

**THE UNIVERSITY OF MANCHESTER - APPROVED ELECTRONICALLY
GENERATED THESIS/DISSERTATION COVER-PAGE**

Electronic identifier: 24798

Date of electronic submission: 22/05/2018

The University of Manchester makes unrestricted examined electronic theses and dissertations freely available for download and reading online via Manchester eScholar at <http://www.manchester.ac.uk/escholar>.

This print version of my thesis/dissertation is a TRUE and ACCURATE REPRESENTATION of the electronic version submitted to the University of Manchester's institutional repository, Manchester eScholar.

A MULTI-LAYERED APPROACH TO INFORMATION EXTRACTION FROM TABLES IN BIOMEDICAL DOCUMENTS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2018

By
Nikola Milošević
School of Computer Science

Contents

Abstract	14
Declaration	16
Copyright	17
Acknowledgements	18
About the author	19
1 Introduction	20
1.1 Table terminology and elements	22
1.2 Tables in biomedical literature	25
1.3 Challenges in automated table processing	29
1.4 Hypothesis and research questions	32
1.5 Scope	32
1.6 Contributions	33
1.7 Thesis structure	35
2 Review of table mining literature	37
2.1 Table description models	37
2.2 Table processing workflow	41
2.3 Table detection	42
2.3.1 Table detection in text (ASCII) documents	43
2.3.2 Table detection in PDF documents	45
2.3.3 Table detection in XML and HTML documents	46
2.4 Functional table analysis	47
2.5 Schema matching	50
2.6 Information retrieval from tables	51

2.7	Table mining applications	52
2.7.1	Knowledge-driven approaches	52
2.7.2	Machine learning approaches	55
2.8	Table navigation for visually impaired people	57
2.9	Conclusion	58
3	Methodology Overview	61
3.1	Introduction	61
3.2	Scope	61
3.3	Table information extraction	64
3.3.1	Table detection	66
3.3.2	Table disentangling	66
3.3.3	Table and cell annotation	67
3.3.4	Information extraction	68
3.4	Evaluation methodologies	71
3.5	Case studies for methodology validation	74
3.6	Data curation and querying interface	75
3.7	Summary	76
4	Disentangling the structure of tables	77
4.1	Model for representing tables	77
4.1.1	Table types	78
4.1.2	Table data representation model	79
4.2	Methodology	82
4.2.1	Reading the articles	83
4.2.2	Identification of functional areas (functional analysis)	83
4.2.3	Identification of inter-cell relationships (structural analysis)	87
4.3	Results and Evaluation	89
4.3.1	Datasets	89
4.3.2	Table disentangling performance	91
4.4	Summary	99
5	Table and cell annotation	101
5.1	Pragmatic analysis	102
5.1.1	Pragmatic analysis case study	103
5.1.2	Evaluation	104

5.2	Semantic tagging	107
5.3	Conclusion and summary	108
6	Information extraction from tables	110
6.1	Cell selection and syntactic analysis	111
6.2	Case studies for cell selection approaches	113
6.2.1	Rule based cell selection and information extraction	113
6.2.2	Machine learning based cell selection	119
6.3	General framework	122
6.3.1	Types of variables	122
6.3.2	Information extraction task specification	124
6.3.3	Defining rules for information extraction	125
6.4	TableInOut: a wizard for information extraction	129
6.5	Summary	130
7	Case studies	132
7.1	Extracting clinical trial baseline characteristics	132
7.1.1	Introduction	132
7.1.2	Methodology	134
7.1.3	Evaluation and results	137
7.1.4	Conclusion	139
7.2	Extracting drug-drug interactions	139
7.2.1	Introduction	139
7.2.2	Methodology	141
7.2.3	Evaluation	147
7.2.4	Conclusion	152
7.3	Summary	152
8	Discussion	154
8.1	Research questions	154
8.1.1	Hypothesis	162
8.2	Challenges	163
8.3	Generalizability	164
8.4	Data curation and querying	166
8.5	Table and cell annotation	168
8.6	Limitations	172

8.7	Availability	173
9	Conclusion and Future perspectives	174
9.1	Summary of thesis contributions	174
9.2	Future work	175
9.3	Final remarks	178
	Glossary	180
	Acronyms	183
A	Case study	200
A.1	INTRODUCTION	200
A.2	BACKGROUND	201
A.3	METHOD	201
A.3.1	Table Decomposition	202
A.3.2	Table Information Extraction	204
A.3.3	Extraction of Number of Trial Participants	204
A.3.4	Extracting Body Mass Index and Weight	206
A.4	RESULTS	207
A.4.1	Table Decomposition Results	207
A.4.2	Number of Patients Extraction Results	208
A.4.3	BMI, Weight and Patient Group Name Extracting Results	208
A.5	CONCLUSION	209
B	Database schema	211
C	TableInOut: Implementation details	214
C.1	Introduction	214
C.2	TableInOut architecture overview	214
C.3	Marvin annotation tool	215
C.4	TableInOut implementation details	218
C.5	Summary	224
D	Guide for writing syntactic rules	225
D.1	Overview	225
D.2	Writing a simple syntactic rule	225
D.3	Adding semantics to the syntactic rule	226

D.4	Conclusion	228
E	TableInOut: Lexical and syntactic rules	229
E.1	Rules for extracting age of patients	229
E.1.1	Rule configuration	229
E.1.2	White list	229
E.1.3	Black list	230
E.1.4	Syntactic rules	230
E.2	Rules for extracting gender	232
E.2.1	Rule configuration	232
E.2.2	White list	232
E.2.3	Black list	233
E.2.4	Syntactic rules	234
F	Library of developed syntactic rules	239
F.1	Syntactic rule - integer	239
F.2	Syntactic rule - float	239
F.3	Syntactic rules - statistical values	240
F.4	Syntactic rules - alternative values (gender)	242

Word Count: 45044

List of Tables

2.1	Approaches to table detection	44
2.2	Approaches to functional analysis (mainly head detection)	48
4.1	Description of the table entities in the table data representation model	82
4.2	Description of the cell entities in the table data representation model .	82
4.3	Statistical data about tables in the datasets. Mean, standard deviation and range are presented per document	91
4.4	Overview of the evaluation dataset	92
4.5	Evaluation of the recognition of structural table types	92
4.6	Evaluation of functional table analysis on the evaluation dataset . . .	93
4.7	Evaluation of functional table analysis on 20 drug-drug interaction ta- bles from DailyMed	94
4.8	Evaluation of header classifiers based on content for detecting header in drug-drug interaction tables	96
4.9	Evaluation of structural table analysis on the evaluation dataset. Inter- cell relationships are evaluated.	96
5.1	Weighted averages for all classes of the pragmatic classification us- ing different content feature sets (each content feature separately and all features combined). The evaluation was done using 10-fold cross validation.	105
5.2	Results of the four-class pragmatic classification experiments on the PMC clinical trial tables. Training and evaluation was performed us- ing the 10-fold cross-validation on 186 "baseline characteristic", 60 "inclusion/exclusion", 239 "adverse event" and 153 "other" tables. . .	106
5.3	Distribution of tables in PMC clinical trial dataset based on their prag- matic class	106
6.1	Results of information extraction for number of patients	114

6.2	Results of information extraction for age of patients, including mean, standard deviation and range	115
6.3	Results of extraction for gender distribution of the patients variable using rule based approach. Evaluation performed on the clinical trial dataset	117
6.4	Results of information extraction for adverse events	118
6.5	Results of selecting cells associated to the patient number variable using various machine learning approaches	120
6.6	Results of selecting cells associated with the age of patients variable (cumulative statistical values such as mean, standard deviation and range) using various machine learning approaches	120
6.7	Results of the selecting cells associated with gender distribution variable using various machine learning approaches	120
6.8	Categories of information that need to be described in order to specify a table information extraction task	125
6.9	Examples of common syntactic patterns and variables that are often represented by them	127
7.1	Examples of lexical cues for extracting given variables	136
7.2	Evaluation of the target variables extracted from Asthma and COPD clinical trials. (TP - true positives, FP - false positives, FN - false negatives)	137
7.3	Functional analysis evaluation of the original TableDisentangler methodology on the DailyMed subset	148
7.4	Machine learning header detection using various algorithms and 10-fold cross validation on the created dataset	148
7.5	Machine learning header detection evaluation for the DailyMed subset	148
7.6	Evaluation of potential drug-drug interaction pairs from tables in DailyMed.	149
A.1	Accuracy of table decomposition system	207
A.2	Performance of extracting total number of patients	208
A.3	Performance extracting BMI, weight and patient groups from PMC clinical trial documents (TP - true positives, FP - false positives, FN - false negatives)	208

List of Figures

1.1	An example of Mesopotamian tablet quoting in the tabular structure the wages of different categories of workers (circa 2350 BC)	21
1.2	Example of a table showing parallel description in structured and compact way (PMC 31582)	22
1.3	Example of table with data that is not mentioned later in text. Table taken from (PMC 29053)	23
1.4	Table elements: Header, stub and body. Table example source: Yildiz et al. (2005)	24
1.5	Elements of complex table (PMC 29053)	24
1.6	Number of published articles by year that are indexed in MEDLINE .	26
1.7	Excerpt of the article presenting table and its reference from the text. Text and tables often present complementary, but not redundant information. Information presented in the text are addition to the information presented in the table (PMC113263)	27
1.8	Example of XML representation and its visualisation (PMC113263) .	30
1.9	Examples of some possible structures of tables	31
2.1	XML representation of table (as used by PMC) and it's HTML visualisation in a web browser (PMC 2410054)	39
2.2	Table ontology as presented by Doush & Pontelli (2013)	41
2.3	Example of three ASCII free text tables: Generated classification output from the Weka toolkit	42

3.1	High-level architecture of the proposed methodology	
	1. Retrieved documents are sent to the table mining engine. 2. The table mining engine uses knowledge sources to extract information from the table. 3. Extracted information is stored in a data store. 4. Data curators review and correct extracted information. 5. Users submit queries to the query interface in natural language. 6. Queries are processed and normalized. 7. Using normalized queries, the data store is queried. 8. Relevant extracted information is presented to the user.	62
3.2	Example of a table presenting values using different presentation patterns (PMC 29047)	63
3.3	High level overview of the methodology	64
3.4	Overview of the methodology steps as they are executed	65
3.5	Example of a table (PMC 65527) and extracted information to the proposed extraction template	69
3.6	Syntactic analysis infers the implicit meaning from the value presentation pattern (upper row) or link to the explicitly stated meaning in the navigational cells (lower row)	71
3.7	Confusion matrix with graphical explanation of true positives, false positives, false negatives and true negatives	72
4.1	Example of a list table (PMC 161814)	78
4.2	Example of a list (one-dimensional) table with multiple columns (PMC 420259)	78
4.3	Example of a matrix table (PMC 65527)	79
4.4	Example of a table with tree like super-row structure. This table has two super-row levels in its stub (PMC 32172)	79
4.5	Example of a multi-table (PMC 57003)	80
4.6	Proposed table representation model	81
4.7	Example of cascading referencing of the header relationships (PMC 270060). The cell with the value 56 is linked to the header "Intervention", which is linked to the upper header "Pre-intervention".	83
4.8	Overview of the methodology for automatic table structure disentangling	83
4.9	An example of PMC XML table and its visual representation	84

4.10	Functional and structural analysis on an example table. The diagram shows step by step labelling of a table. During functional analysis, functional areas are labelled (header - yellow, stub - blue, super-row - orange, data - green). During the structural analysis, related cells are found for each cell. The example shows related cells of data cell with a content 8.	88
4.11	Number of PMC articles and the number of PMC articles with at least one table in XML. Statistics of PMC clinical trial dataset	90
4.12	Number of tables per year and average number of tables in PMC clinical publications	91
4.13	Example of the DailyMed table containing caption in the header cell. Document SetID: a7a2a4e1-9ecd-4e59-82b5-2068b5e50164	95
4.14	Example of the part of the table presenting drug-drug interactions from the DailyMed dataset. Document SetID: 6C08B50E-CC9F-4C49-D7AE-F0FDDCB10199	95
4.15	Example of the XML and table having mislabelled header (PMC 406425)	97
4.16	Example of the table that was falsely classified as multi-table due to the presence of horizontal lines. The table is actually a matrix table. (PMC 3381636)	98
5.1	Examples of tables for each pragmatic class defined in a clinical trial case study	103
6.1	Workflow diagram of the information extraction steps	111
6.2	Extracted number of patient variable and filled template from an example table (PMC 1947993)	114
6.3	Example of a table and extracted values for the age variable (PMC 1906819)	116
6.4	Example of table presenting baseline characteristics as number of people having certain conditions (PMC 2147028)	119
6.5	Variable types, with their subtypes and the example variables for each defined subtype.	123
6.6	Example of one syntactic rule with its semantics for extracting gender distribution of the participants.	129
6.7	Example of one syntactic rule with its semantics for extracting statistical values	129

7.1	Example of the baseline characteristic table from asthma clinical trial presenting FEV1, PEP and Asthma Quality of Life Questionnaire (AQLQ) variables (PMC 2228375)	133
7.2	Workflow of the methodology used for extracting variables from clinical trial documents about asthma and COPD	135
7.3	Example of the table in which row containing "Age" was not recognised as super-row and therefore age ranges (in a row bellow) were not extracted (PMC 3528484)	138
7.4	Workflow diagram	142
7.5	Example of a table in which both caption and footer are inside the table cells (DailyMed setID: 524c025b-809b-440f-a756-e3518d7c92db) . .	144
7.6	Workflow of the modified methodology for functional and structural analysis of DailyMed documents.	144
7.7	ATC coding system	145
7.8	Example of a table presenting multiple interacting drugs per cell (SetID: b9df447c-b65b-45b9-873a-07a2ab6e2d1f)	147
7.9	Example of a drug-drug interaction table with super-rows. Often super-rows were not correctly recognised and their content extracted as an interacting drug (SetID: f02310a3-92ea-9ec4-f218-38ddb8eb0334) . .	150
7.10	Example of a drug-drug interaction table that changes overrides the way of data presentation defined in the header in rows 5 and 6 (SetID: f02310a3-92ea-9ec4-f218-38ddb8eb0334)	151
8.1	Curation interface for checking and improving quality of data after the functional analysis. On the left is original table, while on the right is the same table with functional annotations (colours). The interface was implemented as an independent project (Su 2016).	167
8.2	The interface for querying structurally analysed table data (Tang 2016)	168
8.3	Format of the proposed annotation schema	169
8.4	RichAnnotator tool index page, showing sample XML document with annotated concepts	170
8.5	RichAnnotator tool - annotation screen automatically finds XPath of selected item. Annotator has only to fill the concept details.	171
A.1	Example of the table (PMC 29053) and the decomposition XML output for one cell from that table	203

A.2	Workflow of table decomposition method	204
A.3	Example of a clinical trial demographic table that contains information about patients BMI (PMC 58836)	205
B.1	Database schema used for string information about tables, cells, cell functions and inter-cell relationships with annotations	212
C.1	Workflow of information extraction from tables using TableInOut . . .	215
C.2	Folder structure of a project in TableInOut	218
C.3	TableInOut project management screen	219
C.4	Example of task management screen containing 6 variables and rules for each variable. This set-up was used for case study described in Section 7.1.	220
C.5	TableInOut database management screen	221
C.6	TableInOut task definition screen	221
C.7	Lexical and semantic rule definition screen. User can define lexical cues just by stating them or by stating [word] as a prefix; semantic cues can be stated as annotations ids, usually referring to concept ids in certain vocabulary, using [annID] prefix or as annotation types, referring, for example, to UMLS semantic types of annotation, using [annType] prefix.	222
C.8	TableInOut syntactic rule definition screen	223

Abstract

A MULTI-LAYERED APPROACH TO INFORMATION EXTRACTION FROM TABLES IN BIOMEDICAL DOCUMENTS

Nikola Milošević

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2018

The quantity of literature in the biomedical domain is growing exponentially. It is becoming impossible for researchers to cope with this ever-increasing amount of information. Text mining provides methods that can improve access to information of interest through information retrieval, information extraction and question answering. However, most of these systems focus on information presented in main body of text while ignoring other parts of the document such as tables and figures.

Tables present a potentially important component of research presentation, as authors often include more detailed information in tables than in textual sections of a document. Tables allow presentation of large amounts of information in relatively limited space, due to their structural flexibility and ability to present multi-dimensional information.

Table processing encapsulates specific challenges that table mining systems need to take into account. Challenges include a variety of visual and semantic structures in tables, variety of information presentation formats, and dense content in table cells. The work presented in this thesis examines a multi-layered approach to information extraction from tables in biomedical documents.

In this thesis we propose a representation model of tables and a method for table structure disentangling and information extraction. The model describes table structures and how they are read. We propose a method for information extraction that consists of: (1) table detection, (2) functional analysis, (3) structural analysis, (4) semantic tagging, (5) pragmatic analysis, (6) cell selection and (7) syntactic processing and extraction. In order to validate our approach, show its potential and identify remaining challenges, we applied our methodology to two case studies. The aim of the

first case study was to extract baseline characteristics of clinical trials (number of patients, age, gender distribution, etc.) from tables. The second case study explored how the methodology can be applied to relationship extraction, examining extraction of drug-drug interactions.

Our method performed functional analysis with a precision score of 0.9425, recall score of 0.9428 and F1-score of 0.9426. Relationships between cells were recognized with a precision of 0.9238, recall of 0.9744 and F1-score of 0.9484. The information extraction methodology performance is the state-of-the-art in table information extraction recording an F1-score range of 0.82-0.93 for demographic data, adverse event and drug-drug interaction extraction, depending on the complexity of the task and available semantic resources.

Presented methodology demonstrated that information can be efficiently extracted from tables in biomedical literature. Information extraction from tables can be important for enhancing data curation, information retrieval, question answering and decision support systems with additional information from tables that cannot be found in the other parts of the document.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

- i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University’s policy on presentation of Theses

Acknowledgements

Every seminal work, such as this thesis, is influenced and helped in various ways by many people. First of all, I would like to thank my parents for their support during my PhD and overcoming the issues caused by my absence from their vicinity. I would like to express a great amount of gratitude to my supervisor – Dr Goran Nenadić, for his support and help on my journey from being an engineer to becoming a scientist. My thanks also go to my industrial supervisors from AstraZeneca – Rob Hernandez and Cassie Gregson – who provided me with a significant amount of feedback in the crucial stages of my work. The financial support for this work was provided by the EPSRC and AstraZeneca scholarship, without which my PhD studies would not be possible in this format. Special thanks go to Richard Boyce and his group at the University of Pittsburgh, with whom we managed to apply my table mining approach for extracting drug-drug interactions from U.S. drug labels. The text mining and natural language processing research group also played a great role in my development and this thesis. I would like to thank all members of the group including: Michele Fillanino, Geriant Duck, George Karystianis, Azad Dehghan, Ruth Stoney, Idoia Gomez Paramio and Maksim Belousov. I learned a lot from this talented group of people and they made my office hours seem subjectively shorter and more enjoyable. I would like to thank Winston Cuthbert and Dr Jodi Schneider for proofreading my work and making constructive comments and suggestions. I would also like to thank my flatmates and great friends who made my stay in Manchester unique and unforgettable. I also feel the need to thank all the people from various student societies (such as MUTIS Finance society, Czechoslovak society, Ex-Yugoslav society and Rotaract) who made my time in Manchester easy, enjoyable and fun while helping me learn about new fields and cultures.

About the author

Nikola Milošević was born in Bratislava, Slovakia on the 7th of December 1986. He finished his undergraduate and master studies at the School of Electrical Engineering, University of Belgrade, Serbia. During his master's, Nikola got interested in the field of text mining, which lead to master project named "Sentiment analysis of sentences in Serbian". Nikola has 4 years of professional experience, working as software engineer across industries (Finance, Telecommunications, Software-as-a-Service). From 2012, he is involved in open source community as project leader of OWASP Seraphimdroid project and leader of OWASP Serbia and OWASP Manchester local chapters. In 2013, he won a first price on Startup Weekend Belgrade with a text mining related project. Nikola often participates in hackathons and other computer science related challenges. Some of notable ones are BLAH2 (Biomedical Linked Annotation Hackathon), GreatUniHack, BLAHmuc (Biomedical Linked Annotation Hackathon in Munich), iSec Cyber Security iPuzzle. He was a speaker and his work was presented in a number of conferences and computer science related events such as NLDB 2016 (21st International conference on applications of natural language to information systems), BIOSTEC 2016 (9th International conference on biomedical engineering systems and technologies), BelBi 2016 (Belgrade BioInformatics conference), bSides Manchester 2014 and 2017, Technical analyst's Machine Learning Techniques 2016, etc. In 2014, Nikola received funding for PhD project from EPSRC (Engineering and Physical Sciences Research Council) and AstraZeneca, which led to this dissertation.

Chapter 1

Introduction

Throughout history, presenting information in tabular formats has been a significant feature of the written language. Archaeologists have uncovered tabular structures dating back to the Bronze Age (3500BC-1600BC) in Mesopotamia and Crete. In the ancient Mesopotamian city of Uruk, horizontal and vertical lines were carved on clay tablets to separate distinct zones of meaning (Dilger & Rice 2010). The example of Mesopotamian tablet can be seen in Figure 1.1. In the Minoan civilization in Crete, there are table-like structures on tablets for games, but some of them presented information that was probably recorded by bureaucrats (Whittaker 2013). Ancient Egyptians used hierarchical structuring of tablets or pergaments in such a way that the top part presented images of the divine world, the second part presented images from the ruler's life, while on the bottom were images of enemies (Dilger & Rice 2010). Tables, as we know them today, were developed about 4600 years ago, already in the days of handwritten documents, and usually used for bureaucratic records, such as recording the number of workers in a workplace (Long 2010). The modern English word "table" originates from the Latin "tabula", meaning "a board, plank; writing table; list, schedule; picture, painted panel," originally "small flat slab or piece" usually used for inscriptions or for games (Hoad 1993). In the 15th century, tables could be found in many publications in mathematics and natural sciences (Smith & Ginsburg 1937). Throughout the history, tables had an important place in presenting data in written documents and with the new typesetter tools that made the creation of tables easier, their role increased (Long 2010). Today, tables are used in a variety of documents, but they are prominent in scientific literature.

Tables are used as an appropriate format for storing a potentially large amount of factual or statistical data in a structured way, in particular multidimensional data.

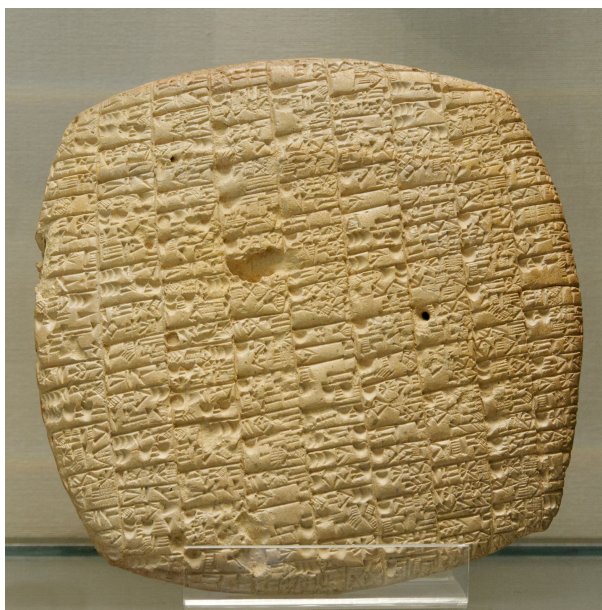


Figure 1.1: An example of Mesopotamian tablet quoting in the tabular structure the wages of different categories of workers (circa 2350 BC)

Various table layouts provide flexibility for structuring data and for storing information in a compact way. If the body of the article is referred to as unstructured text, tables are semi-structured textual parts of the article.

Tables have two main uses: to present data and to present a short parallel description in a compact and structured way that otherwise would have to be expanded and listed in the text (Alley 1996). An example of the table presenting parallel descriptions can be seen in Figure 1.2. Had the authors wished to present this data in the text body, they would have been required to repetitively expound on their otherwise concise description.

In the textual parts of the document, authors may discuss or highlight important findings from the data, but they will usually not repeat the data presented in the table. Figure 1.3 shows an example of a table presenting numeric data that are not mentioned in the textual part of the document. The mention of this table in the article says the following: *"Table 2 shows the total respiratory heat changes of the ventilated gases with the three systems used for conditioning of ventilatory gases. Total respiratory heat loss was significantly less with the HH than with either HME ($P < 0.01$)"* (PMC 29053). The article highlights the significant difference in table's data and discusses it, but does not mention the numerical values. Tables present important features of the documents for presenting information and therefore cannot be ignored.

Table 1

Comfort scale [12]

Variable	Score				
	1	2	3	4	5
Alertness	Deeply asleep	Lightly asleep	Drowsy	Fully awake and alert	Hyper alert
Calmness/agitation	Calm	Slightly anxious	Anxious	Very anxious	Panicky
Respiratory response	No coughing and no spontaneous respiration	Spontaneous respiration with little or no response to ventilation	Occasional cough or resistance to ventilator	Actively breathes against ventilator or coughs regularly	Fights ventilator, coughing or choking
Physical movement	No movement	Occasional, slight movement	Frequent, slight movements	Vigorous movement limited to extremities	Vigorous movements including torso and head
Mean arterial blood pressure	Blood pressure below baseline	Blood pressure consistently at baseline	Infrequent elevations of 15% or more (1-3 during observation period)	Frequent elevations of 15% or more above baseline (more than 3 during observation period)	Sustained elevation of 15% or more
Heart rate	Heart rate below baseline	Heart rate consistently at baseline	Infrequent elevations of 15% or more above baseline (1-3 during observation period)	Frequent elevations of 15% or more above baseline (more than 3 during observation period)	Sustained elevation of 15% or more
Muscle tone	Muscle totally relaxed, no muscle tone	Reduced muscle tone	Normal muscle tone	Increased muscle tone and flexion of fingers and toes	Extreme muscle rigidity
Facial tension	Facial muscles totally relaxed	Facial muscle tone normal, no facial muscle tension evident	Tension evident in some facial muscles	Tension evident throughout facial muscles	Facial muscles contorted and grimacing

Figure 1.2: Example of a table showing parallel description in structured and compact way (PMC 31582)

1.1 Table terminology and elements

The Oxford English Dictionary defines a table as: "an arrangement of numbers, words or items of any kind, in a definite and compact form, so as to exhibit some set of facts or relations in a distinct and comprehensive way, for the convenience of study, reference, or calculation".

A table is considered information bearing element of the document, usually characterized by grid-like appearance. It is a static element of the document that presents information and should not be altered by the reader (unlike for example form). A table is presenting information to the reader by organising a set of meaningful elements on the page so the relationships between those elements, and the manner in which combinations of elements interact, is demonstrated to the reader (Hurst 2000). Some information in the table is assumed by the author to be known to the user. This information is then used to introduce new information and relationships between information

Table 2

Total respiratory heat exchanges

Inspired gas-conditioning device	Total respiratory heat loss (cal/min)
HH	
45 min	52.3 ± 17.2 (31.3–80.8)*
24 h	51.7 ± 16.4 (30.4–77.8)*
Hydrophobic HME	
45 min	100.1 ± 19.1 (83.7–133.8)
6 h	111.2 ± 50.1 (68.3–230.0)
24 h	108.5 ± 21.8 (86.2–151.1)
Hygroscopic HME	
45 min	92.3 ± 16.4 (64.6–111.9)
6 h	102.6 ± 51.7 (73.2–194.0)
24 h	99.8 ± 28.9 (71.3–147.1)

Values are expressed as mean ± standard deviation (range). * $P < 0.01$ versus hydrophobic and hygroscopic HME.

Figure 1.3: Example of table with data that is not mentioned later in text. Table taken from (PMC 29053)

to the reader. Tables are usually not interpreted on their own, but rather with other presented information within textual part of the document, including table description, that makes the context of the table.

Basic table element is cell. **Cell** is the basic grouping within a table. Cells usually contain only one value, word, phrase or concept and are divided by horizontal and vertical lines. **Column** is a set of vertically aligned table cells. **Row** is a set of horizontally aligned table cells.

With respect to structural function, there are three types of table elements:

- **Table descriptors**, which textually describe the table and its data and often provide the table data context. These include table captions and footers.
 - **Title** or **caption** describes the table content and subject.
 - **Footer** provides more detailed information about the table and is usually placed below the table. Footer often presents the legend for symbols used in the table or observations about the table data.
- **Navigational (access) cells** describe and label data cells. Headers, stubs and super-row cells are referred together as navigational cells.

- **Header (column header)** is usually top-most row (or set of multiple top-most rows) of a table and defines the columns' data. In some cases, header does not have to be in the top-most rows, however, it still defines and categorizes columns' data bellow it (e.g. in multi-tables).
- **Sub-header** or **super-row** creates an additional dimension of the table and additionally, describes table data. The sub-header row is usually placed between data rows, separating them by some dimension or concept.
- **The stub (row header)** is typically the left-most column of the table, usually containing the list of subjects or instances to which the values in the table body apply.
- **Table body (data cells)** contains the table's data. Data cells are placed in the body of the table. Cells in the body represent the value of things (variables) or the value of relationship defined in headers, sub-headers and stub.

Described table elements are presented in Figures 1.4 and 1.5.

	Amount of samples	Recall	Precision
Lucid tables	50	0.84	0.97
Complex tables	100	0.92	0.95

Figure 1.4: Table elements: Header, stub and body. Table example source: Yildiz et al. (2005)

Wright produced notable work examining the ease of use of different types and layouts of tables, defining table types by their dimensionality and the way information is presented (Wright 1968, Wright & Fox 1970, Wright 1977). From the dimensionality perspective, Wright defined list tables (one-dimensional tables, containing only the list of items in the table) and matrix tables (containing two dimensions and data arranged in the matrix) (Wright 1968). From the way of the presenting information perspective, Wright differentiate between explicit and implicit tables. An explicit table has information presented in an explicit way, while an implicit table is a table in which a reader has to do something more than just to look for an item (e.g. a reader needs to calculate the value of money in a different currency using the exchange rate presented in the table). From the ease of use perspective, Wright showed that explicit tables are much easier to read than implicit tables. She showed that tables could be quite hard for humans to read and that the easiest tables are explicit list tables. If the values are

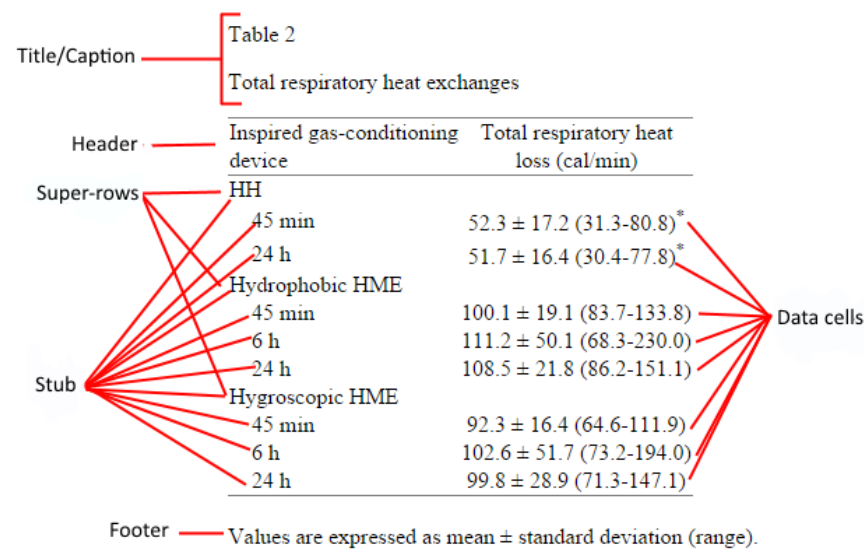


Figure 1.5: Elements of complex table (PMC 29053)

not explicit and the person had to perform additional cognitive processes to find or calculate the value of interest, the probability of mistakes increases and reading speed decrease. Understanding how to use two or multi-dimensional tables proved to be challenging (Wright 1977). However, it improves with training and learning (Wright 1968).

Wright’s experiments showed that making columns within a table readably distinguishable does aid the user. A set of her experiments showed that abbreviations and acronyms in table cells are hindrances and they make table more difficult to read. One of the conducted experiments showed that reading was faster and errors were fewer with the vertical as compared to horizontal tables (Wright & Fox 1970).

Generally, Wright’s research shows that tables are complex structures for humans to understand and the more cognitive operation a user has to perform while reading it, the more a user’s reading will be slow and error prone.

1.2 Tables in biomedical literature

The number of published biomedical research papers is growing exponentially (see Figure 1.6). MEDLINE, a database of biomedical citations, contains over 27 million references from approximately 5,600 journals in 30 languages. In 2015, over 806,000 new citations were added to MEDLINE database¹. On average, over 2,200 scientific

¹https://www.nlm.nih.gov/bsd/stats/cit_added.html

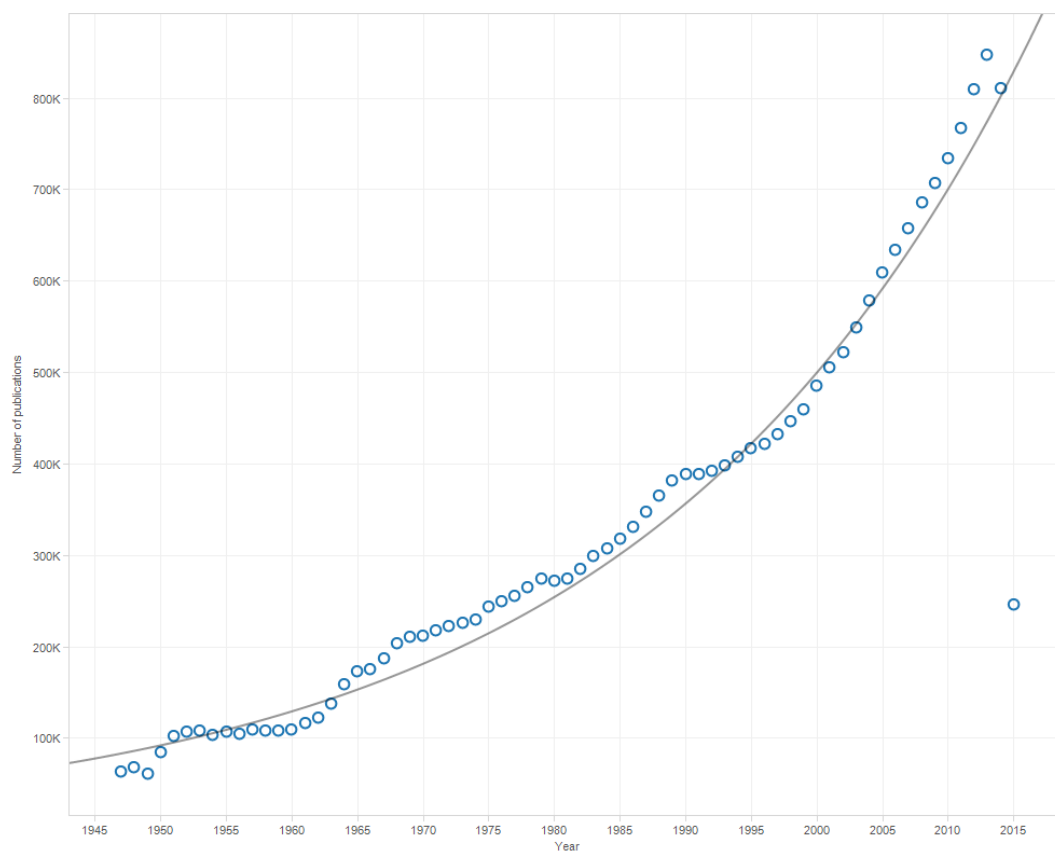


Figure 1.6: Number of published articles by year that are indexed in MEDLINE

papers are published every day in the biomedical domain. It is impossible for the scientists and professionals, who need to keep track with the state-of-the-art in the field, to cope with this amount of published research. Text mining and natural language processing can provide the means to explore this enormous amount of knowledge. The aim of text mining is to process, extract and curate relevant information from the literature. In recent years, a notable progress have been made in biomedical text mining (Cohen & Hersh 2005, Allahyari et al. 2017, Kilicoglu 2017). Text mining can assist with recognising named entities (finding names of diseases, genes, proteins, etc.), linking related entities (finding drug-drug interactions, protein-protein interaction, adverse events to a drug, etc.), text classification (e.g. classify the field of study), retrieving relevant texts or answer questions. However, text mining methods typically ignore lists, tables, and figures. On the other hand, tables are one of the main means of presenting data in scholarly articles. As it can be seen from the example presented in Figure 1.7, tables usually present complementary information to the information presented in the text, but usually there is no repetition. Some information (e.g. experimental

settings, definition of scales, detailed results of the experiments) is presented only in tables (Abeel et al. 2010) and this information would be lost if not processed. Many of these are key information for reproducing, repeating and understanding the presented research. Such information needs to be integrated with other information extracted from research literature, otherwise the meaning can be lost or misrepresented. In order for text mining to be useful, it is necessary for text mining systems to process all components of a research paper in order to gather the same information as a human reader from the same document. Due to tables in the literature and the information stored in them being important for understanding of the literature as a whole, table processing needs to be one of the key components of literature text mining methodologies.

patient was found to have an ovarian primary cancer after enrollment. Patient characteristics for the 95 eligible patients are listed in table 1. The median age was 64. The median ECOG performance score was 1. All patients had a histologic diagnosis of adenocarcinoma. The major sites of metastatic disease were liver, lung, soft tissue and lymph node. None of the patients remain on study; 71 pa-

rolled onto the study were assessable for toxicity. Mucositis, nausea/vomiting and diarrhea were the most commonly observed toxicities and are listed in table 3. The hematologic toxicities are also shown in table 3 and in general were infrequently observed. Overall the regimen was well tolerated.

Page 3 of 7

(page number not for citation purposes)

BMC Cancer 2002, 2

<http://www.biomedcentral.com/1471-2407/2/9>

Table 1 Patient Characteristics

Total Patients	95	100%
Median Age Years (range)	64 (34–84)	
Sex		
Female	49	52%
Male	46	48%
Race		
African American	21	22%
Caucasian	74	78%
Primary Disease		
Colon	82	86%
Rectal	13	14%
Performance Status		
0	35	37%
1	52	55%
2	8	8%

ance score, gender and number of metastatic sites. Female patients, patients with fewer metastatic sites and patients with an ECOG performance 0 had a statistically significant improved survival. Survival by metastatic site is shown in Table 6. Those patients without lung metastasis and those without intra-abdominal metastasis had longer survival times. In the multivariate analysis, presence of lymph nodes, lower LDH levels female sex, better performance status and fewer metastatic sites were statistically predictive of longer survival (Data not shown).

Table 3: Toxicity

Hematologic Toxicity	# of Patients	Grade II (%)	Grade III (%)	Grade IV (%)
Toxicity				

Figure 1.7: Excerpt of the article presenting table and its reference from the text. Text and tables often present complementary, but not redundant information. Information presented in the text are addition to the information presented in the table (PMC113263)

In the PMC database, a database of open access publications in the biomedical domain, more than 72% of research articles, available in XML, contain tables. On average there are 3.1 tables per document, with 80 cells on average. Similarly, drug labels presented in the DailyMed database, maintained by the National Library of Medicine,

contain, on average, 4.1 tables. The statistics detailed here, strongly support the argument that tables are used frequently in the biomedical field.

Tables may appear in various domains and play important role in disseminating information. Table structures are domain independent, however, the content of the table is domain dependent. Certain steps of table processing may be domain-independent, however, in order to fully understand information presented in a table and perform information extraction, it is necessary to have background information on the given domain. Tasks, such as information extraction and knowledge acquisition require domain-specific knowledge sources. Domain-specific vocabularies, topologies, and ontologies can significantly contribute to extracting the relevant information. It was quite extensively invested in developing knowledge resources in the biomedical domain, such as UMLS, SNOMED, PubMed, ATC, etc. (Li et al. 2017, Moore & Holmes 2016, Ofoghi et al. 2014). These resources can be integrated with each other and used to annotate raw text data, which proved to be helpful for information extraction and other semantic processing of textual data. Since semantic resources are well established and developed in the biomedical domain, this domain can be used in order to evaluate and test table information extraction methodology. Biomedical researchers can also benefit from text and table processing, which can result in better treatments and more rapid research on drugs.

Research in table processing was focused on detecting tables in various document formats (Ng et al. 1999, Yildiz et al. 2005, Son et al. 2008) and detecting functional areas (headers, stubs, super-rows and data cells) in tables (Tengli et al. 2004, Silva 2010, Jung & Kwon 2006). Some research has also been done to support table information retrieval (Hearst et al. 2007, Liu 2009), knowledge discovery (Wong et al. 2009), information extraction (Embley et al. 2005, Mulwad et al. 2013) and question answering (Sun et al. 2016). However, most of these studies are limited to the certain set of standardised tables.

Currently, there is lack of concentrated effort to develop a methodology for table mining (consolidating all steps of table processing and information extraction from tables) that can complement text mining methods. The work presented in this thesis focuses on developing a methodology for information extraction from tables in biomedical documents. Biomedical domain was selected because certain parts of information extraction methodology require domain knowledge. Biomedical domain is one of the most vibrant research domain, contributing to the welfare, health and quality of life of people around the world.

1.3 Challenges in automated table processing

Tables store information in a semi-structured manner. They may contain both numeric and textual information, including whole sentences and paragraphs. Processing individual tabular cell content presents, at least, similar challenges to processing textual information. However, since tables structure information within their own perimeters and rely on visual relationships between tabular cells, they pose additional challenges. These challenges include:

Table detection. In many types and formats of documents this task can be challenging (eg. ASCII free text or PDF). The recognition of tables in documents can be either a research goal in its own right, or the first step in an integrated system such as those created for information extraction or information retrieval (Hurst 2000).

Representation for visualization. Tables are primarily used in a way that data can be easily viewed. Most methods and languages that support describing tables, including XML, HTML or LaTeX, are designed with the focus on visualisation. In mark-up languages, tables contain a lot of information about what a table should look like (see Figure 1.8), but very little about how the table entries relate (Thompson 1996, Hurst & Douglas 1997). Since the focus is on visualisation and visual representation, a table author only needs to focus on the visual appearance of the table, ignoring description of the functions of areas or relationships. Therefore, reading and computational analysis of tables described in this manner require a method that is able to disentangle visual structure before further analysis.

