Disentangling the Structure of Tables in Scientific Literature

DOI:
10.1007/978-3-319-41754-7_14

Document Version
Accepted author manuscript

Disentangling the structure of tables in scientific literature

Nikola Milosevic[1], Cassie Gregson[2], Robert Hernandez[2], and Goran Nenadic[1,3]

[1]School of Computer Science, University of Manchester,
Kilburn Building, Oxford road, M13 9WJ, Manchester, UK
[2]AstraZeneca Ltd, Riverside, Granta Park, Cambridge, CB21 6GH, UK
[3]Health eResearch Centre, Manchester
{nikola.milosevic,g.nenadic}@manchester.ac.uk

Abstract. Within the scientific literature, tables are commonly used to present factual and statistical information in a compact way, which is easy to digest by readers. The ability to "understand" the structure of tables is key for information extraction in many domains. However, the complexity and variety of presentation layouts and value formats makes it difficult to automatically extract roles and relationships of table cells. In this paper, we present a model that structures tables in a machine readable way and a methodology to automatically disentangle and transform tables into the modelled data structure. The method was tested in the domain of clinical trials: it achieved an F-score of 94.26% for cell function identification and 94.84% for identification of inter-cell relationships.

Keywords: table mining, text mining, data management, data modelling, natural language processing

1 Introduction

Tables are used in a variety of printed and electronic documents for presenting large amounts of factual and/or statistical data in a structured way [1, 21, 25]. They are a frequent option in written language for presenting large, multidimensional information. For example, in experimental sciences, tables are usually used to present settings and results of experiments, as well as supporting information about previous experiments, background or definitions of terms. Tables are, in particular, widely used in the biomedical domain. However, while there have been numerous attempts to automatically extract information from the main body of literature [20, 11, 27], there have been relatively few attempts to extract information from tables.

One of the main challenges in table mining is that the existing models used to represent tables in the literature are focused on visualisation, rather than on content representation and mining. For example, tables in PubMedCentral (PMC)[1] are presented in XML with tags describing rows, cells, header and body

[1] http://www.ncbi.nlm.nih.gov/pmc/

of the table. However, these tags are used only for formatting and there is no guarantee that cells labelled as headers are also semantically headers of the table. Therefore, the table layout structure and relationships between cells make preprocessing and decomposition necessary before machine understanding tasks can be performed.

Hurst [10] introduced five components of table processing: *graphical* (a basic graphical representation of the table, e.g. bitmap, rendered table on screen or paper), *physical* (a description of the table in terms of physical relationships between its basic elements when rendered on a page), *functional* (the purpose of areas of the table with respect to the use of the table by the reader), *structural* (the organisation of cells as an indication of the relationships between them), and *semantic* (the meaning of the text in the cell). Following Hurst, we differentiate five steps of table processing:

1. **Table detection** locates the table in document.
2. **Functional analysis** detects and marks functional areas of tables such as navigational (e.g. headers, stubs, super-row) and data cells.
3. **Structural analysis** determines the relationships between the cells. For each cell in the table, it finds related header(s), stub(s) and super-row cells.
4. **Syntactic analysis** looks at the value of the cells at the syntactic level, for example, by identifying whether the value is a numeric expression.
5. **Semantic analysis** determines the meaning of data and attempts to extract and represent specific information from tables.

In this paper, we focus on functional and structural analysis of tables in clinical literature available openly in PMC. The aim is to facilitate further syntactic and semantic processing of tables by providing a model to capture necessary information about cells' functions, relationships and content.

2 Background

There are three main areas of table processing in literature: (1) table detection, (2) functional analysis and (3) table mining applications, which include information retrieval, information extraction, question answering and knowledge discovery.

Table detection is a hard problem for graphical document formats because it may involve visual processing of the document in order to find visual structures recognisable as tables [2]. Other formats can be also challenging. For example, table tags exist in HTML, but they are often used for formatting web page layout. Previous work focused on detecting tables from PDF, HTML and ASCII documents using Optical Character Recognition [13], machine learning algorithms such as C4.5 decision trees [17] or SVM [22, 19], and heuristics [26].

