Extracting Patient Data from Tables in Clinical Literature

Case Study on Extraction of BMI, Weight and Number of Patients

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

[1] *School of Computer Science, University of Manchester, Manchester, U.K.*

[2] *AstraZeneca plc, Cambridge, U.K.*

[3] *Health e-Research Center, University of Manchester, Manchester, U.K.*

Keywords: Text Mining, Table Mining, Information Extraction, Natural Language Processing, Clinical Trials.

Abstract: Current biomedical text mining efforts are mostly focused on extracting information from the body of research articles. However, tables contain important information such as key characteristics of clinical trials. Here, we examine the feasibility of information extraction from tables. We focus on extracting data about clinical trial participants. We propose a rule-based method that decomposes tables into cell level structures and then extracts information from these structures. Our method performed with a F-measure of 83.3% for extraction of number of patients, 83.7% for extraction of patient's body mass index and 57.75% for patient's weight. These results are promising and show that information extraction from tables in biomedical literature is feasible.

1 INTRODUCTION

The amount of published scientific research is accelerating: the number of published papers is growing at a double-exponential pace (Hunter and Cohen, 2006). MEDLINE contains over 25 million references[1] and it is impossible to cope with this amount of published research.

Text mining provides tools and methods to deal with large numbers of articles in biomedicine. However, these efforts have been focused mainly on the processing of unstructured text and most of them ignored lists, tables and figures.

Tables are used for storing large amounts of factual or statistical data in a structured, concise and human-readable way (Tengli et al., 2004). They also provide a way for storing multidimensional data. The visual layout of a table often describes relationships between the items in the table. Because of the variety of layouts, it is challenging to perform analysis of data in this form.

In biomedicine, important experimental information, such as the settings and the results of experiments, interactions between substances, drug side effects, information about arms and patients, are usually stored in tables. In PMC database, more than 72% of research articles contain tables. We manually found that some of the documents in the database do not contain the whole article in XML format (scanned documents, containing only parts in XML). Also, we calculated that the PMC articles contain on average 2.72 tables.

In this paper, we present a method for table decomposition and a case study on extracting information from tables in biomedical literature. The aim of our study is to examine the feasibility of information extraction about patients from tables in clinical literature. Our case study performed extraction of number of patients, body mass indexes (BMI) and weight of patients from tables.

2 BACKGROUND

Hurst (Hurst, 2000) was among the first to examine tables from the text mining perspective. He proposed a model of tables with five components: graphical, physical, functional, structural and semantic. Also, Hurst created one of the first table mining engines. He split the process of table mining into three parts: table detection, functional analysis and information extraction.

The table detection step examines how to correctly detect tables in the documents. Work has been done in detecting tables from PDF, HTML and

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

Milosevic, N., Gregson, C., Hernandez, R. and Nenadic, G.
Extracting Patient Data from Tables in Clinical Literature - Case Study on Extraction of BMI, Weight and Number of Patients.
DOI: 10.5220/0005660102230228
In *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016) - Volume 5: HEALTHINF*, pages 223-228
ISBN: 978-989-758-170-0

ASCII documents using Optical Character Recognition (Kieninger and Strieder, 1999), machine learning algorithms such as C4.5 decision trees (Ng et al., 1999) and SVM (Son et al., 2008) or heuristics (Yildiz et al., 2005).

The second step is functional analysis and it examines the purpose of areas of the table. The aim of this step is to identify which cells contain raw data and which contain navigational data. Approaches using machine learning methods like C4.5 decision trees (Chavan and Shirgave, 2011) or CRF (Wei et al., 2006) were used.

The final step is semantic processing. In this step, relationships and semantics of the table elements are analysed. Semantic processing of the tables are used for information retrieval (Hearst et al., 2007; Divoli et al., 2010), information extraction (Mulwad et al., 2010; Wong et al., 2009) and question answering systems (Wei et al., 2006).

So far, no work has been conducted on extracting information from tables in clinical literature.