Variety of tables structural layouts and visual relationships. There is no “*common*” table structure. The combination of cell arrangement, their spanning, content, and function (headings or data) determine how the table is read and understood. Cells can span over several other cells both horizontally and vertically. Some of the examples of tabular structures are presented in Figure 1.9. Cells in a table are visually related, presenting multiple dimensions and annotations of the data, in contrast to linear textual information. Tables are flexible in their structure, providing authors with means to shape them according to their data presentation needs. Table structure makes automated detection of functional areas (functional analysis of table) and resolving inter-cell relationships (structural analysis of table) challenging. Table layouts and their visual relationships are not specific to any domain. Complex tables can be found

```

<table frame="hsides" rules="groups" class="rendered small default_table">
  <thead>
    <tr>
      <td align="left" rowspan="1" colspan="1">Total Patients</td>
      <td align="center" rowspan="1" colspan="1">95</td>
      <td align="center" rowspan="1" colspan="1">100%</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td align="left" rowspan="1" colspan="1">Median Age Years (range)</td>
      <td align="center" rowspan="1" colspan="1">64 (34-84)</td>
      <td rowspan="1" colspan="1"></td>
    </tr>
    <tr>
      <td align="left" rowspan="1" colspan="1">Sex</td>
      <td rowspan="1" colspan="1"></td>
      <td rowspan="1" colspan="1"></td>
    </tr>
    <tr>
      <td align="left" rowspan="1" colspan="1">Female</td>
      <td align="center" rowspan="1" colspan="1">49</td>
      <td align="center" rowspan="1" colspan="1">52%</td>
    </tr>
    <tr>
      <td align="left" rowspan="1" colspan="1">Male</td>
      <td align="center" rowspan="1" colspan="1">46</td>
      <td align="center" rowspan="1" colspan="1">48%</td>
    </tr>
  </tbody>
</table>

```

Total Patients	95	100%
Median Age Years (range)	64 (34-84)	
Sex		
Female	49	52%
Male	46	48%

Figure 1.8: Example of XML representation and its visualisation (PMC113263)

in any domain. However, domain specific knowledge is useful for detecting functional areas or resolving relationships within the table. Evaluation and creation of new machine learning-based approaches to disentangle visual layouts will be simplified if gold standard annotated corpora existed. Several annotation schemas have been proposed for annotating textual resources and over the time they have been standardised. At the moment there is almost no commonly accepted annotation schema for tables, neither research on how table content should be annotated, while preserving the structure and relationships between the cells.

Figure 1.9: Examples of some possible structures of tables

Use and integration of biomedical knowledge sources. In biomedical domain, there are well-established semantic resources that can help information extraction. These resources are in form of vocabularies, topologies and ontologies. In order to positively contribute to the information extraction task, these resources have to be used and integrated correctly. The use of semantic resources for information extraction from tables in biomedical research is in its beginnings, and it is challenging to select and integrate correctly resources that will help with extracting the variables of interest. The integration of semantic resources has to deal with the specifics of the given resource, but also to take into account the visual and structural relationships between the cells in the table.

Variety of value presentation formats. Values in cells can be presented using various syntactic representation formats. While some authors may present mean and standard deviation in one cell using the plus-minus (\pm) sign (i.e. 12 ± 2), some will use brackets (i.e. $12(2)$) and some will use two separate cells. Extraction of these values requires knowledge of possible value presentation patterns article authors most frequently use in tables. The value presentation formats may be different in different research domains. Often same representation format can be used for presenting different things in different domains. For example, presentation, such as 12 ± 2 , in biomedical literature would usually indicate mean or median with standard deviation, while in computer science domain it may be often used for mean and standard error.

The goal of table mining is to make table information easily accessible and to interpret tables automatically. The process of making published information from literature in various sources structured, managed, searchable and easily accessible in the future, while maintaining value, is commonly referred as data curation (Choudhury 2008). If done manually, data curation is a laborious and expensive task. Automated or assisted curation can speed up curation process by more than 70% and help make information computationally interpretable (Alex et al. 2008).

In order to successfully process table and extract information from them, these challenges need to be addressed.

1.4 Hypothesis and research questions

The main hypothesis explored in this thesis is whether a multi-layered approach to mining information from tables can facilitate large-scale semi-automated extraction

and curation of data stored in tables in the biomedical domain. Specifically, we will aim to answer the following research questions:

1. What levels of processing are needed to disentangle table's internal structure from its visual representation? What levels of processing are necessary for extracting information from tables?
2. What information and knowledge about data is necessary in order to design and implement information extraction from tables?
3. What levels of table processing would benefit from rule-based approaches rather than machine-learning, given the typical short text available in tables?
4. How can table information extraction benefit from domain specificity? Which steps of the methodology are domain dependant and which ones are domain independent?
5. Can the surrounding text that refers to a table help in interpreting the table's data?
6. What levels of accuracy would facilitate efficient data curation on a large-scale to support information extraction?

To demonstrate and validate the approach, we will perform two case studies. In the first case study, we will focus on the clinical trials literature, where tables are often used to present data that detail a trial's settings and outcomes. In the second case study, we will examine whether the developed methodology can be applied for extracting drug-drug interaction from drug label documents.

1.5 Scope

The general scope of this thesis is to investigate a table information extraction methodology in the biomedical domain. Mainly, the aim was to develop a methodology that can be applied for extracting numerical and categorical information from the biomedical literature. Literature mining is important because new research and knowledge is reported in it. However, because of the large amount of published literature, it is necessary to allow easy access to the knowledge using text and table mining. We mainly focus on clinical trial documents as a sub-domain of the biomedical literature. The

clinical trial literature is one of the most relevant sub-fields for drug discovery and disease treatment research (Lalnunpuii 2013). Tables in clinical trial documents may give a valuable insight into general characteristics of a clinical trial such as demographics of the participants, trial arms, names and the side effects of the tested drugs. Clinical trial documents may also contain information about interactions between biomedical substances, such as drug-drug or protein-protein interactions. Clinical trial literature consists of traditional publications and therefore this dataset was used for initial validation of the methodology.

The second biomedical literature sub-domain on which we have evaluated our methodology is the drug label documents, on which we will present a case study demonstrating extraction of drug-drug interactions that are often represented in tables. In the United States, companies are required by law to report potential drug-drug interactions on product labels. The drug labels are presented in a different format and since it is a different dataset, tables may have specific features and challenges that have to be overcome. Because of the differences between these dataset DailyMed was used to evaluate generalizability of the methodology

Biomedical documents are published in a number of formats, including HTML, XML, and PDF. Most well known biomedical domain databases, such as MEDLINE, PMC and DailyMed, keep their documents in an XML format. Our focus therefore is the processing of biomedical documents in the XML format. For other common formats in literature, such as PDF, there are a number of tools for converting documents into XML, such as pdf2xml, pdftohtml, pdfextract, SectLabel, PDFX and easyPDF SDK (Constantin 2014). These tools might be used more or less successfully in a preparatory step for the PDF format. More specifically, we use clinical trial articles from the PMC database for the experiments and evaluations related with clinical domain and drug labels from DailyMed database for experiments and evaluations related with drug-drug interaction from drug labels case study.

1.6 Contributions

The research presented in this thesis has made the following contributions:

- A model of tables and articles that is suitable for table mining, annotation and curation, making tables structured and machine readable, while preserving all necessary information and annotations.

- A method for automated disentangling of table structure. The method that is presented in this thesis covers a wide range of table structures. Disentangling of different table structures makes our information extraction method the first end-to-end methodology for information extraction from tables.
- A step-by-step, end-to-end methodology for extracting information from tables in scientific literature (with the main focus on biomedicine). The methodology has multiple layers, including disentangling the structure of tables, annotating functional areas in the tables, resolving inter-cell relationships, classifying tables by their pragmatics, semantics and finally, extracting information based on manually crafted lexical cues and syntactic rules.
- Modelled common value presentation patterns for the most common numerical variables and created a reusable library of patterns with the semantic mapping of the values.
- A method and reusable rule sets for extraction of baseline characteristics from clinical trial publications and drug-drug interactions from structured drug labels. By doing these case studies, we devised, to the best of our knowledge, the first method for extracting this information from tables.

The described methodology for table disentangling and information extraction are available as open source tools called TableDisentangler² and TableInOut³, which are published under GNU General Public Licence (GPL) v3.

Publications and presentations

Parts of the work presented in the thesis have been published in the following papers:

1. Milošević, N., Gregson, C., Hernandez, R. and Nenadić, G., 2016, June. Disentangling the Structure of Tables in Scientific Literature. In International Conference on Applications of Natural Language to Information Systems (pp. 162-174). Springer International Publishing. DOI: 10.1007/978-3-319-41754-7_14
2. Milošević N., Gregson C., Hernandez R. and Nenadić G. (2016). 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

²<https://github.com/nikolamilosevic86/TableDisentangler>

³<https://github.com/nikolamilosevic86/TabInOut>

Joint Conference on Biomedical Engineering Systems and Technologies ISBN 978-989-758-170-0, pages 223-228. DOI: 10.5220/0005660102230228

3. Milošević N., Gregson C., Hernandez R. and Nenadić G. (2016). Hybrid methodology for information extraction from tables in the biomedical literature. In Proceedings of Belgrade BioInformatics conference – BelBi 2016, pages 74–78, ISBN: 978-86-7589-108-6

Parts of the work from this thesis were also presented in the following conferences/events (abstracts):

- **Poster:** *Extraction of drug-drug interactions from drug product labeling tables* presented at AMIA Joint Summits in San Francisco, March 27 - 30, 2017
- **Poster:** *Supporting clinical trial data curation from literature using table mining* – presented at the Postgraduate Summer Research Showcase at the University of Manchester, June 2015
- **Oral presentation:** *Supporting clinical trial data curation and integration with table mining*, presented at FARR Institute International Conference on Data Intensive Health Research and Care in St. Andrews, August 2015

1.7 Thesis structure

The thesis has been organised into nine chapters. The background research review of the field of table mining and table understating that was conducted so far is presented in the next chapter. The third chapter (Methodology Overview) gives a brief overview of the models and methodology presented in this thesis. The fourth chapter gives a detailed explanation of the model of tables, including the data model for persevering and curating tables. The same chapter also discusses details on functional analysis of tables and structural analysis of relationships between cells. Chapter 5 presents pragmatic table analysis and semantic tagging of the table content. In the Chapter 6, we examine and compare two possible approaches for cell selection (machine learning-based and heuristic-based approach) and syntactic analysis of the cell content. At the end of this chapter, we generalise the findings into a framework and multi-layered methodology for information extraction. The seventh chapter presents two case studies of information extraction from tables using the previously described framework. The eighth chapter discusses the models, the approaches presented and answers the

research questions. The last chapter concludes the thesis and gives an outline of the future research.

Chapter 2

Review of table mining literature

Tables have been studied from various perspectives, including their creation and editing (Wang & Wood 1993, Long 2010), ergonomics, table models and mining. In this chapter, we review research describing the proposed models of tables and automated processing methodologies, such as table detection, functional table analysis, table information retrieval, information extraction, knowledge discovery and question answering.

2.1 Table description models

Tables are viewed and manipulated for several different purposes (i.e. creating/editing, reading, mining). This have led to the specific models that give an insight about important tables characteristics from these viewpoints. A table model is a representation of organisation (layout), structure and content of tables.

Tables can be considered at three levels of description: abstract, physical and logical (Long 2010).

- **The abstract level** encapsulates the communicative intent of the author (i.e. relationships between the data) (Wang & Wood 1993).
- **The physical level** consists of pixels, lines, and text located in documents or other display devices. They are referred to as layout structures of tables (Haralick 1994, Hurst 1999).
- **The logical level** describes the arrangement and content of the table elements. Tables at the physical level are usually described and created on a logical level

using some descriptive language, such as HTML, XML or LaTeX (Haralick 1994, Nagy 2000, Wang & Wood 1993).

There may be a number of models for each table description level. At the abstract level, tables can be viewed as a set of functions associating labels in a table (Long 2010). The label was defined by Wang & Wood (1993) as the access cell, which was later adopted by other authors in the area. The abstract level of description is usually used for a linear table representation and access. It can be used for both creation and mining. Wang & Wood (1995) defined an abstract model for table editing and formatting using mathematical and logical representations and operators. Similar models were discussed by other authors (Embley et al. 2006, Douglas et al. 1995, Hurst & Douglas 1997). Some authors refer to this form of tables as canonical forms (Embley et al. 2006).

At the logical level, tables can be modelled by trees, graphs, grids and hierarchies. Wang and Wood's model keeps the logical structure of multi-dimensional tables by defining a table item as a node in the tree. Each data cell is represented as a leaf node of the tree, identified by the sequence of nodes representing access cells or labels (Wang & Wood 1993). Nagy & Seth (1984) represented tables as X-Y trees. An X-Y tree is a top-down method for page layout analysis. The basic assumption is that layout elements are generally laid out in a rectangular blocks. Blocks are grouped and adjutant to each other, having one dimension in common. The document is split into successive smaller blocks by making cuts along white space areas. The result can be presented in a tree where the root is the whole document, while nodes represent smaller blocks of the document (Cesarini et al. 1999). Nagy and Seth's X-Y cut algorithm uses recursion to split tables by alternating horizontal and vertical cuts along the column and row delimiters. The graph model directly encodes geometric relationships between the cells. Cells are represented as vertexes with an edge between neighbouring cells (Amano & Asada 2002, 2003). The type of edge represents the relationships between cells (e.g. adjutant cells, spanning cells with different height, etc.). Alternatively, a grid, where spanning cells are split into the size of the smallest cell in the table and content is typically copied into newly formed split cells, can also represent tables. By doing so, the table is represented as a perfect grid. Cells that were split are called virtual cells, while the original cells are called real cells (Ramel et al. 2003). This model is usually used for processing, rather than table visualisation since it simplifies the structure without modifying data. However, it modifies the original, visual representation of the table.

A number of table description models at the logical level have been standardised.

The Organization for the Advancement of Structured Information Standards (OASIS) created a standard to represent tables in SGML/XML documents called the CALS Model (Bingham 1995). The CALS model was designed to handle a variety of military technical documents. It allows encoding of geometric and formatting features such as cell alignment, borders, and cell orientation. It also allows for splitting tables into multiple sections, such as the heading area, the body area and the footer area with the use of specific tags. It also allows naming cells, columns and rows, so they can be referred to not only by indexes but also by a name. However, ambiguities were identified in CALS' semantic representation, which led to several interoperability problems between the vendors (Severson & Bingham 1995). As the industry moved in 1998 towards the XML format, OASIS switched its focus to this format, incorporating certain elements of the CALS table model standard (Walsh 1999) to the XML, which became the most popular format for table serialisation, since it allows encoding the geometric structure, as well as functions of table's entities (Zanibbi et al. 2004). Also, it allows easy transformation to HTML which is suitable for visualisation and higher level processing (Hurst 2003). The example of XML table representation can be seen in Figure 2.1.

<pre> ▼<table-wrap id="tbl1" position="float"> <label>Table 1</label> ▼<caption> <p content-type="table-title">Baseline demographic and disease characteristics</p> </caption> ▼<table border="1" frame="hsides" rules="groups" width="85%"> ▼<colgroup> <col align="left"/> <col align="char" char="."/> </colgroup> ▼<thead valign="bottom"> ▼<tr> <th align="left" charoff="50" valign="top">Number of patients enrolled</th> <th align="center" char="." charoff="50" valign="top">21</th> </tr> </thead> ▼<tbody valign="top"> ▼<tr> <td align="left" charoff="50" valign="top">Median age (range)</td> <td align="center" char="." charoff="50" valign="top">57 (36–2) years</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">Sex</td> <td align="center" char="." charoff="50" valign="top">15</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">Performance status</td> <td align="center" char="." charoff="50" valign="top">5</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">1</td> <td align="center" char="." charoff="50" valign="top">13</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">2</td> <td align="center" char="." charoff="50" valign="top">3</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">Brain metastasis previously resected.</td> <td align="center" char="." charoff="50" valign="top">6</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">Male</td> <td align="center" char="." charoff="50" valign="top">15</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">Female</td> <td align="center" char="." charoff="50" valign="top">6</td> </tr> ▼<tr> <td align="left" charoff="50" valign="top">...</td> <td align="center" char="." charoff="50" valign="top">...</td> </tr> </tbody> </table> </pre>																							
<p>Table 1</p> <p>Baseline demographic and disease characteristics</p> <table> <tr> <th>Number of patients enrolled</th><th>21</th></tr> <tr> <td>Median age (range)</td><td>57 (36–2) years</td></tr> <tr> <td>Sex</td><td></td></tr> <tr> <td>Male</td><td>15</td></tr> <tr> <td>Female</td><td>6</td></tr> <tr> <td>Performance status</td><td></td></tr> <tr> <td>0</td><td>5</td></tr> <tr> <td>1</td><td>13</td></tr> <tr> <td>2</td><td>3</td></tr> <tr> <td>a</td><td></td></tr> <tr> <td>Brain metastasis previously resected.</td><td></td></tr> </table>		Number of patients enrolled	21	Median age (range)	57 (36–2) years	Sex		Male	15	Female	6	Performance status		0	5	1	13	2	3	a		Brain metastasis previously resected.	
Number of patients enrolled	21																						
Median age (range)	57 (36–2) years																						
Sex																							
Male	15																						
Female	6																						
Performance status																							
0	5																						
1	13																						
2	3																						
a																							
Brain metastasis previously resected.																							

Figure 2.1: XML representation of table (as used by PMC) and its HTML visualisation in a web browser (PMC 2410054)

Hurst (2000) did an extensive study of tables, table understanding and information extraction from tables. He proposed a model of tables, which has five components:

- *Graphical* – representation that describes how the table is rendered on screen, e.g. bitmap,
- *Physical* – a description of the table in terms of physical relationships between its basic elements when rendered on the page,
- *Functional* – the purpose of areas of the table with respect to the use of the table by the reader,
- *Structural* – the organization of cells as an indication of the relationships between them,
- *Semantic* – the meaning of text in the cell, the relationship between the interpretations of cell content, and the meaning of the structure in the table.

In the functional component of the table model, Hurst defined two types of cells — access cells (navigational cells such as headers and stubs) and data cells (containing table data). Hurst defined a **reading path** as a path which reader takes through an array of cells when using the table to locate or read a particular piece of information. The model developed in this thesis used Hurst’s model as a baseline, additionally specifying and extending it.

Doush and Pontelli proposed an ontology of spreadsheets’ components that contains an ontological model of tables. Their ontology was developed in order to help their system for non-visual navigation through spreadsheets for visually impaired people. The ontology describes tables and different kinds of cell components (header, title, data, empty cell) with relationships between these components (Doush & Pontelli 2010, 2013). However, certain table elements, such as super-rows, were not part of the table ontology. The ontology can be seen in the Figure 2.2. The presented ontology served as a baseline for a data model for disentangling tables and storing information from the tables in a manner that machines can process, described in Chapter 4.

The presented models of tables are the base of the approaches to detect, disentangle table structure and facilitate further mining of information in tables. However, most of these models do not capture all table types (e.g. Wright’s model captures only list and matrix tables, but not multidimensional tables) and all table elements (often some element such as super-rows or footers are not included in the model because the approaches were designed for a subset of simple tables or it does not appear in a given domain/document type). Models for computational representation are often focused on visualisation, and lack features for capturing semantics of the information presented in

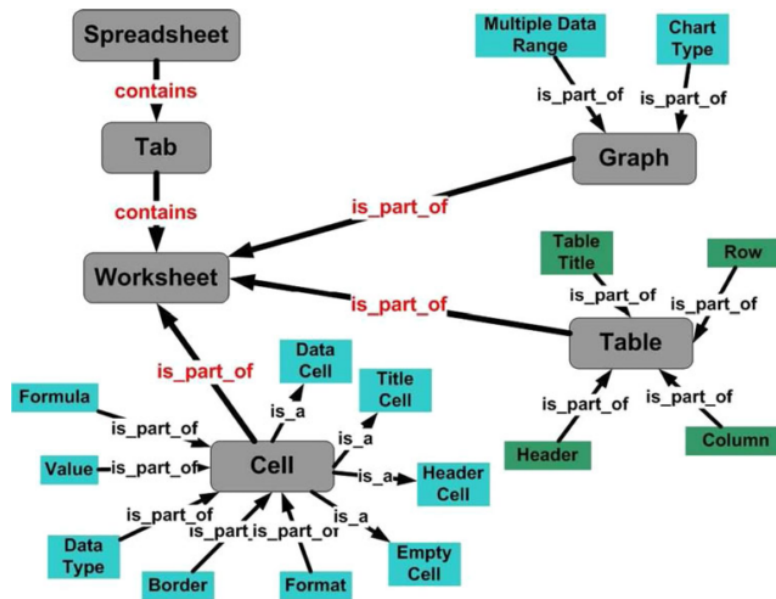


Figure 2.2: Table ontology as presented by Doush & Pontelli (2013)

the table and their relationships. There is still no consensus about a complete table model, as models usually lack certain table elements (such as super-rows) or features for semantic table description or annotation.

2.2 Table processing workflow

Similarly to natural language processing, which is a complex task requiring lexical, syntactic and semantic analysis (Ananiadou & McNaught 2006), table processing is a multi-layered task. If natural language is used in a table's content, tables require similar processing layers as natural language. However, additional layers are necessary in order to process and disentangle the structure of tables.

Hurst (2000) identified four layers of table processing:

- Table detection – locating tables in the document.
- Functional analysis – classifying areas in the table according to their functions (e.g. header, stub, data cell).
- Structural analysis – inferring relationships between the cells.
- Semantic analysis – understanding the information presented in the table (e.g. information extraction, information retrieval, question answering, etc.).

Rastan et al. (2015) proposed a TEXUS system with a model that views table processing as a set of consequent tasks: *Document Converting*, *Locating*, *Segmenting*, *Functional Analysis* and *Structural Analysis*. In their model, they grouped tasks into table extraction tasks (document converting, locating and segmenting) and table understanding tasks (functional analysis and structural analysis). TEXUS does not include semantics in their workflow, since their goal was to transform tables to Wang’s abstract representation (Wang & Wood 1993).

In this thesis we will use the workflow defined by Hurst, since the workflow defined by TEXUS can be mapped to Hurst’s workflow (locating = table detection, segmenting+functional analysis = functional analysis, structural analysis = structural analysis). TEXUS adds document converting to the workflow that may be necessary for certain document formats. The workflow proposed by Hurst also includes semantic analysis that can be applied to structurally analysed and disentangled tables in order to retrieve or extract information, answer questions, or discover knowledge.

2.3 Table detection

The first challenge in table processing is to detect tables in documents. This task can be challenging in some formats, like PDF, HTML web pages and ASCII text documents. Data from PDF files can be obtained using Optical Character Recognition (OCR) (Kieninger & Strieder 1999, Rus & Summers 1994, Green & Krishnamoorthy 1995, Chandran & Kasturi 1993) or by interpreting binary content of the file (Yildiz et al. 2005, Constantin et al. 2013). HTML documents often use the <table> tag. Unfortunately, not all elements that start with <table> tags are genuine tables that contain structured data, as the same tags are often used for HTML page formatting. In free text documents, tables are structured by using empty spaces or set of special characters (see Figure 2.3).

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.840	0.011	0.977	0.840	0.903	0.862	0.962	0.953	findings
	0.600	0.129	0.714	0.600	0.652	0.493	0.807	0.735	settings
	0.791	0.240	0.586	0.791	0.673	0.514	0.892	0.751	support-knowledge
Weighted Avg.	0.741	0.121	0.768	0.741	0.746	0.629	0.886	0.816	

Figure 2.3: Example of three ASCII free text tables: Generated classification output from the Weka toolkit

Table 2.1 presents the main approaches to table detection with associated citations,

which are discussed below.

2.3.1 Table detection in text (ASCII) documents

Table detection in free text ASCII documents was prevalent at the end of the 1990s and beginning of 2000s. Later, with the emergence of rich text document formats, markup formats, and portable document format (PDF), ASCII documents ceased being used for presenting semi-structured information. However, parts of the approaches developed for detecting tables in ASCII documents are useful also in other document types. The presented approaches consist of machine learning algorithms or heuristics.

Ng et al. (1999) used two machine learning algorithms – C4.5 decision trees and the back-propagation algorithm for artificial neural networks with nine features for table boundaries detection. Each feature has its representation for previous, current and next horizontal line. Therefore, each training instance consists of a set of 27 feature values. Their system could also detect if some vertical line is the first line of a column, within the column, last line of the column, or outside any column. As a dataset for testing their approach, they used Wall Street Journal news. A similar approach was used for the horizontal lines and rows. Their features include the number of white spaces in the line, the number of leading spaces, the number of segments with multiple continuous space characters, whether a cell contains special characters, alphanumeric characters, etc. C4.5 provided higher accuracy on recognizing table boundaries and columns, while back-propagation performed better at recognizing table rows. Silva et al. (2003) used C4.5 decision trees to detect tables in both PDF (converted to ASCII using the pdftotext Linux utility) and ASCII documents presenting companies financial statements. They implemented a similar approach to Kieninger (1998), Pyreddi & Croft (1997), Ng et al. (1999), that used alignment of characters, words, non-space characters, and white spaces as features, but they also used a number of each line's inner spaces as a feature for machine learning algorithm. However, they also used two heuristics to improve their algorithms: no matter how many inner spaces a line has, it is not in a table if the lines around it are plain text; on the other hand a line with no inner spaces is not plain text if table lines surround it. They reported recall of 99,4%, but precision and F1-score were not reported.

Heuristics were also used in combination with statistical cross-correlation to detect tables (Pande 2002). Cross-correlation is a standard statistical signal detection procedure that is useful for determining the similarity of two signals. Pande (2002) explored the cross-correlation concept as a way of computing similarity measures between lines

Approach	Citation	Description	ASCII	XML	PDF
Heuristic	Chandran & Kasturi (1993)	OCR detecting line			X
	Green & Krishnamoorthy (1995)	structures of the document			
	Rus & Summers (1994)	OCR and space density graphs			X
	Kieninger & Strieder (1999)	OCR detecting word blocks in same logical unit	X		X
	Pande (2002)	Cross-correlation between lines	X		
	Yildiz et al. (2005)	Heuristics about position of elements			X
	Gatterbauer et al. (2007)	CSS2 visual box and visual pattern recognition		X	
	Fang et al. (2011)	Element position heuristics			X
	Nurminen (2013)	PDF transformed to grayscale image and then edges detected			X
	Constantin (2014)	Graphical model of article elements			X
	Constantin et al. (2013)				
Machine learning	Kasar et al. (2015)	Attribute relational graph based on regular expressions			X
	Ng et al. (1999)	ANN and C4.5, features about characters and white spaces	X		X
	Kieninger (1998)				
Hybrid	Pyreddi & Croft (1997)				
	Wang & Hu (2002)	SVM with structural, content type and word group features		X	
	Son et al. (2008)				
Hybrid	Liu (2009)	Element positions, caption string matching, font size, matching heuristics and ML (CRF and SVM) to detect sparse lines (table rows)			X
	Silva (2010)	Combination of HMM, Decision trees and heuristics			X

Table 2.1: Approaches to table detection

of plain text or between aggregates of lines. Their research have been performed on a general domain, on documents and books stored in Odessa digital library.

2.3.2 Table detection in PDF documents

Since the initial release of the portable document format (PDF) in 1993 (Bienz et al. 1993), computational representation of tabular data moved towards this format. At the physical level, we can observe some of the heuristics which can distinguish data cells from access cells, such as font type, font color, cell background color, cell spanning, alignment, line art, etc (Hurst 2000). However, PDF is a challenging format to extract structures from, as it normally does not save the structure of their graphical objects (Corrêa & Zander 2017).

Pdf2table recognises tables in PDF documents by using pdf2html. Pdf2html returns PDF elements in XML with the position of these elements. Pdf2table uses obtained XML information and extracts tables. Tables are extracted from PDF documents using heuristics about content positions. Since the approach makes errors, the authors created a graphical user interface which gives the user the ability to make adjustments on the extracted data (Yildiz et al. 2005).

Similarly, pdfx is a rule-based system designed to reconstruct the logical structure of scholarly articles in PDF (Constantin et al. 2013). The system output is an XML document that describes the logical structure in terms of title, sections, figures, tables, references, etc. The system carries a two-stage process in which firstly, it constructs a geometrical model of the article's content to determine the spatial organization of textual and graphical elements and then identifies logical units of discourse based on their discriminative features (font styles, text positioning, lists of cue words, contextual features, etc.) (Constantin 2014). This system is successful in recognizing tables, but the transformation of the table into the XML format faces a number of challenges. For table extraction and transformation they reported F1-score of 13.27% to 57% depending on the dataset used (Elsevier, PMC or Luong et al.).

One of the approaches to detect tables in PDF documents is by using OCR. A number of studies use OCR to recognize tables using white space density graphs (Rus & Summers 1994) or by detecting line structures in documents (Green & Krishnamoorthy 1995, Chandran & Kasturi 1993). Alternatively, the T-Recs system is able to read both ASCII, PDF or paper scanned documents. This approach detects word-block clusters and look for the words that belong to the same logical unit. T-Recs identifies horizontally overlapping words that form the rows of the tabular structure (Kieninger

& Strieder 1999).

Approaches for table detection were compared in a table competition that was organised during the 12th International Conference on Document Analysis and Recognition (ICDAR 2013). The goal of the competition was to objectively compare state-of-the-art techniques in table recognition in a standardized way, across several input formats, including PDF. The competition asked participants to return the bounding boxes of the table as well as information about the content, and location (column and row number) of every cell in the table. Seven academic systems (Nurminen 2013, Fang et al. 2011, Liu 2009, Yildiz et al. 2005, Silva 2010, Stoffel et al. 2010, Gobel et al. 2013) and four commercial products were submitted. The reported accuracy range for table recognition was between 58.5% and 87.7% (Gobel et al. 2013). Several of these approaches used positions of elements in a PDF document to detect graphic ruling lines and white space delimiters (Fang et al. 2011, Yildiz et al. 2005, Stoffel et al. 2010). Nurminen (2013) converted PDF into a grayscale image and used image processing (comparing adjutant pixels) in order to find delimiters and tables. Silva (2010) used machine learning (HMM and decision trees) in combination with heuristics to classify lines of the document and locate tables.

Kasar et al. (2015) proposed a methodology in which the user first specifies a set of key fields in the document image (commonly PDF). These fields are transformed into graphs, where nodes represent cells with their features and edges represent spatial relationships between them. Graph matching based on regular expressions and cosine similarity with the content selected by the user is used to find other similar structures on the document image. The extracted graphs are analysed to find the borders of the overall table structure. The authors reported pattern retrieval precision and recall of 97.1% and 94.99% and cell recognition accuracy of 99.3%. However, this approach heavily depends on the content of the tables and similar tables are extracted only when the parts of the content are similar.

2.3.3 Table detection in XML and HTML documents

As structured logical representations, XML and HTML documents provide tags that describe tables in the documents. However, these tables are not always used for presenting data and in many cases are used to describe the layout. As a result, there is a necessity to differentiate tables that present data and information from tables that format the page layout.

For solving this task, machine learning and heuristic methods were developed. Machine learning methods typically used Support Vector Machines (SVM) in order to discriminate data tables from the formatting ones (Wang & Hu 2002, Son et al. 2008). For example, Wang & Hu (2002) used three feature groups (structural, content type and word group). Structural features were the average number of columns, a standard deviation of the number of columns, the average number of rows, standard deviation of the number of rows, the average cell length, standard deviation of cell length and the average length consistency. They modelled cell content into seven content types (Image, Form, Hyperlink, Alphabetical, Digit, Empty and Others) and their features included a histogram of content type for a given table and the average content type consistency. The third feature group was word group features where they calculated TF-IDF measure with some adjustments. Son et al. (2008) used structural and content features, with two classifiers - one for each group of features. They found that structural features were more important than content type and word group features.

An interesting approach focused on analyzing tables using visual representation similar to ones that browsers use to draw tables (taking into account CSS and how people view tables) using the CSS2 visual box (Gatterbauer et al. 2007). The authors argue that it is difficult to detect tables from web documents only by analysing tree-like HTML structure and that not all tables are inside `<table>` tags. Their approach was to render pages and analyse visual two-dimensional patterns of the rendered page by the set of heuristics in order to detect tables. They reported 81% recall and 68% precision for table detection. However, the execution time needed to render the tables and analyze them was over 5 seconds per table.

2.4 Functional table analysis

The second step in table processing is the functional analysis of cells. During this step, the data area should be distinguished from the header areas and other access cells. This is a challenging task, in particular when dealing with complex tables that have several subheads (super-rows) and/or multi-row headers.

Approaches to this task consider using heuristics or machine learning classification techniques. Most of the approaches recognize headers but often fail to recognize super-rows and stubs (column heads). Table 2.2 lists some relevant approaches to functional analysis of tables, which are discussed below.

Approach	Citation	Description	ASCII	HTML	PDF
Heuristic	Hu et al. (2000 <i>a</i>) Hu et al. (2000 <i>b</i>)	Detecting headers based on spacial and syntactic rules	X		
	Tanaka & Ishida (2006)	Formal representation given by humans generalized into RDF		X	
	Embley et al. (2016)	Block algebra and cell constraints		X	
Hybrid	Tengli et al. (2004)	Decision trees and heuristics about empty cells		X	
	Silva (2010)	Uses set of machine learning algorithms to determine the function of each line			X
Hybrid table classification	Jung & Kwon (2006)	C4.5 decision tree and heuristic rules to find genuine tables. Set of heuristics to find header		X	
Heuristic header detection	Chavan & Shirgave (2011)	Uses combination of rules and decision trees to find genuine tables. Then it uses a set of rules (about font and additional tags) to find headers		X	
Machine learning	Wei et al. (2006)	CRF to annotate function of the line	X		

Table 2.2: Approaches to functional analysis (mainly head detection)

Tanaka & Ishida (2006) present a heuristic approach that gives a formal representation of generalized table structure based on adjacency of cells and iterative structures. The approach is based on human interpretation of table structure, table generalization and relation extraction from tables. Once humans interpret the table structures, the table structures are automatically generalized. In this approach, users provide the structure of the table and description of the relationships. The algorithm then extracts data from the table following given descriptions that use a set of RDF statements describing the relation between data in a structure. The RDF structure is then represented as a connected table and generalized by finding repetitive blocks.

For recognition of table structure from ASCII documents, hierarchical clustering was used to identify a likely grouping of words and build a binary tree representation of the table (Hu et al. 2000*a,b*). The algorithm identifies potential headers based on spatial (the header for each column is roughly aligned with the column; hierarchical headers are placed such that the high-level header is above its subsidiary headers and centred horizontally with regard to the columns represented by the subsidiary headers) and simple syntactic rules (every phrase in a header line must be associated with at least one column; if a phrase in a header line is associated with more than one column, then each subsidiary column must already have its own header assigned). Based on the detection of cell function, they built a directed acyclic graph representation of the table, which they were able to visualize and edit in a graphical user interface.

Tengli et al. (2004) used a machine learning approach in order to differentiate navigational cells (access cells) from data cells on tables collected from universities websites (using the Common Data Set, CDS tables). The features used were cell similarity, the number of cells and type of cells. They also extracted headers, indexed them and, if the relative string-edit-distance was less than a set threshold, they merged them together. They used learned labels for header detection but also used some heuristics and transformations for detecting super-rows. For example, if the row contained empty cells and before and after it there were non-empty data cells, or a row contained just one non-empty cell – it was assumed to be a super-row label. They reported 91.4% F1-score. However, the Common Data Set tables are relatively standardised and Universities have to follow given guidelines for present information in these tables. Therefore, they are not representable set for possible table layout structures.

Jung & Kwon (2006) presented an example of a hybrid approach to functional analysis that filters easy detectable non-genuine tables by using a set of heuristics about empty rows, tables without rows or columns, one-dimensional tables and fraction of

hyperlinks and images. After this filtering, machine learning is applied with a set of features such as presence of <caption>, <th> tags, border options, inner tables, numeric data, fraction of empty cells, fraction of cells including , <a>, <input> tags, fraction of cells containing text, symbols, numeric data and probability of the presence of the header, etc. Consistency features like the standard deviation of the number of columns, the number of rows, length of columns and rows are also used. Priority weighting is assigned to these tags, so if some cells contain more of them, the probability of that cell belonging to the header will be higher (Chavan & Shirgave 2011). Cell similarity can be an indication of the header since headers are the origin of the columns' similarity and sets the pattern of the body cells' content that is followed in the column. Jung & Kwon (2006) reported 95% F1-score for distinguishing table types and 82.1% accuracy in extracting table headers from genuine tables.

2.5 Schema matching

Schema matching is a field of research that aims to generate correspondences between the elements of multiple information schemas. Schemas could be in any format (XML, SQL, ontology, etc.). The goal is to find semantically equivalent or related elements in the other schemas. The field of schema matching for tables is related to functional analysis because in order to find a match between table and database or ontology schema, it is important to recognize the table headers. Also, some of the work extends schema matching into the applications of information extraction or question answering domains. A number of approaches have been proposed, including linguistic matching, thesauri, structure or graph based matching (Bernstein et al. 2011). Several authors proposed approaches to match tables, usually from XML, HTML or spreadsheet into the structured or linked data database (Cafarella, Halevy, Wang, Wu & Zhang 2008, Shigarov 2015, Embley et al. 2016).

WebTables (Cafarella, Halevy, Wang, Wu & Zhang 2008, Cafarella, Halevy, Zhang, Wang & Wu 2008) system attempted to improve the relevance of search and improve database management systems by extracting tables from web pages into a relational database. Their system filters non-relational tables using machine learning and classifies tables with headers. For tables that do not contain a header, they used a reference-matching technique which tries to find a label for the column based on the data and previously extracted tables with headers. Since each table can have its own schema of labels (headers), each table is considered as a relational database, with labels (headers)

and an array of tuples (data). They developed a keyword search over the corpus of 154 million tables that provided higher relevance than solutions based on traditional information retrieval techniques.

Shigarov (2015) created a rule engine (called CELLS) that utilizes spatial, typographical and natural language information in order to transform information from tables in a spreadsheet format into a relational database. Embley et al. (2016) used block algebra together with a cell constraint model to describe table layouts in order to determine headers and extract table data from HTML and spreadsheet formats into the SQL or RDF structure suitable for querying. Unlike the majority of approaches, they considered both row and column headers for their recognition task.

Mulwad et al. (2010, 2013) utilize knowledge bases in order to cluster the entities into the labels that are likely to describe them and use semantics to map web tables into the relational or semantic schemas.

2.6 Information retrieval from tables

Information retrieval considers a task of finding and presenting relevant information to the user. Search engines, such as Google, Bing or PubMed, are examples of information retrieval systems. The majority of popular search engines lack specific support for tables, since tables present a challenge because of diverse media formats, different table layouts, cell types or value presentation patterns (Liu 2009).

The BioText Search engine (Hearst et al. 2007, Divoli et al. 2010) retrieves information from texts, abstracts, figure captions and tables from PubMedCentral. Different indexing weights are assigned to the various document elements (title, text, abstract, table captions, table data; figure and image captions). If a table search is selected, different indexing weights are used compared to when the text search is chosen. For table search, maximal weights are assigned to table captions and table data, while article text and image captions have a low weight.

Liu (2009) created a system called TableSeer that is focused on indexing and ranking tables in scientific articles in PDF format using OCR and proposed an algorithm called TableRank. TableRank considers multiple features of a table and the document it appears in, and aggregates these features to determine the final ranking of the table with respect to a query. The approach consists of five elements: a table crawler, a table metadata extractor, a table metadata indexer, a table-ranking algorithm, and a table search query interface. In summary, TableSeer crawls scientific documents from

the digital libraries, identifies the documents with tables, detects each table using a document page box-cutting method (an OCR method), extracts the metadata for each identified table, ranks the matched tables against the end user's query with the TableRank algorithm, and displays the ordered results in a user-friendly interface. TableRank tailors the traditional vector space model to rate the $\langle \text{query}, \text{table} \rangle$ pair by replacing the document vectors with the table vectors. It uses the standard TF-IDF and cosine similarity measure. However, parts of tables in the table vector are weighted differently; for example, a table title would have higher importance than table data during the search. TableRank also uses some document features, such as the number of citations, impact of a journal or conference and publication year in order to rank the results. Compared to BioText, TableSeer returns whole documents with relevant tables, while BioText extracts and shows only potentially relevant tables.

2.7 Table mining applications

One of the challenging tasks for table processing and developing table mining applications is the extraction of data and its relationships from the table. Relationships, especially in complex tables can be fuzzy and ambiguous.

Several approaches have been designed for information mapping and storing information from tables. Information from tables can be mapped to predefined, structured database/knowledge sources, or stored in the form of attribute-value pairs. Processed table information allows development of information retrieval, information extraction, knowledge discovery or question answering applications.

The main methodologies for development of table mining applications are:

- Heuristic and knowledge-driven – uses a knowledge source and, typically, a set of heuristics defined by experts.
- Machine learning driven – learns to process and extract information from previously annotated data.

2.7.1 Knowledge-driven approaches

A number of information extraction approaches used heuristics in combination with ontologies to extract instances of classes (Embley et al. 2005, Quercini & Reynaud 2013, Hignette et al. 2009, Jannach et al. 2009). For example, Embley et al. (2005)

used an ontology to recognize expected attribute names (header categories) and data values from tables. They used specific domains of car advert and phone sales tables. Using the ontology, it was possible to identify table attribute (navigational) cells and associate them with value cells. If the value cell is empty, this method can distinguish whether the cell is missing or the cell has a value based on internal factoring by observing a pattern of empty cells in a column. However, their approach only works for top-level tables whose attributes are at the top of columns, while their approach is not able to deal with complex tables (Embley et al. 2005).

Hignette et al. (2009) used fuzzy similarity measures (each cell is annotated with multiple concepts sorted by relevance) to annotate table relations (annotate cells, then columns, and at the end relations between columns) based on a pre-determined domain ontology. The ontology can be instantiated using tables on the web and a set of heuristics for table structure and tokens in a table (Jannach et al. 2009). Each application of these approaches needs to have its own ontology to annotate table data.

Tables can also be interpreted using existing Linked Data Knowledge Bases. Mulwad et al. (2010, 2013) presented a method that comprises four steps. In the first step, ontology classes are associated with columns. According to Mulwad et al. (2010), in a typical, well-formed table, each column contains data of a single syntactic type (e.g. numbers) that represent entities or values of a common semantic type (e.g. number of people, yearly salary in US dollars, etc.). The column's header, if present, may name or describe the semantic type. The approach is to map each cell value to a ranked list of classes and then to select the one which best characterizes the entire column. To utilize this approach, the Wikitology knowledge base was queried. In the next step, the algorithm links table cells to entities from the Linked Open Data cloud (the Linked Open Data cloud was used for annotating and normalizing entities in the cells). In the third step, the algorithm tries to identify relations between table columns by generating a set of candidate relations from the relations that exist between the concepts associated with the string in each row of at least two columns. DBPedia was queried to identify relations. In the final step, they developed a template for annotating and representing tables as linked RDF. They reported that 66.12% of table cell strings were correctly linked. The algorithm performed quite well in linking Persons (83%) and Places (80%), but quite poorly in linking data like movies, nationality, songs, types of business and industry. This may have been due to the sparseness of data for these types of entities in the knowledge base. This approach works for tables that Mulwad et al. (2010, 2013) call well-formed (having a category per column with category definition

or label in the first row). Well-formed tables are a subset of tables and present quite restricting definition of tables and therefore this term or definition is not used further in this work.

Limaye et al. (2010) also used knowledge sources (e.g. YAGO, DBPedia) to annotate cells, columns and relations between columns, with the help of machine learning and word similarity measures, such as cosine similarity. A similar approach by Quercini & Reynaud (2013) used knowledge sources for annotated named entities (such as restaurants, theatres, museums) in tables but in case an item was not known, a search engine was queried for that item. They retrieved snippets from the search engine and classified them using a classifier in order to retrieve named entity class.

A small seed ontology can be enriched using tables, which was shown with the example of geopolitical entries from tables (Tijerino et al. 2003). In this approach, relatively simple tables were analyzed, entities were extracted and top-level entities were matched with entities in the seed ontology.