Functional analysis examines the purpose of areas of the table. The aim here is to differentiate between cells containing data and cells containing navigational information, such as headers, stubs or super-rows (see Figure 1). To solve this problem, several machine learning methods like C4.5 decision trees [12, 5],

conditional random fields [23] and hierarchical clustering [9] were used on web and ASCII text tables in general domain. Hurst also developed a system that is able to perform text mining analysis according to his model [10]. His system was composed of several components performing smaller tasks such as detection, functional analysis and structural analysis of tables by using rule based and/or machine learning approaches.

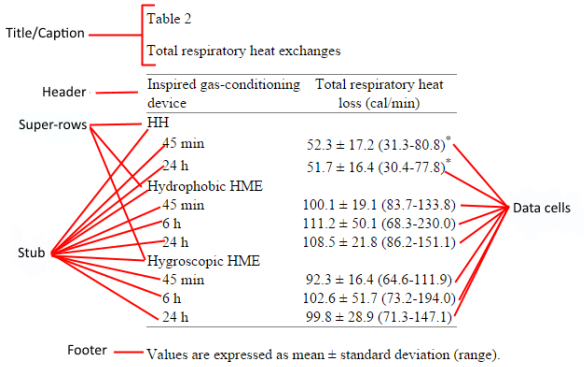

Fig. 1. Table components (PMC29053): Cell – the basic grouping within a table; Caption – a textual description of the table's content; Footer – additionally describes data or symbols used in the table; Column – a set of vertically aligned cells; Row – a set of horizontally aligned cells. Header – top-most row (or set of several top-most rows) that defines data categories of data columns; Stub or column header – typically the left-most column of the table that categorizes the data in the rows. Super-row – groups column headers and data by some concept.

There are several *applications* that use tables. For example, the BioText Search engine [8, 6] performs information retrieval from text, abstracts, figures and tables in biomedical documents. Wei et al. [23] created a question-answering system that looked for answers in tables, using CRF and information retrieval techniques. Few attempts were made to extract information using linked data and databases [16, 24] with machine learning methods trained on a standardized set of tables [21]. There have been several approaches to semantically annotate columns using external resources, such as search engines [18] or linked data resources [14]. *WebTables* used a hypothesis that tables in web documents could be viewed as relational database tables [4]. They created a huge corpus database that consists of 154 million distinct relational tables from the world wide web that can be used for schema auto-complete, attribute synonym finding or joint graph traversal in database applications. Most of these approaches restricted themselves to simple tables because they lacked functional or structural processing steps.

Doush and Pontelli [7] created an spreadsheet *ontology* that included a model of table for screen reader's purposes. Their model considered tables that can be

found in spreadsheets (usually simple matrix tables). They differentiate between data and header cells, but they do not include other possible cell roles such as stub and super-row cells.

3 Table Model

Since current table models focus mainly on visualization for human readers, we here propose a new model for computational processing which is comprised of two components:

- Table types: common table structural types that determine the way of reading the table;
- Data model: models the table structure and data in a way that data can be automatically processed by the machine (including visualisation).

3.1 Table types

We define three main structural table types with several sub-types based on the table's dimensionality:

- **One-dimensional (list) tables** are described by a single label. The label is usually placed in the header (see Figure 2 for an example). One-dimensional tables may have multiple columns, representing the same category, where multi-column structure is used for space saving purposes.
- **Two-dimensional** or *matrix tables* have data arranged into a matrix categorised usually by two labels: a column header and row header (stub). In our model, these tables may have multiple layers of column or row headers (see Figure 4 for an example).
- **Multi-dimensional tables** contain more than two dimensions. We identify two types of multi-dimensional tables:
  - **Super-row tables** contain super-rows that group row headers below them (see example in Figure 1). A super-row table can have multiple layers of super-rows, forming a tree-like structure. This structure is typically visually presented with an appropriate number of white spaces in front of each stub's label.
  - **Multi-tables** are tables composed of multiple, usually similar tables, merged into one table. In some cases, headers of concatenated tables inherit some categorisation from the header of the first table.