3 METHOD

We aim to extract information from tables about participants of the clinical trials such as their number, BMI and weight. The method we propose is composed of two parts: table decomposition into structures that are more suitable for further processing and information extraction. We propose a way to decompose table into cell-level data structures while maintaining information about relationships between elements of the table. Table decomposition, viewed through Hurst's model, represent functional and structural table analysis. The second part considers information extraction from the tables, which corresponds to semantic analysis in Hurst's model.

Data

Our dataset had 2517 documents collected as a clinical trial publications from PubMedCentral (PMC)[2]. Out of these documents 568 had no XML presentation of tables. They had a reference to the image of scanned table. The total number of tables in our dataset was 4141.

Firstly, we conducted a manual analysis on a small sample of 70 PMC documents with 217 tables. Based on our analysis we were able to create rules to identify structure, decompose tables in structured manner and extract information.

3.1 Table Decomposition

Table decomposition contains five steps.

In the first step, the algorithm is locating table with its meta-data such as caption and footer. These data are stored in particular XML tags.

In the second step, our algorithm locates headers and stubs of the table. Cells that are inside the *thead* tags are labelled as header cells. The left-most column cells are labelled as the stub cells. If this column has row-spanning cells, then the following column is also labelled as part of the stub. Row-spanning cells are usually used to group and categorise other stub cells in the following column. The first column with no row-spanning cells outside header will be the last column labelled as the stub. Similarly, complex headers with column-spanning cells are labelled, if there is no *thead* tag. If there is no *thead* tags, our method is checking whether the table does not have a header by checking similarity of value types between first five rows. Since the table might have multiple layers of headers, five was the optimal number of rows for this check, since it indicates in an unambiguous way separation between types. If the cell in the first row has different type (i.e. text) than the following rows (i.e. numeric), the first cell is labelled as part of header. If all five cells have values of same type, the table has no header. Types of cells could be empty, numeric (integer or floating point number), partially numeric (number with special characters and punctuations) and string.

In the third step, spanning cells (recognised by appropriate XML attribute) are split and the content of the cell is copied to all the newly created cells(Chen et al., 2000).

The fourth step is classification of the table by number of dimensions. Navigational paths are read differently for one, two or multi-dimensional tables. Our algorithm identifies three types of tables using heuristics rules. **List (one dimensional) tables** contain a list of items in one or more columns (space saving reasons). They can be recognized if it has only one column, the header is spanning through all the columns or if there is same header for all the columns. **Matrix (two dimensional) tables** contain data arranged in simple matrix of cells (Example can be seen in Figure 3). **Super-row (multi-dimensional) tables** are similar to matrix tables, but the presence of super-rows (Tengli et al., 2004) changes the way they are read (Example can be seen in Figure 1). Super-rows are usually presented as a row inside data part of the table that is spanning through all columns or a row with a value only in one cell.

In the last step, our method is iterating through all data cells and trying to find the correct navigation

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

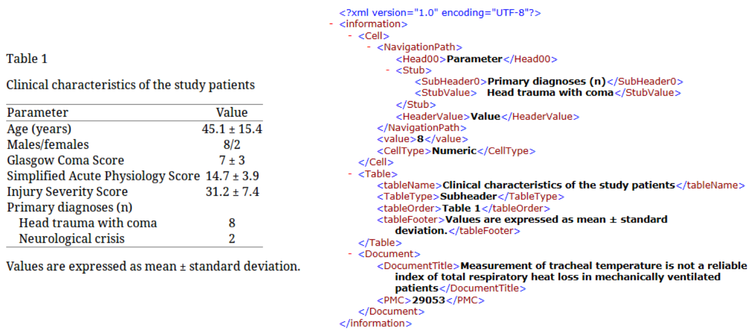

Table 1

Clinical characteristics of the study patients

Parameter	Value
Age (years)	45.1 ± 15.4
Males/females	8/2
Glasgow Coma Score	7 ± 3
Simplified Acute Physiology Score	14.7 ± 3.9
Injury Severity Score	31.2 ± 7.4
Primary diagnoses (n)	
Head trauma with coma	8
Neurological crisis	2