In order to improve a heuristic method that uses ontologies and thesauri for information extraction, tables can be simplified to a single standard type that contains only one header row (Wang 2013). If some cells contain column-spanning or row-spanning, those cells are appended to the next header cells (over which it spans) and deleted. However, this approach does not involve header detection and simply appends spanned cells to the following ones. The extraction algorithms read the properties from a first table row and look up in the ontology. If the property is found in the ontology, the values are stored in the database. If not found in the ontology, the method looks for synonyms in the thesaurus. If the property is not matched, then it creates a new value in the ontology. Using this method, ontology term matching resulted in average accuracy of 93.6%.

Van Assem et al. (2010) used the ontology of units of measure and related concepts (OUM) to annotate tables. The approach was of a classical text annotation task, firstly extracting the content of the cells and annotating it using a set of crafted rules.

Tables can be represented as attribute-value pairs, especially if reduced to the grid structure by dropping column-spanning and row-spanning cells, while duplicating content to the required number of cells. A cell can be a value of more than one attribute and may act as an attribute in one case and a value in another. As a result, multiple attribute-value pairs can be merged to represent the actual meaning of table (Chen et al. 2000). Data from web tables can be clustered to build concept-term relations using overlapping triplets from tables. Triplets contain three entities of the same class

(e.g. [Apple, Avocado, Peach] – fruit class triplet) that occur in the same table (usually in the same column). It is possible to make concept-term clusters in an unsupervised way. The content of three rows can be taken as a data triplet from each column. The assumption is that the context for each term is stored in a triplet. Triplets can be clustered by concept since each triplet represents instances of the same concept. Tables are labelled with the help of a Hyponym Concept dataset, which includes data on terms and concepts they occur in, with counts for each concept. However, this approach can only be used for a narrow set of matrix tables (Dalvi et al. 2012).

Google have patented a method to extract relational tables from lists on the Web (Elmeleegy et al. 2014). The aim was to transform lists into multi-column relational data. They firstly split individual lines into multiple fields, and then compare the splits across multiple lines to identify and fix incorrect splits and bad alignments. For each table a calculated extraction score should reflect the confidence in table's quality.

ChemDataExtractor (Swain & Cole 2016) recently presented a method for information extraction of chemical entities from literature that is able to process both text and tables. It focuses only on tables where data about a chemical entity is in one row, utilizing a rule-based parsing grammar tailored for extracting certain properties. Extracted data is mapped into a predefined data model. The overall results range from 85% F1-score to 92% for various sub-tasks. However, no results for information extraction from tables only have been reported. The methodology is also limited to the pre-described type of simple tables although it can extract information from XML, HTML and PDF documents.

2.7.2 Machine learning approaches

One of the main applications of information extraction is a question-answering system. Wei et al. (2006) designed a question-answering system that answers questions whose answers were in tables. The system analysed ASCII free text documents and tagged table lines with functional tags (e.g. title, header, superheader (super-row), data row, table caption, nontable, etc.) using the conditional random fields (CRF) algorithm. They used white space (the number and length of white spaces and gaps), text (the number of cells on a line, certain types of string more typically in some parts of tables, type of character on a line) and separator features (special characters, successive characters) from current, previous and following lines. The system created an XML document for each cell with data, metadata and table captions related to that cell's information. Cell documents were created using a set of heuristics about header cells. Cell documents

are ranked using information retrieval methodology (TF-IDF and cosine similarity), in order to find the document that contains the answer to the query. The work of Wei et al. (2006) provided a baseline for the development of the data structure for storing table data and its annotations (described in detail in Chapter 4).

Information can be extracted from tables to semantic triplets of the form $\langle p, s, o \rangle$, where p is a predicate or relation, s is the subject of the predicate and o is its object (Crestan & Pantel 2010). However, extracting the subject from the table can be challenging. In attribute-value tables (tables containing only 2 columns with attributes in the first and values in the second column), normally one column is devoted to the attribute names (mapping to predicates, p) and another column to the values of the attributes (mapping to the objects, o). Extracting predicate and object is generally straightforward in attribute-value tables. There are mainly three places where the protagonist (subject) could be found: within the table (occasionally found in the table with a generic attribute such as name or model), within the document or within the HTML `<title>` tag and anchor text pointing to the page. Crestan & Pantel (2010) used N-grams (1-12 grams) and anchor text (obtained from a commercial search engine's web link graph) in combination with the Gradient Boosted Decision Tree (GBDT) regression model. They reported 40% precision, which they considered a good starting point as they also reported a 97% chance to find the correct protagonist in the top 100 ranked candidates.

Recently, Sun et al. (2016) proposed a three-step approach for answering questions that have answers stored in a table. In the first step they generated a set of candidate chains containing a topic (label or header) and a value. In the second step, they utilized search engine snippets to filter out irrelevant candidate chains. In the last step they used deep neural networks in order to rank remaining answers.

Gene mutation extraction from tables is one of the rare applications of table mining in the biomedical domain (Wong et al. 2009). PubMed was crawled following links to HTML articles containing tables. Since these tables are labeled with table tags, the approach did not have to deal with table detection. The extraction task is grounded in the specific content of the Mismatch Repair (MMR) Database — a database of known genetic mutations related to cancer, along with links to research papers from which the database has been constructed. From the database and its links to papers, a collection of tables related to cancer mutations was constructed. Then, the MMR database records themselves were used as a gold standard for evaluating the techniques. Column headers

were detected using `<hr>` tag and row headers were detected by checking if the top-left cell was empty. They classified columns/rows (depending on whether the table was horizontal or vertical) into relevant entities and used a heuristic for classification of headers matching the header string to the names of the classes. The second approach was to build a more informed classifier for the class "Mutation" using an NER system called MutationFinder (Caporaso et al. 2007). They applied MutationFinder to the text in the table cells and identified which table-vector contained at least one mutation mention. They used a set of machine learning algorithms with different sets of features (cell bag-of-words, header bag-of-words, cell and header bag-of-words, basic features like header string, average median cell length and are data numeric).

Xu & Wang (2015a) created a classifier for drug-associated side effects, using the lexicons for side effects and drugs. Firstly, the extracted tables were classified using machine learning into two classes – related to drug-side effects and unrelated to drug-side effects. Secondly, the associations were extracted using rules and manually curated lexicons of drugs and side effects. The study showed that 84.7% of side effect extracted from tables from articles published in Journal of Oncology were not reported in FDA's drug labels.

Silva (2010) claims that table mining is a complex, multi-layered problem and no one algorithm is capable of accurately treating all tables. She applied several algorithms in order to extract information from tables in financial statements, such as SVM, heuristics and graphical approaches such as Markov random fields.

Since successful information extraction depends on gold standard corpora, Shmanina et al. (2016) developed a corpus of tables in full-text biomedical literature for information extraction and relation extraction. They developed an annotation guideline and a dataset comprising of 83 annotated tables with UMLS concepts, cell groups, and relationships between cells (associated_with, property_of). Even on request, the dataset was not made available to the author.

2.8 Table navigation for visually impaired people

One of the most important applications of table mining is table navigation for the visually impaired. Screen readers allow for basic navigation through a table, reading to the user content of cell, row or column (Yesilada et al. 2004). Some screen readers on iPad read the content of the cells under a reader's finger (Ahmed et al. 2010). For table reading, it is unnecessary to add semantics or analyze the function of the table areas.

However, some approaches included the use of a table ontology and utilized functional analysis of the cells (Doush & Pontelli 2013, 2010). Some of the approaches for table navigating for visually impaired people are available in commercial products such as JAWS¹.

2.9 Conclusion

In the past, various models and approaches for table processing have been presented. Typically however, these models only describe a table's visual structure and have been designed mainly for presenting information. In order to obtain a model that describes the meaning of the table values and their relationships, further processing is necessary. Models that link data cells with their navigational cells or table data with the meaning described in some ontology are rare and often incomplete (e.g. do not include all table types or table elements). Apart from abstract, physical and logical table models, there is a need for a semantic model that describes relationships between cells and content with their semantics.

Processing and understanding tables involves several steps: detecting tables, analyzing functional areas, structure and table semantics. Successful approaches across different media types (HTML, PDF, text documents) have been proposed for table detection. Similarly, a number of heuristic and machine learning-based approaches have been proposed for functional analysis. However, most functional analysis approaches have focused on detection of column headers, while detection of other functional areas, such as row headers (stubs) or super-rows are rare. The majority of table processing research is undertaken in these two areas. With satisfactory results for table detection and functional analysis (header detection), these problems could be considered solved. Table mining relies on table detection and functional analysis, which makes these two research fields attractive to researchers. Detecting inter-cell relationships adds semantics by linking a certain data cell to the related navigational cells. Information retrieval, information extraction and knowledge acquisition approaches have also been presented. Table understanding (tasks such as information retrieval, information extraction or knowledge acquisition) relies on previous steps that can be applied to a wide variety of table layouts, as their main purpose is to add semantics to the tables. The semantics of the table content have to be domain specific. Even some of the seemingly domain independent tasks, such as functional analysis may benefit from

¹<http://www.freedomscientific.com/Products/Blindness/JAWS>

semantics and domain specific approaches. Work, done so far, was either performed in various diverse domains or tried to find solution for general domain. These attempts produced limited results. Certain solutions are available only for some narrow domains (e.g. car sales, gene mutations, etc.) or they are limited by a specific table type. There is a need to consolidate the efforts and present a complete information extraction work-flow for one domain that is generalizable with certain modification to other domains.

Specifically, the survey presented in this chapter highlights the following challenges:

- Resolving structural relationships between table cells: relationships between the cells depend on functional analysis. However, only a few approaches have been presented on how to disentangle inter-cell relationships.
- Handling complex tabular structures: many approaches disregard complex table types (multiple tables merged, or tables with spanning cells or super-rows).
- Semantic analysis of tables and its applications to knowledge discovery, information extraction, question answering: some of these applications have been explored, but many lack a domain specific solution, such as biomedicine.
- Lack of resources: we noticed that most of the approaches have not published data or software implementations. Some were not available even upon request. The rarely available implementations are not maintained and often rely on deprecated libraries.

Consequently, we see three gaps for table-mining research:

1. Structural analysis and inter-cell relationship resolution.
2. Creating a table models for storing and representing semantics of information stored in tables.
3. Table understanding and semantic analysis, including tasks and applications requiring a level of semantic understanding of data. Capturing common table and value representation patterns and their semantics can be one of the examples that facilitate table understanding.

The steps are dependent on each other, so it would be difficult to tackle table understanding and applications of table mining without resolved inter-cell relationships and

without a good model to store multidimensional semantic information about the table. The described directions are reflected in our research questions presented in Section 1.4. Structural analysis and inter-cell relationship resolution refer to the question on the levels of processing needed for disentangling a table's internal structure from its visual representation. The table model for storing and representing the semantics of the table data is reflected in the same question but expanded with questions about processing levels, necessary knowledge and information for designing an information extraction system. In the case of table understanding and semantic analysis, this thesis focuses on information extraction tasks. Our research questions ask about what layers of processing and knowledge of the data is required for building a table information extraction system, as well as which layer can benefit from machine learning approaches compared to rule-based approaches, and what accuracy levels can facilitate successful data curation.

Table mining is a complex, multi-layered problem and no one algorithm is capable of accurately treating all tables (Silva 2010). Thus, coordination between different table processing approaches, whether these are alternative or sequential to each other, is fundamental. Analysing table function, structure or detecting tables is not much dependent on domain knowledge and can be done without any domain specific resource. However, table interpretation is dependent on context knowledge and therefore table interpretation may use domain ontologies and lexicons.

The biomedical domain is an active research domain for text mining, with a strong focus on literature mining and information extraction from the literature. To-date, only a few approaches have been presented for mining information from tables in the biomedical domain. Text mining of tables in the biomedical domain is also important, as advances in biomedical research and better access to biomedical information can have an impact on societal health, quality of life and mortality. Therefore, this research focuses on developing a table mining and information extraction approach for the biomedical domain.

Chapter 3

Methodology Overview

3.1 Introduction

Literature processing has a goal to extract, store and maintain relevant information from the articles and facilitate querying and usage of presented information. The process of collection, storage, maintaining, annotating and integration of the data in order to maintain the value the information over time is called *data curation* (Yakel 2007). This thesis focuses on information extraction and data curation from tables presented in biomedical documents in XML format. Generally, semi-automated data curation systems consist of four components: an information extraction engine, a data store, a data curation interface and a query interface (Alex et al. 2008). Each of the four parts of the system may be supported by various knowledge sources such as lexicons, thesauri, databases or linked data sources.

An architectural overview of the proposed approach is shown Figure 3.1. This approach first processes documents using an automated information extraction system. After the data is extracted, human expert curators check and adjust the extracted data using a data curation interface to assure the validity of extracted data. Users may access the data using a query interface. In the following sections, these components will be described in more details, with the focus on information extraction methodology.

3.2 Scope

A methodology that facilitates information extraction and curation from tables in the biomedical literature requires a model for preserving table information, an information extraction engine and data querying engines.

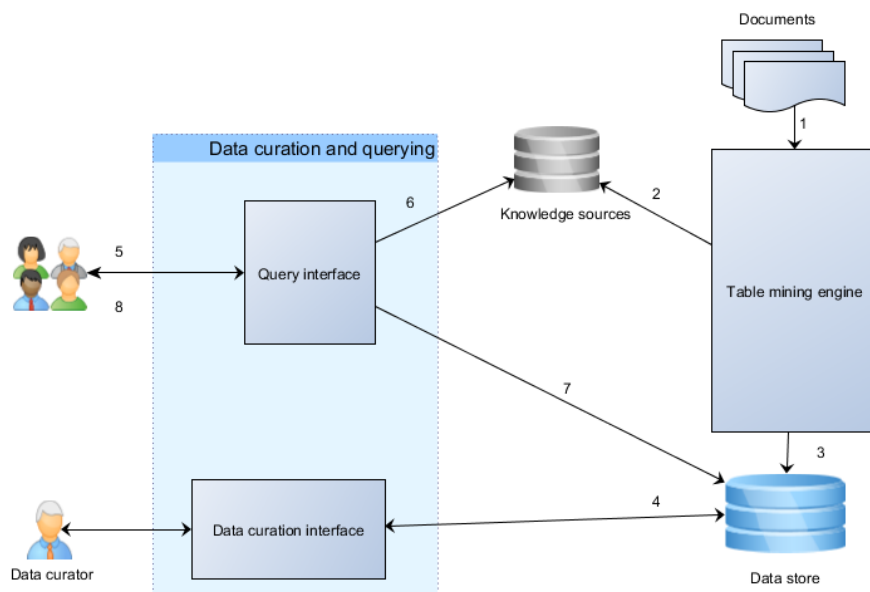


Figure 3.1: High-level architecture of the proposed methodology

1. Retrieved documents are sent to the table mining engine. 2. The table mining engine uses knowledge sources to extract information from the table. 3. Extracted information is stored in a data store. 4. Data curators review and correct extracted information. 5. Users submit queries to the query interface in natural language. 6. Queries are processed and normalized. 7. Using normalized queries, the data store is queried. 8. Relevant extracted information is presented to the user.

The data presented in tables are often unreachable by automated literature processing systems. Therefore, there is a need to make these data reachable and searchable.

The context of the presented information is stored in navigational cells (headers, stubs, and super-rows) and table descriptors (table caption and footer). In order to make table information searchable, this information needs to be complemented in a data model with information from related navigational cells and table descriptors. For example, if we consider the table presented in Figure 3.2, and cell with content "12/4" in order to interpret the content, it is necessary to consider its stub and header (gender distribution for dexmedetomidine arm). Transforming information from an article to a data model that preserves cell information with its context and allows for querying of data, highlights the need for cell function recognition (functional analysis) and disentangling relationships between cells (structural analysis).

Once the data is transformed and stored in a format that allows querying, it is possible to make the following queries:

- Find relevant tables constrained by the content of the table and the article (e.g. find tables containing "dexmedetomidine", find tables containing adverse events

in the caption and "cancer" in the article title, etc.).

- Find cells constrained by their content or the content of related navigational cells (e.g. find cells containing "sex" in stub, find cells containing "dexmedetomidine", etc.)

Table 1

Patient characteristics and operative details for the dexmedetomidine and placebo groups

	Dexmedetomidine	Placebo
Type of surgery (cardiac/general)	9/7	10/7
APACHE II score for general surgical patients (median interquartile range)	10 (5)	10 (2)
Parsonnett score for cardiac surgery patients (median interquartile range)	10 (6)	13 (7)
Age (years)	63 ± 15	63 ± 15
Sex (male/female)	12/4	11/6
Wean-extubation time (h)	2.6 ± 2.7	2.4 ± 2.5
Intubation time (h)	10.5 ± 4.5	9.0 ± 4.4
Total infusion (intubation + extubation; h)	17.5 ± 3.7	15.1 ± 4

Figure 3.2: Example of a table presenting values using different presentation patterns (PMC 29047)

Data represented in the data model that connects navigational cells with the data cells and makes possible querying the table data is useful for information retrieval. However, semantics are still missing from the data. The aim of information extraction is to extract variables of interest with values and metadata from the literature tables. In a given example from Figure 3.2, the methodology should be able to infer that the row with "Sex" in the stub is related to gender distribution: 12 is the number of male participants and 4 is the number of female participants. Also, the methodology should infer that the values are related to the dexmedetomidine arm of the trial. Therefore, an information extraction template for extracting information from tables should contain the name of the extracted variable, the value of the variable, unit of measure, variable metadata (information from navigational cells, such as name of the clinical arm, patient ID, etc.), document metadata (reference to the document and the table), etc. Information extracted in this manner allows for more granular queries (e.g. number of males in dexmedetomidine arm) and performing certain operations on the data (e.g. calculating a number of participants from the reported number of male and female participants, retrieving studies in which a variable was greater/lower than a specified value, etc.).

3.3 Table information extraction

This part of the methodology aims to extract information from the source documents automatically. At a higher level, the methodology contains four parts:

1. Table detection – determines the location of table in the document.
2. Table disentangling – decomposes the table in the article into cells, while inferring their function and relationships to other cells. This part is usually task and domain independent.
3. Table and cell annotation – annotates, normalises and enriches the information in cells and table. Performs pragmatic and semantic tagging of cells and tables. Since this part uses specific knowledge bases and machine learning models, it is typically domain dependent but task independent.
4. Information extraction – extracts the variables of interest into the data store.

The graphical presentation of these high-level methodology parts and how they form an information extraction methodology workflow can be seen in Figure 3.3.

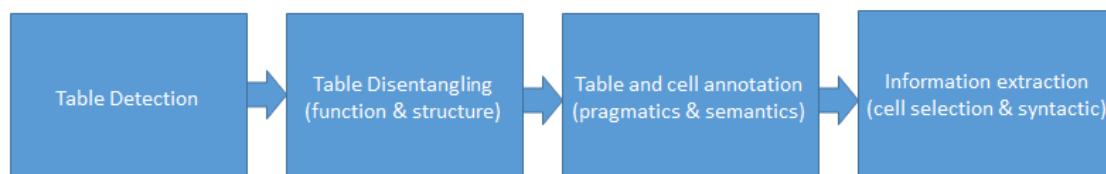


Figure 3.3: High level overview of the methodology

On a lower level, our approach consists of seven steps: (1) table detection, (2) functional analysis, (3) structural analysis, (4) semantic tagging, (5) pragmatic analysis, (6) cell selection and (7) syntactic processing and extraction. Table detection is the first part of the methodology. Functional and structural analysis answer how cells are related to each other. Semantic tagging and pragmatic analysis form the third part of the high level methodology. The process is finishing by information extraction that consists of cell of interest selection and syntactic processing. Detailed overview of the methodology is presented on Figure 3.4. Each step of the methodology is described below.

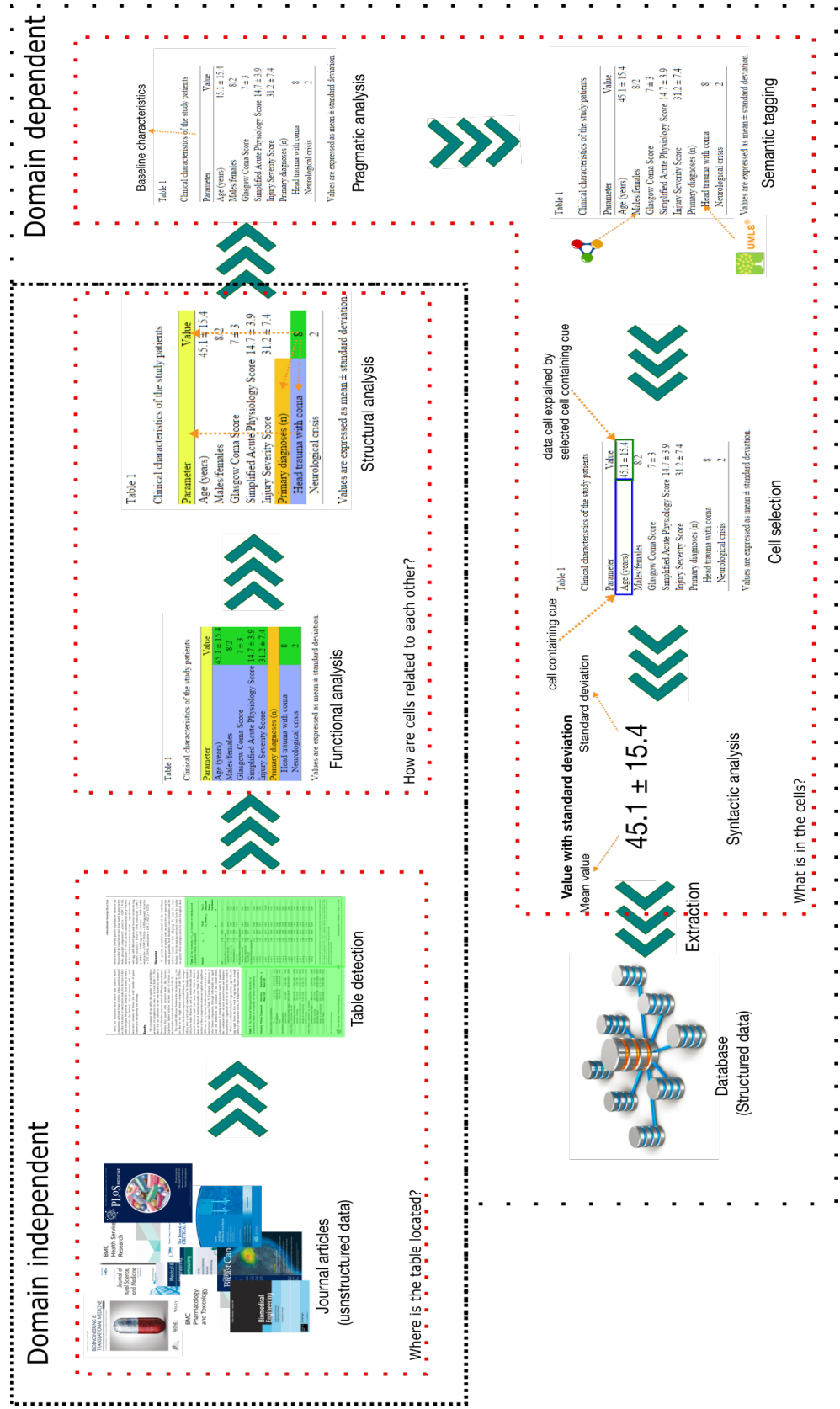


Figure 3.4: Overview of the methodology steps as they are executed

3.3.1 Table detection

In table mining, it is first necessary to identify table mentions in documents. In some types of documents, this task might be trivial. For example, in many XML formats it is possible to identify tables by extracting the content of a specific XML tag. However, in some types of XML documents and in HTML it is a much harder task. Research on the identification of tables from HTML was described in Section 2.3 and is outside the scope of this thesis. Our methodology is developed for documents in PMC and DailyMed database, where tables have been identified by locating appropriate *table* tags.

3.3.2 Table disentangling

Functional processing

The second step in the methodology is a detection of navigational and data areas. In literature, this task is called *functional area detection* or *functional table analysis* since it determines the function of cells in a table. In this task, it is important to distinguish header, stub and super-row cells from data cells. Research that addressed this task was discussed in Section 2.4. Many authors have used machine learning to approach this problem. However, the challenge is that there is no publicly available annotated dataset or system. Therefore, our methodology mainly follows a heuristic-based approach that considers the arrangement of cells, spanning cells, a presence of special characters (e.g. horizontal lines) or empty cells, cell similarity, etc. However, this approach could not distinguish functional areas in cases where tables did not make any distinction between navigational cells and data cells. Thus, only semantics can distinguish functional areas. In order to detect functional areas in datasets in which headers are only semantically distinguishable, the methodology uses a machine learning classifier based on cell content.

Structural processing

The aim of structural analysis is to map each data cell in the table to the related navigational cells (i.e. header, stub, and sub-header). Structural analysis, through Hurst's model (Hurst 2000), can be seen as finding reading paths. This task is highly dependent on functional analysis of the table. If headers, stubs and super-rows are correctly recognized, our methodology can relate data cells to their navigational cells. Also, since

tables can have multiple layers of headers, stubs or super-rows, the method needs to relate lower layers of navigational cells with the higher layers that describe them in a cascading way. We used a set of heuristics about cell function, structure, content, position and table structure to disentangle table structures and inter-cell relationships.

3.3.3 Table and cell annotation

Pragmatic classification of the table

Our methodology labels the table with the class that represents the category of domain-specific information stored in it (for example, in our clinical trial case studies, we used the following four classes: baseline characteristics, adverse events, inclusion/exclusion criteria, other). The classification algorithm infers what the table is used for and how its context contributes to the meaning. In linguistics, a study of author's intent and how context contributes to meaning is called pragmatics (Liu 2005). Therefore, this classification is called pragmatic classification of tables. It is intended to narrow the scope when extracting information and exclude information that would be mapped to the same concept but is used for a different purpose (e.g. the number of patients at the beginning of the trial and the number of patients that survived until the end of the trial). Pragmatic classification is performed using the machine learning methodology with table content features.

Semantic tagging

Data in table cells are usually presented as strings and numerals. Semantic tagging normalizes and enriches data. In this step, our methodology makes relationships between words or phrases and concepts in knowledge sources. Binding data to knowledge sources is useful since knowledge sources contain semantic information that can be used to automatically determine the meaning of the cell, group of cells or the whole table. This method enriches data by annotating cell content using named entity recognizers and vocabularies such as UMLS (Bodenreider 2004). For this purpose, annotation software called "Marvin" was developed that is able to annotate text by mapping it to multiple knowledge sources. Marvin can be used both as a library or standalone application. At the time of writing, Marvin supports annotations using WordNet, MetaMap, DBpedia and custom SKOS thesauri.

3.3.4 Information extraction

The aim of the information extraction task is to extract user-defined variables, their values and metadata from tables in literature (see Section 3.2). Here we firstly describe extraction template, then we describe variable description framework that is used for developing information extraction rules and methodology. At the end we describe the information extraction steps, namely cell selection and syntactic analysis.

Extraction template

Since tables present multidimensional data, an information extraction template should reflect the multi-dimensionality of the information, the variable's value and present necessary metadata and context attached to the variable. Additionally, the template should retain a bond to the article from which the information is extracted. We propose the following extraction template:

(VariableName, VariableSubCategory, ValueComponent, Context, Value, Unit)

- *VariableName* is the name of the variable that should be extracted. It can be linked with a certain ontology (e.g. Ontology of Clinical Research (OCRe) (Sim et al. 2014) or UMLS).
- *VariableSubCategory* is used only for variables when there are multiple subcategories that have values (e.g. ethnicity and number of participant presented as number of White, Asian, Hispanic and Black people).
- *ValueComponent* parameter presents the name of the value component of the extracted variable's value, obtained by analysing its presentation pattern. For example it may be *Value* if the cell presents a single value, *Range:Min* if the extracted value is minimum in the range, *Range:Max* for the maximum in the range, *Percentage* for values presenting percentage, *Mean* for mean values, and *SD* for standard deviation. In the case when a cell presents a range, two rows in the template should be extracted, one for the minimum and one for the maximum.
- The *Context* is the parameter that describes the value's context. It can be, for

example, a clinical trial arm for tables presenting cumulative baseline characteristics of patients, or a patient identifier for tables presenting baseline characteristics for each patient separately.

- The *Value* is the extracted value for the given variable from the table.
- The *Unit* parameter is only applicable for numeric variables, where it is used to specify the unit of measure in which the value is expressed. For example, body mass can be presented using a singular unit (gram), multiples (kilogram) or sub-multiples (milligram) (Van Assem et al. 2010). Each variable should have defined a default unit (if it exists, usually it is singular unit) and that unit is used if it is not otherwise specified in the table.

Additionally, the template should retain a bond to the article and the table from which the information is extracted. Example of the table and several populated template rows can be seen in Figure 3.5.

Clinical characteristics of the ITT sample (n = 40)

	Whole Group mean \pm SD	Group 1 (n = 16) mean \pm SD	Group 2 (n = 24) mean \pm SD	P
Age (years)	53.0 \pm 11.0	57.1 \pm 10.4	50.3 \pm 10.7	ns
Sex (female/male)	25/15	10/6	15/9	ns
Previous ECT (yes/no)	12/28	6/10	6/18	ns
Unipolar/bipolar depression (yes/no)	34/6	13/3	21/3	ns
Duration of current episode (wk) ^a	54.3 \pm 34.5	47.8 \pm 34.3	58.6 \pm 34.6	ns
Pre-ECT HDRS score	25.3 \pm 6.8	28.6 \pm 5.4	23.0 \pm 6.9	0.0093
Psychotic features (DSM-IV)(yes/no)	5/35	3/13	2/22	ns
Pre-ECT MMSE score	27.2 \pm 2.3	26.0 \pm 2.6	28.1 \pm 1.6	0.0053
Number of antidepressant trials during episode	2.2 \pm 1.4	1.9 \pm 1.2	2.3 \pm 1.5	ns
Prior adequate antidepressant treatment (yes/no)	34/6	14/2	20/4	ns

Extractions:

Variable name	Variable Sub-Category	Value Component	Context	Value	Unit
Age	-	Mean	Whole Group mean \pm SD	53	years
Age	-	SD	Whole Group mean \pm SD	11	years
Gender	Male	Number	Whole Group mean \pm SD	15	people
Gender	Female	Number	Whole Group mean \pm SD	25	people

Figure 3.5: Example of a table (PMC 65527) and extracted information to the proposed extraction template

Framework for information extraction from tables

This thesis proposes a framework for information extraction from tables that relies on variable description. The variable description includes:

- Variable identifier - Name or ontological identifier of the variable that should be extracted from the data.
- Pragmatic class of the table - Pragmatic or context class of the table narrows the scope of the information search and reduces the number of false positives.
- Lexical and/or semantic cues - These cues help determine whether a certain cell contains variable or its value. Lexical and semantic cues are defined using whitelist (cues that indicate an existence of the value in a certain cell) and blacklist (cues that indicate that cue is not in the cell).
- Functional cues - Indicate in which functional areas of the table cues should be looked for and in which functional areas variables and their values are presented.
- Syntactic patterns - Indicate how the complex value presentation patterns associated with the target variable should be disentangled and which part of the cell with a value should be extracted target variable.
- Unit of measures - Indicates in which unit of measurement the value is presented, if applicable.

Based on this description, information extraction task is designed and executed.

Information stored in tables can be numerical (e.g. number of patients, BMI, average age, etc.), categorical (e.g. positive/negative, grade I/grade II/grade III, etc.) or textual (e.g. definition of terms or scales). Presentation patterns for variables of the same type are often similar. For example, statistical variables such as BMI (body mass index), age, FEV1 (forced expiratory volume exhaled at the end of the first second of forced expiration) are usually presented as mean, standard deviation and/or value range. Tables presenting them often have a similar structure. Therefore, it is possible to use these cell content presentation patterns and develop a set of rules that can be applied to extract many variables. The proposed framework generalizes table information extraction using patterns that commonly appear in tables (structural, semantic/reading, syntactic).

Cell selection

Once the knowledge from or the variable description or recipe for the information class is provided by the user, the framework method can extract the defined variables and their values. The method uses rule-based method that firstly selects cells that contain

cues in given functional areas (headers, stubs, super-rows or data cells). Once cells are selected, they are analyzed against the blacklist of cues. If the blacklist exists in a given functional area, the considered cell is discarded.

Syntactic analysis

Before the information is extracted, the syntax of the selected cell is analysed against a set of syntactic patterns. These patterns are pre-defined to inform the method how to disentangle and interpret the content of the cell. Cells often contain complex value presentation patterns and represent multiple information (see example in Figure 3.6). Authors usually use same or similar value presentation patterns to present similar information (e.g. *variable value*, *mean*, *standard deviation*, *percentage*, *alternative values*, etc.). Patterns provide the way to extract atomic information and to provide the value presentation semantics. For example if the value is presented as 16 ± 3.2 , it is possible to determine that the first value is mean or median, while the second is standard deviation or standard error. In order to exactly specify the semantics of each value component, the methodology looks at the related access cells. Based on these patterns, information is extracted and stored to the database.

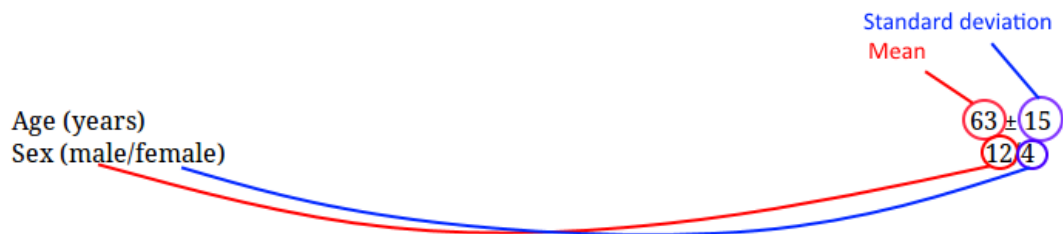


Figure 3.6: Syntactic analysis infers the implicit meaning from the value presentation pattern (upper row) or link to the explicitly stated meaning in the navigational cells (lower row)

3.4 Evaluation methodologies

As evaluation metrics for information extraction and classification are often used precision, recall, and F1-score. Precision is a measure that evaluates how many true/relevant instances are among the retrieved (classified as positive) instances. The recall is a measure that evaluates the number of true/relevant instances that have been retrieved

(or classified) among the total number of true/relevant instances. In order to calculate precision and recall, it is necessary to introduce the following metrics:

- True positive (TP) - a number of true/relevant instances that were retrieved. The number of instances that are relevant/true and as well indicated by the algorithm that is relevant.
- False positive (FP) - a number of irrelevant instances that were retrieved as relevant. The number of instances that are irrelevant/false, but indicated as relevant/true by the algorithm.
- False negative (FN) - a number of true/relevant instances that were not retrieved. The number of instances that are relevant/true, but algorithm indicated them as irrelevant/false.
- True negative (TN) - a number of false/irrelevant instances that were not retrieved. The number of instances that are irrelevant/false and indicated as such by the algorithm.

A visual explanation of these metrics can be seen in Figure 3.7.

		Prediction outcome	
		P'	N'
Actual value	P	True Positive	False Negative
	N	False Positive	True Negative

Figure 3.7: Confusion matrix with graphical explanation of true positives, false positives, false negatives and true negatives

Formulas for calculating precision and recall are presented in Equations 3.1 and 3.2. F1-score is a metric that combines precision and recall (Feldman & Sanger 2007).

The formula for calculating F1-score is presented in Equation 3.3.

$$Precision = \frac{TP}{TP + FP} \quad (3.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.2)$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3.3)$$

Precision and recall are commonly used measures in information extraction and information retrieval communities because they deal with different types of errors and can be viewed for each variable. Therefore, they can provide a valuable insight into the performance of the system.

Other measures are also used in information extraction community, such as accuracy. However, since accuracy is a measure of correctly classified instances over all instances, in some cases it can provide a misleading insight, especially with unbalanced datasets.

Silva (2010) argued that precision and recall are not the best measures of table and so proposed new measures:

- Completeness – a proportion of completely identified elements with respect to the total number of real elements; for example, in order to be completely identified, a column must contain all of its cells.
- Purity – a proportion of fully detected elements with respect to the total number of detected elements; a pure element is one whose components belong to only one original element.

Silva's measures are more lenient than precision and recall, as they measure partial match to the expected template. Therefore, it may be that these measures were invented in order to boost the results of the presented method.

In this work, we will follow the majority of information extraction community and precision, recall, and F1-scores as our evaluation metrics. The insights of these metrics depend on how the data is selected. The evaluation data should be a representable sample. In case there is a selection bias or the evaluation dataset is too small, even these measures can give misleading insights about the performance of the evaluated system.

For evaluating and fine-tuning of the methodology, usually, initial data set is split into a training and testing (validation) set (a method often referred as *holdout validation*). A training set is used either in machine learning for training the model or to fit the rules. A testing set should be unseen until the model is created or satisfactory rules are created. Then the method is evaluated against unseen testing set (Kohavi et al. 1995). The testing set should remain representative of the whole dataset. In case it is not representative, the evaluation might be biased. Therefore, it is necessary to select representative and large enough sample for both training and testing set.

Leave-one-out cross-validation, N-fold cross-validation (most often 10-fold cross-validation) is a method in which evaluation dataset is split into N parts, trained on N-1 parts and tested on the last part. This is performed N times, each time testing on the different part of the dataset and training on the other parts. Using this method, the whole dataset is used for testing, without any data point being used for training and testing in the same validation cycle (therefore method remaining unbiased in that regard (Kohavi et al. 1995)). This method can be used with relatively small datasets, where the split dataset may not be representative (Cawley & Talbot 2003). However, since all the data is used for evaluation, the performance of all distinct kinds of data points will be captured. It is still necessary to ensure that the whole evaluation set is representative.

3.5 Case studies for methodology validation

To test and validate our method, we used two datasets consisting of clinical trial documents and drug labels. The first dataset consists of clinical publications stored as open access in PubMedCentral¹. The second dataset consists of drug labels. We have designed three case studies in order to develop and validate our methodology.

The first case study was designed for the development of the methodology. In this case study, we wanted to extract number of patients, their age, gender and adverse events that happened during the described clinical trial. We used clinical trial articles from PubMedCentral database for this study.

The second and third case studies were designed to validate the methodology. The second case study was designed in collaboration with AstraZeneca in order to take into account industrial needs for information extraction from biomedical tables. The goal of the study was to extract relevant information from the clinical trials about asthma and

¹<http://www.ncbi.nlm.nih.gov/pmc/>

chronic obstructive pulmonary disease (COPD). We extracted reported measurements from several lung function tests (spirometry) tests, such as forced expiratory volume in one second (FEV1) and peak expiratory flow (PEF). Additionally we extracted results of several quality of life questionnaire such as asthma quality of life questionnaire and Saint George respiratory questionnaire.

In the last case study, we tried to validate methodology on extracting drug-drug interactions from tables in drug labels. The dataset was created by selecting drug labels that report drug-drug interaction in DailyMed database², maintained by the US National Library of Health. The motivation for this case study was twofold – to examine generalisation of the methodology on different datasets (XML formats) and to explore application of information extraction method for relationship extraction.

3.6 Data curation and querying interface

The information extraction engine is the first step in our semi-automated data curation pipeline. For potentially erroneous extractions, a data curation interface is necessary that would allow human experts to check the extracted data and if necessary, correct, add or delete information. Data curation is a key for practical applications in which data quality is important. The main purpose of the data curation engine is to provide a user-friendly interface for human experts to review and edit previously extracted information.

There are two points at which a data curation interface can be useful: after structural disentangling and after information extraction. Information extraction is dependent on the structural disentangling part of the process. Because of this dependency, high-quality information after disentangling will ensure higher information extraction efficiency. At this step, the user can correct wrongly labelled, functional annotations. A similar method applies to information extraction. In order to achieve high quality of extracted information, a curator should review, check and correct extracted information. This can be done by reviewing data in the database (for an expert user), or by using a web interface crafted for this purpose.

When the data are extracted and reviewed by human experts, they may be used by a broader user base. Then, users can query the data store in order to find relevant information.

²<https://dailymed.nlm.nih.gov/dailymed/>

3.7 Summary

An overview of the information extraction and data curation from biomedical literature tables' methodology is presented in this chapter. Further details about the methodology and its evaluation will be provided in the following chapters. The methodology consists of two main parts: an information extraction engine and a data curation and query interface. The methodology for information extraction contains seven steps: (1) table detection, (2) functional analysis, (3) structural analysis, (4) pragmatic analysis, (5) semantic tagging, (6) cell selection and (7) syntactic analysis and information extraction. Since functional and structural table analysis is crucial for further table mining tasks, we propose a curation interface able to visualize tables with functional areas that a curator can correct and ensure the quality of data. In order to perform information extraction, it is necessary to use and develop tools that perform these steps well. We propose an annotation step, consisting of semantic tagging and pragmatic analysis (pragmatic annotation of the table) that will help further table understanding tasks and information extraction. After table and cell annotation, it is possible to use annotated data for multiple tasks, such as information retrieval, table querying, information extraction and table reading for visually impaired people. In this thesis, we focus on information extraction, for which we propose a flexible and easy-to-use, rule-based framework. As part of the framework, we propose a recipe and generic corresponding methodology for biomedical literature information extraction. Our recipe prescribes the knowledge users need to feed into the corresponding methodology to perform successful information extraction. In the following chapters, we describe our methodologies in more detail: functional and structural analysis in Chapter 4, pragmatic table analysis and semantic tagging in Chapter 5 and cell selection and syntactic analysis of the content in Chapter 6.

Chapter 4

Disentangling the structure of tables*

Disentangling table structure comprises of reading the table from a presentation document, transforming it to a computational representation, distinguishing functional areas and resolving functional relationships between table cells. The disentangled structure is stored in a computational format, such as a relational database.

In this chapter, we first describe a model for computational representation and handling of tables. Then we describe a method to locate navigational areas of the table in the PMC XML format and resolve functional relationships between table cells.

4.1 Model for representing tables

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

- Table types that includes common structural types that determine the way of reading the table;
- A data model that represents the table structure and data in such a way that data can be automatically processed (including mining and visualization).

* Parts of this work have been published in Milošević, N., Gregson, C., Hernandez, R. and Nenadić, G. Disentangling the Structure of Tables in Scientific Literature. In International Conference on Applications of Natural Language to Information Systems (pp. 162-174). 2016, June. Springer International Publishing. DOI: 10.1007/978-3-319-41754-7_14

4.1.1 Table types

We define three main structural table types with several sub-types based on table dimensionality. The identified table types were motivated by the work by Wright & Fox (1970), who identified one-dimensional (list) and two-dimensional (matrix) tables. We extended this work and our table types include:

- **One-dimensional (list) tables** are described by a single label. The label is usually placed in the header (see Figure 4.1). One-dimensional tables may have multiple columns representing the same category, where a multi-column structure is used to save space (see Figure 4.2).