3.2 Data Model

The proposed data model captures necessary information for semantic understanding of tables to facilitate further processing and knowledge gathering. We have extended the spreadsheet ontology for tables [7] by adding entities that are

Table 2

Examples of the objectives of the cards

- Describe a situation before interpretation
- Devise the interpretation of a situation as a hypothesis
- Search for different interpretations of the same situation
- Identify the cognitive and behavioural consequences of the different hypotheses
- Search for a link between the interpretation given for a situation and a personal real-life experience
- Put the hypotheses in hierarchical order in terms of their probability
- Search for arguments for or against a hypothesis
- Conceive a way of testing a given hypothesis in reality

Fig. 2. Example of a list table (PMC161814)

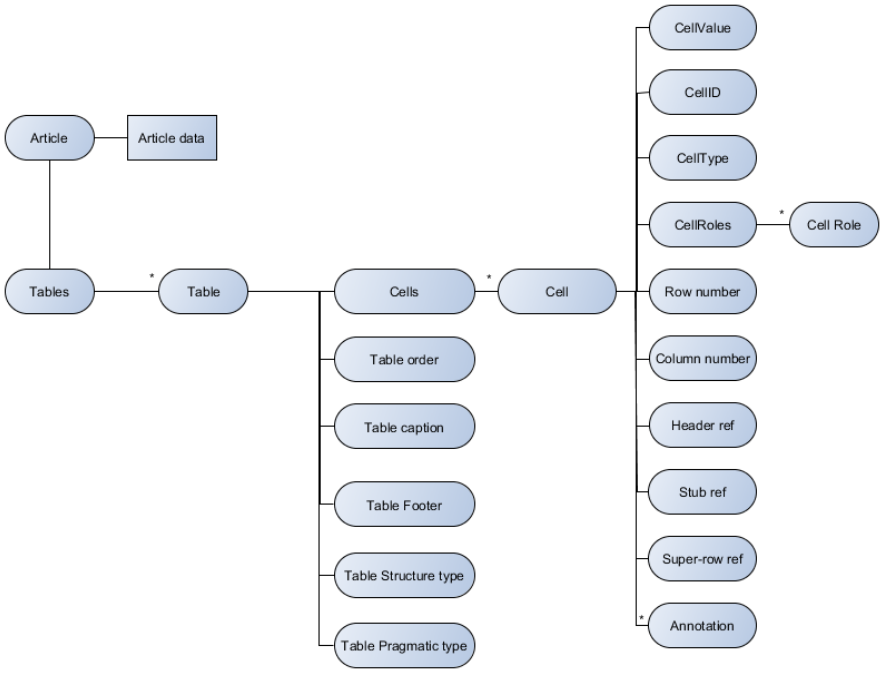

Fig. 3. Hierarchy of the proposed table model

not specific for navigation in screen readers, so it contains information that can aid text mining from tables and visualization.

The model has article, table and cell layers (see Figure 3), which are arranged in a tree-like instantiation, with the *Article* node as the top element, containing the article information (i.e. title, reference, authors, text) and a list of tables. The article layer also stores where tables are mentioned within the document. Caption, footer, order of the table in the document and its structural type (dimensionality of the table, as defined above) are stored in the *table layer*. The table node also contains a list of table's cells. Finally, in the *cell layer*, the model stores the information about each cell including its value, function and position

in the table. In the cell layer, the model also stores information regarding structural references to the navigational cells (headers, stubs and super-rows). The references to navigational cells are set by the ID of the closest cell in the navigational area. If navigational cells contain multiple layers, we apply a cascading style of cell referencing, where lower layers (closer to the data cell) reference the higher order layer (see Figure 4). In this layer, the model further captures any possible annotations of the cell content, which might be added during the table processing. For example, annotations may be syntactic, giving information about the type of value inside the cell, or semantic, mapping to a knowledge source (e.g. ontology or thesauri such as UMLS [3]).For each annotation, we record the span positions of annotated parts in content, concept names, ids in the lexicon or ontologies with which the text was annotated, name of the annotation knowledge source, its version, and environment description.

Table 1

Reported and audited outcome data

Trial phase Audited	Pre-intervention		Post-intervention	
	Intervention	Control	Intervention	Control
Patients seen	307	209	418	237
Referrals	56	39	80	63
Number of practices	27	25	27	25

Fig. 4. Example of cascading referencing of the header relationships (PMC270060). The cell with the value 56 is linked to the header "Intervention", which is linked to the super-header "Pre-intervention".

