the parse tree structure, just as in the case of ordinary 1D grammars.

The operators "–>" and "|->" control the generation direction. In term of table analysis, a nonterminal represents a matrix of tokens and a terminal represents a single token. Sub-matrices generated by a horizontal rule will have same height but not necessarily same width; similarly, submatrices generated by a vertical rule will have same width but not necessarily same height. In other words, a matrix is partitioned into two halves by the binary production rule.

Probabilities are placed on each rule, as in ordinary 1D SCFGs. They are used to eliminate parses falling below a threshold, which also helps to reduce the time complexity in practice.

Parsing with two-dimensional grammars can be conceptualized most easily via parse tree examples. Figure 2 shows a complete parse tree for parsing the table in Table 7 into a flat model. Figure 3 is a portion of a parse tree for parsing the table in Table 8 into a nested model, while Figure 4 is a portion of parse trees for parsing Table 7 into a dimensional model. The following is the grammar fragment that gives the parse tree as Figure 2:

```
T1-1H |-> FlatModel
FlatModel |-> FlatSchema Records
FlatSchema --> CompositeAttribute FlatSchema
FlatSchema --> CompositeAttribute
Records |-> Record Records
Records |-> Record
Record --> Data Record
Record --> Record
```

Note that the internal nodes of the parse trees serve to label subregions with data models, thus assigning a semantic interpretation specifying the exact logical relations between table elements. None of the previous models construct declarative parse trees like these, which are necessary for many types of subsequent analysis, including information extraction applications.

4 Experimental Method

To the best of our knowledge, unfortunately none of the table corpora mentioned in previous work are available to the public. Thus, it was necessary to construct a corpus for our experiments. We collected a large sample of tables by issuing Google searches with a list of random keywords, for example, *census age*, *confusion matrix*, *data table*, *movie ranking*, *MSFT*, *school ranking*, *telephone plan*, *tsunami numbers*, *weather report*, and

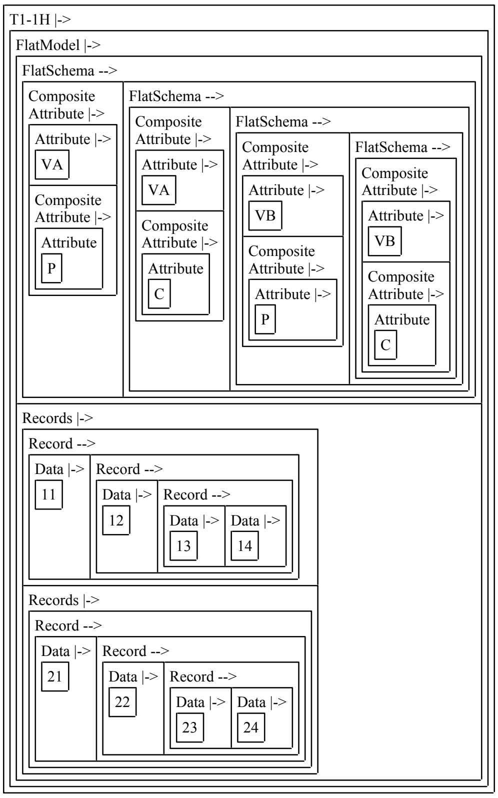

Figure 2: A parse tree for a flat model.

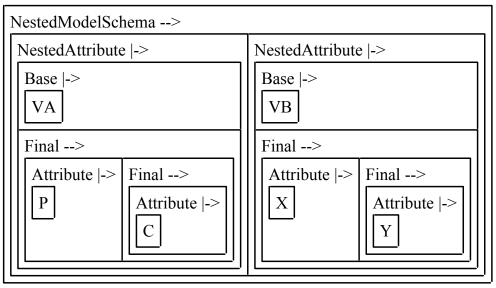

Figure 3: A partial parse for a nested model.

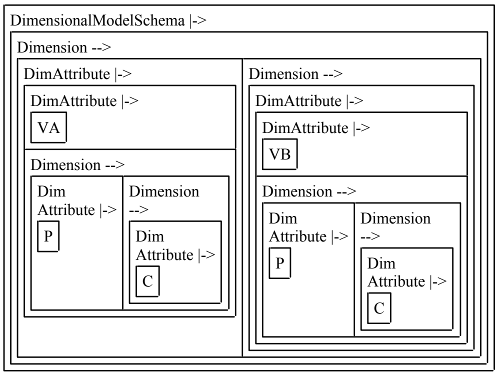

Figure 4: A partial parse for a dimensional model.

so on. Tables were extracted from the collected sample, automatically cleaned, and tokenized into two-dimensional array of tokens.

Table 7: Example table for Figures 2 and 4.

VA		VB	
P	C	P	C
11	12	13	14
21	22	23	24

Table 8: Example table for Figure 3.

VA		VB	
P	C	X	Y
11	12	13	14
21	22	23	24

Table 9: Example table showing a floor legend.

6	School of Business & Management
5	Department of Biochemistry
4	Classrooms 4202 - 4205
3	Department of Computer Science
3	Department of Mathematics

For the blind evaluation, a human annotator independently manually annotated a randomly chosen sample of 45 tables from the collection. All tables in the evaluation sample were previously unseen test cases, never inspected prior to the construction of the two-dimensional grammar.