Table 1

Inclusion and exclusion criteria

General inclusion criteria:

- Psychological problems (ICPC chapter P)
- Symptoms of general exhaustion and burn-out (A01, A04)
- Musculoskeletal pain (ICPC chapter L)

Figure 4.1: Example of a list table (PMC 161814)

Table 1

Twenty epithelial characteristics and their descriptions used to evaluate uterine biopsies using SEM

Epithelial Characteristics – Graded 0–3	
1. Epithelial abundance – the amount of epithelium found within the sample	11. Cell separation – at times cells are observed to be separate rather than tightly clustered together
2. Tissue heterogeneity – the variability of tissue surfaces within the sample	12. Denuded apices – cell surfaces are devoid of surface modifications such as microvilli, cilia (excluding uterodomes)
3. Cell heterogeneity – the variability of the appearance of the cell types within each field	13. Flattened cells – degree to which cells display a flattened topography
4. Gland abundance – the relative number of glands observed within each field	14. Deflated cells – whether cell apices appear to have collapsed or withered
5. Gland opening – the types of gland opening, whether wide, raised, narrow	15. Apical protrusion – the degree to which the cell surface protrudes into the lumen of the uterus
6. Cilia groups – the relative number of ciliated cells clustered together	16. Uterodomes – shape (see Uterodome Assessment)
7. Single cilium – presence of these indicate a senescent or atrophying epithelium	17. Uterodomes – abundance (see Uterodome Assessment)
8. Microvilli height – the relative length of microvilli, from short and blebbed to long	18. Cell borders – may be obvious or deeply recessed between cells
9. Microvilli density – relative number of microvilli per cell, from few to many	19. Secretion – the presence of secretory product within the field or on cell surfaces
10. Apical membrane defects – include observation of porosity and degeneration	20. Plicae – are microvillous folds or ridges on the cell surfaces

Figure 4.2: Example of a list (one-dimensional) table with multiple columns (PMC 420259)

- **Two-dimensional or matrix tables** have data arranged into a matrix, usually categorized by two labels: a column header and row header (stub). Example of a matrix table can be seen in Figure 4.3. In our model, these tables may have multiple layers of column or row headers. One header layer may specify the headers above it (see Figure 4.7).
- **Multi-dimensional tables** contain more than two dimensions. We identify two types of multi-dimensional table:

Clinical characteristics of the ITT sample (n = 40)

	Whole Group mean \pm SD	Group 1 (n = 16) mean \pm SD	Group 2 (n = 24) mean \pm SD	P
Age (years)	53.0 \pm 11.0	57.1 \pm 10.4	50.3 \pm 10.7	ns
Sex (female/male)	25/15	10/6	15/9	ns
Previous ECT (yes/no)	12/28	6/10	6/18	ns
Unipolar/bipolar depression (yes/no)	34/6	13/3	21/3	ns
Duration of current episode (wk) ^a	54.3 \pm 34.5	47.8 \pm 34.3	58.6 \pm 34.6	ns
Pre-ECT HDRS score	25.3 \pm 6.8	28.6 \pm 5.4	23.0 \pm 6.9	0.0093
Psychotic features (DSM-IV)(yes/no)	5/35	3/13	2/22	ns
Pre-ECT MMSE score	27.2 \pm 2.3	26.0 \pm 2.6	28.1 \pm 1.6	0.0053
Number of antidepressant trials during episode	2.2 \pm 1.4	1.9 \pm 1.2	2.3 \pm 1.5	ns
Prior adequate antidepressant treatment (yes/no)	34/6	14/2	20/4	ns

Figure 4.3: Example of a matrix table (PMC 65527)

- **Super-row tables** contain super-rows that group row headers below them (see example in Figure 4.4). 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.

		All cases (N=2,105)	Controls (N=11,500)	Adjusted RR* (95%CI)
Super-row Level 1	Formulation/dose as instructed			
Super-row Level 2	Plain			
	≤ 150 mg	201	626	1.9 (1.6-2.2)
	300 + mg	30	69	2.6 (1.6-4.2)
	Enteric-coated			
	≤ 150 mg	24	39	3.5 (2.0-6.1)
	300 + mg	32	103	1.8 (1.2-2.8)

Figure 4.4: Example of a table with tree like super-row structure. This table has two super-row levels in its stub (PMC 32172)

- **Multi-tables** are composed of multiple, usually similar tables, merged into one table. In some cases, headers of concatenated tables inherit some categorization from the header of the first table. An example of a multi-table can be seen in Figure 4.5.

4.1.2 Table data representation model

The proposed table data representation model captures necessary information for the understanding of tables to facilitate further processing and knowledge gathering. We have extended the data model by Wei et al. (2006) and the spreadsheet ontology for

Comparison of hydrocephalus CSF samples grouped according to clinical characteristics

	Congenital	Acquired	p
n	30	35	
Age (years)	0.35 (0 – 14.3)	0.4 (0 – 11.2)	> 0.1
Protein (mg/dL)	271 (36–853)	203 (14 – 1284)	> 0.1
NGF (pg/mL)	149 (< 1 – 2025)	238 (< 1 – 1876)	> 0.1
NGF/protein ratio	1.03 (0.04 – 14.82)	1.04 (0.05 – 59.09)	> 0.1
Low NGF (< 10 pg/mL)	5 (17 %)	4 (11 %)	> 0.1
Undetectable NT-3	6 /17	11 /14	0.029

	Cultures positive	Cultures negative	p
n	8	57	
Age (years)	0.35 (0.1 – 2.4)	0.4 (0 – 14.3)	> 0.1
Protein (mg/dL)	241 (88 – 1100)	248 (14 – 1284)	> 0.1
NGF (pg/mL)	473 (43 – 2025)	200 (< 1 – 1876)	> 0.1
NGF/protein ratio	0.96 (0.20 – 7.47)	1.04 (0.04 – 59.09)	> 0.1
Low NGF (< 10 pg/mL)	0 (0 %)	9 (16 %)	> 0.1
Undetectable NT-3	2 /2	15 /29	> 0.1

	First sample	Repeat sample	p
n	42	23	
Age (years)	0.4 (0 – 14.3)	0.3 (0 – 11.2)	> 0.1
Protein (mg/dL)	251 (15 – 1100)	210 (14 – 1284)	> 0.1
NGF (pg/mL)	149 (<1 – 2025)	279 (< 1 – 1876)	> 0.1
NGF/protein ratio	0.98 (0.07 – 59.09)	1.07 (0.04 – 16.43)	> 0.1
Low NGF (< 10 pg/mL)	7 (17 %)	2 (9 %)	> 0.1
Undetectable NT-3	9 /19	8 /12	> 0.1

	Pressure low	Pressure elevated	p
n	7	58	
Age (years)	1.1 (0.1 – 4.8)	0.3 (0 – 14.3)	> 0.1
Protein (mg/dL)	149 (15 – 467)	253 (14 – 1284)	> 0.1
NGF (pg/mL)	87 (< 1 – 886)	234 (< – 2025)	> 0.1
NGF/protein ratio	0.50 (0.20 – 59.09)	1.07 (0.04 – 16.43)	> 0.1
Low NGF (< 10 pg/mL)	1 (14 %)	8 (14 %)	> 0.1
Undetectable NT-3	2 /3	15 /28	> 0.1

Figure 4.5: Example of a multi-table (PMC 57003)

tables (Doush & Pontelli 2013) by adding additional entities that are not specific for navigation in screen readers, as table types, annotations and/or references to the navigational cells.

The model has article, table and cell layers (see Figure 4.6), which are arranged in a tree-like instantiation with the *article* node as the top element, containing article information (i.e. title, references, authors, text) and a list of its tables. The *article* layer also stores where tables are mentioned within the document.

The *table* layer stores caption, footer, the order of the table in the document, its structural type (dimensionality of the table, as defined in Section 4.1.1), pragmatic type and sentences referring to the table. The table node also contains a list of the table’s cells.

Finally, at the *cell* layer, the model stores the information about each cell including its value, function and position in the table. At the cell layer, the model also stores

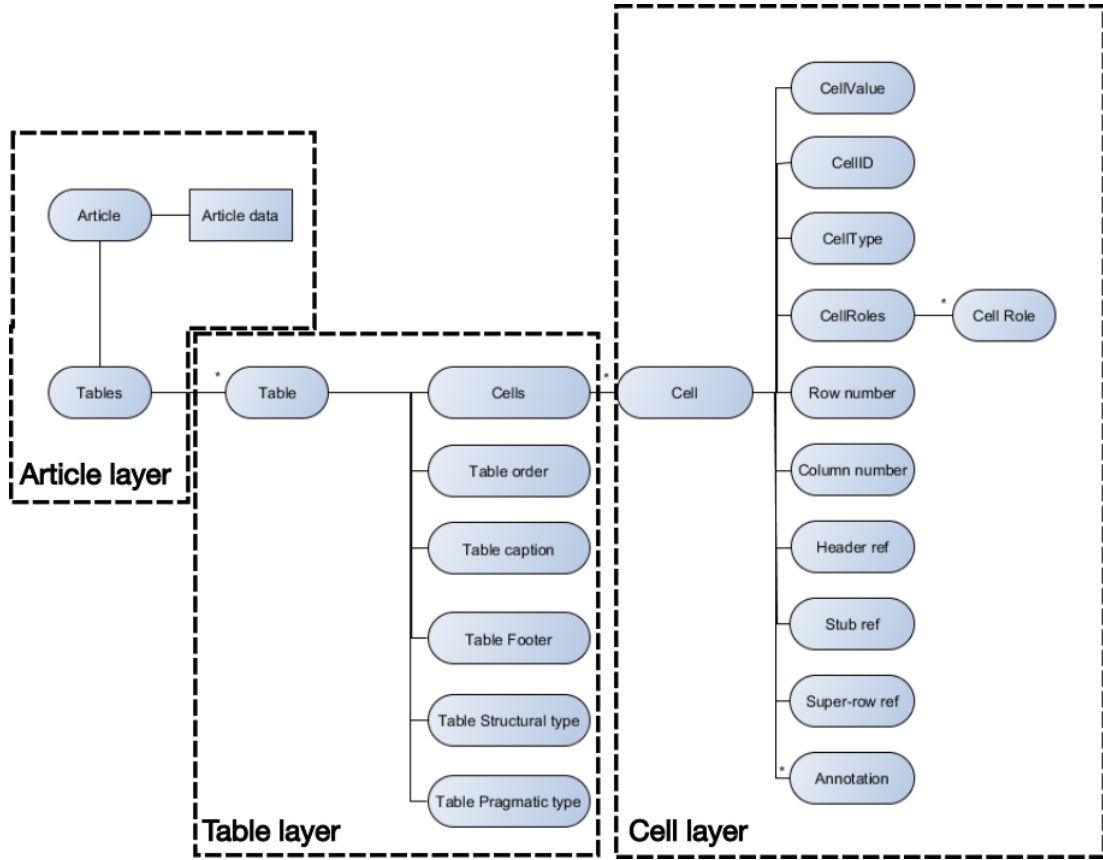


Figure 4.6: Proposed table representation model

information regarding its structural references to the navigational cells (headers, stubs, and super-rows). 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 4.7). The references to navigational cells are set by the ID of the closest cell in the navigational area. In this layer, the model further captures any possible annotations of the cell content, which might be added during the table processing. 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 (Bodenreider 2004)). For each annotation, the model can record the span positions of annotated parts in content, concept names, IDs in the lexicon or ontologies with which the text was annotated, a name of the annotation knowledge source, its version, and environment description. Name of the knowledge source, its version and environment description are information about the provenance of the annotations. In the case of annotating with multiple knowledge sources and different versions, this information can be important to distinguish the sources of annotations so

as to use the most appropriate.

Entities from the table layer are summarised in Table 4.1, while the entities from the cell layer are presented in Table 4.2.

Name of the entity	Description
Cells	A collection of table cells
Table order	Order/position of the table in the document
Table caption	The caption of the table describing the table and its content
Table footer	The footer of the table presenting additional information about the data
Table structural type	The structural type of the table (list, matrix, super-row, multi-table)
Table pragmatic type	Domain-dependent pragmatic type describing the type of information presented in the table

Table 4.1: Description of the table entities in the table data representation model

Name of the entity	Description
Cell value	The content of the cell
Cell ID	Unique ID of the cell in the table
Cell type	Content type of the cell (numeric, partially numeric, text, empty)
Cell roles	Functional role of the cell in table (header, stub, super-row, data) One cell can have multiple roles (e.g. super-row and stub)
Row number	Number of the row in which cell is located
Column number	Number of the column in which cell is located
Header reference	Reference to the closest header cell, if available
Stub reference	Reference to the closest stub cell, if available
Super-row reference	Reference to the closes super-row, if available
Annotation	Annotations of the cell and its content.

Table 4.2: Description of the cell entities in the table data representation model

We note that spanning cells in the 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.

4.2 Methodology for automatic table structure disentangling

We propose a methodology that automatically performs the functional and structural analyses 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. The overview diagram of the methodology is presented in Figure 4.8.

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

Figure 4.7: Example of cascading referencing of the header relationships (PMC 270060). The cell with the value 56 is linked to the header "Intervention", which is linked to the upper header "Pre-intervention".

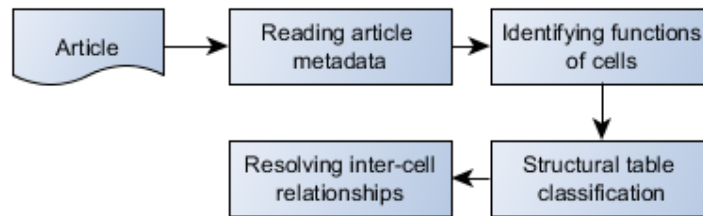



Figure 4.8: Overview of the methodology for automatic table structure disentangling

4.2.1 Reading the articles

The first step of our methodology is to read the article XML files, locate and extract article metadata and locate the tables. The methodology of locating metadata and tables in the article can be dependent on the format of the article. Therefore, our architecture contains an abstract "reader", that goes through the set of documents locating article metadata and tables. For practical reasons, the responsibility of the reader is to extract table data from the document into a data structure (cell matrix), which is then further processed. We focus on XML formats for presenting literature in the biomedical domain, and readers for PMC and DailyMed documents were implemented, because of the two case studies evaluated in this thesis. It is possible to extend the methodology to other formats by implementing other reader methods.

4.2.2 Identification of functional areas (functional analysis)

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



```

<table-wrap position="float" id="T1">
  <label>Table 1</label>
  <caption><p>Clinical characteristics of the study patients</p></caption>
  <table frame="hsides" rules="groups">
    <thead><tr><td align="left">Parameter</td><td align="center">Value</td></tr></thead>
    <tbody>
      <tr>
        <td align="left">Age (years)</td>
        <td align="center">45.1 ± 15.4</td>
      </tr>
      <tr>
        <td align="left">Males/females</td>
        <td align="center">8/2</td>
      </tr>
      <tr>
        <td align="left">Glasgow Coma Score</td>
        <td align="center">7 ± 3</td>
      </tr>
      <tr>
        <td align="left">Simplified Acute Physiology Score</td>
        <td align="center">14.7 ± 3.9</td>
      </tr>
      <tr>
        <td align="left">Injury Severity Score</td>
        <td align="center">31.2 ± 7.4</td>
      </tr>
    </tbody>
  </table>
  <table-wrap-foot><p>Values are expressed as mean ± standard deviation.</p></table-wrap-foot>
</table-wrap>

```

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

Values are expressed as mean ± standard deviation.

Figure 4.9: An example of PMC XML table and its visual representation

Header identification. Table headers can be recognized through both visual and positional features (first row, font emphasis, empty cells, spanning cells, line below it) or by semantic features. In certain datasets (e.g. PMC) it is possible to rely exclusively on visual features, while in others (e.g. DailyMed), authors may not use visual features to emphasize headers and the only way to recognize a header is to rely on the content of the table and its semantics.

Our main header detection approach relies on visual and positional emphases in the table. Headers in XML documents are often marked using a *thead* XML tag. If *thead* tag exists, we assume that it is correctly used.

We examine syntactic similarity of the cells over the column in tables that do not have annotated headers. This is performed using a window that takes five cells from the column and checks whether the content in the window 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 two 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 span over several columns. If they do, we assume that the header contains the next rows, until we reach the 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.

The previously described rules are able to detect headers in tables that emphasize headers in a certain way. For PMC articles it is common to have emphasised headers. However, not all datasets present headers in this manner. For example, drug labels presented in DailyMed database do not have emphasis features in many tables. However, since in our case studies, we were mainly interested in drug-drug interaction tables, these tables have in headers relatively standardized vocabulary (e.g. drug, effect, type of interaction, clinical class, interacting agent). The header vocabulary is rarely used in other functional areas of the table. One approach could be to extract common words used in headers (a common header vocabulary) and use them to recognize headers. A more complex and accurate approach would be to use machine learning with content features of table functional areas. However, by using a vocabulary or machine learning with features based on table content, we may lose our approach's domain independence. A domain independent approach may look at the tags' attributes. In many cases, header cells that were not labelled using *thead* tag had attribute or class of the tag pointing out that they may be a header (i.e. word "first" as a class of *td* XML tag). However, these attributes and classes are dataset specific. The presented approach is able to disentangle the majority of tables presenting numerical, factual information, even if the table is not labelled according to common practices. However, if the authors did not make a separation between header and data areas and the table presents textual data, there is a way to distinguish headers by recognizing and learning vocabulary often used in headers. In such tables, text semantics differentiates whether certain cells describe data or present data.

A methodology for handling table datasets that do not emphasize headers relies on combining the described methodology with a machine learning using the content

of the header cells. Initially, our rule-based methodology tries to perform functional analysis. Then we apply header detection using machine learning. We use machine learning algorithms to learn probable combinations of words that are prevalent in that domain as headers. The classifier takes into account words (e.g. using a bag-of-words model) from the top-most row (or multiple rows, up to three rows), as features. Since algorithm is classifying on a cell level, it may happen that cell may be classified as a different class from the majority in a given row. Rows in tables may be either fully header rows, or non-headers. We added a heuristic that detected cells classified as data cells inside a row that contained a majority of header cells and fixed their label. A similar heuristic detected cells labelled as headers in data rows.

Stub identification. We use a heuristic that marks cells that are in the left-most column as stub. However, if cells in the left-most column are row spanning, the stub area contains the next columns, until the first column with no spanning cells is identified.

Super-row identification. Super-rows are rows that group and categorize stub labels. They can have multiple layers. In order to recognize 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 non-empty content, they are labelled as super-rows.
- In some cases, spanning cells can 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.
- A table may have multiple layers of super-rows. Authors usually present a subgroup of relationships with leading blank spaces (indentation) at the beginning of the grouped elements (e.g. Figure 4.4). The number of blank spaces often determines the layer of categorisation (i.e. the first layer usually has one blank space, the second has two, etc.). In other words, the 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 categorises 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.

Cells that were not identified as headers, stubs or super-rows are labelled as data cells.

4.2.3 Identification of inter-cell relationships (structural analysis)

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 a multi-table. If it contains super-rows, it is a super-row table. If the table has only one dimension (table containing only one column or header spanned over all columns), it is a list table. Otherwise, it is matrix table.

The method further decides which inter-cell relationships to search for depending on class. 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 relationship 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. One additional relationship of data cells is with the stub-head cell, which can further describe the stub. Navigational cells are related to the higher layers of navigational cells as defined in the cascading referencing model (as explained in Section 4.1.2 and in Figure 4.7).

An example of the decomposition is presented in Figure 4.10. The decomposition of the table is a basis for further analysis. Our definition of the task makes a graph of related cells. However, this graph can be linearized, presenting table data with labels describing them in a linear representation. The linear model, also referred to by other authors as canonical representation of tables, was introduced by Wang & Wood (1993) and later used by Hurst (2000), Embley et al. (2006), Douglas et al. (1995) and others. Linear or canonical form presents data in the linear form, one line per data cell consisting of navigational cells (stub-head, headers, stubs and super-rows) related to the given data cell and the data cell.

In the presented methodology, relationships are disentangled for each cell. For each cell, outputting related labels in a certain order or format would output a table in canonical form. For the data cell coloured green in a bottom table of the Figure 4.10, canonical form will look in the following manner:

```
([sh]Parameter:)([sr]Primary diagnoses (n):)([s]Head trauma with coma:)([h]Value:)([d]8)
```

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.



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.



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.

Figure 4.10: Functional and structural analysis on an example table. The diagram shows step by step labelling of a table. During functional analysis, functional areas are labelled (header - yellow, stub - blue, super-row - orange, data - green). During the structural analysis, related cells are found for each cell. The example shows related cells of data cell with a content 8.

In the example, the output in square brackets represents cell function (i.e. sh - stub header, sr - super-row, s - stub, h - header, d - data). Next to the cell function is presented the content of the relevant cell, related to the described data cell.

However, our data model contains additional entities, such as annotations. It is also able to reflect and recreate the visual structure of tables, since the arrangement of the cells are included in the model. The proposed data model can be converted to multiple table representations, including Wang’s canonical or visual representation. The model is implemented as relational database. A database schema that reflects our data model and stores disentangled tables after functional and structural disentangling is presented in Appendix B.

4.3 Results and Evaluation

4.3.1 Datasets

We have created two datasets that reflect two different data sources. These two datasets are motivated by the case studies about information extraction in clinical trial documents and extraction of drug-drug interactions in drug labels. These case studies were briefly described in Section 3.5.

For the first case study, we collected a clinical trial dataset by filtering 2014 PMC data for clinical trial publications. Clinical trial publications are useful in drug discovery and drug management, often reporting important information in tables, such as participant characteristics, adverse events or results. Therefore, clinical trial publications were chosen as an appropriate case study. The dataset was created by mapping PMC articles with MEDLINE citation that contained word "clinical" in publication type. This dataset contains 6,109 articles with 14,009 tables. For the second case study, we created a drug-drug interaction dataset containing 1,284 structured product labels from DailyMed that had tables present in the Drug Interaction section. This dataset was created to test the methodology in a different domain and with a different document structure.

The performance of the table disentangling process has been mainly analysed from the perspective of the clinical trial dataset. However, we have also evaluated a small sample of drug label documents in order to present challenges for the table disentangling methodology when applied to the different dataset. Our evaluation has been performed on sample of 100 clinical trial documents and sample of 30 drug labels.

Clinical trial dataset

The clinical trial dataset contains 6,109 PMC articles from the clinical domain. The articles were published between 1965 and 2013. However, articles published before 1997 are not always transformed to XML. Their XML contains metadata but the body of the article is often in scanned pictures. Therefore, tables from these articles cannot be extracted without the use of OCR, which lies outside the scope of this thesis. In total there are 4,435 articles (72% of total number) that were presented with the body of the article in XML and that contain at least one table. Figure 4.11 shows the number of articles in the dataset and the number of articles presented in XML containing tables, per year of publication. From 2002 onwards, approximately 85% of the articles presented are in XML and contain tables. Between 1997 and 2002, it is apparent that the process of presenting PMC articles in XML was in an adaptation phase. From 1997 to 2000, only about 10% of articles were presented in XML. In 2001, the percentage of articles presented in XML grew to about 20%, and finally in 2002, this presentation format was adapted to the majority of PMC articles, with 83% of articles presented in this manner.

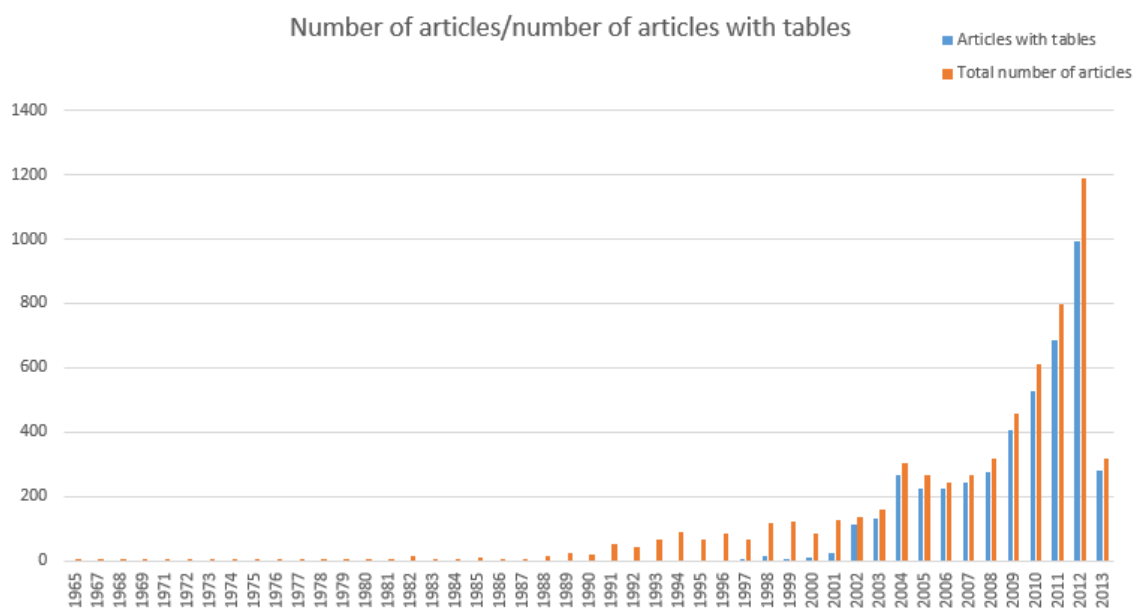


Figure 4.11: Number of PMC articles and the number of PMC articles with at least one table in XML. Statistics of PMC clinical trial dataset

There are 14,009 tables in this dataset. On average there are 3.12 (with standard deviation 0.3) tables per PMC article. Therefore, in clinical trial publications there is a relatively constant number of tables per publication over time. The yearly distribution

of tables and the yearly average number of tables per document since 1997 can be seen in Figure 4.12.

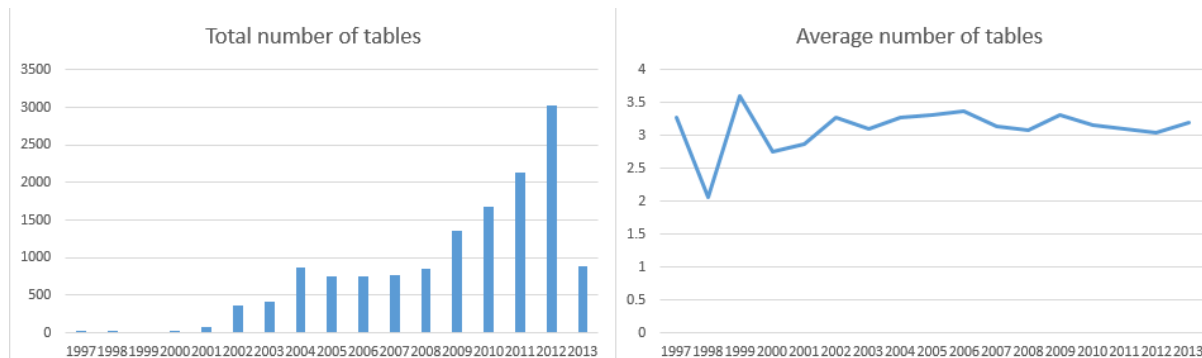


Figure 4.12: Number of tables per year and average number of tables in PMC clinical publications

Drug-drug interaction product label dataset

The drug-drug interaction product label dataset was created in order to perform a case study on drug-drug interaction information extraction. The dataset documents were downloaded from the DailyMed website¹ and only documents with tables presenting drug-drug interactions were selected. The dataset contains 1,284 structured product label (SPLs) documents published between January 2006 and January 2015. It contains 16,211 tables, where each document contains at least one table. The documents in this dataset have between 1 and 37 tables, however, over 90% of documents contain between 4 and 23 tables.

Statistical information about datasets are presented in Table 4.3.

Dataset	No. documents	No. tables	Mean	Standard deviation	Range
Clinical trials	6,109	14,009	3.12	0.32	0 - 15
Drug labels	1,284	16,211	12.62	6.55	1 - 37

Table 4.3: Statistical data about tables in the datasets. Mean, standard deviation and range are presented per document

4.3.2 Table disentangling performance

We evaluate table disentangling performance on the clinical trial and drug-drug interaction product label datasets. Our system was able to process 11,156 tables from the

¹<https://dailymed.nlm.nih.gov/>

clinical trial dataset (79.6%). The unprocessed tables were not in the XML format and processing of non-XML tables is out of the scope of this thesis. Table 4.4 presents the numbers of tables identified as belonging to different types. It is interesting that matrix tables make up over 49.7% of tables and super-row tables over 43.5%, while list and multi-table are quite rare at around 7%.

We performed the evaluation on a random subset of 30 articles containing 101 tables from PMC dataset. The evaluation sample contained tables from each table type and was manually evaluated. The detailed information about the evaluation dataset and the performance on structural table type recognition is given in Table 4.4. The evaluation was performed manually by the author.

	Overall	List tables	Matrix tables	Super-row tables	Multi-table
Number of tables	11,156	79 (0.7%)	5,546 (49.7%)	4,852 (43.5%)	679 (6.1%)
Number of evaluated	101	6	50	28	17

Table 4.4: Overview of the evaluation dataset

Table 4.5 presents the evaluation of the recognition of structural table types. We have performed evaluation on each table structural type (class). Tables that were correctly classified as the target class were considered *true positives (TP)*. Tables that were classified as the target class but were of another class were considered *false positives (FP)*, while tables that were of the target class and labelled differently were considered *false negatives (FN)*. We calculated precision, recall and F1-score. Overall, our methodology for recognition of table structure produced a micro-averaged F1-score of 0.921. The most common case of misclassification was in super-row tables, in which super-rows were separated using horizontal lines. Since horizontal lines commonly separate headers of multi-table tables, they were often sources of confusion.

Table type	TP	FP	FN	Precision	Recall	F-score
List	6	0	0	1.000	1.000	1.000
Matrix	48	1	2	0.979	0.960	0.969
Super-row	23	3	5	0.885	0.821	0.852
Multi-table	16	4	1	0.800	0.941	0.865
Overall (micro-averaged)	93	8	8	0.921	0.921	0.921

Table 4.5: Evaluation of the recognition of structural table types

The results for the functional and structural analyses are presented in tables 4.6 and 4.9. We have performed evaluation at the cell-level. Associations to the right roles and navigational relationships (headers, stubs, super-rows) were considered true positives

(TP). Association to the incorrect roles or relationships were considered false positives (FP) while missing association were considered false negatives (FN).

	TP	FP	FN	Precision	Recall	F-Score
Cell role – header	1,041	35	260	0.9670	0.8000	0.8760
List	1	0	0	1.0000	1.0000	1.0000
Matrix	469	9	0	0.9810	1.0000	0.9900
Super-row	275	18	20	0.9385	0.9322	0.9353
Multi-table	296	8	240	0.9736	0.5522	0.7047
Cell role – stub	1,250	87	22	0.9349	0.9827	0.9582
List	0	0	7	N/A	N/A	N/A
Matrix	407	1	3	0.9975	0.9926	0.9951
Super-row	488	17	4	0.9663	0.9910	0.9789
Multi-table	355	69	8	0.8372	0.9779	0.9022
Cell role – super-row	414	102	66	0.8023	0.8625	0.8313
List	12	7	0	0.6315	1.0000	0.7742
Super-row	359	26	27	0.9324	0.9300	0.9312
Multi-table	43	63	37	0.4057	0.5375	0.4624
Cell role – data	3,709	167	41	0.9569	0.9891	0.9727
List	31	7	6	0.8157	0.8378	0.8266
Matrix	1,438	1	12	0.9993	0.9917	0.9955
Super-row	1,517	11	21	0.9928	0.9863	0.9895
Multi-table	723	148	2	0.8300	0.9972	0.9060
Overall	6,414	391	389	0.9425	0.9428	0.9426

Table 4.6: Evaluation of functional table analysis on the evaluation dataset

For the functional analysis, the method archived an overall micro-averaged F1-score of 0.9426, with the lowest performance on identification of multi-table super-row areas. The results are comparable to previously reported methods. For example, Hurst (2000) combined Naive Bayes, heuristic rules and pattern-based classification archiving an F1-score of around 0.92 for functional analysis on tables in ASCII format on general domain. Similarly, Tengli et al. (2004) reported an F1-score of 0.914 for the table extraction task in which they recognized labels and navigational cells from tables in Common Data Set (CDS) retrieved from the websites of the Universities, while Wei et al. (2006) reported an F1-score of 0.9 for detecting headers using CRF on the general domain. Cafarella, Halevy, Wang, Wu & Zhang (2008) detected navigational cells with precision and recall not exceeding 0.89 and Jung & Kwon (2006) reported 0.821 accuracy extracting table headers, both evaluating on tables from the web in HTML format.

We have tested this approach on other dataset, namely on drug label documents

available on DailyMed. Table 4.7 presents the results of the functional analysis of 20 randomly selected drug-drug interaction tables from the drug labels dataset, as described in Section 4.3.1.

	TP	FP	FN	Precision	Recall	F-Score
Cell role – header	61	39	32	0.6100	0.6559	0.6321
Cell role – stub	309	0	0	1.0000	1.0000	1.0000
Cell role – super-row	49	6	45	0.8909	0.5213	0.6578
Cell role – data	675	18	104	0.9740	0.8664	0.9171
Overall	1,094	63	181	0.9455	0.8580	0.9014

Table 4.7: Evaluation of functional table analysis on 20 drug-drug interaction tables from DailyMed

The header and super-row detection results were significantly lower than those for the PMC dataset, for which the approach was initially developed. There were 8 tables in which the header was not recognized at all. These tables did not have any separation between the header and the data area, no header XML tag, while headers and the cells below them were textual (see example in Figure 4.14). In one table, our methodology marked all the cells in the table as header cells (due to syntactic similarity of the cells). There were 125 tables (about 10% of tables from the DailyMed dataset) that contained *thead* tag, but in the tagged area they presented a caption (see Figure 4.13). In those tables, often the second row – the first not marked as part of the heading area of the table – was the actual table header. Overall performance was not affected much, since stub cells were recognized in all cases. The DailyMed tables we analysed typically had simpler stub structures than PMC tables. This is because all the tables analysed were matrix, super-row or multi-tables with only one stub column.

We created a learning dataset by extracting headers classified by our initial methodology. The headers were then manually checked and labelled as headers or non-headers. This data was then used to train and test a machine learning classifier (using 10-fold cross validation). The dataset contained 600 cell instances, 300 labelled as headers and 300 labelled as non-headers. The machine learning approaches is compared to the simple baseline approach that labels only the first row in the table as header. In Table 4.8, we present the results of a machine learning algorithm for classifying headers of drug-drug interaction tables from the DailyMed drug labels.

The baseline approach is recognising the header with an F1-score of 0.595. On the other hand, the rule-based methodology detected headers with an F1-score of 0.632, while the random forest algorithm produced an F1-score of 0.922. As this algorithm

Table 4 Drugs Tested in <i>In Vitro</i> Binding or <i>In Vivo</i> Drug Interaction Testing or With Post-Marketing Reports	
Drugs with a known interaction with colesevelam	Cyclosporine ^c , glyburide ^a , levothyroxine ^a , and oral contraceptives containing ethinyl estradiol and norethindrone ^a
Drugs with postmarketing reports consistent with potential drug-drug interactions when coadministered with WELCHOL	phenytoin ^a , warfarin ^b
Drugs that do not interact with colesevelam based on <i>in vitro</i> or <i>in vivo</i> testing	cephalexin, ciprofloxacin, digoxin, warfarin ^b , fenofibrate, lovastatin, metformin, metoprolol, pioglitazone, quinidine, repaglinide, valproic acid, verapamil

Figure 4.13: Example of the DailyMed table containing caption in the header cell.
Document SetID: a7a2a4e1-9ecd-4e59-82b5-2068b5e50164

```

<table border="1" cellpadding="2" cellspacing="1">
<caption ID="TableII">Table II. Clinically significant drug interactions with theophylline*.</caption>
<colgroup>
<col />
<col />
<col />
</colgroup>
<tbody>
<tr valign="bottom"><td><content styleCode="bold">Drug</content></td>
<td><content styleCode="bold">Type of Interaction</content></td>
<td><content styleCode="bold">Effect**</content></td></tr>
<tr valign="top"><td>Adenosine</td><td>Theophylline blocks adenosine receptors.</td>
<td>Higher doses of adenosine may be required to achieve desired effect.</td>
</tr>
<tr valign="top"><td>Alcohol</td><td>A single large dose of alcohol (3 mL/kg of whiskey) decreases theophylline clearance for up to 24 hours.</td>
<td>30% increase</td></tr>
<tr valign="top"><td>Allopurinol</td><td>Decreases theophylline clearance at allopurinol doses ≥600 mg/day.</td>
<td>25% increase</td></tr>
<tr valign="top"><td>Aminoglutethimide</td><td>Increases theophylline clearance by induction of microsomal enzyme activity.</td>
<td>25% decrease</td></tr>
<tr valign="top"><td>Carbamazepine</td><td>Similar to aminoglutethimide.</td>
<td>30% decrease</td></tr>
<tr valign="top"><td>Cimetidine</td><td>Decreases theophylline clearance by inhibiting cytochrome P450 1A2.</td>
<td>70% increase</td></tr>
<tr valign="top"><td>Ciprofloxacin</td><td>Similar to cimetidine.</td>
<td>40% increase</td></tr>
<tr valign="top"><td>Clarithromycin</td><td>Similar to erythromycin.</td>
<td>25% increase</td></tr>
<tr valign="top"><td>Diazepam</td><td>Benzodiazepines increase CNS concentrations of adenosine, a potent CNS depressant, while theophylline blocks adenosine receptors.</td>
<td>Larger diazepam doses may be required to produce desired level of sedation. Discontinuation of theophylline without reduction of diazepam dose may result in respiratory depression.</td>
</tr>
</tbody>
</table>

```

Drug	Type of Interaction	Effect**
Adenosine	Theophylline blocks adenosine receptors.	Higher doses of adenosine may be required to achieve desired effect.
Alcohol	A single large dose of alcohol (3 mL/kg of whiskey) decreases theophylline clearance for up to 24 hours.	30% increase
Allopurinol	Decreases theophylline clearance at allopurinol doses ≥600 mg/day.	25% increase
Aminoglutethimide	Increases theophylline clearance by induction of microsomal enzyme activity.	25% decrease
Carbamazepine	Similar to aminoglutethimide.	30% decrease
Cimetidine	Decreases theophylline clearance by inhibiting cytochrome P450 1A2.	70% increase
Ciprofloxacin	Similar to cimetidine.	40% increase
Clarithromycin	Similar to erythromycin.	25% increase
Diazepam	Benzodiazepines increase CNS concentrations of adenosine, a potent CNS depressant, while theophylline blocks adenosine receptors.	Larger diazepam doses may be required to produce desired level of sedation. Discontinuation of theophylline without reduction of diazepam dose may result in respiratory depression.

Figure 4.14: Example of the part of the table presenting drug-drug interactions from the DailyMed dataset. Document SetID: 6C08B50E-CC9F-4C49-D7AE-F0FDDCB10199

classified the cell level, certain cells in header rows were classified as headers, while some cells in data rows were also classified as headers. We also manually evaluated header detection for 50 random drug-drug interaction tables. The manual evaluation resulted in a slight drop of performance. The algorithm performed with 0.748 precision and 0.871 recall producing a 0.805 F1-score. These were still significantly better results than our initial, rule-based methodology for the drug label dataset. Consequently, we used these algorithms in our drug-drug interaction extraction case study,

	Precision	Recall	F-Score
Baseline	0.894	0.447	0.596
Naive Bayes	0.796	0.778	0.775
Bayesian Networks	0.797	0.727	0.709
SVM with SMO	0.917	0.913	0.913
C4.5 decision tree	0.715	0.715	0.715
Random forest	0.923	0.922	0.922

Table 4.8: Evaluation of header classifiers based on content for detecting header in drug-drug interaction tables

as presented in Section 7.2.

The results of structural table analysis is presented in Table 4.9.

	TP	FP	FN	Precision	Recall	F-Score
References – header	5,402	768	47	0.8755	0.9913	0.9298
List	7	0	0	1.0000	1.0000	1.0000
Matrix	2,076	15	3	0.9930	0.9985	0.9960
Super-row	2,501	61	6	0.9761	0.9863	0.9895
Multi-table	818	692	38	0.5417	0.9556	0.6915
References – stub	4,982	147	0	0.9710	1.0000	0.9855
Matrix	1,788	14	0	0.9922	1.0000	0.9961
Super-row	2,057	95	0	0.9558	1.0000	0.9774
Multi-table	1,137	38	0	0.9670	1.0000	0.9835
References – Super-row	1,663	78	269	0.9552	0.8607	0.9055
List	29	0	6	1.0000	0.8285	0.9062
Super-row	1,456	66	215	0.9566	0.8713	0.9112
Multi-table	178	12	42	0.9368	0.8091	0.8682
Overall	12,047	993	316	0.9238	0.9744	0.9484

Table 4.9: Evaluation of structural table analysis on the evaluation dataset. Inter-cell relationships are evaluated.

For the structural relationships, we have counted the number of relationships a cell has. Each cell can have at most one relationship with the header, the stub and the super-row. Evaluation has been done for each class. The system achieved an overall micro-averaged F1-score of 0.9484 for the structural analysis task (see Table 4.9). By comparison, system by Hurst (2000) scored 0.8121 recall and 0.8514 precision (on general domain). It is important to note that input data in Hurst’s system were perfectly formatted, while the PMC data is sometimes not. To the best of our knowledge, there is no other system that attempted to perform the combined task of functional and structural table analysis.

Error analysis

During the error analysis, we identified misleading markup and complex tables unique to a specific paper in the evaluation set as the main reasons for errors. In PMC documents, XML markup features such as spanning cells, head tags and breaking lines are often misused to make tables look visually appealing. In some tables for example, a head tag is used to label only the first row of a multi-row header, while a horizontal line divides the actual header from the body of the table (see example in Figure 4.15). Although we have applied heuristics that can overcome some of the issues, some of the misleading XML labelling remains challenging.

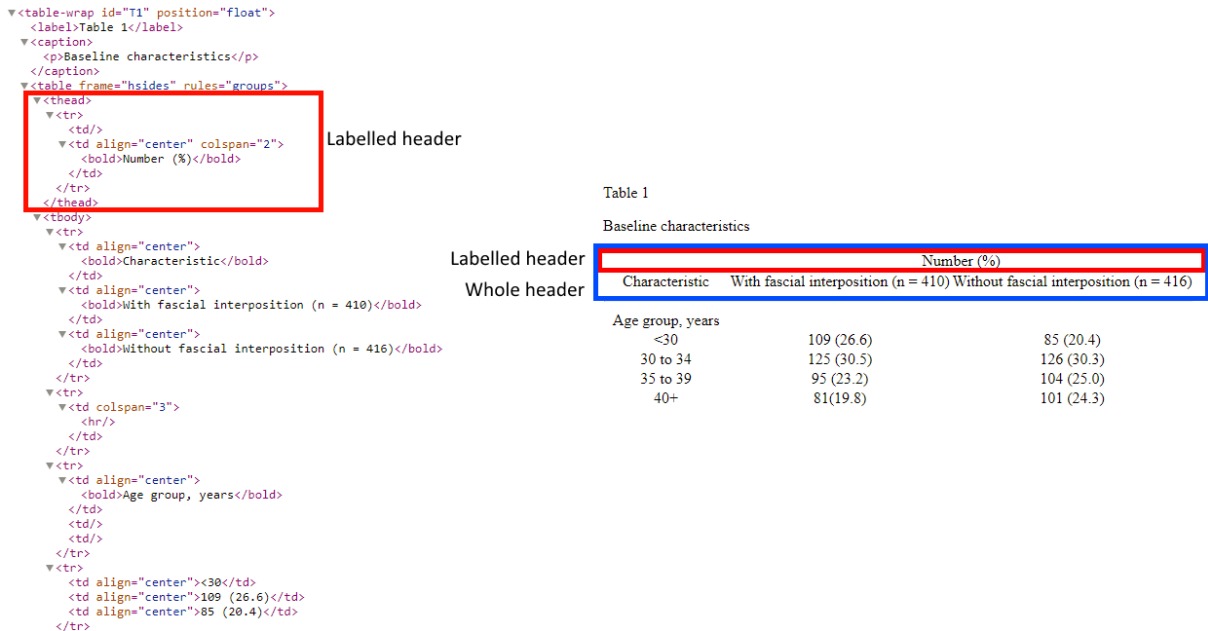


Figure 4.15: Example of the XML and table having mislabelled header (PMC 406425)

Furthermore, there are table structures that are not only complex but also their structure is unique to a specific paper and thus difficult to generalize.