We note that spanning cells in our model are split and the content of a cell is copied to all cells that were created in the splitting process. Column, row numbers and cell ids are assigned after the splitting of spanning cells.

4 Methodology for automatic table disentangling

We propose a methodology that automatically performs the functional and structural analysis of tables in PMC documents. The method uses a set of heuristic rules to disentangle tables and transform them into the previously described table model.

4.1 Identification of functional areas

The aim of functional analysis is to identify functional areas (headers, stubs, super-rows, data cells) within the table.

Header identification. In most PMC documents, headers are marked using a *thead* XML tag. If *thead* tag exists, we assume that it is correctly labelled. For

tables that do not have a header annotated, we examine syntactic similarity of the cells over the column. This is performed by using a window that takes five cells from the column and checks whether the content has the same syntactic type (i.e. string, single numeric value (e.g. 13) numeric expression (e.g. 5 ± 2), or empty). If all cells are of the same syntactic type, we assume that the table does not have a header. However, if the cells are syntactically different, for example, the first 2 cells are strings while the rest are numeric, we move the window down until it reaches the position where all cells in it have the same syntactic type. The cells above the window are marked as header cells. The window size of five cells is chosen based on experimental experience. We have encountered tables that had up to four rows of headers, so the window size needs to be large enough to capture syntactic type differences. The algorithm then marks as header only rows with all cells marked as headers.

Another heuristic for determining header rows is to check whether some of the first row cells spans over several columns. If they do, we assume that the header contains the next rows, until we reach first one with no spanning cells.

Headers in multi-tables are usually placed between horizontal lines. Only the first header is usually marked with *thead* tags. If multiple cells between the lines have content, these cells are marked as header cells. However, if only one cell has content, these cells are classified as super-rows.

Stub identification. Stubs or column headers are usually cells in the left most column. However, if cells in the left-most columns are row-spanning, the stub contains the next columns, until the first column with no spanning cells is identified.

Super-row identification. Super-rows are rows that group and categorise stub labels. They can have multiple layers. In order to recognise super-rows, our method uses the following heuristics:

- A super-row can be presented as cells that span over the whole row. If these cells have regular content, they are labelled as super-rows.
- A tables may have multiple layers of super-rows. Authors usually present subgroup of relationships with leading blank spaces (indentation) at the beginning of the grouped elements. The number of blank spaces determines the layer of categorisation (i.e. the first layer has usually one blank space, the second has two, etc.). In other words, indentation level visually structures the super-row and stub layers. The row with a label that has less blank spaces than the labels in a stub below, is categorising them, and is therefore considered their super-row. Since there can be multiple levels of super-rows, we used a stack data structure in order to save the associated super-rows of the currently processed cell.
- In PMC documents, it is usual for spanning cells to be presented as a column with multiple cells where only one cell has content (usually the leading one). Rows with only one cell with content are labelled as super-rows.

4.2 Identification of inter-cell relationships

Once the functional areas are detected, we further attempt to identify relationships between cells. Using the detected functional areas, the method classifies tables into one of the four structural classes (one-dimensional, matrix, super-row, multi-table). This classification is based on a set of rules about the functional areas of the table. For example, if the table contains multiple headers, it is classified as multi-table. If it contains super-rows it is a super-row table. If table has only one dimension, it is a list table. Otherwise, it is matrix table.

Depending on the class, the method decides which relationships to search for. For example, data cells in one-dimensional tables can contain only headers, in matrix tables they contain relationships with stubs and headers, while in super-row tables they contain an additional relationships with the super-rows, which may be cascading with multiple layers. Data cells are related to header cells above, stub cells on the left and super-rows above. Navigational cells are related to the higher layers of navigational cells as defined in the cascading referencing model.

Detected functions and relationships of cells can be stored in an XML file and mySQL database according to our model.