Values are expressed as mean ± standard deviation.

```
  <?xml version="1.0" encoding="UTF-8"?>
- <information>
  - <Cell>
    - <NavigationPath>
        <Head00>Parameter</Head00>
      - <Stub>
          <SubHeader0>Primary diagnoses (n)</SubHeader0>
          <StubValue>   Head trauma with coma</StubValue>
        </Stub>
        <HeaderValue>Value</HeaderValue>
      </NavigationPath>
      <value>8</value>
      <CellType>Numeric</CellType>
    </Cell>
  - <Table>
      <tableName>Clinical characteristics of the study patients</tableName>
      <TableType>Subheader</TableType>
      <tableOrder>Table 1</tableOrder>
      <tableFooter>Values are expressed as mean ± standard
        deviation.</tableFooter>
    </Table>
  - <Document>
      <DocumentTitle>Measurement of tracheal temperature is not a reliable
        index of total respiratory heat loss in mechanically ventilated
        patients</DocumentTitle>
      <PMC>29053</PMC>
    </Document>
  </information>
```

Figure 1: Example of the table (PMC29053) and the decomposition XML output for one cell from that table.

path. Navigation path is a path through the navigational cells (header, stub, super-rows) that logically annotates the data from the data cell. In list tables only the header value is part of navigation path. For matrix tables, the algorithm has to read the header cell in the same column as the given cell, the stub cell in the same row as the cell and the header value for stub's column. Since the super-row table may have a number of super-row levels in a tree-like structure, we created a stack structure that stores current super-row paths, as the algorithm iterates through the cells. For this kind of table, our method reads a header value for the stub (stub's label), all levels of super-rows above the item of interest, the stub value and the header value above the cell.

Data retrieved from the tables are stored in the XML elements (see Figure 1).

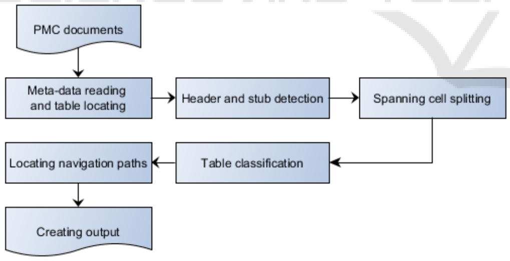
PMC documents
Meta-data reading and table locating
Header and stub detection
Spanning cell splitting
Locating navigation paths
Table classification
Creating output

Figure 2: Workflow of table decomposition method.

A work-flow diagram of our method can be seen in Figure 2.

3.2 Table Information Extraction

We performed two case studies on information extraction from tables. The first study's objective was to extract the total number of patients, while the second had to extract BMI and weight of patients from a clinical trial publication. In the second task, the participant group names had to be extracted together with the appropriate mean BMI or weight. For example, table shown in Figure 3 has two participant groups.

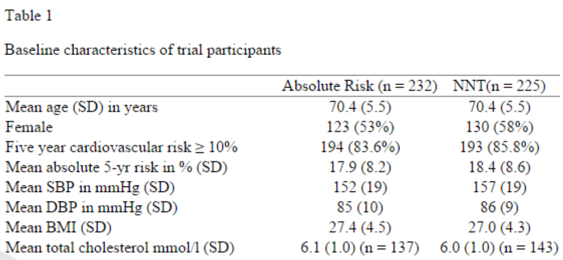

Table 1

Baseline characteristics of trial participants

	Absolute Risk (n = 232)	NNT(n = 225)
Mean age (SD) in years	70.4 (5.5)	70.4 (5.5)
Female	123 (53%)	130 (58%)
Five year cardiovascular risk ≥ 10%	194 (83.6%)	193 (85.8%)
Mean absolute 5-yr risk in % (SD)	17.9 (8.2)	18.4 (8.6)
Mean SBP in mmHg (SD)	152 (19)	157 (19)
Mean DBP in mmHg (SD)	85 (10)	86 (9)
Mean BMI (SD)	27.4 (4.5)	27.0 (4.3)
Mean total cholesterol mmol/l (SD)	6.1 (1.0) (n = 137)	6.0 (1.0) (n = 143)

Figure 3: Example of a clinical trial demographic table that contains information about patients BMI (PMC58836).