Our method made a 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 the XML structure (F1-score for header detection in multi-tables was 0.69). Example of table with misclassified cells due to multiple horizontal lines can be seen in Figure 4.16. Errors recognizing headers or super-rows cause a large amount of consequent false links in structural analysis, since relationships in the subsequent rows are wrongly annotated. However, multi-tables are relatively rare so this did not heavily affect the overall results.

Table 3.

Postoperative complications.

	Group I Open Varicocele (65)	Group II Lap Varicocele (128)
Postoperative pain (> 12 years old)	(56 patients)	(94 patients)
• No narcotic inject.		
• 1 injection	46 (82%)	82 (87%)
• 2 injections	10 (18%)	11 (11.7%)
• > 2 injections	-	1 (1.1%)
Wound erythema	6 (9.2%)	3 (2.3%)
Wound infection	1 (1.5%)	1 (0.8%)
Hydrocele	3 (4.6%)	3 (2.3%)
Recurrent varicocele	7 (10.8%)	5 (3.9%)

Figure 4.16: Example of the table that was falsely classified as multi-table due to the presence of horizontal lines. The table is actually a matrix table. (PMC 3381636)

Limitations

The presented approach also has some limitations. The approach is developed for XML documents and it embeds the tags and attributes that are used in the particular dataset. It assumes that the majority of tables will be marked with appropriate header tags or that there will be a clear differentiation between the header and the data area, such as a horizontal line or a change in cell type (for example from textual to numeric). However, this is not the case even with all table data presented in the PMC XML format. Also, drug-drug interaction tables presented in drug label documents available through the DailyMed website do not follow this assumption.

In the datasets that do not emphasise functional areas of the table in its XML format, it is possible to use machine learning. However, using machine learning, the approach is no more domain independent. Machine learning approach learns what words are often used in particular functional areas in the given document subset or sub-domain. From the presented results, it can be concluded that machine learning can help in recognising functional areas of the table, however, in order to be effective, it has to be trained on a relatively narrow domain. Machine learning approaches can generalise over multiple document stores, however, not over multiple domains. On the other hand, rule-based approaches based on the layout and emphasis features (e.g. font size, boldness, arrangement of lines, etc.) can be generalised over multiple domains, but it is document store specific. Therefore, an approach can be either generalized for

multiple domains in a single document store or over multiple document stores in one domain. Unfortunately, different document stores from the same publishers may utilize different attributes and tags in tables. Examples of this include the PMC and the DailyMed document stores, which are both maintained by the US National Library of Medicine.

4.4 Summary

In this chapter, we have presented a model to computationally represent tables found in biomedical scientific literature. We also presented a domain-independent methodology to disentangle tables and add annotations to functional areas (functional analysis) and relationships between table cells based on table structure and emphasis features (structural analysis). In performing functional and structural analysis, the method transforms tables from a presentation to the structured format in which information presented in tables can be queried, analyzed and mined. Also, data in the structured format can be easily transformed to a canonical form as described by Wang & Wood (1993) and later used by Hurst (2000), Embley et al. (2006), Douglas et al. (1995).

The evaluation has shown that the table structure can be identified with high F1-scores (above 0.94 on the PMC clinical trial documents, around 0.90 on DailyMed drug-drug interaction table dataset). Even though we performed the main evaluation on the PMC clinical trial documents, the proposed approach can be extended to include the DailyMed documents, HTML or any other XML-like format. However, the method is currently limited to XML formats. Other formats, such as PDF are possible to process using this methodology after they have been converted to XML format.

The method can be extended for datasets that have to use a different approach for some part (such as approach based on lexical cues or semantics of the data in order to distinguish functional areas during functional analysis). Usually, XML documents contain emphasis features in some form. However, each document store may use different tags or attributes in order to achieve emphasis (e.g. visual interpretation of attributes or classes may be defined in a separate file, such as a css file). Therefore, a reader for each dataset has to be developed. Alternatively, machine learning with lexical and semantic features can be applied to help distinguishing functional areas. However, machine learning has to be trained on a narrow domain in order to be effective. Therefore, the methodology can be either developed and generalised for multiple

domains in a single document store (using emphasis and layout features) or for multiple document stores within a single narrow domain (using machine learning and lexical or semantic features).

The proposed model can serve as a basis to support information retrieval, information extraction and question-answering applications and assist visually impaired people to read tables. The following chapters will present a general methodology for information extraction from tables in biomedical literature that uses the presented approach. The methodology can also be used as a basis for semantic analysis and querying of tables. For example, screen readers for visually impaired people could enable easy navigation through tables by providing information about a cell's relationships and functions.

Chapter 5

Table and cell annotation

Table processing tasks, such as information extraction from tables require a multi-layered approach, consisting of multiple processing steps or layers. In the previous chapter, we examined functional and structural processing, that disentangle a table's structure and lay the foundation for further table mining tasks.

In this chapter, we examine further annotation steps. Table and cell content annotations normalise table data, map table data to a certain knowledge source or classification and reduce search space for further tasks (e.g. information extraction, information retrieval or question answering). Similarly to text annotation (Aronson & Lang 2010), annotation of the table content is domain dependent, but it is task independent. Table annotations can be useful in multiple table mining tasks. In this chapter, we examine the pragmatic analysis (pragmatic processing) and cell content annotation using knowledge sources – semantic tagging.

The idea of pragmatic analysis is to classify tables based on the information they present. Certain variables are often grouped together across the documents. For example, authors usually group into one table information presenting baseline characteristics, adverse reactions, inclusion and exclusion criteria or results of the the experiment. This grouping can be identified, used for narrowing the scope for information extraction and reducing false positive matches. It can be also reused for multiple information extraction tasks.

On the other hand, the main idea of the semantic tagging of the cell content is to map content of cells to the concepts in the domain specific knowledge sources. By doing this, table content is normalised (e.g. various synonyms or different terminologies are mapped to the same concept) and by using ontologies content obtains an additional semantic layer that can be further exploited in the later tasks (e.g. filtering by higher

level concept in ontology or semantic type).

5.1 Pragmatic analysis

Pragmatics is the study of how context and the way information is communicated contributes to meaning (Leech 2016). In the case of tables in the literature, we consider pragmatics to analyse the author’s intentions regarding the context and the purpose of the table.

Usually, authors intentionally group certain information, such as demographic information, adverse events or inclusion and exclusion criteria. The main purpose of pragmatic analysis is to identify a target table where the information of interest is located, narrowing the search space for information of interest and reducing the number of false positives. Generally, pragmatic analysis can be seen as a task of finding topics (topic modelling) of the information presented in a table. Since the classes (topics) are associated with the whole table, we design table pragmatic analysis as a table level annotation task.

Pragmatic analysis of the tables can be performed using rule-based and machine learning approaches, depending on the structure of the analysed documents. For example, in drug labels, it is possible to develop rules that will select only tables in a certain section (e.g. *drug interactions*, *adverse reactions*, *dosage and administration*, etc.). The drug labels are well structured into topic related sections where relevant tables can be found. However, in different scientific publications tables presenting the same variable group (e.g. baseline characteristics, or adverse reactions) can be in different sections. Therefore, it is not possible to select relevant tables based on rules.

For documents where it is challenging to develop a rule-based approach for pragmatic analysis, we propose a machine learning classification method that analyses captions and the variables presented in a table, with an aim to determine the purpose of a given table and the types of information stored in it. Since the proposed methodology utilises supervised machine learning, firstly, it is necessary to define the classes of tables and manually annotate a set of table to be used as training set. The classes of tables should reflect variable groups that are commonly presented together (e.g. baseline characteristic variables, such as number of patients, their age, gender, weight, height, body mass index are commonly presented in one table). Defining pragmatic table classes may take into consideration potential future tasks (e.g. information extraction or retrieval). Once the pragmatic classes are defined, it is necessary to annotate

a set of tables that are later used for training machine learning method. The machine learning considers content of the caption, footer, referring sentences to table, cells and their function as features.

5.1.1 Pragmatic analysis case study

Table 2		
Demographic and Medical Data		
	Mean \pm SD	Range
Gender (% female) (n = 25)	83%	
Education (years)	13.6 \pm 3.0	6 to 20
Age (years)	49.7 \pm 13.9	20 to 69
Time Since Diagnosis (years)	1.8 \pm 1.5	0.8 to 5.3
PaO ₂ mmHg	62.6 \pm 13.5	38 to 97
Most recent 6 minute walk (meters)	455 \pm 132	227 to 877
New York Heart Functional Class (N)		
Class 1	0	
Class 2	3	
Class 3	20	
Class 4	0	
Supplemental Oxygen (N)		
2 Liters per minute	7	
3 Liters per minute	8	
4 Liters per minute	4	

Table 4			
Adverse Injection Site Events			
Adverse Event*	Mild	Moderate	Severe
Bruising	2	0	0
Inflammation	1	1	0
Mass	3	3	1
Pain	3	6	1
Pigmentation Change	2	1	0
Site Reaction NOS	2	1	0

Table 1	
Inclusion and exclusion criteria of the study	
Inclusion criteria	Exclusion criteria
Female sex	Pregnancy
Age older than 21 years	Recurrent disease
Not pregnant	Previous mastectomy
Suspicious lesion of the breast (on palpation or mammography)	Fine needle aspiration within 1 week prior to scintimammography
Recommendation for excisional biopsy after mammography	Core biopsy during the previous 4 weeks
Informed consent of the patient	Previous chemotherapy
	Medically unstable patient (severe arrhythmia, heart failure or recent surgery)

Table 1	
Neurocognitive Battery for Telephone Administration	
COGNITIVE DOMAIN	TEST INSTRUMENT
Attention	WMS-III: Digits Forward
Concentration/Working Memory	WMS-III: Digits Backward
	WMS-III: Letter-Number Sequencing
Executive Function	Hayling Sentence Completion Test
Reasoning	WAIS-III: Similarities
Language / Crystallized Intelligence	WAIS-III Vocabulary
Memory	WMS-III: Logical Memory I & II

Figure 5.1: Examples of tables for each pragmatic class defined in a clinical trial case study

In order to test and evaluate how different parts of the table contribute to pragmatic classification, we designed a case study of clinical trial articles. Since the information extraction case study, mentioned in Section 3.5, considers mainly extraction of baseline characteristics (patient number, age, FEV1, PEF, etc.) and adverse reactions, table classes reflect the requirements of the case study. Possible tables classes for the clinical trial case study are "baseline characteristics", "adverse events", "inclusion/exclusion criteria" and "other" (see examples in Figure 5.1). Our dataset for pragmatic classification contained 186 tables labelled as baseline characteristic tables,

60 inclusion/exclusion criteria tables, 239 adverse event tables and 153 classified as others. The tables were randomly selected from the PMC clinical trial dataset and were manually reviewed and classified by the author. As features, we used words and semantic annotations from the caption, column; row headings and sentences referring to a table, the number of rows and number or columns that the table has. A number of machine learning algorithms were tested, including Naive Bayes, SVM, decision trees, random tree and random forest in Weka toolkit (Hall et al. 2009). The evaluation was performed using the 10-fold cross-validation.

Furthermore, we analysed how the content of various table areas (headers, stubs, data, super-row, caption) and referring sentences contribute to pragmatic classification. For this, we developed separate classifiers for each feature and evaluated them. We have also created a classifier that combines the proposed features.

5.1.2 Evaluation

The results of the pragmatic classification experiments are presented in Table 5.1.

Caption and stubs are good features for the pragmatic analysis. This is expected as a caption's purpose is to describe the table and its content. Caption often describes what information is grouped together. On the other hand, stubs contain concept names that, when grouped together, can help identify a table's pragmatic type. Other features were not as successful in predicting pragmatic type. The header information usually contains names of clinical arms or drugs, which is not as relevant for the pragmatic analysis. When a header is used only as a content feature, it achieves F1-scores between 0.618 and 0.66 depending on the algorithm used. The data cells' content presents concept values, but little can be concluded from these values without the descriptions from a table's navigational areas. As expected, F1-scores for data cells as the only content feature are lower, in a range of 0.551-0.587. Referring sentences sometimes describe tables but more often, they analyse or compare the results or just refer to the table (e.g. "See table X"). From the analysis, comparison or reference often cannot infer the purpose of the table without additional information. Referring sentences' F1-scores range between 0.573 and 0.626. Super-rows can be as good classification feature as stubs (of which they are a part) however, many tables do not contain super-rows. Therefore, when only super-rows are used as content features, the F1-score produces a range of 0.373 - 0.490.

The final classifier used some of the content features, such as stub, caption, header content and quantitative features, such as number of columns, number of rows and

Algorithm	Precision	Recall	F-Score
Caption text			
Naive Bayes	0.901	0.902	0.901
Bayesian Networks	0.907	0.905	0.906
SVM	0.930	0.930	0.930
C4.5 Decision tree	0.926	0.925	0.926
Random Forests	0.889	0.889	0.888
Header text			
Naive Bayes	0.687	0.654	0.660
Bayesian Networks	0.682	0.634	0.642
SVM	0.648	0.631	0.635
C4.5 Decision tree	0.659	0.612	0.620
Random Forests	0.646	0.628	0.618
Stub text			
Naive Bayes	0.821	0.796	0.801
Bayesian Networks	0.841	0.802	0.807
SVM	0.808	0.772	0.776
C4.5 Decision tree	0.821	0.779	0.783
Random Forests	0.803	0.776	0.780
Super-row text			
Naive Bayes	0.568	0.477	0.461
Bayesian Networks	0.696	0.440	0.490
SVM	0.526	0.448	0.373
C4.5 Decision tree	0.691	0.508	0.476
Random Forests	0.694	0.537	0.514
Data cell content			
Naive Bayes	0.573	0.556	0.551
Bayesian Networks	0.572	0.568	0.567
SVM	0.604	0.586	0.587
C4.5 Decision tree	0.560	0.551	0.551
Random Forests	0.603	0.592	0.587
Referring sentence			
Naive Bayes	0.726	0.590	0.618
Bayesian Networks	0.698	0.618	0.625
SVM	0.682	0.625	0.626
C4.5 Decision tree	0.630	0.575	0.573
Random Forests	0.675	0.622	0.617
Combined content features			
Naive Bayes	0.873	0.871	0.872
Bayesian Networks	0.865	0.864	0.864
SVM	0.915	0.914	0.914
C4.5 Decision tree	0.883	0.880	0.881
Random Forests	0.917	0.915	0.916

Table 5.1: Weighted averages for all classes of the pragmatic classification using different content feature sets (each content feature separately and all features combined). The evaluation was done using 10-fold cross validation.

order of the table in the article. We used an SVM classifier with the results published in Table 5.2.

Algorithm	Precision	Recall	F-Score
Naive Bayes	0.943	0.943	0.943
Bayesian Networks	0.938	0.939	0.938
C4.5 decision trees	0.944	0.945	0.944
Random tree	0.905	0.903	0.904
Random Forests	0.948	0.948	0.948
SVM	0.967	0.966	0.966

Table 5.2: Results of the four-class pragmatic classification experiments on the PMC clinical trial tables. Training and evaluation was performed using the 10-fold cross-validation on 186 "baseline characteristic", 60 "inclusion/exclusion", 239 "adverse event" and 153 "other" tables.

The PMC clinical trial dataset has 6,558 articles containing 12,787 tables. The distribution of tables, according to our pragmatic classification model, is presented in Table 5.3. Articles often present tables with participant baseline characteristics. On the other hand, inclusion and exclusion criteria are rarely presented in tables (more often found in text rather than tables).

Table type	Number of table (percent)
Baseline characteristics	2,803 (21.92%)
Adverse events	633 (4.95%)
Inclusion/Exclusion	82 (0.47%)
Other	9,291 (72.66%)

Table 5.3: Distribution of tables in PMC clinical trial dataset based on their pragmatic class

The pragmatic classification in this case study is crafted towards information extraction of baseline characteristic, inclusion/exclusion and adverse event variables. The majority of table fall under "other" class (72.6%), however, this is not a problem, as the tables needed for the further information extraction step are tagged. The number of classes can be extended in case other information is required for extraction.

We also tested an approach in which we defined the pragmatic classes more broadly ("*experimental settings*", "*experimental results*", "*supporting knowledge*" that included literature review, definitions of scales, terms or examples, and "*others*"). Adopting this approach, the machine learning algorithm's best performance produced an F1-score of 0.85, approximately 10% worse than specifically defined pragmatic classes.

The broader classes have a larger vocabulary of terms and cues used in tables, which makes it more difficult to learn class terminology.

5.2 Semantic tagging

The purpose of semantic tagging is to enrich the content of a cell by mapping its content to a concept in a knowledge source such as an ontology, domain specific vocabulary or terminology. Enriching cell content with concepts from knowledge sources normalizes the cell content by mapping synonyms and different terminological items to the same concept in the knowledge source. Further, semantic tagging enriches data with information detailing relationships between terms and concepts, as defined in a given knowledge source. This allows knowledge-driven text mining approaches to be developed. Semantic resources allow for normalization at more general levels of representation (Nédellec & Nazarenko 2005). For example, it is possible to query or filter cells with more general concepts than those presented in the table (e.g. a pharmacological substance is hierarchically above particular names of drug substances, so it is possible to select all pharmacological substances without naming them). Mapping of the content to ontology, domain specific terminologies and vocabularies prove to be significantly useful for further text and table mining tasks, especially information extraction and knowledge discovery (Xu et al. 2010, Mulwad et al. 2013, Limaye et al. 2010).

In order to perform semantic tagging of table cells, we developed a semantic tagging methodology that utilizes pre-existing knowledge sources. At the time of writing, the developed method supports tagging with UMLS (Bodenreider 2004), DBPedia (Lehmann et al. 2015), WordNet (Miller 1995) and vocabularies in Simple Knowledge Organisation System (SKOS) format (Bechhofer & Miles 2009). The tagging methodology iterates through the table cells and tags the content with a selected knowledge source. When UMLS is applied, the method sends the content of the cell to the MetaMap server (Aronson 2001), which returns annotations. Word sense disambiguation is performed by the MetaMap. When DBPedia is applied, the method queries defined SPARQL interface. The queries are made for unigrams, bigrams and trigrams. Longer sequence of tokens are not queried, as they are rare in DBPedia and it significantly increases speed of the tagging.

SKOS vocabularies and WordNet are similarly queried. However, if there are multiple entities for the same sting, a modification of the Lesk algorithm (Banerjee &

Pedersen 2002) is used in order to disambiguate. Lesk algorithm looks at the definition of the concept and the context window around the term in the text. It counts the words that appear in both the context and the definition. Disambiguation is typically performed by selecting the concept that has the highest count of words appearing in both the context and the definition. However, this may be biased towards the concepts with longer definitions. Therefore, we divide the number of matching words with the number of tokens in the definition and disambiguate based on the value of this ratio.

Since we perform semantic tagging with already existing knowledge sources, that have been evaluated on a number of applications, we use them as they are and do not provide additional evaluation.

5.3 Conclusion and summary

In this chapter, we described a methodology for annotating tables by utilising pragmatic table classification and semantic tagging. The purpose of table and cell level annotations is to enrich data presented in a table with semantic knowledge from a given semantic resource. The enriched information can be used for normalisation and advanced semantic querying of tabular data. Querying may take into account relationships between entities and terms in a semantic knowledge source. Pragmatic annotations and semantic tags can contribute significantly to the performance of text mining tasks, such as information extraction, information retrieval and question answering. Such annotation can significantly contribute to the simplification of information extraction rules (because of the semantic generalisation that can be exploited).

For pragmatic analysis, we have proposed a machine learning methodology that uses table content as features. The content of caption and stub cells contributes most to the effective pragmatic analysis. Caption describes a table, while stub cells enumerate variables that are presented in the table, therefore these areas describe a table in pragmatic terms. The performance of pragmatic classification was dependent on the specificity of pragmatic classes: the more specific class - the better classifier.

Semantic tagging and pragmatic analysis are optional steps for annotation of tables and cells. In many cases, it is possible to perform text mining tasks, such as information extraction, relying solely on lexical cues and rules. However, in other cases, semantic tags and pragmatic classification of a table can significantly contribute to performance and simplification of the rules. Rules can be simplified by exploiting semantic relationships between the entities (e.g. using a high level entity, instead of

naming all possible target entities and their lexical variations). Utilisation of semantic tagging and pragmatic analysis depends on the data and performed task.

Table and cell annotation steps are domain dependent, as they use domain knowledge and ontologies to tag data. Previous steps, grouped to table disentangling, are usually domain and task independent. Table and cell annotations are still task independent, since the same annotations can be used for multiple tasks, including information extraction, knowledge discovery, information retrieval and question answering. In the next chapter, we discuss task-dependent processing steps.

Chapter 6

A framework for information extraction from tables

This Chapter describes a framework for information extraction from tables in the biomedical literature. As stated in Chapter 2, multiple surveys indicated that methods developed for information extraction from text under-performed when applied to tables. This is because the structure of a table, functional areas in the table and relationships between cells play roles in understanding information in tables. Consequently, a specific methodology for information extraction from tables is required.

In Chapter 3, we gave overview of 7 steps of the methodology for information extraction from tables: (1) Table detection, (2) Functional analysis, (3) Structural analysis, (4) Pragmatic analysis, (5) Semantic tagging, (6) Cell selection and (7) Syntactic analysis and information extraction. In this chapter, we explore the last two steps of the methodology - cell selection and syntactic analysis with information extraction. At the end, the methodology extracts information to populate the proposed template, that was described in Section 3.3.4:

(VariableName, VariableSubCategory, ValueComponent, Context, Value, Unit)

Firstly, we explore two possible approaches for selecting the cells that contain a target variable - machine learning-based and rule-based. We evaluate these two methodology over a set of case studies for extraction of baseline characteristic variables, such as number of patients, age, gender and adverse events. We experimented with numerical (*number of patients, age, gender distribution*) and categorical (*adverse event*

names) variables. At the end, we generalise findings from the provided exploratory case studies into a framework for information extraction from tables.

The framework presented in this chapter is based on rules that take into account a table's structure, arrangement of functional areas, variable types and common syntactic value presentation patterns. Table authors in our domain often use common patterns and structures to present certain variables (e.g. statistical variables, numeric variables). For example, if users want to present the mean value and standard deviation, there is a set of commonly used patterns (e.g. 15 ± 2 , $15 (2)$). Similarly, lexical cues are usually from a closed set and presented only in certain functional areas of a table. Therefore, we hypothesise that these patterns can be modelled and rules that take into account these patterns can be reused or modified for other similar variables.

6.1 Cell selection and syntactic analysis

In the last part of the methodology, cells are analysed and information is extracted. Information extraction has two sub-steps. In the first, relevant cells are selected by analysing whether they contain the target variable. This analysis step can be performed by using either heuristics or machine learning. The second step performs analysis over the cell's value, disentangles the components of the presented values, and extracts them by filling the extraction template. The diagram of our methodology is presented in Figure 6.1.

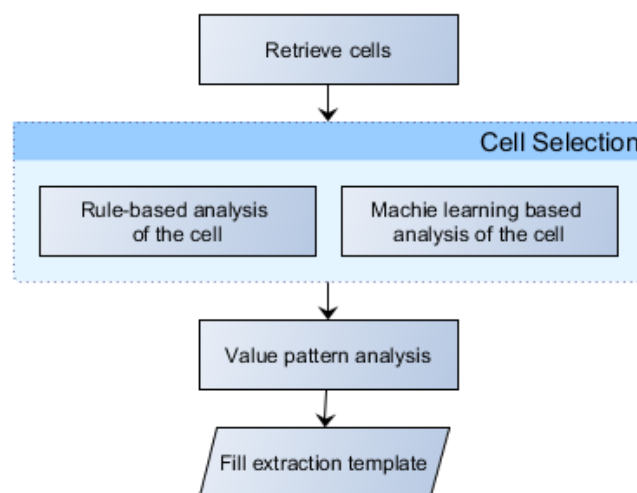


Figure 6.1: Workflow diagram of the information extraction steps

1) Cell selection This step selects cells that contain values of the target variable by analysing the content of the cells and the related navigational cells (cell's context). Two approaches for cell analysis are considered:

- **A heuristic based approach** selects only cells that contain a certain lexical cue in their context. The content of the cell and related navigational cells are analysed. The method looks for lexical cues that suggest the existence of information in the target cell (whitelist) or for the cues that will discard the cell because it does not contain a target variable or information (blacklist). It can also analyse whether the value presentation pattern matches the usual pattern for presenting that kind of information by using regular expressions. The heuristics need to be crafted manually based on the previously crafted information description and improved by using insights gleaned from the data.
- **Machine learning based approach.** We modelled the problem as a classification task: if the cell contains the target variable, classification returns a positive class, and negative class otherwise. Features for each cell contain cell content, the content of its header, stub and super-row number, cell role (e.g. header, stub, super-row, data) and the position of the cell in the table grid. The content of the cell and its navigational areas are stemmed using Porter stemmer (Porter 1980) and then tokenized before the bag-of-word methodology is used.

2) Syntactic pattern analysis and value extraction By this step, the method knows that the cell contains the target variable or its value. However, information can be presented in various formats. During this step, the content of a cell is analysed and the value is searched based on a number of possible information presentation patterns. In the case of numeric information, these patterns can be crafted using regular expressions. These patterns may also have a logic of how to translate the presented information to the extraction template. For example, if the presented value for patient age is "18.3 (16-27)", the logic should be able to conclude that number 18.3 is a mean value, while 16-27 is a range in which the first value is the minimum and the second is the maximum patient age. After the pattern analysis is performed, the value populates the information extraction template.

6.2 Case studies for cell selection approaches

6.2.1 Rule based cell selection and information extraction

As a case study, we implemented and evaluated a rule-based methodology for the following variables:

- Total number of patients
- Patient age statistics (mean, standard deviation and range)
- Gender distribution (number of male and female participants)
- Names of adverse events

The extraction of number of patients. We created rules based on 100 randomly selected, baseline characteristic tables extracted from clinical trial publications in the PMC training set. We first selected only tables presenting trials' baseline characteristics. In these tables, we checked the caption of each table for patterns starting with a number followed by a lexical cue in its vicinity (e.g. *patients, subjects, individuals, participants*, etc.). If a pattern was found, the number was extracted as the total number of trial participants. We also selected cells containing lexical clues and phrases in their stub. The header usually represents an arm - or a treatment group, which maps to *context* in our extraction template. The number in a cell is extracted as a candidate for a number of participants in that group. The candidates are checked against a blacklist of cues that determine that value is not the number of participants (*p-value, %, mean, median*). If the header, stub, and cell content do not contain these words, the value is extracted. We also selected header cells containing the letter "n" and the number (e.g. "*n = 19*"). The number next to the letter "n" was extracted along with the content of the cell without the expression that is regarded as the participant group name (context). An example of extracted values and filled template from a baseline characteristic table can be seen in Figure 6.2.

The evaluation (testing) set contained another 100 randomly selected baseline characteristic tables from the clinical trial papers. The manual evaluation results for information extraction of the number of the patients are presented in Table 6.1. As true positive, we considered values that were correctly extracted (with filled template). False positives were values that were not presenting the correct value of the target variable

Table 1

Patient characteristics of 131 patients treated with high dose accelerated radiotherapy with or without chemotherapy

Patient characteristics	Number of patients	Group		
		I	II	III
Total number	131	56	26	49
Male/female	89/42			
mean age (years)	66	62	61	71
Elderly (>70 years)	55	16	7	32
Performance (ECOG scale)				
0/1/2	28/76/27			

Variable name	Variable Sub-Category	Value Component	Context	Value	Unit
Number of patients	-	Number	Number of patients	131	people
Number of patients	-	Number	Group I	56	people
Number of patients	-	Number	Group II	26	people
Number of patients	-	Number	Group III	49	people

Figure 6.2: Extracted number of patient variable and filled template from an example table (PMC 1947993)

(in this case number of patients), but were extracted by the methodology. False negatives were values that were present in a table, presenting the value of the target variable, but not extracted by the methodology.

	Precision	Recall	F-Score
Training	0.900	0.839	0.868
Testing	0.894	0.791	0.839

Table 6.1: Results of information extraction for number of patients

Our algorithm extracted 4,355 values from 6,558 documents. As some tables presented a number of patients per clinical trial arm or participant group, there were only 1,699 documents (26%) presenting the number of patients in tables.

The errors in extraction appeared because we did not compile an exhaustive lexical cue list. This caused both false positives and false negatives. Certain words were missing in the blacklist (e.g. for the phrase "number of patients excluded" - the word

"excluded" in a given example), producing false positives. Also, non-standard abbreviations, that were not included in the whitelist, produced false negatives (*Num. patients, No. patients, N patients, # patients, with possible changes in word order*). Most of the cues or abbreviations that we did not capture were not present in the training set, while some were specific to a given paper.

Patients' age extraction. A similar methodology was applied for the extraction of patient age statistics. The algorithm selected candidate cells based on lexical cues in stubs and super-rows. These candidates were then filtered with a lexical cue blacklist. Once the right candidates were selected, we extracted variables against a set of regular expression-based presentation patterns (*mean \pm standard deviation, min-max, mean (min-max), etc.*). Age may be presented in several units (*years, months, weeks, days*), so we checked stub and header values for the appearance of cues. When some of these units are mentioned in the navigational area, that unit is recorded in the template unit field. Otherwise, "year" is recorded as the default unit. An example of a filled extraction template from a baseline characteristic table can be seen in Figure 6.3.

We evaluated extraction of patient age (mean, standard deviation and range) using the same training and testing dataset as for the extraction of the patient number. The results can be seen in Table 6.2. During the evaluation, we considered a filled template as true positive if the complete extraction template was filled correctly including variable name, variable value, its value component, context and unit. We considered an entry as false positive if an incorrect value was extracted or if parts of the template were incorrectly populated. The entry was considered false negative if the value that should have been extracted was not extracted.

	Precision	Recall	F-Score
Training	0.927	0.792	0.854
Testing	0.936	0.838	0.884

Table 6.2: Results of information extraction for age of patients, including mean, standard deviation and range

Age was presented in 1,944 documents (30% of all documents). The method extracted 13,182 values for the patient age variable, 6,125 instances of mean age, 2,475 instances of standard deviation and 2,291 instances of the age range. Compared to the *total number of patients* variable, age is more commonly present in tables.

We encountered a couple of tables that contained the age variable, which were not recognized as baseline characteristic tables, by the pragmatic analysis component

Table 1

Patients' characteristics

Characteristic	Group 1 (Jaeger-Wirth score < 3)	Group 2 (Jaeger-Wirth score = 3)
Sex	13 female, 14 male	5 female, 8 male
Age (years)	34 (range 17–47)	38 (range 25–64)
Height (cm)	175 (range 160–189)	174 (range 164–181)
Weight (kg)	76.85 (range 54–100)	76.25 (range 60–102)
Body mass index	25 (range 19–34)	25 (range 21–31)

Extractions:

Variable name	Variable Sub-Category	Value Component	Context	Value	Unit
Age	-	Mean	Group 1 (Jaeger-Wirth score < 3)	34	years
Age	-	Range: Min	Group 1 (Jaeger-Wirth score < 3)	17	years
Age	-	Range: Max	Group 1 (Jaeger-Wirth score < 3)	47	years
Age	-	Mean	Group 2 (Jaeger-Wirth score = 3)	38	years
Age	-	Range: Min	Group 2 (Jaeger-Wirth score = 3)	25	years
Age	-	Range: Max	Group 2 (Jaeger-Wirth score = 3)	64	years

Figure 6.3: Example of a table and extracted values for the age variable (PMC 1906819)

of the methodology. One of the tables contained the cue "age" in an unexpected context (*HT age* – duration of hypertension). In two tables value presentation formats were unexpected, so our algorithm was only able to extract the mean value and missed standard deviations presented in these tables. Four tables presented age groups together with a number of trial participants in each of the group (e.g. *18-25 – 10 patients*, *25-35 – 15 patients*, etc.). The algorithm misinterpreted these numbers as the participants' mean ages. In three tables, the super-row or the second header of multi-table were not recognized correctly which led to false negatives.

Matching patterns and extracting the right values, once the value is recognized, is an important part of the process. We evaluated the performance of pattern extraction for the extraction of patient age. The patient age statistics can be presented as the mean value, its standard deviation and/or the range of ages. However, these three values can be combined and presented using various syntactically different formats. During the pattern matching evaluation, we examined the cells recognized as correctly containing

the age of patients.

The patterns were matched and extracted with 0.994 precision, 0.9575 recall and an F1-score of 0.975. In the testing set, we encountered one, new presentation pattern and several patterns that did not appear in the training data that included some special characters (central dot (\cdot) instead of dot ($.$). e.g. $5 \cdot 34$). Overall, pattern matching was reliable, accurate and reusable and if developed for certain types of presentations or value groups (such as aggregated statistical data), it can be applied to other variables that present information in the same manner.

Gender distribution extraction. In order to extract gender distribution, the methodology looked for gender related cues (*gender, sex, male, female, m, f, etc.*) in table headers and stubs. The blacklist included cues such as *p value* or *change*. The syntactic rules looked for the following patterns:

- Four values presenting the absolute number and percentage of male and female participants (e.g. 34/24 (58%/42%))
- Two values presenting the number of male and female participants (e.g. 34/24, 34:24)
- Two values presenting a single category as absolute value and the correspondent percent (e.g. 34 (58%))
- A single value presenting the value for one of the sub-category, either male or female

We note that the order of the variable and separators may be different than in examples. However, we included multiple combinations in our rules.

The results of gender distribution extraction are featured in Table 6.3.

	Precision	Recall	F-Score
Training	0.965	0.914	0.939
Testing	0.948	0.840	0.891

Table 6.3: Results of extraction for gender distribution of the patients variable using rule based approach. Evaluation performed on the clinical trial dataset

The process for extracting the gender distribution of patients produced similar errors to those described earlier, including presentation patterns missed, missing lexical cues that either approved or discarded cells and errors accumulated from previous steps

(e.g. disentangling, pragmatic classification, annotation). The lexical and syntactic rules for extracting age and gender distribution variables are available in Appendix E.

Adverse event names extraction. As it is a categorical variable, we adopted a slightly different approach for extracting adverse reaction names compared to previously examined, numerical variables. Categorical variables do not utilise presentation patterns as numerical variables do. First, we selected tables that the pragmatic classification step classified as containing adverse events. Second, we used UMLS semantic type annotations of the content of the cells in order to recognize whether a certain column contained adverse events. The methodology annotated the content of cells with semantic types using MetaMap. We checked whether cells in a certain column contained phrases annotated as *"Sign or Symptom"* or *"Disease or Syndrome"*. In cases where multiple cells in the same columns contained these annotations, the content of all cells in that column - except the header - were extracted as adverse event names.

We performed an evaluation of detecting names of adverse events over 35 documents in both the training and the testing set. We considered extraction as true positive if it contained the correctly extracted name of adverse event from the table. In case the extracted name was not an adverse event, it was considered a false positive. In case the adverse event name is not extracted from the table, we considered it as false negative. The results are presented in Table 6.4.

	Precision	Recall	F-Score
Training	0.945	0.906	0.925
Testing	0.883	0.962	0.921

Table 6.4: Results of information extraction for adverse events

MetaMap annotated 7,701 instances of adverse events in cells, while 4,974 adverse event instances, that were not annotated by MetaMap, were extracted by our methodology from 6,558 clinical articles with 12,787 tables.

The extraction of adverse event names performs better than the extraction of numeric variables (the number of patient and age). The possible reason is that the semantic resource (the UMLS concept tagger from MetaMap) helped with the rule creation and generalisation. Errors appeared in columns that contained mixed content, among which were also adverse events. Additionally, one table listed clinical conditions on admission, with cell content annotated as *"Sign or Symptom"* or *"Disease or Syndrome"* semantic types, which were recognized by our approach as adverse events (see example in Figure 6.4).

Baseline clinical characteristic of the 156 patients with cerebral malaria in both treatment arms on admission.

Variable	Placebo N = 80	Mannitol N = 76	P value
Female	42(52.5%)	34(44.7%)	0.33
Fever	79(98.8)	76 (100%)	0.33
Convulsions	79(98.8%)	75(98.7%)	0.97
Duration of coma	7.0 (IQR3.5–12.0)	6.0 (5.0–12.0)	0.79
Blantyre coma score 1/5	13(16.2%)	10(13.2%)	0.59

Figure 6.4: Example of table presenting baseline characteristics as number of people having certain conditions (PMC 2147028)

6.2.2 Machine learning based cell selection

Another approach to extracting information from tables uses machine learning in order to detect cells containing a given variable. Similarly to a rule-based approach, the information is then extracted using patterns. We implemented detection of cells containing the number of patients; information about patient age and gender distribution. Our aim was to explore to what extent a machine learning system can help extract target variables compared to lexical cues.

We created a training dataset using 100 randomly selected baseline characteristic tables from PMC. For the variable referring to number of patients, there were 147 positively labelled cells. The number of cells presenting age of the patients was 272, while there were 204 cells presenting the gender variable. The dataset was highly imbalanced since the whole dataset contained 13,610 cells. We performed three machine learning experiments. In the first one, we balanced the dataset for each learning task, so it contained the same number of negatively labelled cells as positively labelled ones (under sampling). For this technique, we performed learning on under-sampled data. The second approach consisted of learning from the unbalanced dataset. In the third approach, we used cost-sensitive classification and experimentally adjusted the weights for the best performance. As features, we used the content of the current cell and content of the navigational cells referring to the current cell. The assumption was that machine learning will be able to learn cues associated with target variables and therefore be able to successfully select cells. In order to make it easier for machine learning algorithm to learn presentation patterns of numeric variables, we changed numeric symbols to the "x" symbol.

We performed 10-fold cross-validation on this clinical trial dataset. In this case, we evaluated only whether algorithm was able to select the right cell (no syntactic analysis

and value/metadata extraction was performed). Therefore, if the cell is classified as containing the value of interest for a given variable, it is considered a true positive. If algorithm did not select the cell that should have been selected, it is considered a false negative, while if algorithm selects wrong cell, it is considered a false positive. The results are presented in Tables 6.5, 6.6 and 6.7.

Algorithm	Under-sampled (147 instances of each class)				Whole unbalanced dataset				Cost-sensitive classification			
	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy
Naive Bayes	0.054	0.952	0.103	0.821	0.173	0.701	0.277	0.960	0.266	0.483	0.343	0.980
Bayesian Nets	0.101	0.912	0.182	0.911	0.292	0.517	0.373	0.981	0.512	0.422	0.463	0.989
C4.5 dec. trees	0.070	0.905	0.130	0.869	0.893	0.510	0.649	0.994	0.714	0.782	0.747	0.994
Random tree	0.066	0.585	0.119	0.906	0.580	0.544	0.561	0.991	0.573	0.585	0.579	0.991
Random Forests	0.214	0.932	0.348	0.962	0.935	0.490	0.643	0.994	0.797	0.667	0.726	0.995
SVM	0.085	0.918	0.155	0.892	0.850	0.463	0.599	0.993	0.754	0.626	0.684	0.994

Table 6.5: Results of selecting cells associated to the patient number variable using various machine learning approaches

Algorithm	Under-sampled (272 instances of each class)				Whole unbalanced dataset				Cost-sensitive classification			
	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy
Naive Bayes	0.089	0.930	0.162	0.879	0.205	0.819	0.327	0.957	0.254	0.754	0.381	0.969
Bayesian Nets	0.128	0.918	0.224	0.920	0.419	0.743	0.536	0.984	0.504	0.684	0.581	0.987
C4.5 dec. trees	0.092	0.795	0.165	0.899	0.886	0.591	0.709	0.994	0.783	0.801	0.792	0.995
Random tree	0.074	0.871	0.136	0.900	0.628	0.573	0.573	0.990	0.628	0.573	0.599	0.990
Random Forests	0.213	0.947	0.348	0.963	0.945	0.503	0.656	0.993	0.883	0.661	0.756	0.995
SVM with SMO	0.180	0.614	0.278	0.963	0.955	0.743	0.836	0.996	0.895	0.801	0.846	0.996

Table 6.6: Results of selecting cells associated with the age of patients variable (cumulative statistical values such as mean, standard deviation and range) using various machine learning approaches

Algorithm	Under-sampled (204 instances of each class)				Whole unbalanced dataset				Cost-sensitive classification			
	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy
Naive Bayes	0.075	0.929	0.139	0.834	0.155	0.675	0.252	0.942	0.167	0.584	0.260	0.952
Bayesian Nets	0.099	0.934	0.179	0.876	0.475	0.584	0.524	0.985	0.813	0.528	0.640	0.991
C4.5 dec. trees	0.119	0.929	0.210	0.899	0.912	0.685	0.783	0.994	0.839	0.766	0.801	0.994
Random tree	0.081	0.959	0.150	0.843	0.739	0.746	0.742	0.992	0.739	0.746	0.742	0.992
Random Forests	0.155	0.990	0.218	0.922	0.953	0.624	0.755	0.994	0.893	0.807	0.848	0.996
SVM with SMO	0.122	0.909	0.909	0.897	0.903	0.756	0.823	0.995	0.833	0.812	0.823	0.995

Table 6.7: Results of the selecting cells associated with gender distribution variable using various machine learning approaches

Precision, recall, and F1-score presented in the tables are measured on positive class. Since the data is unbalanced, the weighted average does not provide representative results.

The datasets contain a small number of positive class instances. Due to the small number of positive instances, balancing data by under-sampling does not perform well. Many machine learning algorithms rely on probabilistic distribution of classes and assume same costs for misclassification of classes (He & Garcia 2009). However, if the data represent the realistic distribution of classes, some of the algorithms are able to cope with the data relatively well. The results for some algorithms (such as decision trees, random forests and SVM) for learning from the whole dataset, are much better than with under-sampled data. By using cost-sensitive classification and assigning larger costs to positive rather than negative class, it is possible to improve these results. By experimentally tuning costs, we managed to improve the F1-scores by almost 10% (see Tables 6.5, 6.6, 6.7).

When a machine learning approach was compared with a rule-based approach, it was observed that a simple rule-based approach with a whitelist and blacklist of lexical cues usually produced similar or better F1-scores. Development of a machine learning model is more complex and time-consuming than crafting whitelists and blacklists as typically, it is necessary to annotate several thousand cells and perform a number of experiments to find the most suitable costs. Due to the imbalanced data, it was also necessary to perform additional data processing (such as cost-sensitive classification). Also, the machine learning approach performed similarly or worse than the rule-based approach. Consequently, it appears more straightforward to develop a rule-based selector for cells containing the variable of interest.

The proposed approach represents the state-of-the-art in table information extraction from XML documents, without any restriction on table structure. Even though some of the previous approaches reported slightly better performance (Embley et al. 2005, Wang 2013), they were limited to standardized tables with pre-defined table structures.