5 Results and Evaluation

5.1 Data set

We collected a data set by filtering PMC data for clinical trial publications. We mapped MEDLINE citations, with clinical trial publication type ("Clinical Trial", "Clinical Trial, Phase I", "Clinical Trial, Phase II", "Clinical Trial, Phase III" and "Clinical Trial, Phase IV"), to the PMC and extracted full text articles. The data set contains 2,517 documents in XML format, out of which 568 (22.6%) had no tables in XML (usually containing only reference to a scanned image). The data set contains a total of 4,141 tables, with 80 cells per table on average.

Clinical trial publications are rich in tables, containing, on average, 2.4 tables per article. Biomedical literature, as a whole, contains on average 1.6 tables per article (30% less than clinical literature).

5.2 Table disentangling performance

The system was able to process 3,573 tables from the data set (86%). Table 1 presents the numbers of tables identified as belonging to different types. It is interesting that matrix tables make over 55% of tables, along with super-row tables (over 42%), while list and multi-table are quite rare (around 2%).

We performed the evaluation of the functional and structural analyses on a random subset of 30 articles containing 101 tables. The evaluation sample contains tables from each table type and has been evaluated manually. The detailed information about the evaluation data set and the performance on structural table type recognition is given in Table 1.

	Overall	List tables	Matrix tables	Super-row tables	Multi-table
Number of tables	3573	27 (0.76%)	1974 (55.24%)	1517 (42.46%)	55 (1.54%)
Number of evaluated	101	6	51	27	17
Accuracy of structural type recognition	92.07%	100%	96%	96.3%	70.6%

Table 1. Overview of the dataset and accuracy of the recognition of structural table types

	TP	FP	FN	Precission	Recall	F-Score
Cell role – header	**1041**	**35**	**260**	**96.70%**	**80.00%**	**87.60%**
List	1	0	0	100.00%	100.00%	100.00%
Matrix	469	9	0	98.10%	100.00%	99.00%
Super-row	275	18	20	93.85%	93.22%	93.53%
Multi-table	296	8	240	97.36%	55.22%	70.47%
Cell role – stub	**1250**	**87**	**22**	**93.49%**	**98.27%**	**95.82%**
List	0	0	7	N/A	N/A	N/A
Matrix	407	1	3	99.75%	99.26%	99.51%
Super-row	488	17	4	96.63%	99.10%	97.89%
Multi-table	355	69	8	83.72%	97.79%	90.22%
Cell role – super-row	**414**	**102**	**66**	**80.23%**	**86.25%**	**83.13%**
List	12	7	0	63.15%	100.00%	77.42%
Super-row	359	26	27	93.24%	93.00%	93.12%
Multi-table	43	63	37	40.57%	53.75%	46.24%
Cell role – data	**3709**	**167**	**41**	**95.69%**	**98.91%**	**97.27%**
List	31	7	6	81.57%	83.78%	82.66%
Matrix	1438	1	12	99.93%	99.17%	99.55%
Super-row	1517	11	21	99.28%	98.63%	98.95%
Multi-table	723	148	2	83.00%	99.72%	90.60%
Overall	**6414**	**391**	**389**	**94.25%**	**94.28%**	**94.26%**

Table 2. Evaluation of functional table analysis

	TP	FP	FN	Precission	Recall	F-Score
References – header	**5402**	**768**	**47**	**87.55%**	**99.13%**	**92.98%**
List	7	0	0	100.00%	100.00%	100.00%
Matrix	2076	15	3	99.30%	99.85%	99.60%
Super-row	2501	61	6	97.61%	98.63%	98.95%
Multi-table	818	692	38	54.17%	95.56%	69.15%
References – stub	**4982**	**147**	**0**	**97.10%**	**100.00%**	**98.55%**
Matrix	1788	14	0	99.22%	100.00%	99.61%
Super-row	2057	95	0	95.58%	100.00%	97.74%
Multi-table	1137	38	0	96.70%	100.00%	98.35%
References – Super-row	**1663**	**78**	**269**	**95.52%**	**86.07%**	**90.55%**
List	29	0	6	100.00%	82.85%	90.62%
Super-row	1456	66	215	95.66%	87.13%	91.12%
Multi-table	178	12	42	93.68%	80.91%	86.82%
Overall	**12047**	**993**	**316**	**92.38%**	**97.44%**	**94.84%**