Extracted information will be: [Absolute Risk (n = 232): BMI: 27.4 (4.5)] and [NNT (n = 225): BMI: 27.0 (4.3)].

3.3 Extraction of Number of Trial Participants

The number of participants is a numerical value and there is a limited set of trigger words to indicate its appearance in table. The number of patients could be presented in different places in the table and it may not be presented as a single (overall) number, but also as a number of participants per each arm of trial.

The table caption usually presents the total number of the clinical trial participants. We extract the number of participants using two rules. The first rule is looking for a number, followed by one of the trigger words (subject, patient, person, individual, people, infant) in either singular or plural in its vicinity. The trigger word does not need to be the word next to the number, since in some cases the authors may want to specify the participants more (e.g. 16 1-month-old infants, 1239 blood donors). The second rule is looking for a pattern consisting of letter n, the equals sign and a number (e.g. n=19).

There are several ways to store the number of clinical trial participants in navigational cells of the table. One way is to store the total number of patients in a stub, while the other is storing it in the header. Usually, in stubs and headers, the number of patients are presented in the form of mathematical expression (e.g. n = 19). In stubs, we are often expecting the total number of patients in one cell. Since header may have values per arm in each column, we created a list of candidates. Firstly, all the values are added to the list. If the content of some cell contained the word ”overall”, ”total” or the phrase ”all patients”, that value is considered as the total number of participants. However, if such cell does not exist, we check if the stub's header cell has a value for number of patients. If none of this is the case, the values from the header columns are summed (example of this can be seen in Figure 3).

Also, the number of patients may be placed in the body of the table. Similarly to headers, data cells may present the number of patients in parts (e.g. per arm), as single total number, or, in some tables, they may contain both partial and total numbers. Since the data cells may contain only numerical values, looking for trigger words and patterns has to be done in the appropriate stub cells. We have defined trigger phrases which our method searches for in the stub (Number of patients, Num. of participants, etc.). If found, values from the data cells are extracted and added to the list of candidates. Headers also need to be analysed (check if header value contain words ”overall”, ”total” or ”all patients”) in order to determine if there is some cell presenting the total number of participants. If there is no such column, the summed value represents the total number of participants.

3.4 Extracting Body Mass Index and Weight

The second case study extracts information about BMI and mean weight of trial participants. This task is much more complex because we want to extract information, together with the participant group names in which these values were measured.

For the BMI extraction, our approach is to look in the stub of the table for trigger phrases ”body mass index” or ”bmi”. If a table contains these trigger phrases, values from the table body are extracted. However, we also checked whether the value is in the appropriate range (15-40). If the value is not in this range, it does not represent mean BMI value, but other value such as BMI change, standard deviation, etc. If there is more then one column with BMI values, the headers are probably the names of the participant groups. To identify header cells that do not represent participant group names, list of terms is created with tokens such as ”range”,”p*”,”±”,”T”,”p-value”,”p* value”,”%”,”significance”. Appearance of these words indicates that the column does not contain BMI values.

Using these heuristics it is not possible to obtain only arm names, but rather patients groups, since the authors may create demographic tables where they divide patients either by treatment (placebo, penicillin), location (Paris, Toulouse), follow-up period (data on enrolment, 1 week and 1 month after treatment) or outcomes (survivors, non-survivors).

Similarly, weight of patients was also extracted. In this case trigger phrases were ”weight” and ”bodyweight”. Since tables can present a number of different measures related to weight, a stop list was introduced, which had the role of discarding entries if the stub contains a word from the list near the trigger phrases. Stop list contained words like ”loss”, ”gain” and ”change”. In this case, we were not able not define the range of values since values may be in different measurement units (g, kg, lb) and a wide variety of values is possible.