Each tokenized table was tagged by the human judge with a list of types T relevant to the table. The relevance is defined as follows: a data model is relevant to a table if and only if the human would agree that such a data model would naturally be hypothesized as an interpretation for that table (analogously to the way that word senses are manually annotated for WSD evaluations). Each type is a tuple of the form (R, O, S), where R is the relevant data model, O is the reading orientation of R, and S is a boolean saying if a schema (i.e. attributes) exist in the table. Thus, Table 2 would be tagged as {(flat, vertical, true)} while the table in Table 4 would be tagged as {(flat, horizontal, true), (flat, vertical, true), (dimensional, _, true)}. But Table 9 may be tagged as {(flat, horizontal, false)}. The exceptions are that both the nested model and the dimensional model always have a schema, while the dimensional model does not have orientation. In cases where multiple legitimate readings were possible, the table was tagged

Table 10: Experimental results.

Precision	Recall
0.60	0.80

with multiple types. A total of 92 relevant types were generated from the tokenized tables.

We processed the tokenized tables with the two-dimensional SCFG parser, and computed the precision and the recall rates against the judge's lists of tags for all the test cases.

5 Results and Discussion

The experimental results are summarized in Table 10. All tables could be parsed; in general, it is very rare for any table to be rejected by the parser, since the grammar permits so many different configurations that can be recursively composed.

Unfortunately it is impossible to compare results directly against previous models, since neither those models nor the data they evaluated on are available.

Moreover, it is difficult to compare with previous models as our evaluation criteria are more stringent than in earlier work. Most previous work evaluated the performance in terms of the (vaguer and less demanding) criteria of number of correct attribute-value pairings. Such an evaluation approach gives unduly high weight to large repetitive tables, and neglects structural errors in the analysis of the table. In contrast, our approach gives equal weight to all tables regardless of how many entries they contain, requires semantically valid structural analyses, and yet still accepts any parse that yields the correct attribute-value pairings (since the tagging of the test set includes all legitimate types when there are multiple valid alternatives).

The fact that precision was lower than recall is due to the fact that many tables were wrongly interpreted as tables without schema or in wrong orientations. The current grammar has difficulty distinguishing attributes from values. Significant improvement can be obtained by using constraints to limit the number of incorrect parses, a strategy we are currently implementing.

6 Conclusion

We have introduced a framework to support a more linguistically-oriented approach to finer interpretation of tables, using two-dimensional stochastic CFGs with Viterbi parsing to find appropriate semantic interpretations of textual tables in terms of different data models. This approach yields a concise model that at the same time facilitates broader coverage than existing models, and is more easily scalable and maintainable. We also introduce a cleaner and richer data model to represent semantic interpretations, and illustrate how it systematically captures a wider range of table types. Without such a data model, the right attribute-value relations caanot be extracted from a table, even if surface elements like "header" and "data" are correctly labeled as previous models attempted to do. Our experiments show that even without other ontological and linguistic knowledge, excellent semantic interpretation accuracy can be obtained by parsing with a two-dimensional grammar based on these data models, by using a wide variety of surface features in the terminal symbols. We plan next to extend the model by incorporating ontological and linguistic knowledge for additional disambiguation leverage.

References

Surajit Chaudhuri and Umesh Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1), March 1997.

Gennaro Costagliola, Andrea De Lucia, and Sergio Orefice. Towards efficient parsing of diagrammatic languages. In *AVI '94: Proceedings of the workshop on Advanced visual interfaces*, pages 162–171, New York, NY, USA, 1994. ACM Press.

Jerome Feder. Plex languages. *Information Sciences*, 3:225–241, 1971.

Joshua T. Goodman. *Parsing Inside-Out*. PhD thesis, Harvard University, 1998.

Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 1 edition, 2000.

Matthew Francis Hurst. *The Interpretation of Tables in Texts*. PhD thesis, The University of Edinburgh, 2000.

Matthew Hurst. Classifying table elements in html. In *The 11th International World Wide Web Conference, Hawaii, USA*, 2002.

K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–36, 1990.

Richard Power, Donia Scott, and Nadjet Bouayad-Agha. Document structure. *Computational Linguistics*, 29(4):211–260, Dec 2003.

Donia Scott. Layout in NLP: The case for document structure (invited talk). In *41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, Aug 2003.

Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. McGraw-Hill, 4th edition, 2002.

H. L. Wang, S. H. Wu, I. C. Wang, C. L. Sung, W. L. Hsu, and W. K. Shih. Semantic search on internet tabular information extraction for answering queries. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, pages 243–249, New York, NY, USA, 2000. ACM Press.

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

Yingchen Yang and Wo-Shun Luk. A framework for web table mining. In *WIDM '02: Proceedings of the 4th international workshop on Web information and data management*, pages 36–42, New York, NY, USA, 2002. ACM Press.

Yingchen Yang. Web table mining and database discovery. Master's thesis, Simon Fraser University, August 2002.

M. Yoshida, K. Torisawa, and J. Tsujii. A method to integrate tables of the world wide web, 2001.