6.3 General framework for information extraction from tables

After performing and evaluating experiments, it is possible to generalise the findings into the general framework for information extraction from tables in the biomedical domain. The framework defines the possible variable types, information required for task specification (information extraction recipe) and the means to define lexical, semantic and syntactic rules.

Components of the information extraction framework are mostly task and domain dependant. However, some of the components of this framework can be developed for one domain and transferred to the other domain. Variable types are domain independent. Task specification recipe is, as well, domain-independent if viewed conceptually. However, the majority of recipe components, its rules or values will be domain dependant. Syntactic analysis and rules can be often transferred to other, especially related domains. However, it is not possible to assume that same presentation patten will have the same meaning in any domains (e.g. 15 ± 2 will be mean/median and standard deviation in the biomedical domain, while in computer science it may be the mean and standard error). Machine learning models, if they are trained using bag-of-words model are domain specific. Trained model cannot be transferred to the other domain, but training methodology is usually transferable.

6.3.1 Types of variables

Our model of table information contains five variable types whose extraction methodologies slightly differ. They are grouped into two high-level types: numerical and textual variables. There are three numerical subtypes: (1) Single numerical value, (2) Aggregated statistical value and (3) Categorized numeric values. There are two textual subtypes: (4) Categorical and (5) Free text information classes. The diagram of the identified information groups is presented in Figure 6.5.

Numeric variables

The numerical variable types contain these three groups/subtypes:

Group 1 – Single numeric. The first group represents the values represented as a single numerical value (e.g. 15, 24.3). In demographic tables in the clinical trial

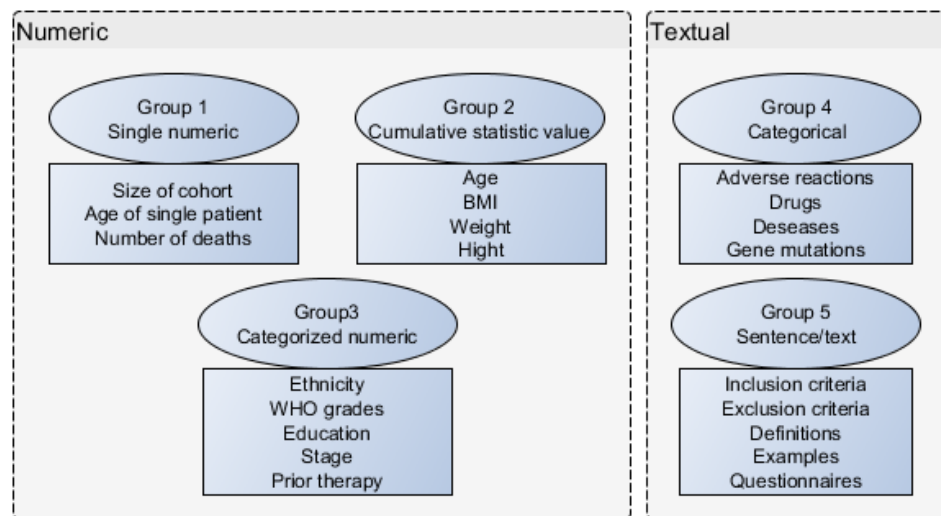


Figure 6.5: Variable types, with their subtypes and the example variables for each defined subtype.

literature, this may be the size of a cohort if we examine tables presenting aggregated data about an entire cohort or age, BMI, weight or the height of a single patient if the data is presented per participant. Individual measurement results are often presented using a single numerical subtype.

Group 2 – Aggregated statistical values. Demographic data in tables are often presented cumulatively, for the whole cohort or for the groups participating in trial arms. In such cases, values are usually presented as the mean value with optional standard deviation or range (e.g. 15.3 ± 2.1 , $24(14 - 35)$, $16 \pm 2(14 - 17)$). Examples of information from this group are BMI, weight, height and/or age of patients in aggregated demographic tables.

Group 3 – Categorized numeric values. Values in this group have multiple subcategories and are presented as numbers, means, ranges or percentages per subcategory. Examples of such values are ethnicity (e.g. *number of White, Asian, Black, Hispanic, etc.*) or the number or percentage of patients with a certain stage of disease, adverse reactions, etc. For this subtype it is necessary to define possible categories and the mapping between the category names and cues identifying them in tables. Numerical values categorized by two categories are a special case since they can be presented in a single cell (e.g. $27/28$). An example of such information is gender of participants

in clinical trial. In some cases values are presented in multiple rows - typical for this group - while in other cases binary categorized values are presented using special presentation pattern (i.e. explicit pattern, such as "*male/female - 22/14*" or implicit, such as "*female(%) - 14 (39%)*"). Category cues are usually in navigational areas (headers or stubs) but in some cases can be in data cells (e.g. *14 M, 18 F*).

Textual variables

Textual information can be grouped into the two groups:

Group 4 – Categorical values. Categorical values are controlled words or short phrases, such as names of diseases, adverse reactions, drugs, institutions, etc.

Group 5 – Free text. The last group presents free-text information. Examples of such information are inclusion and exclusion criteria, the definition of terms or scales and examples of questions asked in a questionnaire. They are longer phrases, sentences or even paragraphs of texts stored in tables. Following extraction, they can be further mined using standard text mining techniques. However, free-text variables are outside the scope of this thesis.

6.3.2 Information extraction task specification

We introduce a description template that defines the information that needs to be defined in order to successfully implement the information extraction from tables' methodology. The template contains eight description categories that need to be defined. The categories are presented Table 6.8.

For some of the description categories, it is useful to define a default value as tables often present values without corresponding units. Unit is therefore one of the values for which it is useful to define the default value. However, it is also necessary to define a procedure for extracting and checking the unit. Another default value that needs to be defined is the semantics of the values extracted during pattern analysis. It is necessary to define the meaning of the values that are parts of the numerical expression. For example, the value may present the mean or median with standard deviation using the same pattern. We aim to identify whether the value is the mean or median by checking the content of the stub cell in the same row. However, the majority of tables present the mean without explicitly stating the word "*mean*". In this case, the default value

Descriptor name	Description	Example
Variable identifier	Name of the variable to be extracted. It can be mapped to an ontology	e.g. Age, BMI, Gender, Adverse event. e.g. mapping to baseline characteristics in OCRE
Table's pragmatic type	Pragmatic type of a table in which information is likely to appear	Pragmatic types can be for example tables with Baseline Characteristics Adverse events, etc.
Categories * Only for categorical variables	A list of possible categories	e.g. for Grade: Grade I, Grade II, etc. Ethnic group: Asian, Black, Caucasian, Hispanic, etc.
Cues		
Lexical cues	Set of lexical cues that determine whether the value is present in that or related cells	Lexical cue for number of patients can be "n=%d", "number of patients" in stub and number in data cell, etc.
Functional cues	Description of functional regions in the table where information may appear.	Number of patients may be in caption, header or data cell
Semantic cues	Set of semantic cues such as semantic types and higher level concept names	i.e. list of semantic types indicates the presence of the value (i.e. Sign or Symptom UMLS semantic type may indicate adverse event in table)
Value type/pattern (Syntactic cue)	Description of the value type and its pattern with the way to extract it	Whether the value is single number, range, percentage, etc.
Unit of measure * Only for numeric variables	Defines the default and possible unit of measure for numeric variables	e.g. default is gram (g), but kilogram (kg) and milligram (mg) may appear

Table 6.8: Categories of information that need to be described in order to specify a table information extraction task

assumes the meaning of the value as "*mean value*" (since it is used more frequently), if not explicitly stated.

6.3.3 Defining rules for information extraction

Cell selection using lexical and semantic rules

Selecting cells in which a variable and its value are presented is done using a defined set of lexical and semantic cues. The rule contains four parts: a list of cues for the

whitelist, a list of keywords for the blacklist, functional areas where the cues should be searched for (search functional areas), a functional area where the value of the variable (information that should be extracted) is located (target functional areas). A functional location of the cue (header, stub, super-row, target) is defined for both lists. In other words, a definition is provided as to whether the cue should be searched for in the header, stub or super-row or in the target cell. Also, it is defined whether the value should be extracted from the header, stub, super-row related to the cell where the cue is found or the data cell. In case a cue is searched in navigational areas (headers, stubs, super-rows) and information should be extracted from the data cell, the information is extracted from all data cells that are related to the matched navigational cell. Cues in both whitelist and blacklist may be lexical or semantic.

Lexical cues are defined as a set of words in a whitelist and a blacklist. Table cells are iterated and tested against the defined lexical rules. The presence of the cue from the whitelist signals that the target cell potentially contains a value for the variable of interest. The cell is then tested against cues from the blacklist. If the cell or its navigational cells (as defined in a part of the rule defining where to look for these cues) contain cues from the blacklist, the selection is discarded. Otherwise, information is extracted from the selected or related cell, depending on the defined search and target functional areas.

Semantic cues are defined similarly to lexical cues. However, instead of words or phrases that are searched for, we use annotations. Annotations can be searched for in headers, stubs, super-rows of the target cell or in the target cell itself. Again, an annotations whitelist and blacklist is used. The method uses two layers of annotations: annotation id and annotation description. In the case of UMLS annotations, annotation ids were UMLS concept ids, while annotation descriptions were semantic types. Therefore, it was possible to create whitelists and blacklists consisting of UMLS concept ids and semantic types for the UMLS annotated data. This method iterates through table cells, selects cells using signals from whitelists and discards cells containing cues from the blacklist. It is also possible to combine lexical and semantic cues while creating cue lists (black or white).

In this step, the method also selects the unit and context for the numerical variables. A set of possible units for the given variable has to be defined as well as the default value. The method searches the cell and its navigational areas (header, stub, super-row) for mention of the unit. If a unit is found, it is extracted and if not, the default unit is used.

In our method, the context is extracted as the concatenated value of navigational cells relevant to the target cell that did not contain cues from the white list.

Syntactic rules and syntactic processing

The role of syntactic processing is to analyse the content of the selected cell with the value, disentangle the value and identify its components (populating *ValueComponent* from the extraction template). For example, the syntactic processing reveals whether the extracted value is the mean, median, standard deviation, range, percentage, etc.

The value patterns are common for certain types of information. For example, age, BMI, FEV1 and many other variables present overall statistics for certain population (*average, mean, standard deviation, range*). If the rules are developed for one variable, they can be reused for others. In this way, it is possible to create a library of common value presentation patterns. Examples of common numerical presentation patterns are presented in Table 6.9.

Pattern	Presentation examples	Variables
Single value	65	Number of patients, number of people with certain adverse event, etc.
Floating point value	0.05	p-value
Aggregate statistical value	18 ± 2 12-18 12.1 (2.4) 18 ± 2 (15-20)	Age, FEV1, PEF, BMI weight, height of patients in cohort
Alternatives	12/17	Gender distribution, blood pressure
Percentage	18 (55%) 55%	Gender distribution Percentage of people with certain effect

Table 6.9: Examples of common syntactic patterns and variables that are often represented by them

Syntactic processing is performed using a rule-based methodology. The methodology uses regular expressions for disentangling cell content. Syntactic rules map values to their descriptions.

A definition of a syntactic rule contains three components: (1) the rule's name, (2) the rule's regular expression and (3) a set of semantic assignments (descriptions) for each component of the regular expression.

Value components (*e.g. mean, standard deviation, range-min, range-max, etc.*)

can be assigned to each regular expression component. The aim of syntactic processing is to assign semantics to each value component based on the value presentation pattern and cues in the pattern and navigational cells related to the cell. Therefore, a set of possible but distinct semantic assignments can be listed with the regular expression defining the rule and giving possible meanings to each extracted value. Often, a value's semantics can be induced from the value presentation pattern. For example, if a table's cell contains BMI values of 20-37, it is likely that the value is the range, with a minimum value of 20 and maximum value of 37. However, for some value presentation patterns, additional information in the navigational part of the table is necessary. One example is a pattern like 16 ± 4 . The first value could be either the mean or median. The navigational cells' content for these data cells will determine through mention, whether the value is mean or median. If the definition is not mentioned, a default assignment of the value's meaning can be used by applying the most common one. In other cases, multiple values are presented with explicitly described semantics of each value part in navigational areas. For example, if the gender value is presented as 15:14, navigational cell's would describe which value presents the number of male participants and which one is the number of female participants. We allow for each extracted regular expression group to define a set of keywords or synonyms with their order of appearance that are looked for in navigational areas. The semantic assignment contains a group number, ordered groups of keywords (or synonyms) and the semantic assignment. Each keyword group is a comma-separated list of strings. A semantic assignment value is separated by the arrow symbol (\rightarrow). Figure 6.6 provides an example of a rule in our descriptive language based on regular expressions, that can disentangle a pattern such as "15:14" for gender. According to the rule, in case any cue from the list linked to the number of male participant variable (*male, m, Male, M, men, males, Males*) appears before any cue linked to the number of female participants variable (*female, f, fem, Fem, women, Women, females, Females*), the first value is associated with the number of male participant variable. In case a cue from the list linked to female participants is appearing first, the first value is the number of female participants, while the second value is the number of male participants. In case none of the cues appear, as default, rule assigns the first value to the number of male participants, while the second is assigned to the number of female participants. A more detailed description of rule format is given in Appendix D.

Another example of the rule definition for statistical values (range, mean, median and standard deviation) can be seen in Figure 6.7.

```

+GetMaleFemaleRule Rule name
(\d+) [/:\\, ]{1,} (\d+) Rule regular expression
1:male,m,Male,M,men,Men,males,Males;female,f,F,fem,Fem,women,Women,females,Females->male
1:female,Female,f,F,fem,Fem,women,Women,females,Females;male,m,Male,M,men,Men,males,Males->female
1->male Default assignment of semantics for the first group in regular expression
2:male,m,Male,M,men,Men,males,Males;female,f,F,fem,Fem,women,Women,Females,females->female
2:female,Female,f,F,fem,Fem,women,Women,females,Females;male,m,Male,M,men,Men,males,males->male
2->female

```

Ordered list of possible cues divided by semi-column symbol

Figure 6.6: Example of one syntactic rule with its semantics for extracting gender distribution of the participants.

```

+GetMean1
(\d+\.\d*) [ ? ] * [ \-+ ] * ( \d+\.\d* ) [ ] * [ ] *
(\d+\.\d*) [ ? ] * [ \pm ] [ ? ] * ( \d+\.\d* ) [ ] *
1->range_min
2->range_max
3:median,Median->median
3->mean
4->SD

```

Figure 6.7: Example of one syntactic rule with its semantics for extracting statistical values

In the case of the cell with the content "12 – 18(16 ± 4)", the rule from Figure 6.7 would say that 12 is minimum value of a range, 18 is a maximum value, 16 is mean or median (in case median is mentioned in stub or header of the cell) and number 4 is standard deviation.

For categorical variables, syntactic analysis depends on the user's definition of possible categories for that variable. Patterns can be defined as possible representations of the given category (e.g. synonyms) that algorithm matches and extracts from the cells' content.

For textual variables, syntactic analysis have to be complemented with further lexical and semantic analysis in order to extract more granular information from the cell content. However, this is outside the scope of this thesis.

6.4 TableInOut: a wizard for information extraction

In order to test the presented methodology, we developed software called TableInOut. TableInOut is a tool in which user can specify table information extraction task, lexical, semantic, syntactic rules and perform the extraction. Using the TableInOut and

presented methodology we reproduced and in some cases were able to improve the results for extracting age, gender distribution and adverse events presented in Section 6.2.1. The implementation details are presented in Appendix C. The tool was used in COPD and asthma case studies presented in the following chapter.

6.5 Summary

In this chapter, we firstly explored two different methods for selecting cells that contain the value of the target variable. One method is based on lexical and semantic rules, while the other is based on machine learning. The evaluation showed that it is more straight forward to use the rule-based methodology, as machine learning requires advanced methods in order to deal with unbalanced data and large amount of annotated data.

We also presented a framework for information extraction from tables. The framework contains a recipe for describing the table information extraction task, a variables model and a step-by-step table processing method.

Information in tables can be categorized by two broad categories (textual and numerical) or five narrower categories (single numerical, cumulative statistic values, categorized numerical, textual categorical and free text). As each of the variable categories is different, the information extraction methodology may differ for each of them. We defined information that anyone developing information extraction rules for certain target variables in tables needs to know (a recipe for describing the table information extraction task), such as binding to an ontology, functional, lexical, syntactic and semantic cues, possible and default units of measure and pragmatic type. Once this information is defined, one can easily craft rules and iteratively improve them.

The method aims to automate as many steps as possible. Table detection and functional and structural analysis in the PMC dataset are generic and there is no need for new rules in these processes. Semantic tagging can be performed by many tools, vocabularies or ontologies. There are two layers for which users need to define rules – lexical and syntactic. Lexical cues are specific to the target variable and need to be separately crafted for each information class. The syntactic rules we propose in our approach can be reused for extracting multiple variables. Simply, syntactic patterns are defined per type of value presentation. Multiple variables can use the same presentation patterns (e.g. age, FEV1 and BMI are represented using statistical patterns representing mean, median, standard deviation and ranges). The methodology of

defining and parsing these rules is generic, so it can be applied to a wide range of data formats and tables.

The framework provides an easy-to-use methodology for non-expert users who do not necessarily need programming skills to develop extraction rules. The framework also makes it easier to develop extraction methods for variables. Our framework provides the complete table information extraction pipeline, with special consideration for the biomedical domain. The method can be extended to other domains by using different models for classifying tables or different vocabularies for annotations however, the base methodology remains the same for all domains.

Chapter 7

Case studies

7.1 Extracting clinical trial baseline characteristics from Asthma and COPD studies

7.1.1 Introduction

Asthma and chronic obstructive pulmonary disease are two obstructive airway disorders that represent major global causes of death and disability (Welte & Groneberg 2006).

Chronic obstructive pulmonary disease (COPD) represents a group of lung diseases characterized by poor, long-term airflow. It includes diseases such as emphysema (damaged air sacs in the lungs) and chronic bronchitis (long-term inflammation of the airways). COPD can cause difficulties for everyday activities, such as moving and walking. In 2015, more than 174 million people (2.4% of global population) suffered from COPD (Vos et al. 2016). COPD was the cause of death for more than 3.1 million people in 2015 and in comparison to 2005, more people suffered and died from COPD (Wang et al. 2016). COPD is predicted to become the third most common cause of death by 2020 (Welte & Groneberg 2006).

On the other hand, asthma is a disease characterized by chronic airway inflammation with increased airway responsiveness. The symptoms of asthma include wheezing, coughing, dyspnoea and airway obstruction over short time periods. Asthma was known in the times of Hippocrates (460-370BC). It is estimated that 130 million people worldwide suffer from asthma. It has been also estimated that about 7% of the UK adult population have asthma (Bourke & Burns 2015) and the number of people suffering from asthma increased by 9.5% from 2005 to 2016 (Vos et al. 2016). More

than 400,000 people worldwide die annually from asthma (Wang et al. 2016). Factors that cause asthma can be genetic, but also environmental (Bourke & Burns 2015).

Both asthma and COPD are diagnosed using lung function tests, most commonly those using spirometry. In both diseases, forced expiratory volume in one second (FEV₁), peak expiratory flow (PEF) and FEV₁/vital capacity (VC) are reduced (Bourke & Burns 2015). These tests, in combination with patient activity, are indicators for diagnosing COPD and asthma.

Table 1

Patient demographics and baseline characteristics

Characteristic	Budesonide pMDI (<i>n</i> = 217)	Budesonide/formoterol DPI (<i>n</i> = 229)	Budesonide/formoterol pMDI (<i>n</i> = 234)
Males/females, <i>n</i>	68/149	89/140	94/140
Mean age (range), years	40 (12–79)	39 (11–78) [*]	40 (12–78)
Smokers, <i>n</i> (%)	14 (6)	11 (5)	13 (6)
Median time since diagnosis (range), years [†]	10 (0–70)	9 (1–63)	8 (1–58)
Mean morning PEF (range), l/min	318 (109–638)	321 (93–668)	326 (89–715)
Mean FEV₁(range)			
% predicted [†]	71 (45–91)	69 (50–90)	71 (39–92)
1	2.01 (0.85–4.25)	2.09 (1.05–3.75)	2.07 (0.94–4.12)
Mean ICS at entry (range), µg/day [†]	759 (400–1600)	774 (500–1600)	776 (400–1600)
LABA use at entry, <i>n</i> (%)	32 (15)	33 (14)	30 (13)
Reliever medication use (range), inhalations/day	2.0 (0.0–14.5)	1.8 (0.0–11.3)	2.1 (0.0–11.4)
Reliever medication-free days (range), %	29 (0–100)	34 (0–100)	29 (0–100)
Total asthma symptom score (range), 0–6	2.1 (0.4–5.7)	2.0 (0.0–6.0)	1.9 (0.0–5.3)
Nights with awakenings (range), %	33.1 (0–100)	32.1 (0–100)	29.2 (0–100)
Symptom-free days (range), %	10 (0–80)	12 (0–100)	12 (0–100)
Asthma-control days (range), %	8 (0–80)	10 (0–89)	10 (0–100)
AQLQ(S) (range), 1–7	4.80 (1.8–6.8)	4.62 (1.8–7.0)	4.70 (1.4–7.0)

^{*}One patient was 11 years and 354 days old at the enrolment visit.

[†]Deviations from inclusion criteria not considered sufficiently significant to justify exclusion of data from the full analysis. AQLQ(S), Asthma Quality of Life Questionnaire (standardised version); DPI, dry-powder inhaler; FEV₁, forced expiratory volume in 1 s; ICS, inhaled corticosteroid; LABA, long-acting β₂-agonist; PEF, peak expiratory flow; pMDI, pressurised metered-dose inhaler.

Figure 7.1: Example of the baseline characteristic table from asthma clinical trial presenting FEV₁, PEF and Asthma Quality of Life Questionnaire (AQLQ) variables (PMC 2228375)

Therefore, the results of lung function test are baseline characteristics that show the severity of the disease in COPD and asthma related clinical trial studies. Since both diseases are chronic, often drug and treatment development focus on improving quality of life for affected subjects. Many studies are measuring and comparing quality of life in subjects of different clinical arms. Usually, well adopted quality of life tests are used, such as the asthma quality of life questionnaire (AQLQ), and the asthma control questionnaire (ACQ) for asthma and the St. George Respiratory Questionnaire (SGRQ) for COPD. An example of a table presenting FEV₁, PEF and AQLQ variables

is presented in Figure 7.1.

In our case study we extract the following variables:

1. Forced expiratory volume in one second (FEV1)
2. Peak expiratory flow (PEF)
3. Asthma quality of life questionnaire (AQLQ)
4. Asthma control questionnaire (ACQ)
5. Saint George respiratory questionnaire (SGRQ)
6. Age of participants
7. Gender distribution of trial participants

The first two variables (FEV1 and PEF) are lung function tests. The questionnaire variables (AQLQ, ACQ, SGRQ) are produced from quality of life questionnaires. Gender distribution and age of participants are general baseline characteristics. All variables in this case study are usually presented as statistical values for the whole population or for a clinical trial arm population. In some cases, results are presented for each participant.

The aim was to extract the variables, their values and metadata to the extraction template (*VariableName, SubCategory, ValueComponent, Context, Value, Unit*).

7.1.2 Methodology

The workflow diagram of the methodology used in this case study can be seen in Figure 7.2.

Data collection

Our collaborators from AstraZeneca provided a set of 148 articles related to COPD and asthma (25 related to asthma and 123 related to COPD). Curators manually extracted information about baseline characteristics from these documents. However, only 28 articles were available as open access. In the newest PMC subset¹ (January 2017), available for download and processing, containing more than 700 thousands articles, only 12 articles (10 about COPD and 2 about asthma) matched articles provided by

¹<https://www.ncbi.nlm.nih.gov/pmc/tools/ftp/>

AstraZeneca. The remaining 16 articles were obtained manually, by downloading them from the PMC website.

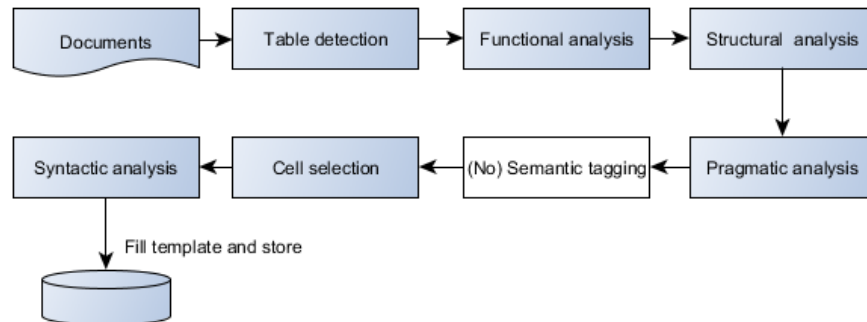


Figure 7.2: Workflow of the methodology used for extracting variables from clinical trial documents about asthma and COPD

The data obtained from AstraZeneca was used as test data. For training data, we used 30 articles about COPD and asthma, selected by searching the PMC public subset from January 2014.

Table detection

Since the documents are in PMC format, table detection is looking for the relevant XML tags.

Functional and structural processing

The obtained documents were processed using the table disentangling methodology described in Chapter 4. This methodology infers the functional areas of the table and relationships between cells.

Pragmatic analysis

Pragmatic processing applies a model that takes into account a table's caption, footer and navigational areas. We previously developed such a model that contains four classes: baseline characteristics, adverse events, inclusion/exclusion criteria and other (for details see Section 5.1). For this case study, the task was to extract baseline characteristics. A previously developed model was used in order to retrieve relevant tables.

Semantic tagging

Semantic tagging was not used in this case study, since we were able to identify lexical cues for selecting the cells presenting target variables.

Information extraction rules (lexical and syntactic processing)

For crafting information extraction rules and performing information extraction, we used methodologies described in details in Chapter 6.

While performing case studies as described in Chapter 6, we developed the extraction rules for age and gender distribution of trial participants. For other variables, we initially applied an intuitive approach for listing cues that we intuitively thought would indicate the presence of a certain variable. The rules were evaluated on a set of 30 documents (different from the selected test documents for case study). During the initial evaluation on that dataset, errors were analysed and rules improved iteratively. During the iterative improvement process, we improved our F1-score for extracting the FEV1 variable by almost 10% (from 0.817 to 0.901). The performance of other variables also slightly improved.

The lexical cues for locating relevant variables in tables are presented in Table 7.1. The cues are searched in navigational areas of tables, while data is extracted from data cells.

Variable	Lexical white list	Lexical black list
Age	Age, age	p-value, p value, p*, P, HT
Gender	Gender, Male, Female, M, F, Sex, M:F, M/F, Boys, Girls, Men, Women	P-value, P, p*
FEV1	FEV1, FEV-1, Forced expiratory volume	change, VC, reversibility, classification, Mild, Moderate, Severe, p-value
PEF	PEF, Peak flow	change, reversibility, p-value, mild, moderate, severe, change, increase decrease
AQLQ	AQLQ, AQLQ(S), Asthma quality of life questionnaire	p-value, p value, p*, P
ACQ	ACQ, Asthma control questionnaire	p-value, p value, p*, P
SGRQ	SGRQ, St. George respiratory questionnaire	p-value, p value, p*, P

Table 7.1: Examples of lexical cues for extracting given variables

All variables in this case study (Age, FEV1, PEF, AQLQ, ACQ, SGRQ), except gender distribution, are presented as cumulative statistical values for the whole population. Usually they are represented as the mean, standard deviation and range. The same syntactic rules for age presented in Chapter 6 were applied for all the six classes. Gender distribution is different, usually presenting as two categories (number of male and number of female participants, often with percentages). In this case study, we also applied the rules developed for extracting gender distribution presented in Chapter 6.

7.1.3 Evaluation and results

We performed a quantitative analysis of the performance of our approach and calculated precision, recall and F1-score for each variable. Since we had access to the data extracted by the curators, we also compared manually curated data with the data obtained using our methodology.

The test set contained 22 articles on asthma and COPD (6 articles out of 28 did not have content. These articles contained reference to the images of the scanned pages) that were quantitatively evaluated for true positives (TP), false positives (FP), false negatives (FN), precision, recall and F1-score. Extracted information was considered true positive if the variable, value component name, value and unit matched. In case any of these were misinterpreted, the instance was considered either false positive (if the metadata or value, that was not supposed to be extracted, was extracted) or false negative (if the metadata or the value, that was supposed to be extracted, was missing). The results per information class are presented in Table 7.2.

Variable	TP	FP	FN	Precision	Recall	F1-score
Age	134	14	38	0.905	0.779	0.837
Gender	147	0	0	1.000	1.000	1.000
FEV1	208	13	25	0.941	0.893	0.916
PEF	10	0	0	1.000	1.000	1.000
ACQ	18	0	0	1.000	1.000	1.000
AQLQ	18	0	0	1.000	1.000	1.000
SGRQ	16	1	0	0.941	1.000	0.969
Overall	551	28	63	0.951	0.897	0.924

Table 7.2: Evaluation of the target variables extracted from Asthma and COPD clinical trials. (TP - true positives, FP - false positives, FN - false negatives)

Extraction of certain variables was perfect (Gender, PEF, ACQ, AQLQ). This is mainly due to the small size of the data set. However, PEF and asthma questionnaires

(ACQ and AQLQ) use quite small and standard set of cues for presenting these variables, which contributed to the good performance. FEV1 and SGRQ also performed well. Age produced lower scores, however they were comparable with results obtained during our previous evaluation (see Section 6.2.1).

The majority of errors were due to complex table structure including indistinguishable super-rows (see an example in Figure 7.3), or multiple rows presented in one XML table cell (content was aligned visually but visual structure was not supported with the structure of XML). The SGRQ extraction picked a p-value, since a p-value keyword for this variable was not on the blacklist. Interestingly, gender distribution in this dataset performed well, while in our previous experiments it performed with a 0.89 F1-score.

Table 1

Baseline demographics and clinical characteristics (safety population)

	Glycopyrronium bromide 12.5 µg OD (n = 89)	Glycopyrronium bromide 25 µg OD (n = 96)	Glycopyrronium bromide 12.5 µg BID (n = 96)	Glycopyrronium bromide 50 µg OD (n = 92)	Glycopyrronium bromide 25 µg BID (n = 96)	Glycopyrronium bromide 100 µg OD (n = 96)	Glycopyrronium bromide 50 µg BID (n = 87)	Placebo (n = 91)
Age (years), mean (SD)	60.2 (7.77)	60.0 (7.98)	60.9 (7.89)	59.2 (8.14)	61.2 (7.80)	62.1 (7.83)	62.2 (7.74)	63.2 (7.67)
Range	47-79	43-81	40-80	43-80	40-80	40-79	49-81	48-78
Sex, n (%)								
Male	56 (62.9)	64 (66.7)	67 (69.8)	56 (60.9)	66 (68.8)	59 (61.5)	59 (67.8)	56 (61.5)

Figure 7.3: Example of the table in which row containing "Age" was not recognised as super-row and therefore age ranges (in a row below) were not extracted (PMC 3528484)

We also manually evaluated expert-curated data and compared it with data obtained via our method. Human curators extracted more variables and were generally more accurate. However, we did find some mistakes in the curated data (e.g. a misplaced decimal point). Also, one article in the curated data had a reference to the wrong PMCID.

On the other hand, our method referenced all navigational areas with the extracted information, so there was no information loss. The curator was able to define variables, e.g. as pre-treatment or post-treatment FEV1, while our method extracted it as FEV1 values, together with stub information that indicated the time of measurement. A manually curated database for gender distribution was standardized, just to present the percentage of female participants. Our method extracted values that were reported, however, with some additional computing, it was usually possible to induce the number

and percentage of both male and female participants.

7.1.4 Conclusion

This case study presented a real task of information extraction of baseline characteristics from clinical trial publications about COPD and asthma. The case study was performed in collaboration with AstraZeneca and reflects real industrial need for table information extraction.

Information extraction rules can be efficiently developed by reusing the methodology for similar variables, developing keyword lists that indicate presence (or absence) of the variable and iteratively improving the rules. In some cases, intuition about how the variable can be presented was enough to develop efficient rules. Some rules took less than ten minutes to develop to the presented performance level: rules for extracting new variables can be efficiently and rapidly developed.

The performance of the algorithm is promising. With an overall F1-score of 0.924, the method facilitates accuracy and speed of data curation from tables in scientific articles. A degree of quality checking and human involvement is still necessary in order to achieve necessary data quality for quality-sensitive disciplines such as medicine, health care and biology. However, these results are promising, especially considering expert curation is imperfect and significantly more expensive to undertake.

Our approach has its limitations. It cannot handle tables that are not in XML format. In this case study, 6 out of 28 documents (21%) were not processed due to this issue. However, publishers are increasing the number of publications available in XML format thus reducing the scale of this problem for our approach.

7.2 Extracting drug-drug interactions from structured product labels

7.2.1 Introduction

Many people are taking multiple drugs. Over 20% of adults are administered with five or more drugs (Guthrie et al. 2015). When multiple drugs are administered, it is possible that one of the drugs may increase or decrease the effect of the other drug. Drug-drug interactions are a key challenge in drug administration and drug development. During drug development, it is important to identify interactions with other chemical

compounds. However, thousands of people are harmed each year by exposure to two or more drugs for which a known potential interaction exists (Magro et al. 2012). In 1994, it was estimated that approximately 700,000 patients in the United States suffered from some kind of adverse drug reaction, while approximately 100,000 died as a consequence of drug adverse reactions or drug-drug interactions (Lazarou et al. 1998). Drug-drug interactions account for more than 30% of all adverse drug reactions (Iyer et al. 2014). The chance an individual may suffer from adverse drug-drug interaction increases exponentially as a new drug is added to his/her regime (Percha et al. 2012). This is especially harmful to the 29.4% of elderly people, who are prescribed with 6 or more drugs simultaneously (Bushardt et al. 2008). Harm to people can be prevented if practitioners know of potential interactions and effects of prescribed drugs. Unfortunately, there is currently no single, complete, structured source of information for these "potential drug-drug interactions" (PDDIs) (Ayvaz et al. 2015). In the United States, one important information source is drug product labelling, which is required by law to contain information regarding clinically significant interactions (US Food and Drug Administration 2014).

All drug product labels in the United States are freely available through the National Library of Medicine's DailyMed website² in a standard format called Structured Product Label (SPL). While easy to access, a major limitation of SPLs is that information regarding PDDIs is provided as unstructured text and tables in diverse formats. Providing a solution to computationally extract PDDI information from the label into an indexed knowledge base would enable a more convenient access to this information. Moreover, extracted PDDI information could be more easily linked to other sources of information and provide a complete picture of the mechanisms, risk factors, clinical implications, and management options of each PDDI.

Structured Product Labels (SPL) is a document markup standard (a variant of XML) approved by Health Level Seven (HL7) and adopted by the United States' FDA as a mechanism for exchanging product and facility information³. SPL documents annotate certain information, such as drug name, ingredient substances or manufacturer. However, they also contain a number of sections with text, figures and tables. Section names and topics are prescribed by the FDA and annotated with Logical Observation Identifiers Names and Codes (LOINC).

²<https://dailymed.nlm.nih.gov/dailymed/index.cfm>

³<https://www.fda.gov/forindustry/datastandards/structuredproductlabeling/default.htm>

Information about PDDI with their effects is stored in tables under the "Drug interaction" section of SPLs. Tables are a convenient format for storing such information because of their flexibility for storing multi-dimensional data in a dense space. However, there is no standard table format and authors can structure tables and their cells in any way they feel appropriate. Using spanning cells, multiple headers and the emphasis of the cell content, it is possible to specify the semantics of the table and how the table should be read.

Within the drug informatics domain, the SPLICER system (Duke et al. 2013) was successfully applied to extract adverse drug events from tables and text written in the Adverse Reactions section of SPLs. Other efforts focus on side effects and drug indications (Fung et al. 2013, Khare et al. 2014, Boyce et al. n.d.). The SIDER (Side Effect Resource) database uses named entity recognition to extract side effects and indications from product labeling, including SPLs (Kuhn et al. 2015). More recently, starting with full-text papers from the Journal of Oncology, Xu & Wang (2015b) extracted drug side effect relationships, which they compare to the SIDER database. They used Support Vector Machines to classify tables in the literature as side-effect-related or not and then used a dictionary-based approach to extract drugs and side effects based on manually curated lexicons. In 2017, the US National Institute of Technology and Standards organized a shared task in which the goal is to extract adverse drug reactions and a number of related entities (drug classes, severity, factors, negations, animals, etc.) from SPLs⁴. We participated in this shared task (Belousov et al. 2017).

In this case study, we report on a hybrid method, combining machine learning and heuristic rules for automatic extraction of PDDI information from tables found within the "Drug Interactions" section of SPLs. With minor modifications due to the nature of extracting entity relations instead of entities, the method relies on the methodology described in Chapters 4, 5 and 6.

7.2.2 Methodology

The methodology contain six steps (see Figure 7.4): (1) table detection, (2) functional analysis, (3) structural analysis, (4) pragmatic analysis, (5) table annotation and (6) information extraction (in this case study, we only selected cells presenting drug-drug interactions, therefore syntactic analysis was not used). Since drug product labels in

⁴<https://bionlp.nlm.nih.gov/tac2017adversereactions/>

the DailyMed database are in an XML format, table detection is trivial. Also, tables containing potential drug-drug interactions are only in the section describing drug interactions. This section is labelled with LOINC (Logical Observation Identifiers Names and Codes) code 34073-7 and therefore pragmatic analysis is also trivial.

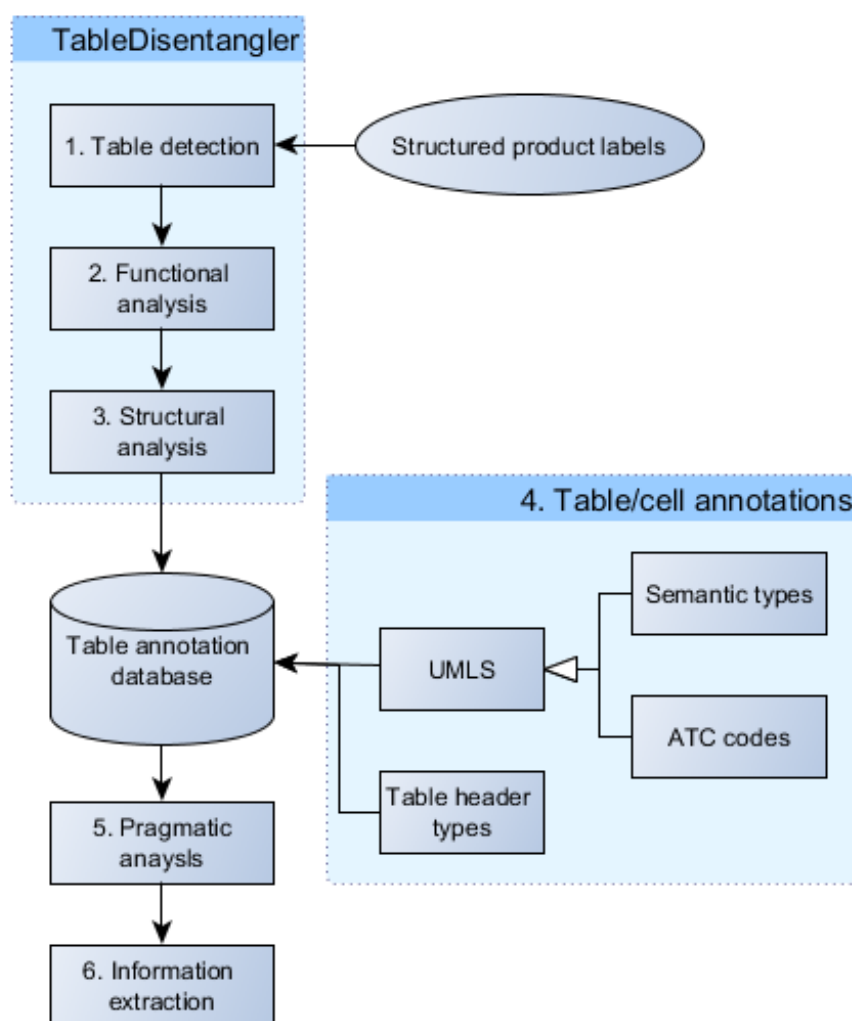


Figure 7.4: Workflow diagram

Table Detection and Extraction

The first step is to enable computational access to data provided in the individual cells of each table. In order to detect tables and extract their content we used TableDisentangler (methodology described in Chapter 4), with a new reader that reads the specific XML structure for SPL documents.

We downloaded structured product labels for all 30,409 prescription drug products as of January 1, 2016, from DailyMed. The full set of SPLs was reduced to a subset of SPLs identified as having at least one table in the Drug Interaction section (section coded with LOINC 34073-7). The data contained 16,211 tables from 1,161 SPL documents. However, only 1,530 tables contained information about drug-drug interactions (they were presented in the drug-interaction section). These SPLs were used as input into TableDisentangler, which parsed and analysed the table content and assigned functional roles and structural relationships to individual cells and annotated the contents of each cell.

Functional and Structural Analysis

Functional analysis determines each cell's functions within each table. Cells are identified as table header, row header, super-row or data cell.

The TableDisentangler methodology is primarily based on emphasis features (see Section 4.2.2). However, the DailyMed dataset does not follow the same emphasis rules, especially for headers. Headers are not divided by horizontal lines and are often not marked with *thead* tags. Approximately 46% of tables presenting drug-drug interactions (565 tables) did not have labelled headers. The caption can be also presented inside *thead* tags, while the actual table header is below, in the body of the table. We queried cell content for cues that indicate caption (word "Table" followed by the number) and found 136 tables containing caption in one of the cells, often labelled as header (see example in Figure 7.5).

We evaluated the performance of the original TableDisentangler header detection algorithm on 20 tables (see Section 4.3.2). The precision was 0.61, while the recall was 0.65. Headers are important for extracting drug-drug interactions since header labels can be used efficiently to craft extraction rules. The evaluation showed that headers have to be treated differently for the DailyMed dataset by taking lexical and semantic cues into account.

Since DailyMed drug labels are different document types from PMC articles, some parts of the methodology, such as header detection, have to be adjusted or changed. We developed a hybrid methodology consisting of a machine learning model and heuristics (see workflow diagram in Figure 7.6). As only the header detection in DailyMed documents is performing with low scores, we used the methodology described in Chapter 4 for classifying stubs and super-rows. Firstly, we TableDisentangler with the standard functional analysis methodology is executed. Secondly, we applied a machine learning

Table 7: Summary of AED Interactions with Oxcarbazepine				
AED Coadministered	Dose of AED (mg/day)	Oxcarbazepine dose (mg/day)	Influence of Oxcarbazepine on AED Concentration (Mean change, 90% Confidence Interval)	Influence of AED on MHD Concentration (Mean change, 90% Confidence Interval)
Carbamazepine	400 to 2000	900	nc ¹	40% decrease [CI: 17% decrease, 57% decrease]
Phenobarbital	100 to 150	600 to 1800	14% increase [CI: 2% increase, 24% increase]	25% decrease [CI: 12% decrease, 51% decrease]
Phenytoin	250 to 500	600 to 1800 >1200 to 2400	nc ^{1,2} up to 40% increase ³ [CI: 12% increase, 60% increase]	30% decrease [CI: 3% decrease, 48% decrease]
Valproic acid	400 to 2800	600 to 1800	nc ¹	18% decrease [CI: 13% decrease, 40% decrease]
¹ nc denotes a mean change of less than 10% ² Pediatrics ³ Mean increase in adults at high oxcarbazepine doses				

Figure 7.5: Example of a table in which both caption and footer are inside the table cells (DailyMed setID: 524c025b-809b-440f-a756-e3518d7c92db)

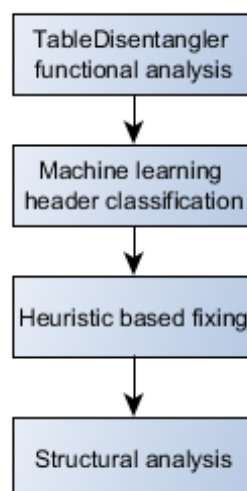


Figure 7.6: Workflow of the modified methodology for functional and structural analysis of DailyMed documents.

algorithm to classify header cells based on their content. In order to train the algorithm we randomly selected 1,000 headers labelled by TableDisentangler from drug-drug interaction tables. For training, we randomly selected 823 labels, 329 headers and 494 non-headers. Thirdly, we used heuristics to post-process functional annotations. We assume that all cells of a certain row have to be either in the header row or outside it. Therefore, if the majority of the cells in some row are classified as headers, then all the other cells in that row are also annotated as part of the header. If a minority of the cells in the row is classified as headers, their annotations are fixed to data cells. Based on experimental experience, we also assume that headers can only be in the top three rows of the table. We noticed manually that there are not many multi-tables among DailyMed drug-drug interaction tables, so it was safe to make this assumption. Relationships between cells rely on functional analysis and so we have not modified our original methodology for structural analysis.