4 RESULTS

4.1 Table Decomposition Results

We have processed all 2517 PMC clinical trial documents. Our method extracted data from 3573 tables. The corpus contained 55.24% of matrix, 0.76% of list and 42.46% of sub-header tables. Since each table has on average 80 cells, it would be impossible to evaluate the whole dataset. We have chosen 100 random tables from each type of tables and evaluated the algorithm's output for them manually by inspecting every table and its cell structures for correctness. If at least one XML cell structure is not read correctly, table is labelled as incorrectly decomposed.

Table 1: Accuracy of table decomposition system.

Class	Tables in dataset	N. Eval.	Accuracy
Matrix tables	1974 (55.24%)	100	89%
Super-row tables	1517 (42.46%)	100	81%
List tables	27 (0.76%)	27	77.7%
Multi-table tables	55 (1.54%)	55	49.1%
Total	3573	282	84.9%

In Table 1, we present the results of our evaluation. Matrix tables were easiest for decomposition and the accuracy would be even higher if our dataset had perfect markup. Due to the non standard XML labelling, our method in some cases was not able to correctly recognize table type or borders of navigational areas. Some of the mislabelling include spanning cells (not using the attribute, but rather using multiple cells) and incorrect labelling of headers with *thead* tags (incorrectly tagging something as a header). Super-row and list tables performed slightly worse. We encountered a small number of tables that actually presented several similar tables merged together (we called them multi-tables). We included a simple algorithm that is able to recognize navigational paths in them based on presence of horizontal lines. However, this algorithm was not good enough to recognize navigational path with high performance. Due to the small number of these tables, they did not affect our overall performance. Overall accuracy of table decomposition was 84.9%.

4.2 Number of Patients Extraction Results

For the extraction of the number of patients, we processed all documents in our dataset. The total number of participants was extracted from 758 documents. For evaluation purposes we randomly selected 50 documents. Our system performed with a F-measure of 83.3%. More detailed statistics can be seen in Table 2.

Table 2: Performance of extracting total number of patients.

Precision	73.53%
Recall	96.15%
F-measure	83.3%

4.3 BMI, Weight and Patient Group Name Extracting Results

For the extraction of BMI and weight, we selected dataset that contains 113 documents, having in at least one of the tables token related to BMI or weight. We separately evaluated the patient group, weight and BMI extraction. The results are shown in Table 3.

Results for BMI and weight are dependent on how the participant groups were recognized, because each extracted value is assigned to the participant group. Participant groups were extracted with a F-measure of 71.32%. They are hard to extract correctly because they may be formed from a wide range of concepts (location, drug, treatment, time, etc.) and may include acronyms or abbreviations. Complex tables, with multiple levels of headers may create additional complexity, since it might be hard to determine where the name of the group ends and where technical or statistical separation of the table's cells starts (ie. mean and standard deviation columns).

BMI has a higher F-measure than participant group extraction. This may look strange, because in order to extract BMIs, the patient group has to be extracted correctly as well. However, defined BMI range made a large contribution to discarding false positives.

Our method for weight extraction performed with high recall but with very low precision. This is due to the fact that the method was matching trigger phrases, but did not have a well crafted stop list, that could help to distinguish actual patient weight from other weight related concepts.

5 CONCLUSION

Information extraction from tables is not extensively researched. However, in many fields, such as biomedicine, it could be useful, due of the importance of the data presented in tables. Information extraction from tables can use some of the established text mining techniques, but due to the challenge of understanding the visual layouts, new approaches have to be developed as well.

We developed a methodology for table decomposition into cell-level data structures. Our method is able to read table data with associated navigational information. Using these structures, it is easier to perform semantic analysis and information extraction. We performed a case study on extracting number of trial participants, BMIs and names of the participant groups from clinical literature. Although we used relatively simple rules for information extraction, our results are promising (F measure for BMI extraction 83.7%, F measure for weight extraction over 57%). Our results indicated that some information classes may be easier to extract, because it is possible to model expected values, while the others remain a challenge.