Table 3. Evaluation of structural table analysis

The results for the functional and structural analyses are presented in Tables 2 and 3. Associations to the right roles and navigational relationships (headers, stubs, super-rows) were considered true positives (TP). Association to the non-existing roles or relationships were considered false positives (FP), while missing association were considered false negatives (FN). For the functional analysis, the method archived an F-score of 94.26%, with the lowest performance on the identification of super-row areas for the multi-tables. Our results are comparable and better than previously reported. For example, Hurst [10] combined Naive Bayes, heuristic rules and pattern based classification archiving F-score of around 92% for functional analysis. Similarly, Tengli et al. [21] reported F-score of 91.4% for the table extraction task in which they recognised labels and navigational cells , while Wei et al. [23] reported an F-measure of 90% for detecting headers using CRF. Cafarella et al. [4] detected navigational cells with precision and recall not exceeding 89% and Jung et al. [12] reported 82.1% accuracy in extracting table headers.

For the task of structural analysis, the system achieved an F-score of 94.84%. For comparison, Hurst's system performed with 81.21% recall and 85.14% precision. It is also important to note that input data in Hurst's system were perfectly formatted, while the PMC data is often not. To the best of our knowledge, there is no other system that attempted to performing the combined task of functional and structural table analysis.

During the error analysis, we identified misleading mark-up and complex tables unique to a specific paper in the evaluation set as the main reasons for errors. In PMC documents, XML mark-up features such as spanning cells, head tags, and breaking lines are often misused to make tables look visually appealing. Although we have applied some heuristics that can overcome some of the issues, some of the misleading XML labelling remains challenging. Furthermore, there are tables that are not only complex in structure, but their structure is unique to specific paper, and thus difficult to generalise. Our method made significant number of errors on multi-tables, since it is challenging to determine whether a row is a new header or just an emphasized row or super-row just by analysing XML structure. Errors in wrongly recognizing headers or super-rows cause high amount of false links in structural analysis, since relationships in the subsequent rows will be wrongly annotated. However, multi-tables are relatively rare, so this did not heavily affect the overall results.

6 Conclusion

In this paper we have presented a model to computationally represent tables found in scientific literature. We also presented a domain-independent methodology to disentangle tables and add annotations about functional areas and relationships between table cells. The evaluation has shown that the table structure can be identified with high F-scores (above 94%) which is encouraging. Even though we performed evaluation on the PMC clinical trial documents, the proposed approach can be extended to HTML or any other XML-like for-

mat. The Implementation of the method is available at `https://github.com/nikolamilosevic86/TableAnnotator`.

Although the results for these steps are encouraging, there are still a number of challenges in table mining, mainly in the semantic analysis of the cell content and the methods to query and retrieve table data.

The proposed model can serve as a basis to support applications in information retrieval, information extraction and question answering. We have already performed several information extraction experiments [15] and in the future we are planning to develop a general methodology for information extraction from tables in biomedical literature that uses the presented approach as its basis. Our methodology can be also used as a basis for semantic analysis and querying of tables. In addition, the model can aid systems in accessibility domain. For example, screen readers for visually impaired people could enable easy navigation through tables by providing information about cell's relationships and functions.

Acknowledgments

This research is funded by a doctoral funding grant from the Engineering and Physical Sciences Research Council (EPSRC) and AstraZeneca Ltd.