Annotation of Cell Content

We annotated cell content using the Unified Medical Language System's (UMLS) and MetaMap program to identify named entities within the table cells (Bodenreider 2004, Aronson 2001). The annotation method stored the MetaMap annotations as Concept Unique Identifiers (CUIs) linked to data from specific table cells. The UMLS Semantic Network provides a semantic type for each CUI, such as Pharmacologic Substance, Clinical Attribute or Therapeutic or Preventative Procedure. These annotations can be further linked to information from UMLS through CUI, such as ATC (The Anatomical Therapeutic Chemical). Using the ATC codes, we can determine on which organ or

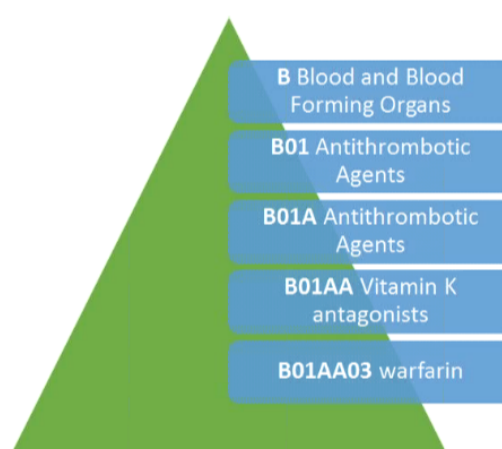


Figure 7.7: ATC coding system

system a drug's active ingredient is acting and whether the cell is describing a single drug or drug group. Drug codes contain seven characters, while drug codes for drug groups or systems on which a drug is acting contain fewer characters. An example of the coding can be seen in Figure 7.7. It is important for users of the drug-drug interaction database to know whether the drug is interacting with the whole drug group or just a single drug ingredient.

Extraction of Drug-drug interactions

Once the tables were annotated, we proceeded with crafting rules for extracting drug-drug interaction. We extracted the drug that the drug label described. This was performed without looking at the table as the document contained XML tag that name the drug SPL refers to. As previously discussed, we only looked for tables presented in the section labelled with LOINC 34073-7. In this case study, we are extracting drugs that interact with the drug the label is about. Therefore, we are dealing with a categorical variable (there is a closed set of possible drugs). The lexical white list for headers contained words "drug", "coadministered" or "co-administered". The header cell should not contain cues like "effect", "dose", "exposure" or "recommendation" (the lexical blacklist). We selected the column defined by the mentioned keywords. Our method extracts cells below the header in the given column unless the column is spanning, is a super-row or the cell is empty.

Extracted cells are saved in the template that stores information about the SPL ID, table id from which the information is extracted, the drug that the SPL is about, the content of the interacting drug(s) cell. Further, the extracted information can be syntactically and semantically analysed in order to obtain one-on-one drug interactions. Often, tables present multiple drugs in one data cell. Authors group cells by drug groups and present multiple drugs from the same group in one cell (see example in Figure 7.8). Our extraction methodology extracts the content of the cell as one interaction entry (as it is in the table). However, in case one wants to obtain the pair of drugs that are interacting, further analysis is necessary. UMLS and ATC annotations provide valuable help in obtaining pairs and recognizing drug groups and individual drugs. However, not all drugs and/or drug groups can be annotated. Therefore, the content must be appropriately split. The content that mixes drug/ingredient names with text (for example about dosage) can be challenging to parse and find the interacting drug. The task involves drug named entity recognition and is beyond the scope of this project.

Table 4 Drugs Tested in <i>In Vitro</i> Binding or <i>In Vivo</i> Drug Interaction Testing or With Post-Marketing Reports	
Drugs with a known interaction with colesevelam	Cyclosporine ^c , glyburide ^a , levothyroxine ^a , and oral contraceptives containing ethinyl estradiol and norethindrone ^a
Drugs with postmarketing reports consistent with potential drug-drug interactions when coadministered with WELCHOL	phenytoin ^a , warfarin ^b
Drugs that do not interact with colesevelam based on <i>in vitro</i> or <i>in vivo</i> testing	cephalexin, ciprofloxacin, digoxin, warfarin ^b fenofibrate, lovastatin, metformin, metoprolol, pioglitazone, quinidine, repaglinide, valproic acid, verapamil

Figure 7.8: Example of a table presenting multiple interacting drugs per cell (SetID: b9df447c-b65b-45b9-873a-07a2ab6e2d1f)

7.2.3 Evaluation

Descriptive Analysis of Table Content

Out of 30,409 prescription drug SPLs, only 1,161 SPLs (3.9%) had tables present in the Drug Interaction section. The selected SPLs contained on average 12.6 tables, while on average 1.24 tables presented potential drug-drug interactions. These 1,161 SPLs included a total of 1,530 tables about drug-drug interactions. Tables in the drug-drug interaction section had 55 cells on average. In this case study, we focused only on tables in the drug-drug interaction section and did not analyze tables in other sections.

Functional and structural analyses

We evaluated the performance of the original TableDisentangler methodology for functional analysis. We randomly selected 20 tables for evaluation and inspected them manually. Cells were considered true positives if their function was correctly annotated. If the correct function was not annotated it was counted as false negative, while if the cell was annotated with incorrect functional annotation, it was considered false positive. The results are presented in Table 7.3.

The performance of header detection, as well as super-rows, significantly dropped in the DailyMed dataset, compared to the PMC dataset for which the methodology was initially developed. For the current case study, it was important to improve header

	TP	FP	FN	Precision	Recall	F-Score
Cell role – header	61	39	32	0.6100	0.6559	0.6321
Cell role – stub	309	0	0	1.0000	1.0000	1.0000
Cell role – super-row	49	6	45	0.8909	0.5213	0.6578
Cell role – data	675	18	104	0.9740	0.8664	0.9171
Overall (micro average)	1,094	63	181	0.9455	0.8580	0.9014

Table 7.3: Functional analysis evaluation of the original TableDisentangler methodology on the DailyMed subset

detection since it was indicating the column that contained drug-drug interactions. We created a dataset from 823 instances: 329 with true header content, 494 with the content of non-header cells. A final year pharmacology student performed annotation. We applied a set of machine learning algorithms on the content of these 823 cells. The 10-fold cross-validation results are presented in Table 7.4.

Algorithm	Precision	Recall	F-score
Naive Bayes	0.588	0.936	0.722
Bayesian Networks	0.559	0.964	0.708
SVM with SMO	0.985	0.821	0.896
C4.5 decision tree	0.944	0.307	0.463
Random forests	0.973	0.875	0.922

Table 7.4: Machine learning header detection using various algorithms and 10-fold cross validation on the created dataset

The algorithm that performed best was the random forest with 97.3% precision and 87.5% recall. We used this model and described post-processing heuristics for the header being in the first 3 rows of the table and the whole row had to be a header on all tables. In this evaluation, cross-validation was performed on cell level. Therefore, cells from the same table appeared in both training and testing sets and the algorithm was able to learn some header terminology. In order to perform a test on a new dataset, we randomly selected 50 tables for training and 50 tables for testing that were manually inspected whether the headers are classified and annotated correctly. The results are presented in Table 7.5.

Dataset	TP	FP	FN	Precision	Recall	F-score
Training data	288	8	41	0.973	0.875	0.922
Testing data	176	59	26	0.749	0.871	0.805

Table 7.5: Machine learning header detection evaluation for the DailyMed subset

As our training data contained only 823 instances, the algorithm did not manage to learn all the possible table header cues. The training data also contained similar entries, while tables selected for testing contained more diverse cues. In short, our training data was not large enough to learn possible cues and achieve performance closer to the training data. However, the results are significantly better than without using machine learning.

Drug-drug information extraction

We used 50 randomly selected tables for rule development and an additional 50 tables for evaluation. The evaluation results are presented in Table 7.6. Our extraction template contained drugs, which the drug label described, interacting drugs and metadata about tables and articles from which data was extracted. If interacting fields contained multiple drugs or drug classes, we assumed correct extraction (true positive). If the algorithm extracted a cell that did not contain interacting drugs or drug classes, we counted it as false positive. If a cell containing interacting drugs or drug classes is missed by the algorithm, it is counted as a false negative.

Dataset	TP	FP	FN	Precision	Recall	F-score
Training data	514	16	128	0.970	0.819	0.888
Testing data	428	45	122	0.904	0.778	0.836

Table 7.6: Evaluation of potential drug-drug interaction pairs from tables in DailyMed.

With an F1-score of 0.877 for the training data and 0.836 for the test data, the results are satisfactory for a drug-drug information extraction task.

However, these scores can be improved with further iterations. In both cases, precision is high and there are not too many false positives. The false positives occurred by collecting rows that described drugs in cells below (usually super-row, see example in Table 7.9) or by selecting the wrong column because of cues missing from the blacklist. The false negatives were more prevalent. A proportion of false negatives were caused by TableDisentangler ignoring cell content after empty HTML characters. Having fixed this issue, the F1-score for training data grew to 0.925, while the F1-score for testing data was 0.904. Other false negatives were mainly caused by changes to table structure in which a new header was presented in a row that overrode the initial header (e.g. see Figure 7.10). For example, the table may present drugs in the first column, while the effect of the interaction is in the second column. However, in the middle of the table a new header presents drugs that increase the effect of some

substance in the first row and drugs that decrease the effect of the same substance in the second column. This is often changed back to the initial table structure by adding a super-row that groups drugs by target organ or disease (see example in Figure 7.10). This way of presenting information is used infrequently.

Table 2: Drug-Thyroidal Axis Interactions		
	Drug or Drug Class	Effect
Drugs that may reduce TSH secretion - the reduction is not sustained; therefore, hypothyroidism does not occur		
	Dopamine/Dopamine Agonists Glucocorticoids Octreotide	Use of these agents may result in a transient reduction in TSH secretion when administered at the following doses: dopamine (≥ 1 mcg/kg/min); Glucocorticoids (hydrocortisone ≥ 100 mg/day or equivalent); Octreotide (> 100 mcg/day).
Drugs that alter thyroid hormone secretion		
	Drugs that may decrease thyroid hormone secretion, which may result in hypothyroidism	
	Aminogluthethimide Amiodarone Iodide (including iodine-containing radiographic contrast agents) Lithium Methimazole Propylthiouracil (PTU) Sulfonamides Tolbutamide	Long-term lithium therapy can result in goiter in up to 50% of patients, and either subclinical or overt hypothyroidism, each in up to 20% of patients. The fetus, neonate, elderly and euthyroid patients with underlying thyroid disease (e.g., Hashimoto's thyroiditis or with Grave's disease previously treated with radioiodine or surgery) are among those individuals who are particularly susceptible to iodine-induced hypothyroidism. Oral cholecystographic agents and amiodarone are slowly excreted, producing more prolonged hypothyroidism than parenterally administered iodinated contrast agents. Long-term amino-gluthethimide therapy may minimally decrease T_4 and T_3 levels and increase TSH, although all values remain within normal limits in most patients.

Figure 7.9: Example of a drug-drug interaction table with super-rows. Often super-rows were not correctly recognised and their content extracted as an interacting drug (SetID: f02310a3-92ea-9ec4-f218-38ddb8eb0334)

The presented approach for extracting drug-drug interactions extracts cells presenting drugs related to the main drug that the drug label is about. However, in order to obtain drug-drug interaction pairs, more work has to be done on splitting the content

Table 2: Drug-Thyroidal Axis Interactions		
	Drug or Drug Class	Effect
Drugs that may reduce TSH secretion - the reduction is not sustained; therefore, hypothyroidism does not occur		
	Dopamine/Dopamine Agonists Glucocorticoids Octreotide	Use of these agents may result in a transient reduction in TSH secretion when administered at the following doses: dopamine (≥ 1 mcg/kg/min); Glucocorticoids (hydrocortisone ≥ 100 mg/day or equivalent); Octreotide (> 100 mcg/day).
Drugs that may alter T_4 and T_3 serum transport - but FT_4 concentration remains normal; and, therefore, the patient remains euthyroid		
	Drugs that may increase	Drugs that may decrease
	serum TBG concentration	serum TBG concentration
	Clofibrate	Androgens / Anabolic Steroids
	Estrogen-containing oral	Asparaginase
	contraceptives	Glucocorticoids
	Estrogens (oral)	Slow-Release Nicotinic Acid
	Heroin / Methadone	
	5-Fluorouracil	
	Mitotane	
	Tamoxifen	
Drugs that may cause protein-binding site displacement		
	Furosemide (> 80 mg IV) Heparin Hydantoins Non Steroidal Anti-Inflammatory Drugs - Fenamates - Phenylbutazone Salicylates (> 2 g/day)	Administration of these agents with levothyroxine results in an initial transient increase in FT_4 . Continued administration results in a decrease in serum T_4 and normal FT_4 and TSH concentrations and, therefore, patients are clinically euthyroid. Salicylates inhibit binding of T_4 and T_3 to TBG and transthyretin. An initial increase in serum FT_4 is followed by return of FT_4 to normal levels with sustained therapeutic serum salicylate concentrations, although total- T_4 levels may decrease by as much as 30%.

Figure 7.10: Example of a drug-drug interaction table that changes overrides the way of data presentation defined in the header in rows 5 and 6 (SetID: f02310a3-92ea-9ec4-f218-38ddb8eb0334)

of the cell and identifying drugs and drug groups. The only attempt made by this approach was identifying drug groups using ATC codes. However, many reported drugs and chemicals are not part of UMLS or ATC. Therefore, we leave identifying individual drugs and drug groups from the identified cell for the future work.

7.2.4 Conclusion

To the best of our knowledge, no other project has attempted to extract PDDI information from tables in SPLs. We found that TableDisentangler could load and conduct a functional and structural analysis of all Drug Interaction section tables. Once tables are disentangled and the headers recognized, the extraction rules based on whitelists and blacklists can successfully extract cells containing drugs and substances that interact with the SPL target drug SPL (with a 0.904 F1-score).

The results suggest that it is feasible to construct scalable rules for extracting PDDI information from tables found in the Drug Interactions section of SPLs. Extracted potential drug-drug interactions provide important information about drug administration that can be used in clinical decision support systems. The method presented here allows population and updating of database that can be used for clinical decision support. The extracted information is indexable and searchable and therefore, can prevent malicious administration of drug combinations in medical decision support systems.

Extracting drug-drug interactions requires relation extraction, where one part is in a table (interacting drug), whilst the other may not be there (the drug the SPL refers to). In this case, it is necessary to expand the methodology to incorporate integration between text and table information extraction. However, the proposed methodology is still useful and followed. The original information extraction approach is limited to extracting information presented only in tables however, information presented in tables will often have context presented elsewhere in the article, as with drug-drug interactions. The approach to deal with such relationships has to include a post-processing relation resolving step.

7.3 Summary

In this chapter, we presented two information extraction case studies. Both case studies present real world problems faced by biomedical domain researchers. The first case study was designed in collaboration with AstraZeneca and its biomedical informatics and text analytics teams. The second case study was designed in collaboration with the University of Pittsburgh and its Biomedical Informatics department.

The first case study presented how clinical trial baseline information can be extracted using the methodology we have developed in this thesis. This case study showed promising performance and ease-of-use for information curation from tables in biomedical literature.

The second case study presented extraction of drug-drug interactions from tables. It was done using the described steps and the recipe we proposed. However, the second part of the information extraction methodology required an extension for resolving relationships with entities that were not presented in the table.

The presented case studies showed power, accuracy, ease-of-use and limitations to the methodology we developed. However, we have presented how it is possible to use the same methodology, to overcome some of the limitations.

Chapter 8

Discussion

This chapter explores our research questions, discusses challenges, generalizability and some limitations of this work.

In this thesis, we designed, developed and used an integrated methodology for information extraction from tables in the biomedical domain that contains the following steps:

1. Table detection (Chapter 3)
2. Functional analysis of tables (Chapter 4)
3. Structural analysis of tables (Chapter 4)
4. Pragmatic analysis of tables (Chapter 5)
5. Semantic tagging (Chapter 5)
6. Cell selection (Chapter 6)
7. Syntactic processing of cell content and extraction of information (Chapter 6)

In Chapter 3 we gave an overview of how the methodology overall works. Initial case studies that helped methodology development are presented in Chapters 5 and 6. In Chapter 7, we presented several case studies in information extraction that validate our methodology.

8.1 Research questions

In Chapter 1, we presented a set of research questions. In this chapter, we will discuss each of them.

What levels of processing are needed to disentangle a table's internal structure from its visual representation?

Compared to text, data in tables is organized using visual structures. These visual structures allow tables to present multi-dimensional information in a dense space. Cells in tables are visually related and, based on these relationships, it can be determined which navigational cell is describing a particular data cell. Therefore, compared to free text data that runs linearly (i.e. one-dimensional), tables contain additional complexity in their visual structure. In order to mine the content of a table, its visual structure needs to be disentangled and computationally modelled.

Once a table is detected, two layers of processing are necessary to disentangle the table's visual representation — functional analysis and structural analysis. As described in Chapter 4, the role of functional analysis is to recognize the functions of the cells in the table. In other words, during the functional analysis, cells are classified as part of the header, stub, super-row or data area of the table. This can be achieved using a set of heuristic rules or machine learning. Different document sources (document databases) may have different styles of tables and sometimes it is necessary to modify the functional analysis methodology to be able to efficiently detect functional areas (see drug-drug interaction extraction case studies in Chapter 7). Structural analysis uses the results of the functional analysis to link cells in the table that are functionally related. In other words, structural analysis finds navigational cells for each cell in the table. If the cell functions are known, this can be done using a set of heuristic rules about the functional areas and structure of the table. These two levels of processing disentangle the tables visual representation and serve as a basis for further table mining processing layers (pragmatic, lexical, syntactic, semantic).

What levels of processing are necessary for extracting information from tables?

As discussed above, two layers of processing — functional and structural analysis — are necessary in order to disentangle a table's visual representation. The table's cell content can be textual, numerical, symbolic or mixed. Therefore, different layers of processing need to be applied, with specific modifications for table data. The textual processing layers that need to be applied to extract table data include:

- Pragmatic analysis – analyses what is the function of the table in the document.

It answers the question of what kind of data is presented in the table or what is the table's purpose in the document.

- Semantic tagging – maps the table content to the relevant concepts in a knowledge resource. Semantic tagging can be achieved by using semantic knowledge sources or databases and semantic taggers.
- Cell selection – Selects the cell containing value of the target variable. In the proposed methodology, a rule-based approach was used. The approach utilises matching of semantic concepts, semantic types or lexical cues in order to select the relevant cells.
- Syntactic analysis – analyses the presentation patterns of a given cell. Often in table processing, syntactic analysis is performed in combination with lexical and semantic analysis of the content, in order to extract the right piece of information from the cell. Syntactic analysis depends on the variable type. If the variable is numerical, it analyses the presentation patterns of the values in the table and assigns meanings to the pattern components. Presentation patterns are often shared in different tables and variables. Some of the patterns are established in certain fields, such as presenting statistical information (mean, standard deviation, ranges, etc.). Therefore, a library of patterns can be reused for the extraction of different variables. On the other hand, if the variable is categorical, the algorithm searches for a possible category in the cell content and extracts it. If the variable is textual, further syntactic, lexical and semantic analysis has to be performed over the text in order to extract more granular information, but this was outside the scope of this thesis

What information and knowledge about data is necessary in order to design and implement information extraction from tables?

Designing and implementing information extraction from tables requires knowledge about variables targeted for extraction, such as background information, lexical and semantic cues and syntactic presentation patterns that are commonly used for that variable. During the design phase, it is necessary to define what type of variable is supposed to be extracted (numerical, categorical, textual). In the case of numerical information, it is necessary to define the possible and default units of measure.

Lexical or semantic cues, as well as syntactic patterns, may be in any functional area and so the designer of the information extraction process has to define where the cue should be searched for and the location of the variable for extraction. Cues and information of interest for extraction do not have to be in the same functional areas. Lexical or semantic cues are commonly located in the navigational area of the table (header, super-row or stub), while the variable values are located in data cells. Syntactic analysis is usually performed on the content of the cell where information of interest is located, but it may also take into account syntactic patterns of the relevant navigational areas (for example, matching values from the pattern with their descriptions in navigational cells). Another specific feature in designing table information extraction can be the use of pragmatics. It is relatively rare to use pragmatics for information extraction from text. However, pragmatics can significantly reduce the search space for information of interest in tables. It classifies tables by the information presented and therefore an information extraction system can only analyse tables that contain relevant groups of information.

What levels of table processing would benefit from rule-based approaches rather than machine-learning, given the typical short text available in tables?

For each layer of table processing, we analysed whether it benefited from machine learning or rule-based approaches. For each step, we proposed the approach that is the most appropriate for that step.

Functional analysis. Both machine learning and rule-based approaches for functional table analysis are detailed in the literature (Yildiz et al. 2005, Fang et al. 2011, Ng et al. 1999, Son et al. 2008, Liu 2009). However, prior work did not analyse in which case a rule-based approach is more appropriate than machine learning-based and vice versa. Our study analysed tables in XML formats in two document databases: PMC and DailyMed. The documents in these databases differ significantly in table structure, emphasis features (e.g. position and alignment of cells, their content, lines, text emphasis, etc.) and markup. These differences provided insights on the benefits and shortcomings of each approach. In datasets where emphasis features can be clearly identified and used to distinguish navigational and data areas, a rule-based approach

performs well (as presented in Table 4.6 in Chapter 4, the F1-score of rule-based functional analysis in the PMC dataset was over 0.94). When tables do not use clear emphasis features in their markup to distinguish navigational and data areas of the table, it is necessary to rely on other features, such as lexical cues and semantics. In such databases, functional analysis can benefit from machine learning or hybrid approaches (a mix of machine learning and heuristic rules) for specific functional areas (typically in a given domain). However, machine learning is domain dependent and requires adjusting and retraining to be applied to the domains for which the algorithm was not trained.

Structural analysis. Structural analysis depends on the results of functional analysis. Once headers, stubs, super-rows and data cells are recognized, the set of rules is able to disentangle relationships between the cells. Therefore, in structural analysis, machine learning was not needed as a set of heuristics based on the composition of functional areas was able to recognize inter-cell relationships.

Pragmatic analysis. A machine learning approach was primarily chosen in this research for pragmatic analysis. Once table disentangling was performed using functional and structural analysis, features for machine learning pragmatic analysis were obtained in a relatively consistent manner (content of navigational areas, captions, footers, referring sentences). Our experiments suggest that a dataset for training a machine learning classifier can be made manually, by labelling around 100 tables for each pragmatic table type. The more specific the pragmatic types, the more accurate the classifier was. In some rare document data stores, it may be possible to craft rules for pragmatic classification based on the position of the table (e.g. in DailyMed).

Semantic tagging. Semantic tagging has a role to enrich and normalise the content of the table, by linking it to the concepts in some semantic knowledge source. Semantic tagging is usually performed using a semantic knowledge source or semantic taggers. Some semantic taggers may apply machine learning approaches (Reeve & Han 2005), while the majority of semantic taggers that we applied use rule-based matching to the knowledge source (Aronson 2001, Miles et al. 2005, Miller 1995).

Cell selection. Cell selection has a role to select target cells that contain the variable or its value. Selecting cells for information extraction can be performed using rules or

machine learning. The process of creating the training dataset for selecting cells for information extraction is a time-consuming process. Also, the created dataset is likely to be highly imbalanced, therefore techniques like cost-sensitive learning have to be applied in order to achieve satisfactory results (e.g. above 0.70 F1-score, see Section 6.2.2). Similar and even better results can be achieved using rule-based approaches and iterative improvement of rules (as demonstrated Chapter 6). The content of the cells in the table is usually short text, often abbreviated, therefore any grammatical features would not be helpful, so a bag of words approach and lexical cue matching approaches provide good results (0.80-0.94 F-scores).

Syntactic analysis. Syntactic analysis is the final step in analysing patterns of value presentation in cells and extracting information of interest. On one hand, values are often presented in a finite set of patterns so it is possible to craft syntactic analysis rules and point to the information to be extracted. On the other hand, it seems more labour intensive to model this as a machine learning task. Syntactic analysis can be modelled as a sequence labelling task, in which cell values are parsed and their meaning recognized (e.g. *mean, standard deviation, range minimum, range maximum, percentage, number of male participants, etc.*). In order to apply this method and train the model it is necessary to have a substantial dataset with a number of examples for each data presentation pattern. Also, semantics of the value are often determined by the pattern in the navigational area (e.g. content "*male/female*" in stub explains the meaning of the value "*19/22*" in the data cell). It is challenging to construct a machine learning approach that is able to analyse and link value descriptions with values in different table areas. Because of these challenges, our methodology used a rule-based approach.

How can table information extraction benefit from domain specificity? Which steps of the methodology are domain dependent and which ones are domain independent?

The task of information extraction is usually seen as domain dependent. Information extraction requires domain knowledge in order to find the variable of interest and to narrow the search scope for the given variable. However, some steps of the methodology may not be domain dependent or may be domain independent in certain special cases.

Our methodology for information extraction from tables has seven steps. *Table detection* is almost in all approaches that can be found in literature domain independent. Since our method is identifying tables based on a particular XML tag, it is domain independent. *Functional analysis* can be domain independent in case tables are emphasizing functional areas such as header. However, not all authors and datasets use emphasis for functional areas. In the case when there is no emphasis, machine learning can help identify functional areas. However, machine learning based approach makes functional analysis domain dependent. *Structural analysis* is based on the output of functional analysis. Once the functional analysis is performed, the structural analysis uses a set of rules to connect related cells. Therefore, structural analysis is domain independent. Domain dependence of the *pragmatic analysis* depends on the structure of the dataset. Some of the datasets contain a particular tag or marker, that indicates the purpose of the table. This tag or marker may not be related to a single table, but in some cases to the whole section and all the tables in the given section would have the same pragmatics. This is a case in DailyMed database. However, some databases or document types do not contain this. For example, in PMC database, tables with different pragmatics or purpose may be in a single section. For these cases, machine learning and the domain specificity can help identify the pragmatics of tables. *Semantic tagging* is always domain dependent, since it usually uses a domain-specific semantic resource. *Cell selection* is also domain dependent, since the cell selection rules contain cues that are representative of the domain-specific variable that should be extracted. *Syntactic analysis* is domain dependent, since different domains may present variables differently, or values may have a different meaning (e.g. a part of the value that has a meaning of a standard deviation in the biomedical domain may have a meaning of a standard error in mathematics or computer science). However, certain domains share value presentation patterns, therefore they can be reused or transferred to the other domain.

How can the data represented in tables be interpreted (i.e. linked to meaning as represented by information extraction slots)?

In this research, we extracted data from tables into structured templates that contained variable identifiers that could be linked to a semantic resource, identifiers of the document and table from which information was extracted and information that additionally described the extracted values and its context (e.g. a clinical arm and unit of measure).

The data in the suggested template format is stored in the structured database. However, it can be easily converted to another format, such as linked data and linked with ontologies and other semantic resources for further value interpretation.

Can surrounding text that refers to a table help interpret the table's data?

Tables are regularly referred to by the surrounding text. In the text, authors highlight and often discuss data in the table. However, the discussion is usually only about a part of the table (e.g. the most significant value from the table). In Chapter 5, we evaluated how much the surrounding text that refers to a table helps in the pragmatic analysis of the table (i.e. describing what information is grouped and presented in the table). When referring text was used for pragmatic classification, the method produced a 0.62 F1-score. When a caption was used to determine the pragmatic type, the method produced a 0.93 F1-score. Similarly, content from other table areas, such as stubs, were more informative about table pragmatics than a referring sentence. Therefore, experiments suggest that sentences referring to the table do not contribute much to understanding and interpretation of table data. For some tables, the referring sentence or text may indicate the most important values in the table for a particular research article topic.

What levels of accuracy would facilitate efficient data curation on a large-scale to support information extraction from clinical trials?

Efficient data curation has two dimensions: speed of curation and accuracy of the automatic curation process.

In terms of curation speed, research by Alex et al. (2008) and Donaldson et al. (2003) claimed that natural language processing-assisted curation from text can speed up the process by 20-70%, depending on the complexity of the task. Similar levels of acceleration can be achieved considering table data. Tables are rich with factual information that is important for reproduction and further research. Automation is required as manual extraction of the table data is a labour intensive and slow process. The main goal of assisted curation is to point a curator to extracted data, where the curator needs to check or improve the data quality.

When considering the level of accuracy that would facilitate efficient data curation on a large-scale, it is desirable to consider human curators' accuracy levels. Human curator accuracy levels can be estimated via inter-annotator agreements for annotating

entities in text and tables. The process of determining the inter-annotator agreement consists of multiple annotators annotating a document and calculating the consistency of their annotations (Brants 2000). Unfortunately, to the best of our knowledge, there is no research that calculates inter-annotator agreement for information extraction in tables. Inter-annotator agreement for annotating protein-related named entities in text produced a 0.849 F1-score while protein-protein interaction produced a 0.6477 F1-score (Haddow & Matthews 2007, Alex et al. 2008). Alternatively, inter-annotator agreement was high for data such as patient date of birth - Cohen's kappa of almost 100% - while cancer staging measures, the number of metastases and other complex information was much lower in terms of inter-annotator agreement (Warner et al. 2013). According to the literature, demographic information has high inter-annotator agreement (e.g. age has agreement of 99-99.7%, race has agreement of 97-99.5%, height 94-98.5%, weight 95.7-98.5%) (Shiloach et al. 2010).

Currently, natural language processing cannot match the performance of human annotators in text or tables. For extracting information about the age of patients, the presented methodology in this work performed with an F1-score of 0.884. A number of patient variables were extracted from the test set with a 0.839 F1-score, adverse events for drugs with a 0.921 F1-score and age distribution a 0.891 F1-score. In the baseline characteristic extraction case study, there were several variables with an F1-score of 1 (ACQ, AQLQ, PEF, SGRQ, Gender). This may be due to a small testing dataset because some of these information classes have a relatively small and standardized form of presentation. The presented results are the state-of-the-art in table information extraction. The scores are lower than human scores however, the difference is lower than 10% - in many cases lower than 5% - so the results suggest that assisted information extraction using table mining is likely to boost the speed and performance of creating curation databases, therefore significantly reducing their cost. With training on larger data sets and crafting more specific rules precision and recall can be improved.

8.1.1 Hypothesis

Since the proposed methodology for information extraction produced reasonable results, we confirmed our suggested hypothesis: that a multi-layered approach to mining information from tables can facilitate large-scale, semi-automated extraction and curation of data stored in tables in a specific biomedical domain. Multi-layer analysis of tables is required because of tables' structural, linguistic and semantic complexity. The

methodology was applied on a large scale in our case studies and we extracted demographic and adverse events information from clinical trials and drug-drug interactions from drug labels.

8.2 Challenges

The work presented in this thesis addresses several challenges in table mining, mainly related to table disentangling and information extraction processes:

- **The use of XML tags and attributes.** Table disentangling relies on how the table XML is read. However, XML used for presenting tables and emphasizing cells or areas in the table can be different for different publishers. Even the same publisher, who manages multiple-document databases, may use different XML tags and attributes for presenting tables in these databases. The U.S. National Library of Medicine maintains both PubMed Central (PMC) – a database of open access articles in biomedicine, and DailyMed – a database of structured product labels of approved drugs in the United States. Although both databases use XML for document representation, the use of attributes and XML features differs significantly. In PMC, the header is marked with *thead* tags in the majority of cases and where it is not, either a horizontal line is present or spanning cells are grouped in upper layers of the header. Thus, straightforward and accurate heuristics can be developed for detecting the header area. The same heuristics do not work with DailyMed data because *thead* and horizontal lines are not used to mark headers, and often captions are misplaced in cells marked with the *thead* tag. Instead of the mentioned features, some cells are distinguished using the *class* attribute and the visual emphasis is achieved by using the combination of *class* attributes and cascading style sheets (CSS). Spanning cells are often represented as multiple cells, where only one is non-empty. While in PMC this is usually the first one, in DailyMed it is often the central one. Because of the stated differences, it is hard to generalize methodology based on emphasis features.

Although it is possible to provide general heuristics about table structure and emphasis, which we did in Chapter 4, an XML reader that utilizes specific attributes and features of that particular data has to be developed for each dataset. We tried to overcome the issue in Chapter 4 and 7 by using machine learning in combination with heuristics. However, in that case the methodology becomes

domain dependent. Publishers can assist this process by standardizing the set of XML features they use and by using features appropriately such as *thead* tags or spanning cells. However, before publishers standardize the use of XML features they use to publish tables, generalizing the approach and developing tools to the wider set of XML-based data sets remains a challenge that requires further development.

- **Semantics of table data.** Table cell content contains text that is often ungrammatical, abbreviated and short. In the cases of short and abbreviated text, simple cue matching techniques perform well. However, semantic analysis relies on functional analysis and disentangled inter-cell relationships. It is often not possible to find the meaning of a certain data cell without knowing which header, stub or super-row cells describe the given cell. Semantic knowledge sources can facilitate understanding of table data and information extraction. If knowledge sources are not available, synonyms, abbreviations and acronyms for the concept that should be extracted have to be specified as rules. Missed terms can reduce the extraction process performance.
- **Syntactic analysis and value presentation patterns.** In this thesis, we have presented an approach to analyse and assign values to various value presentation patterns in table cells using regular expressions and a set of value component assignment rules. The assignment rules search for keywords in navigational areas that facilitate assignment of semantic value. Different variables may use the same value presentation patterns, therefore the rules are reusable, since value types, such as statistical values and single numerical values, can be analysed with the same patterns. Still, with the degree of freedom authors have in terms of value presentation, capturing variable values can be challenging. Sometimes, multiple values are presented in a single data cell, while the description of each value is presented in a navigational cell. Linking values and their descriptions in other table areas can also be challenging.

8.3 Generalizability

Although designed primarily for the biomedical domain, the information extraction methodology and all analysis layers presented in this thesis were designed with generalization in mind.

While we aimed to provide a generalizable framework for table information extraction from documents in XML formats, we only partially managed to generalize certain steps. As previously mentioned, reading XML documents and performing functional analysis depends on XML tags and attributes used in a given dataset. However, we have provided a set of heuristics based on emphasis and table structure. These heuristics are applicable to most of the data sets. However, reading of the XML and tags and attributes with which structure is described in a given dataset has to be implemented for specific datasets. Our heuristics include:

- For recognizing headers:
 - Headers are often emphasized in bold, with a different colour or font compared to the rest of the table
 - Headers are often separated by horizontal lines.
 - Headers are usually on top of the table. Multiple layered tables can be recognized by spanning cells that group concepts below it, until the first row that contains no horizontally spanning cells.
 - Header's content is often of different syntactic type (e.g. text) compared to data cells below it (e.g. numerical).
- For recognizing stubs:
 - A stub is usually the left-most column of the table.
 - In cases where the most left column contains vertically spanning cells, the stub contains multiple columns until the first column without vertically spanning cells.
- For recognizing super-rows:
 - Super-row cells often span cells across the whole row (this may also be presented as a row of cells, with a non-empty cell only in the beginning or in the middle).
 - Cells below super-row in the stub contain indentation

Our experiments suggest that with these heuristics, it is possible to distinguish functional areas in most datasets. In case some of the heuristics are not present in the given dataset, our methodology can be adjusted, for example by using a machine learning or hybrid method.

Semantic tagging that we performed using UMLS is specific to the biomedical domain, but specific tagging tools can be used for other domains.

Pragmatic analysis is performed using a machine learning method, relying on the content of captions, headers, stubs and super-rows. It is a domain dependent task as it uses the content of the cells and caption as features. However, a pragmatic classifier can be trained for other domains and tasks.

The TableInOut method was developed for crafting lexical, semantic and syntactic rules and is domain independent. It relies on functional and pragmatic analysis output as well as semantic tagging. It is based on term and pattern matching so it is domain independent. We successfully applied it to the biomedical domain, obtaining F1-scores of 0.82-0.94 for information extraction, depending on the complexity of the task, tables and quality of semantic annotations. With some domain- and dataset specific modifications, the methodology can be applied to other domains.

8.4 Data curation and querying

Data output from our systems (both *TableDisentangler* and *TableInOut*) is stored in a relational database. The data could also be in other formats, most notably in the form of linked data stored in a triple store. Linked data is a convenient format for linking data from our data store to other knowledge bases and data stores.

Since a linked data format is able to represent any relational data, the data from the relational database can be in the future transformed to linked data (Konstantinou & Spanos 2015).

We have identified places in our methodology where curators can check and improve the quality of data, as well as places where data can be useful for users. The first point, where curators can check the data is after the functional analysis. Detected functional areas are the main feature for structural analysis. Also, information extraction depends on correctly recognized functional areas. We propose a curation interface, in which curator can see the original table and the table in which functional areas are labelled (or coloured). The curator can check and change labelling for each cell, based on which further steps of table processing are performed (see Figure Figure 8.1).

Table data can be queried after the structural analysis. After this step, document, table and cell level information retrieval can be performed. The table data output from structural analysis stored in the form of relational database can be used or this data can be additionally indexed using information extraction techniques, potentially giving

A	B	C	D
N = 15	Mean	SD	Range
Age (yr)	67.5	6.0	56 – 76
BMI (kg/m ²)	26.9	4.9	15.4 – 35.2
Smoking (pack-year)	8.8	13.9	0 – 50
FEV1(L)	0.76	0.3	0.44 – 1.80
FEV1(%)	34	12.4	15 – 63
FVC (L)	1.25	0.5	0.77 – 1.69
FVC (%)	44		26 – 72
FEV%	61	13.7	47 – 77
Arterial pH	7.38	0.05	7.25 – 7.45
PaCO ₂ (kPa)	6.0	1.1	5.5 – 9.9 74.2
PaO ₂ (kPa)	9.0	1.2	5.6 – 12.6
Systolic BP (mmHg)	154	21.8	120 – 190
Diastolic BP (mmHg)	87	14.1	60 – 118

Figure 8.1: Curation interface for checking and improving quality of data after the functional analysis. On the left is original table, while on the right is the same table with functional annotations (colours). The interface was implemented as an independent project (Su 2016).

different weights for different table elements. The following types of queries can be performed over the data:

- Retrieve documents that contain a certain keyword in table.
- Retrieve documents that contain in certain functional area (e.g. header, stub, super-row, data cell) given keyword or concept annotation.
- Retrieve tables that contain certain keyword.
- Retrieve tables whose referring sentence, caption or footer contain given keyword.
- Retrieve tables from document which title, abstract or text contain given keyword.
- Retrieve tables that contain given keyword in a given functional area (e.g. only header, header or stub, etc.).
- Retrieve cells that contain given keyword.
- Retrieve cells that contain given keyword in related cells (e.g. retrieve content of the cells whose header contain "placebo").

- Retrieve cells that contain given annotation in themselves or in related cells (e.g. retrieve cells that have annotation for BMI concept in stub, retrieve cells that are annotated as "symptom or disease").

As evidenced, the queries can be quite specific and provide useful information from tables. We have developed a demonstration web application that demonstrates querying over structurally analysed table data (see Figure 8.2). The web application is available at http://gnteam.cs.manchester.ac.uk/demos/table_explorer/. The web interface was partially developed as an independent project at the University of Manchester (Tang 2016).

Select: ☒ Table ☐ Cell ☐ Count (Table) ☐ Count (Cell)

Table constraints:

Field	Operations	Value	And/Or
Caption	Contains	baseline	And
Caption	Contains	Age	

Submit

Result

idTable	PMCID	TableOrder	TableCaption
30	1079947	Table 1	Characteristics of patients at baseline.
31	1079947	Table 2	Objective sleep quality at baseline, N = 15.
34	1087493	Table 2	Baseline Demographic and Clinical Characteristics
43	1097745	Table 1	Baseline characteristics of study patients, and number of patients with pathologic findings in bicycle or scintigraphy stress-test.
84	116603	Table 2	Baseline comparison between treatment and placebo groups
158	1208942	Table 1	Baseline demographic characteristics of individuals.*
160	1208953	Table 1	Baseline Characteristics
181	122063	Table 1	Summary of Demographic Data and Disease Characteristics at Baseline (all randomized subjects)

Figure 8.2: The interface for querying structurally analysed table data (Tang 2016)

The final point in our methodology where curators can check and improve data quality is after the final step of information extraction. The curation interface should show extracted data in templates and original tables from which the data was extracted. Also, as this is the final step of the presented table information extraction methodology, data can be used for querying and development of medical decision support, knowledge management or question answering systems.

8.5 Table and cell annotation

Annotation of the documents, including their components, such as tables, is a data curation task. The existence of table annotation schemata and annotated corpora, would

facilitate the development of methodologies for table information extraction. Currently, most annotation formats focus on annotating textual documents. A range of both in-line and stand-off annotation formats for annotating text has been developed over time. However, the majority of these formats and annotation tools do not support annotation of XML documents and tables. While some may be used to annotate text in XML, they do not store the structure, which is crucial for table annotation. Lack of annotation tools and formats slows table mining research and makes it more difficult to perform research using techniques that traditionally rely on annotations (such as machine learning training). During the second Biomedical Linked Annotation Hackathon (BLAH2, 2015 in Mishima, Japan), we proposed a modification of PubAnnotations (Kim & Wang 2012), based on XPath that can be successfully utilized to annotate XML documents and tables located in these documents¹. The idea was expounded further during the Biomedical Linked Annotation Hackathon in Munich, Germany the following year, when we began development of an annotation tool based on the proposed scheme.

```
{
  "xml": "<table>
    <tr><td>parameter</td><td>number</td></tr>
    <tr><td>male/famale</td><td>15/18</td></tr>
  </table>",
  "denotations": [
    {"id": "T1", "xpath": "/table/tr[1]/td[1]", "obj": "Header"},
    {"id": "T2", "xpath": "/table/tr[1]/td[1]", "obj": "Stub"},
    {"id": "T3", "xpath": "/table/tr[1]/td[2]", "obj": "Header"},
    {"id": "T4", "xpath": "/table/tr[2]/td[1]", "obj": "Stub"},
    {"id": "T5", "xpath": "/table/tr[2]/td[2]", "obj": "Data"},
    {"id": "T6", "xpath": "substring(/table/tr[1]/td[1],2,5)",
    "obj": "substringEx"}
  ],
  "relations": [
    {"id": "R1", "subj": "T4", "pred": "dataOfHeader", "obj": "T1"},
    {"id": "R2", "subj": "T5", "pred": "dataOfHeader", "obj": "T3"},
    {"id": "R3", "subj": "T5", "pred": "dataOfStub", "obj": "T4"},
  ]
}
```

Figure 8.3: Format of the proposed annotation schema

The basic idea of the annotation schema was inherited from the PubAnnotation

¹<https://docs.google.com/document/d/1aZoT3yMZjN8bv952F1jpHWgMCOxYrtkE57fy219sNvY/edit?usp=sharing>

schema, in which it is possible to annotate denotations or concepts, relations and modifications. Denotations or concepts are simple mentions of something an annotator wants to annotate. The relation is the relationship between two concepts. Modifications modify the concept (negation, hypothetical, etc.). While standard PubAnnotations are designed for annotating text and store the span of the annotated text in its annotation format, the proposed annotation format uses XPath instead of the span. With the use of the XPath, it is possible to point at any part of the XML structure. However, between XML tags there may be multiple words, out of which only a few annotators may want to annotate. This is possible to achieve using the XPath substring function. Using the substring function, we can point to the span inside the content of the selected XML tag. An example of the proposed annotation schema with annotated denotations and relations can be seen in the Figure 8.3. We also developed a proof-of-concept RichAnnotator tool, that is able to make denotation (concept) annotations²³. The current state of the RichAnnotator tool can be seen in Figures 8.4 and 8.5.