The results of our case studies are comparable with state-of-the-art methods in table information extraction. However, not many works report information extraction from tables. Hurst (Hurst, 2000) reported the combined task of functional, structural and relational analysis to have a F score of 83.13%. However, this task matches our table decomposition task, which is just first part of our information extraction method. Gatterbauer et al. (Gatterbauer et al., 2007) created generic information extraction system, but they reported F measure of 52%. Tengli et al. (Tengli et al., 2004) reported the best F measure of 91.4% for information extraction from tables. However, they apply a method on The Common Data Set tables, which is a standardized presentation format for higher education data in the United States. Compared to these tables, tables from PMC are not standardised in any way.

The performance of our method is quite promising

Table 3: Performance extracting BMI, weight and patient groups from PMC clinical trial documents (TP - true positives, FP - false positives, FN - false negatives).

Class	TP	FP	FN	Precision	Recall	F-measure
BMI	72	22	6	76.6%	92.3%	83.7%
Participant group	153	93	27	61.45%	85%	71.32%
Weight	95	133	6	41.66%	94.05%	57.75%

and indicates that information extraction from tables is a feasible task. However, there is a space for advancement. There is still the need for the human curators to control the system and correct mistakes. We believe our system will reduce data curation time for medical documents.

6 AVAILABILITY

Our code and annotated corpus is available on GitHub[3]. Current code is a work in progress and might be subject to changes. The documents user were clinical trial articles from PubMedCentral[4].

REFERENCES

Chavan, M. M. and Shirgave, S. (2011). A methodology for extracting head contents from meaningful tables in web pages. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 272–277. IEEE.

Chen, H.-H., Tsai, S.-C., and Tsai, J.-H. (2000). Mining tables from large scale html texts. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 166–172. Association for Computational Linguistics.

Divoli, A., Wooldridge, M. A., and Hearst, M. A. (2010). Full text and figure display improves bioscience literature search. *PloS one*, 5(4):e9619.

Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B., and Pollak, B. (2007). Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, pages 71–80. ACM.

Hearst, M. A., Divoli, A., Guturu, H., Ksikes, A., Nakov, P., Wooldridge, M. A., and Ye, J. (2007). Biotext search engine: beyond abstract search. *Bioinformatics*, 23(16):2196–2197.

Hunter, L. and Cohen, K. B. (2006). Biomedical language processing: perspective whats beyond pubmed? *Molecular cell*, 21(5):589.

Hurst, M. F. (2000). *The interpretation of tables in texts*. PhD thesis, University of Edinburgh.

Kieninger, T. G. and Strieder, B. (1999). T-recs table recognition and validation approach. In *AAAI Fall Symposium on Using Layout for the Generation, Understanding and Retrieval of Documents*.

Mulwad, V., Finin, T., Syed, Z., and Joshi, A. (2010). Using linked data to interpret tables. *COLD*, 665.

Ng, H. T., Lim, C. Y., and Koo, J. L. T. (1999). Learning to recognize tables in free text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 443–450. Association for Computational Linguistics.

Son, J.-W., Lee, J.-A., Park, S.-B., Song, H.-J., Lee, S.-J., and Park, S.-Y. (2008). 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*, volume 1, pages 368–371. IEEE.

Tengli, A., Yang, Y., and Ma, N. L. (2004). Learning table extraction from examples. In *Proceedings of the 20th international conference on Computational Linguistics*, page 987. Association for Computational Linguistics.

Wei, X., Croft, B., and McCallum, A. (2006). Table extraction for answer retrieval. *Information retrieval*, 9(5):589–611.

Wong, W., Martinez, D., and Cavedon, L. (2009). Extraction of named entities from tables in gene mutation literature. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 46–54. Association for Computational Linguistics.

Yildiz, B., Kaiser, K., and Miksch, S. (2005). pdf2table: A method to extract table information from pdf files. In *IICAI*, pages 1773–1785.

[3] https://github.com/nikolamilosevic86/TableAnnotator

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