References

1. Alley, M.: The craft of scientific writing. Springer Science & Business Media (1996)
2. Attwood, T.K., Kell, D.B., McDermott, P., Marsh, J., Pettifer, S., Thorne, D.: Utopia documents: linking scholarly literature with research data. Bioinformatics 26(18), i568–i574 (2010)
3. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research 32(suppl 1), D267–D270 (2004)
4. Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: Webtables: exploring the power of tables on the web. Proceedings of the VLDB Endowment 1(1), 538–549 (2008)
5. Chavan, M.M., Shirgave, S.: A methodology for extracting head contents from meaningful tables in web pages. In: Communication Systems and Network Technologies (CSNT), 2011 International Conference on. pp. 272–277. IEEE (2011)
6. Divoli, A., Wooldridge, M.A., Hearst, M.A.: Full text and figure display improves bioscience literature search. PloS one 5(4), e9619 (2010)
7. Doush, I.A., Pontelli, E.: Non-visual navigation of spreadsheets. Universal access in the information society 12(2), 143–159 (2013)
8. Hearst, M.A., Divoli, A., Guturu, H., Ksikes, A., Nakov, P., Wooldridge, M.A., Ye, J.: Biotext search engine: beyond abstract search. Bioinformatics 23(16), 2196–2197 (2007)
9. Hu, J., Kashi, R., Lopresti, D., Wilfong, G.: A system for understanding and reformulating tables. In: Proceedings of the Fourth IAPR International Workshop on Document Analysis Systems. pp. 361–372 (2000)
10. Hurst, M.F.: The interpretation of tables in texts. Ph.D. thesis, University of Edinburgh (2000)

11. Jensen, L.J., Saric, J., Bork, P.: Literature mining for the biologist: from information retrieval to biological discovery. Nature reviews genetics 7(2), 119–129 (2006)
12. Jung, S.W., Kwon, H.C.: A scalable hybrid approach for extracting head components from web tables. Knowledge and Data Engineering, IEEE Transactions on 18(2), 174–187 (2006)
13. Kieninger, T., Dengel, A.: The t-recs table recognition and analysis system. In: Document Analysis Systems: Theory and Practice, pp. 255–270. Springer (1998)
14. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. Proceedings of the VLDB Endowment 3(1-2), 1338–1347 (2010)
15. Milosevic, N., Gregson, C., Hernandez, R., Nenadic, G.: Extracting patient data from tables in clinical literature: Case study on extraction of BMI, weight and number of patients. In: Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016). vol. 5, pp. 223–228 (2016)
16. Mulwad, V., Finin, T., Syed, Z., Joshi, A.: Using linked data to interpret tables. In: Proceedings of the First International Conference on Consuming Linked Data-Volume 665. pp. 109–120. CEUR-WS. org (2010)
17. Ng, H.T., Lim, C.Y., Koo, J.L.T.: Learning to recognize tables in free text. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. pp. 443–450. ACL (1999)
18. Quercini, G., Reynaud, C.: Entity discovery and annotation in tables. In: Proceedings of the 16th International Conference on Extending Database Technology. pp. 693–704. ACM (2013)
19. Son, J.W., Lee, J.A., Park, S.B., Song, H.J., Lee, S.J., Park, S.Y.: Discriminating meaningful web tables from decorative tables using a composite kernel. In: Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on. vol. 1, pp. 368–371. IEEE (2008)
20. Spasić, I., Livsey, J., Keane, J.A., Nenadić, G.: Text mining of cancer-related information: review of current status and future directions. International journal of medical informatics 83(9), 605–623 (2014)
21. Tengli, A., Yang, Y., Ma, N.L.: Learning table extraction from examples. In: Proceedings of the 20th international conference on Computational Linguistics. pp. 987–994. ACL (2004)
22. Wang, Y., Hu, J.: A machine learning based approach for table detection on the web. In: Proceedings of the 11th international conference on World Wide Web. pp. 242–250. ACM (2002)
23. Wei, X., Croft, B., McCallum, A.: Table extraction for answer retrieval. Information retrieval 9(5), 589–611 (2006)
24. Wong, W., Martinez, D., Cavedon, L.: Extraction of named entities from tables in gene mutation literature. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing. pp. 46–54. ACL (2009)
25. Yesilada, Y., Stevens, R., Goble, C., Hussein, S.: Rendering tables in audio: the interaction of structure and reading styles. In: ACM SIGACCESS Accessibility and Computing. pp. 16–23. No. 77-78, ACM (2004)
26. Yildiz, B., Kaiser, K., Miksch, S.: pdf2table: A method to extract table information from pdf files. In: IICAI. pp. 1773–1785 (2005)
27. Zhu, F., Patumcharoenpol, P., Zhang, C., Yang, Y., Chan, J., Meechai, A., Vongsangnak, W., Shen, B.: Biomedical text mining and its applications in cancer research. Journal of biomedical informatics 46(2), 200–211 (2013)