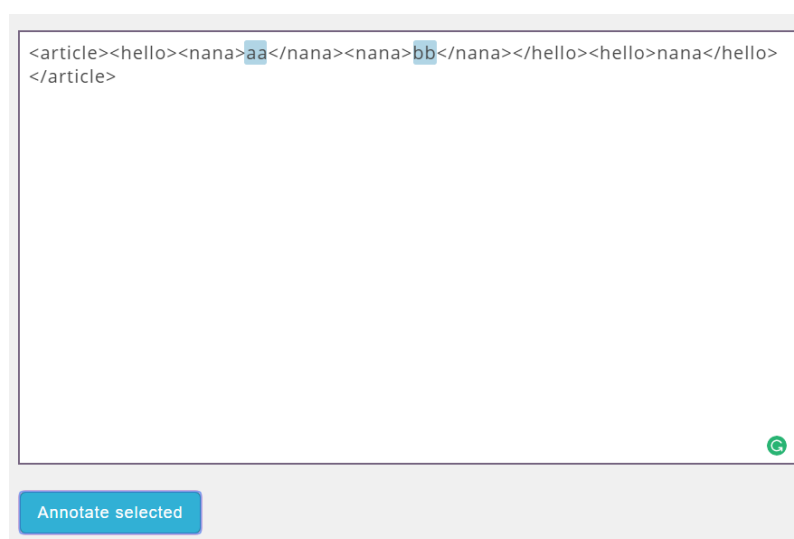


Figure 8.4: RichAnnotator tool index page, showing sample XML document with annotated concepts

Visualization is still at the XML level although it would provide a better user experience if XML was interpreted. Also, the current version of the tool does not support relations and modifications. However, since table annotation was not the focus of this project and we used other means of training machine learning algorithms using table data, the final development of XML and table annotation tool remains a future task.

²<https://gist.github.com/nikolamilosevic86/c94382d4b52705e9ae75dab0eda6381e>,

³<https://github.com/nikolamilosevic86/RichAnnotator>

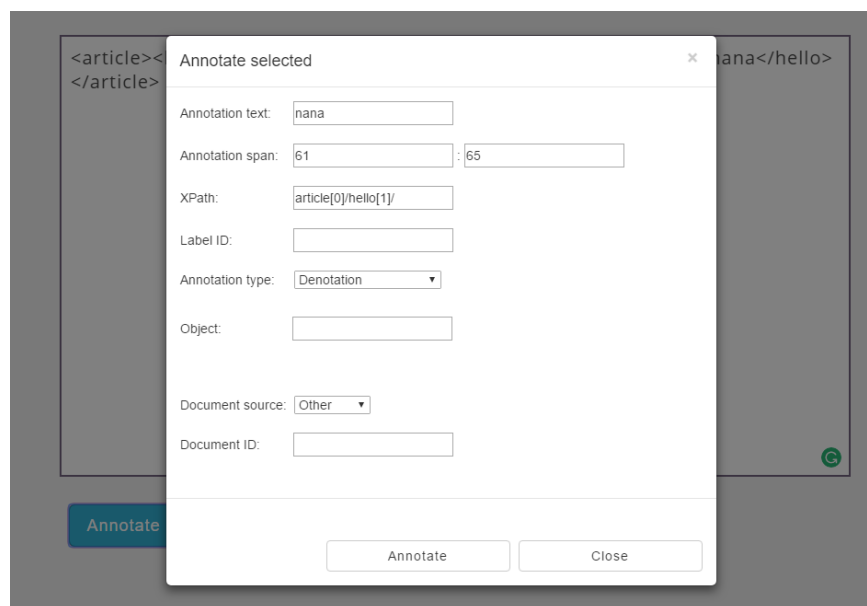


Figure 8.5: RichAnnotator tool - annotation screen automatically finds XPath of selected item. Annotator has only to fill the concept details.

Interestingly, PAULA XML (Potsdam Exchange Format for Linguistic Annotations) followed a similar path, utilizing XPointers for selecting annotated XML tag (Zeldes et al. 2013). However, PAULA XML is an annotation format. Annotations from different annotation tools can be converted to PAULA XML annotation format. However, there is no specific annotation tool that would help annotate structured documents. In late February 2017, W3C released a Web annotation data model recommendation⁴ that uses XPath for selecting annotated data in the document. Hypothes.is is an open source tool that follows W3C recommendations for the data model. The tool is in form of a Chrome add-on and server side application that stores the annotations. Hypothes.is runs its instance of the server side tool, from which annotations are accessible through an API⁵. There is a space for the development of new tools, especially those that allow annotation of other XML formats (specialized formats such as PMC or DailyMed).

⁴<https://www.w3.org/TR/annotation-model/>

⁵<https://web.hypothes.is/>

8.6 Limitations

The work in this thesis was undertaken for analysing tables in an XML format. A significant amount of scientific literature is still made available in other formats, such as PDF (e.g. arXiv.org, certain journals and conference proceedings on IEEE Xplore, SpringerLink, etc.). Databases, such as PMC and DailyMed are growing and converting more papers into XML formats, however, a large amount of scientific literature is still not available in the XML format. While some concepts given in this thesis can probably be applied to the PDF format, certain steps of the methodology, designed and developed in this thesis, are not suitable for PDF or any other non-XML format.

Functional analysis is currently developed for the PMC and DailyMed XML formats. Although both in XML, they had differences in tags and attributes used for emphasizing functional areas. Extending functional analysis for other data sets may require implementation of heuristics or training machine learning models. Developing a generic approach to functional analysis that would not be domain or dataset dependent requires further research. Also, some work is required on improving the performance of the functional analysis. It is important to provide the best performing approach in the early stages of table analysis, such as functional analysis, because the errors from this stage may propagate in the higher analysis layers and negatively affect their performance.

Pragmatic classification using machine learning performs well for narrow, specific table classes. As pragmatic classes and information grouped in pragmatic classes widen, the performance is likely to be weaker. In our methodology, the designer of the task specifies the number and nature of classes. Therefore, the designer of the information extraction task is required to anticipate, at the time of design, what information is required for future extraction. In some datasets, it is possible to determine the pragmatic types based on the section the table is in or some other identifier. However, we consider this practice relatively rare for literature data sets. A machine learning is the best performing method for datasets that do not have labelled table pragmatics. The challenge for designing pragmatic classification is to identify relevant pragmatic classes.

Information extraction using TableInOut is designed for extracting variables and their values. It is possible to develop rules that can extract a number of patients, the age of patients, weight, height, BMI or race. However, extracting multiple values and their relationships requires additional steps, especially if one of the related entities is not presented in the table (e.g. drug-drug interaction). We have proposed a modification

of the method that uses the output of functional, structural analysis, semantic tagging and pragmatic analysis to extract drug-drug interactions from structured product labels. The method modification uses lexical cues for finding one variable in the table, but has to look outside the table and link the variables afterwards.

Precision, recall, and F1-score of functional, structural and pragmatic analysis was over 0.92. This is the state-of-the-art at the moment, however, it also means that almost 10% of cells are falsely classified to the areas they do not belong. F1-score of information extraction task was in the range of 0.82-0.94. This is often considered as good performance. However, 6%-18% of information were missed or are false positives. Some of the errors of information extraction are propagated from the previous steps, such as functional or pragmatic analysis. Also, relatively acceptable levels of errors in information extraction can be propagated further, to other systems that are using information extraction output. This propagation of errors from the lower stages of analysis to the higher is a common problem in text mining systems and may lead to unacceptable levels of errors in the final system.

8.7 Availability

The implementation of the table disentangling method, covering functional, structural and pragmatic analysis is available as TableDisengangler tool at <https://github.com/nikolamilosevic86/TableDisengangler>.

We have created a web application for exploring the data from disentangled tables from clinical trial papers in PMC. The page can be accessed on the following address: http://gnteam.cs.manchester.ac.uk/demos/table_explorer/.

Clinical trial data set retrieved from PMC on which TableDisentangler was run in order disentangle tables and make them browsable using the created web application is available on Mendeley Data at <https://data.mendeley.com/datasets/wk53twxddf/1>.

The implementation of the TableInOut wizard that follows table information extraction framework and allows creation of lexical, semantic and syntactic rules is available at <https://github.com/nikolamilosevic86/TabInOut>.

The code is published under GNU GPLv3 licence. The data set is published under Creative Commons BY 4.0 licence.

Chapter 9

Conclusion and Future perspectives

9.1 Summary of thesis contributions

This thesis proposed, developed and validated a framework for information extraction from tables in the biomedical literature. More precisely, the main contributions are:

- **A model of tables.** Types of tables and common table structures in the biomedical literature were investigated, including the means and patterns of presenting values. We proposed a model of tables according to table dimensionality, which includes list (one-dimensional tables), matrix (two-dimensional tables) and multi-dimensional tables with two sub-types, namely super-row tables (presenting additional dimensions using super-row structures) and multi-tables (consisting of multiple tables merged together). This model of tables assists the analysis of functional areas of the table and disentangling inter-cell relationships.
- **A methodology to disentangle table structure.** The methodology analyses tables based on the arrangement of cells, their spanning, content and content emphasis in order to discriminate functional areas of the table (navigational vs data areas). The methodology is mainly rule-based, however, in the case of a dataset with a lack of emphasis features, it can fall back to a machine learning-based methods that can be integrated with heuristics. Based on the functional areas and table type, the methodology finds which cells are related and in which manner. Disentangling the table structure helps to map a table to a format that can be queried based on its content, functional areas and relationships, and facilitate information extraction.

- **Data model for storing and querying disentangled table data appropriate for further processing.** We presented a data model that allows for querying data disentangled by the previously described methodology. The data in this format can help users find information and create complex queries. This data model can also assist with more advanced tasks such as information extraction, question answering and even navigating tables for visually impaired people.
- **A multilayer approach for information extraction from tables.** We presented a multi-layer, hybrid approach for information extraction from tables that uses table detection, a functional, structural, pragmatic analysis of tables, semantic tagging of table content, target cell selection and syntactic analysis of the cell content. The methodology uses a heuristic, rule-based and machine learning-based approaches to obtain the most efficient results.
- **A library of common data presentation patterns.** We presented a method and have developed a library of the most common, numerical data presentation patterns that map presented values to their components. The components are descriptive and add additional layer of the meaning to the extracted values. To distinguish the meaning of the value, the rule-based method utilizes value/variable description, its pattern and its word order in navigational table areas.
- **Application of the presented methodology to case studies.** We applied the presented model and methodology to two case studies. The aim of the first case study was to extract demographic and other baseline clinical trial characteristics, while the aim of the other one was to explore extraction of relationships by applying methodology to the task of extracting drug-drug interactions. The studies were performed in order to demonstrate the potential of the presented methodology and to identify remaining challenges.

9.2 Future work

A number of areas for future work have been identified in this work and each will be discussed in detail in this section.

1. **Generalization for other document formats.** The major limitation of this work is that the presented methodology only supports documents in an XML format. A large amount of scientific literature is published in PDF and other document

formats are also used frequently. Our methodology provides general heuristic guidelines for functional analysis. However, in order to implement this approach it is necessary to apply optical character (and object, including lines) recognition and other techniques to transform and disentangle tables from visual representation to the appropriate representation for computational handling. Once tables are transformed to the proposed data model and stored in a database, it is possible to apply tools developed in this thesis for information extraction.

2. **Generic extraction of relations.** The methodology presented in this thesis provides a framework for extracting single information with its descriptors from a table. As described in our DailyMed case study, our methodology lacks the capacity to extract relations. Primarily we focus on extraction of single values or entities - the extraction template and recipe lack the capacity to extract multiple concepts with a given relationship. As demonstrated by extending extraction methodology, it is possible to use the table disentangling data model for relation extraction. Drug-drug and protein-protein interaction research from the literature is an emerging field and a large amount of this information is stored in tables.
3. **Integrate the curation system with information extracted from text.** Information extraction from text is an active research area. To date, a number of approaches have been proposed for extracting values, concepts and relations. In this thesis, we propose an approach that is able to help extract value and concept information from tables. However, none of these approaches will extract all the presented information. About 30%-40% of information in clinical trial publications about the age of patients and gender distribution, is presented in tables. In many publications, authors present part of the information in tables and the remainder in text. In order to develop an efficient data curation engine, it is desirable to extract and curate complete information of interest from all document areas. This includes text, tables and figures. Complete information can be obtained only by integrating information from text and tables in publication.
4. **Evaluate effects of assisted curation.** In this work we have not tested the effects on speed and accuracy of machine assisted data curation for table mining. Assumption that it will significantly increase curation speed is based on the literature on assisted curation from text. It is left for the future to design user interface and examine the gains of assisted data curation from tables.

5. **Explore table representations for deep learning.** In the recent years, deep learning and deep neural networks archived successes in many areas, ranging from playing games to natural language processing (Schmidhuber 2015, LeCun et al. 2015). Several text vector representation models significantly improved performance of text classification, named entity recognition and information extraction in text (Mikolov et al. 2013, Pennington et al. 2014). These models are able to handle linguistic context of the words in text. However, they are not designed to handle visual structures and make predictions based on them. Vector representations that would involve both context and structure of the article element should be explored in the future. Also, performance of information extraction using recurrent neural networks in combination with the mentioned representation model should be further explored in the future.
6. **Examine other text mining tasks.** In the past, approaches in information retrieval (Hearst et al. 2007, Liu 2009), information extraction (Embley et al. 2005, Mulwad et al. 2013) and knowledge discovery (Wong et al. 2009, Xu & Wang 2015a) from tables were presented. Some text mining tasks, such as relation extraction, summarization, question answering, topic segmentation and recognition have not been examined. These research fields lack research activity, especially in the biomedical domain that is rich in tables that provide valuable information important for experiment reproduction, evidence synthesis and future research. This thesis provides a foundation with a table and data model for table analysis that can be utilized for higher layers of table analysis to solve said tasks. This is especially true for PMC data, for which we provided methods that perform the functional and structural analyses. We briefly mentioned topic recognition in our pragmatic analysis, which recognizes the main table topic. However, the extraction task designer assigned possible topics manually. An automatic and generic topic analysis mechanism that can automate pragmatic analysis remains a task for future development. Question answering relies on information extraction and information retrieval but also employs a number of specific normalization techniques. The specifics of table data and the influence of table structure on question answering also remain tasks for future development. Table summarization may help with large and complex table reading. Summarization would aid a reader in determining whether or not information he/she is looking for may be stored in the table as well as present the most important findings to a user without going into the table (e.g. statistically significant results). This task requires complex

semantic analysis of the table data, as well as analysis of the surrounding text.

7. **Table annotation format, software, and data.** At the moment there are only few proposed table annotation schemas, while annotated corpora of tables are rare. Annotated corpora would facilitate development of novel table mining methodologies. Development and standardisation of the currently proposed annotation schema, integration of these schemas into easy-to-use software for non-expert user and development of annotated corpora of tables in various domains and for various table mining tasks will be essential for the advancements in the area.

9.3 Final remarks

This thesis examined tables in biomedical literature, with a focus on clinical trial literature and drug labels. It was found that tables are frequent in the biomedical domain, however, it also depends on sub-domain: there are more than 3 tables per article in clinical literature and more than 10 per structured drug label. Vast information about patients, adverse events, procedures and results are stored in tables - out of reach for traditional text mining techniques. Although a number of table processing techniques have been proposed in the past, none provided a complete pipeline for information extraction for all table structures found in biomedical literature. The aim of this thesis is to provide a methodology that facilitates easy-to-use information extraction from all tables in a given domain.

Since tables are a structured document element, containing textual content in its cells, the analysis of tables requires several layers. First, table structure has to be disentangled. Table structure disentangling consists of two layers: functional and structural. In the functional layer, functional areas of the table are recognized while in the structural layer, table structure is analysed and relationships between cells are disentangled. Once the structure is disentangled, analysis of the content can follow. As language contains syntactic, lexical, pragmatic and semantic analysis layers, all these layers can be applied to table content as well. However, in the table, meaning can be spread to several, related cells and so this analysis has to be performed in relation to its structure.

This thesis provides a framework for table analysis and information extraction in particular. Our framework consists of the description of analysis layers, table model, data structures and the recipe for disentangling tables and information extraction. Our methodology for table disentangling is a foundation for table analysis. Functional and

structural analysis is required for any further table mining tasks, such as information extraction, information retrieval, question answering, summarisation, etc. Therefore, our method enables table mining tasks to be performed for the vast variety of table structures. The provided method is a recipe for extracting information from clinical literature with a 0.82-0.94 F1-score, which represents the state-of-the-art in table information extraction performance. The information extraction methodology is iterative, which means that some of the presented performance can be refined and improved.

Finally, tables often contain information that can facilitate future research or enable reproduction of research, so this information should not be overlooked. This thesis enables large-scale, semi-automated data curation that takes into account all available information from all document structures. Semi-automated curation will significantly increase the speed and accuracy of creating curated databases.

Glossary

ATC The Anatomical Therapeutic Chemical (ATC) Classification System is used for the classification of active ingredients of drugs according to the organ or system on which they act and their therapeutic, pharmacological and chemical properties. Drugs are classified in groups at five different levels. The drugs are divided into fourteen main groups (1st level), with pharmacological/therapeutic subgroups (2nd level). The 3rd and 4th levels are chemical/pharmacological/therapeutic subgroups and the 5th level is the chemical substance. 11, 137, 138

Caption describes the table content and subject. Overview of what table is about and what is presented in table.. 23

Cell is the basic grouping within a table. Cells usually contain only one value, word, phrase or concept and are divided by horizontal and vertical lines.. 23

Column is a set of vertically aligned table cells.. 35

DailyMed is a website operated by U.S. National Library of Medicine (NLM) that publishes up-to-date drug labels. The content published on DailyMed is collected from the pharmaceutical companies by U.S. Food and Drug Administration (FDA). The documents are published using HL7 version 3 Structured Product Labelling (SPL) standard in XML format.. 8, 26, 30, 62, 76, 81, 82, 88–90, 92, 132–135, 139, 140, 149, 154, 163, 167

Footer provides more detailed information about the table and is usually placed below the table. Footer often presents the legend for symbols used in the table or observations about the table data.. 35

Functional table analysis is a process of identifying functional areas within table, such as header, stub, super-row and data areas.. 37

Gene Ontology defines concepts/classes used to describe gene function, and relationships between these concepts. 204

Header is usually top-most row (or set of multiple top-most rows) of a table and defines the columns' data. 23

Information extraction is a task of automatically extracting structured information from unstructured or semi-structured machine readable documents. 103

Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). 46

LOINC (Logical Observation Identifiers Names and Codes) is an universal code system for reporting laboratory and other clinical observations. From 1999, it was identified by HL7 as a preferred code set for laboratory test names in transactions between health care facilities, laboratories and public health authorities. 132, 133, 135, 138

MEDLINE is the U.S. National Library of Medicine (NLM) premier bibliographic database that contains more than 27 million references to journal articles in life sciences with a concentration on biomedicine. 25, 30, 81

Navigational cells (access cells) describe and label data cells. Header, stub and super-row cells are referred together as navigational cells.. 23

Pragmatic table analysis is a process of analysing the purpose of the table in document. Also pragmatic analysis analyses what kind of information is presented in a given table.. 94

Row is a set of horizontally aligned table cells.. 35

Semantic analysis is a process of analysing the meaning of the table and its data.. 122

SNOMED-CT is a systematically organised computer processable clinical health terminology distributed around the world by SNOMED International. 204

Structural table analysis is a process of identifying relationships between cells in a table.. 37

Stub (row header) is typically the left-most column of the table, usually containing the list of subjects or instances to which the values in the table body apply.. 23

Super-row creates an additional dimension of the table and additionally, describes table data. The sub-header row is usually placed between data rows, separating them by some dimension or concept.. 23

Syntactic table analysis is a process of analysing the syntax of the cells' content.. 113

Table detection is a process of recognising or location table in the article.. 37

TF-IDF (Term Frequency, inverse document frequency) is a numerical statistical measure that is intended to indicate how important is some term in a corpus of documents. TF-IDF is usually used in information retrieval.. 43, 47, 51

Acronyms

ACQ Asthma Controlled Questionnaire. 125, 129, 153

API Application Programming Interface. 204

AQLQ Asthma Quality of Life Questionnaire. 125, 129, 153

ASCII American Standard Code for Information Interchange. 38–41, 43, 44, 51

BMI Body Mass Index. 116, 120, 163

CALS (Continuous Acquisition and Life-cycle Support. 35

COPD Chronic Obstructive Pulmonary Disease. 124–126, 129, 131

CRF Conditional Random Fields. 39, 44, 51, 85

CSS Cascading Style Sheet. 39, 43, 154

CUI Concept Unique Identifiers. 137

FDA Food and Drug Administration. 52, 132

FEV1 Forced Expiratory Volume in 1 second. 125, 129

FN False Negative. 85

FP False Positive. 85

GUI Graphical User Interface. 205

HMM Hidden Markov Model(s). 39, 42

HTML HyperText Markup Language. 27, 30, 34, 38, 42–44, 46, 51, 54, 62, 92

- ICD-10** International Classification of Diseases 10. 204
- MeSH** Medical Subject Headings. 204
- MMR** Mismatch Repair Database. 52
- OCR** Optical Character Recognition. 38, 39, 41, 47, 82
- PDDI** Potential Drug-Drug Interaction. 132, 133, 143, 144
- PDF** Portable Document Format. 27, 30, 38–42, 44, 47, 51, 54, 92, 163, 166
- PEF** Peak Expiratory Flow. 125, 129, 153
- PMC** PubMed Central. 26, 30, 62, 70, 76, 81, 82, 87–89, 92, 97, 106, 112, 122, 126, 139, 149, 150, 154, 163, 168
- RDF** Resource Description Framework. 43, 44, 46, 49, 204
- SGML** Standard Generalized Markup Language. 35
- SGRQ** St. George Respiratory Questionnaire. 125, 129, 130, 153
- SKOS** Simple Knowledge Organisation System. 63, 204
- SPL** Structured Product Label. 132–135, 138, 139, 143, 144
- SPLICER** Structured Product Label Information Coder and Extractor. 133
- SQL** Structured Query Language. 46, 198
- SVM** Support Vector Machines. 39, 42, 52, 97
- TP** True Positive. 85
- UMLS** Unified Medical Language System. 53, 63, 74, 111, 117, 137, 138, 156, 204, 209
- W3C** World Wide Web Consortium. 162, 204
- XML** eXtensible Markup Language. 27, 30, 34, 35, 39, 41, 42, 46, 51, 62, 70, 76, 77, 82, 84, 87, 88, 90, 92, 114, 130, 133, 134, 138, 149, 154–156, 161, 163, 166

Bibliography

- Abeel, T., Van Landeghem, S., Morante, R., Van Asch, V., Van de Peer, Y., Daelemans, W. & Saeys, Y. (2010), ‘Highlights of the biotm 2010 workshop on advances in bio text mining’, *BMC Bioinformatics* **11**(Suppl 5), I1.
- Agirre, E. & Edmonds, P. (2007), *Word sense disambiguation: Algorithms and applications*, Vol. 33, Springer Science & Business Media.
- Ahmed, F., Islam, M. A., Borodin, Y. & Ramakrishnan, I. (2010), Assistive web browsing with touch interfaces, in ‘Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility’, ACM, pp. 235–236.
- Alex, B., Grover, C., Haddow, B., Kabadjor, M., Klein, E., Matthews, M., Roebuck, S., Tobin, R. & Wang, X. (2008), Assisted curation: Does text mining really help?., in ‘Pacific Symposium on Biocomputing’, Vol. 13, pp. 556–567.
- Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. & Kochut, K. (2017), ‘A brief survey of text mining: Classification, clustering and extraction techniques’, *arXiv preprint arXiv:1707.02919* .
- Alley, M. (1996), *The craft of scientific writing*, Springer.
- Amano, A. & Asada, N. (2002), Complex table form analysis using graph grammar, in ‘International Workshop on Document Analysis Systems’, Springer, pp. 283–286.
- Amano, A. & Asada, N. (2003), Graph grammar based analysis system of complex table form document., in ‘ICDAR’, pp. 916–920.
- Ananiadou, S. & McNaught, J. (2006), *Text mining for biology and biomedicine*, Artech House London.

- Aronson, A. R. (2001), Effective mapping of biomedical text to the umls metathesaurus: the metamap program., *in* 'Proceedings of the AMIA Symposium', American Medical Informatics Association, p. 17.
- Aronson, A. R. & Lang, F.-M. (2010), 'An overview of metamap: historical perspective and recent advances', *Journal of the American Medical Informatics Association* **17**(3), 229–236.
- Ayvaz, S., Horn, J., Hassanzadeh, O., Zhu, Q., Stan, J., Tatonetti, N. P., Vilar, S., Brochhausen, M., Samwald, M., Rastegar-Mojarad, M. et al. (2015), 'Toward a complete dataset of drug–drug interaction information from publicly available sources', *Journal of biomedical informatics* **55**, 206–217.
- Banerjee, S. & Pedersen, T. (2002), An adapted lesk algorithm for word sense disambiguation using wordnet, *in* 'International Conference on Intelligent Text Processing and Computational Linguistics', Springer, pp. 136–145.
- Bechhofer, S. & Miles, A. (2009), 'Skos simple knowledge organization system reference', *W3C recommendation*, W3C .
- Belousov, M., Milosevic, N., Dixon, W. & Nenadic, G. (2017), Extracting adverse drug reactions and their context using sequence labelling ensembles in tac2017, *in* 'Text Analytics Conference', NIST.
- Bernstein, P. A., Madhavan, J. & Rahm, E. (2011), 'Generic schema matching, ten years later', *Proceedings of the VLDB Endowment* **4**(11), 695–701.
- Bienz, T., Cohn, R. & Adobe Systems (1993), *Portable document format reference manual*, Addison-Wesley Reading, MA, USA.
- Bingham, H. (1995), Cals table model document type definition, Technical report, Technical report, OASIS (Organization for the Advancement of Structured Information Standards).
- Bodenreider, O. (2004), 'The unified medical language system (umls): integrating biomedical terminology', *Nucleic acids research* **32**(suppl 1), D267–D270.
- Bourke, S. J. & Burns, G. P. (2015), *Lecture notes: respiratory medicine*, John Wiley & Sons.

- Boyce, R., Gardner, G. & Harkema, H. (n.d.), Using natural language processing to extract drug-drug interaction information from package inserts, *in* 'BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing', pp. 206–213.
- Brants, T. (2000), Inter-annotator agreement for a german newspaper corpus., *in* 'LREC'.
- Bushardt, R. L., Massey, E. B., Simpson, T. W., Ariail, J. C. & Simpson, K. N. (2008), 'Polypharmacy: misleading, but manageable', *Clinical interventions in aging* **3**(2), 383.
- Cafarella, M. J., Halevy, A., Wang, D. Z., Wu, E. & Zhang, Y. (2008), 'Webtables: exploring the power of tables on the web', *Proceedings of the VLDB Endowment* **1**(1), 538–549.
- Cafarella, M. J., Halevy, A. Y., Zhang, Y., Wang, D. Z. & Wu, E. (2008), Uncovering the relational web., *in* 'WebDB'.
- Caporaso, J. G., Baumgartner Jr, W. A., Randolph, D. A., Cohen, K. B. & Hunter, L. (2007), 'Mutationfinder: a high-performance system for extracting point mutation mentions from text', *Bioinformatics* **23**(14), 1862–1865.
- Cawley, G. C. & Talbot, N. L. (2003), 'Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers', *Pattern Recognition* **36**(11), 2585–2592.
- Cesarini, F., Gori, M., Marinai, S. & Soda, G. (1999), Structured document segmentation and representation by the modified xy tree, *in* 'Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on', IEEE, pp. 563–566.
- Chandran, S. & Kasturi, R. (1993), Structural recognition of tabulated data, *in* 'Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on', IEEE, pp. 516–519.
- Chavan, M. M. & 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', IEEE, pp. 272–277.

- Chen, H.-H., Tsai, S.-C. & Tsai, J.-H. (2000), Mining tables from large scale html texts, *in* 'Proceedings of the 18th conference on Computational linguistics-Volume 1', Association for Computational Linguistics, pp. 166–172.
- Choudhury, G. S. (2008), 'Case study in data curation at johns hopkins university', *Library Trends* **57**(2), 211–220.
- Cohen, A. M. & Hersh, W. R. (2005), 'A survey of current work in biomedical text mining', *Briefings in bioinformatics* **6**(1), 57–71.
- Constantin, A. (2014), Automatic Structure and Keyphrase Analysis of Scientific Publications, PhD thesis.
- Constantin, A., Pettifer, S. & Voronkov, A. (2013), Pdfx: fully-automated pdf-to-xml conversion of scientific literature, *in* 'Proceedings of the 2013 ACM symposium on Document engineering', ACM, pp. 177–180.
- Corrêa, A. S. & Zander, P.-O. (2017), Unleashing tabular content to open data: A survey on pdf table extraction methods and tools, *in* 'Proceedings of the 18th Annual International Conference on Digital Government Research', ACM, pp. 54–63.
- Crestan, E. & Pantel, P. (2010), Web-scale knowledge extraction from semi-structured tables, *in* 'Proceedings of the 19th international conference on World wide web', ACM, pp. 1081–1082.
- Dalvi, B. B., Cohen, W. W. & Callan, J. (2012), Websets: Extracting sets of entities from the web using unsupervised information extraction, *in* 'Proceedings of the fifth ACM international conference on Web search and data mining', ACM, pp. 243–252.
- Dilger, B. J. & Rice, J. (2010), *From A to A: Keywords of Markup*, U of Minnesota Press.
- Divoli, A., Wooldridge, M. A. & Hearst, M. A. (2010), 'Full text and figure display improves bioscience literature search', *PloS one* **5**(4), e9619.
- Donaldson, I., Martin, J., De Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G. D., Michalickova, K. et al. (2003), 'Prebind and textomy—mining the biomedical literature for protein-protein interactions using a support vector machine', *BMC bioinformatics* **4**(1), 11.

- Douglas, S., Hurst, M. & Quinn, D. (1995), Using natural language processing for identifying and interpreting tables in plain text, *in* 'Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval', pp. 535–546.
- Doush, I. A. & Pontelli, E. (2010), Non-visual navigation of spreadsheet tables, *in* 'Computers Helping People with Special Needs', Springer, pp. 108–115.
- Doush, I. A. & Pontelli, E. (2013), 'Non-visual navigation of spreadsheets', *Universal access in the information society* **12**(2), 143–159.
- Duke, J., Friedlin, J. & Li, X. (2013), 'Consistency in the safety labeling of bioequivalent medications', *Pharmacoepidemiology and drug safety* **22**(3), 294–301.
- Elmeleegy, H., Madhavan, J. & Halevy, A. (2014), 'Harvesting relational tables from lists on the web'. US Patent 8,732,116.
- Embley, D. W., Hurst, M., Lopresti, D. & Nagy, G. (2006), 'Table-processing paradigms: a research survey', *International Journal of Document Analysis and Recognition (IJDAR)* **8**(2-3), 66–86.
- Embley, D. W., Krishnamoorthy, M. S., Nagy, G. & Seth, S. (2016), 'Converting heterogeneous statistical tables on the web to searchable databases', *International Journal on Document Analysis and Recognition (IJDAR)* **19**(2), 119–138.
- Embley, D. W., Tao, C. & Liddle, S. W. (2005), 'Automating the extraction of data from html tables with unknown structure', *Data & Knowledge Engineering* **54**(1), 3–28.
- Fang, J., Gao, L., Bai, K., Qiu, R., Tao, X. & Tang, Z. (2011), A table detection method for multipage pdf documents via visual separators and tabular structures, *in* 'Document Analysis and Recognition (ICDAR), 2011 International Conference on', IEEE, pp. 779–783.
- Feldman, R. & Sanger, J. (2007), *The text mining handbook: advanced approaches in analyzing unstructured data*, Cambridge University Press.
- Fung, K. W., Jao, C. S. & Demner-Fushman, D. (2013), 'Extracting drug indication information from structured product labels using natural language processing', *Journal of the American Medical Informatics Association* **20**(3), 482–488.

- Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B. & Pollak, B. (2007), Towards domain-independent information extraction from web tables, *in* 'Proceedings of the 16th international conference on World Wide Web', ACM, pp. 71–80.
- Gobel, M., Hassan, T., Oro, E. & Orsi, G. (2013), Icdar 2013 table competition, *in* 'Document Analysis and Recognition (ICDAR), 2013 12th International Conference on', IEEE, pp. 1449–1453.
- Green, E. & Krishnamoorthy, M. (1995), Recognition of tables using table grammars, *in* 'Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval', pp. 261–278.
- Guthrie, B., Makubate, B., Hernandez-Santiago, V. & Dreischulte, T. (2015), 'The rising tide of polypharmacy and drug-drug interactions: population database analysis 1995–2010', *BMC medicine* **13**(1), 74.
- Haddow, B. & Matthews, M. (2007), The extraction of enriched protein-protein interactions from biomedical text, *in* 'Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing', Association for Computational Linguistics, pp. 145–152.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009), 'The weka data mining software: an update', *ACM SIGKDD explorations newsletter* **11**(1), 10–18.
- Haralick, R. M. (1994), Document image understanding: Geometric and logical layout, *in* 'Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on', IEEE, pp. 385–390.
- He, H. & Garcia, E. A. (2009), 'Learning from imbalanced data', *IEEE Transactions on knowledge and data engineering* **21**(9), 1263–1284.
- Hearst, M. A., Divoli, A., Guturu, H., Ksikes, A., Nakov, P., Wooldridge, M. A. & Ye, J. (2007), 'Biotext search engine: beyond abstract search', *Bioinformatics* **23**(16), 2196–2197.
- Hignette, G., Buche, P., Dibia-Barthélemy, J. & Haemmerlé, O. (2009), Fuzzy annotation of web data tables driven by a domain ontology, *in* 'European Semantic Web Conference', Springer, pp. 638–653.

- Hoad, T. F. (1993), *The concise Oxford dictionary of English etymology*, Oxford University Press Oxford.
- Hu, J., Kashi, R., Lopresti, D. & Wilfong, G. (2000a), A system for understanding and reformulating tables, in 'Proceedings of the Fourth IAPR International Workshop on Document Analysis Systems', pp. 361–372.
- Hu, J., Kashi, R. S., Lopresti, D. P. & Wilfong, G. (2000b), Table structure recognition and its evaluation, in 'Photonics West 2001-Electronic Imaging', International Society for Optics and Photonics, pp. 44–55.
- Hunter, L. & Cohen, K. B. (2006), 'Biomedical language processing: what's beyond pubmed?', *Molecular cell* **21**(5), 589–594.
- Hurst, M. (1999), Layout and language: Beyond simple text for information interaction-modelling the table, in 'Proceedings of the 2nd international conference on multimodal interfaces, Hong Kong'.
- Hurst, M. (2003), A constraint-based approach to table structure derivation, in 'Proceedings of the Seventh International Conference on Document Analysis and Recognition-Volume 2', IEEE Computer Society, p. 911.
- Hurst, M. & Douglas, S. (1997), Layout & language: Preliminary experiments in assigning logical structure to table cells, in 'Proceedings of the fifth conference on Applied natural language processing', Association for Computational Linguistics, pp. 217–220.
- Hurst, M. F. (2000), The interpretation of tables in texts, PhD thesis.
- Iyer, S. V., Harpaz, R., LePendou, P., Bauer-Mehren, A. & Shah, N. H. (2014), 'Mining clinical text for signals of adverse drug-drug interactions', *Journal of the American Medical Informatics Association* **21**(2), 353–362.
- Jannach, D., Shchekotykhin, K. & Friedrich, G. (2009), 'Automated ontology instantiation from tabular web sources—the allright system', *Web semantics: science, services and agents on the world wide web* **7**(3), 136–153.
- Jung, S.-W. & Kwon, H.-C. (2006), 'A scalable hybrid approach for extracting head components from web tables', *Knowledge and Data Engineering, IEEE Transactions on* **18**(2), 174–187.

- Jupp, S., Bechhofer, S. & Stevens, R. (2009), A flexible api and editor for skos, in 'European Semantic Web Conference', Springer, pp. 506–520.
- Kasar, T., Bhowmik, T. K. & Belaid, A. (2015), Table information extraction and structure recognition using query patterns, in 'Document Analysis and Recognition (ICDAR), 2015 13th International Conference on', IEEE, pp. 1086–1090.
- Khare, R., Li, J. & Lu, Z. (2014), 'Labeledin: cataloging labeled indications for human drugs', *Journal of biomedical informatics* **52**, 448–456.
- Kieninger, T. G. (1998), Table structure recognition based on robust block segmentation, in 'Photonics West'98 Electronic Imaging', International Society for Optics and Photonics, pp. 22–32.
- Kieninger, T. G. & 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'.
- Kilicoglu, H. (2017), 'Biomedical text mining for research rigor and integrity: tasks, challenges, directions', *Briefings in Bioinformatics* .
- Kim, J.-D. & Wang, Y. (2012), Pubannotation: a persistent and sharable corpus and annotation repository, in 'Proceedings of the 2012 Workshop on Biomedical Natural Language Processing', Association for Computational Linguistics, pp. 202–205.
- Kohavi, R. et al. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection, in 'Ijcai', Vol. 14, Montreal, Canada, pp. 1137–1145.
- Konstantinou, N. & Spanos, D.-E. (2015), *Creating Linked Data from Relational Databases*, Springer International Publishing, pp. 73–102.
- Kuhn, M., Letunic, I., Jensen, L. J. & Bork, P. (2015), 'The sider database of drugs and side effects', *Nucleic acids research* p. gkv1075.
- Lacasta, J., Nogueras-Iso, J., López-Pellicer, F. J., Muro-Medrano, P. R. & Zarazaga-Soria, F. J. (2007), 'Thmanager: An open source tool for creating and visualizing skos', *Information Technology and Libraries* **26**(3), 39.
- Lalnunpuii, C. (2013), Extraction of clinical trial arm information from the literature, PhD thesis, University of Manchester, UK.

- Lazarou, J., Pomeranz, B. H. & Corey, P. N. (1998), 'Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies', *Jama* **279**(15), 1200–1205.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015), 'Deep learning', *Nature* **521**(7553), 436–444.
- Leech, G. N. (2016), *Principles of pragmatics*, Routledge.
- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S. et al. (2015), 'Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia', *Semantic Web* **6**(2), 167–195.
- Lesk, M. (1986), Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in 'Proceedings of the 5th annual international conference on Systems documentation', ACM, pp. 24–26.
- Li, D., Azoulay, P. & Sampat, B. N. (2017), 'The applied value of public investments in biomedical research', *Science* **356**(6333), 78–81.
- Limaye, G., Sarawagi, S. & Chakrabarti, S. (2010), 'Annotating and searching web tables using entities, types and relationships', *Proceedings of the VLDB Endowment* **3**(1-2), 1338–1347.
- Liu, S. (2005), 'What is pragmatics', *Eprint* (<http://www.gxnu.edu.cn/Personal/szliu/definition.html>) .
- Liu, Y. (2009), Tableseer: Automatic Table Extraction, Search, and Understanding, PhD thesis, The Pennsylvania State University.
- Long, V. (2010), An agent-based approach to table recognition and interpretation, PhD thesis, Macquarie University Sydney, Australia.
- Magro, L., Moretti, U. & Leone, R. (2012), 'Epidemiology and characteristics of adverse drug reactions caused by drug–drug interactions', *Expert opinion on drug safety* **11**(1), 83–94.
- Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv:1301.3781* .

- Miles, A. & Bechhofer, S. (2009), 'Skos simple knowledge organization system reference'.
- Miles, A., Matthews, B., Wilson, M. & Brickley, D. (2005), Skos core: simple knowledge organisation for the web, *in* 'International Conference on Dublin Core and Metadata Applications', pp. 3–10.
- Miller, G. A. (1995), 'Wordnet: a lexical database for english', *Communications of the ACM* **38**(11), 39–41.
- Moore, J. H. & Holmes, J. H. (2016), 'The golden era of biomedical informatics has begun'.
- Mulwad, V., Finin, T. & Joshi, A. (2013), Semantic message passing for generating linked data from tables, *in* 'International Semantic Web Conference', Springer, pp. 363–378.
- Mulwad, V., Finin, T., Syed, Z. & Joshi, A. (2010), 'Using linked data to interpret tables.', *COLD* **665**.
- Nagy, G. (2000), 'Twenty years of document image analysis in pami', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1), 38–62.
- Nagy, G. & Seth, S. (1984), Hierarchical representation of optically scanned documents, *in* 'Proceedings of International Conference on Pattern Recognition', Vol. 1, pp. 347–349.
- Nédellec, C. & Nazarenko, A. (2005), Ontology and information extraction: a necessary symbiosis, *in* 'Ontology Learning from Text: Methods, Evaluation and Applications', IOS Press.
- Ng, H. T., Lim, C. Y. & 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', Association for Computational Linguistics, pp. 443–450.
- Nurminen, A. (2013), 'Algorithmic extraction of data in tables in pdf documents'.
- Ofoghi, B., López-Campos, G., Martín-Sánchez, F. J. & Verspoor, K. (2014), Mapping biomedical vocabularies: a semi-automated term matching approach., *in* 'ICIMTH', pp. 16–19.

- Pande, A. (2002), Table understanding for information retrieval, Master's thesis, Virginia Polytechnic Institute and State University.
- Pennington, J., Socher, R. & Manning, C. (2014), Glove: Global vectors for word representation, *in* 'Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)', pp. 1532–1543.
- Percha, B., Garten, Y. & Altman, R. B. (2012), Discovery and explanation of drug-drug interactions via text mining, *in* 'Pacific symposium on biocomputing. Pacific symposium on biocomputing', NIH Public Access, p. 410.
- Porter, M. F. (1980), 'An algorithm for suffix stripping', *Program* **14**(3), 130–137.
- Pyreddi, P. & Croft, W. B. (1997), A system for retrieval in text tables, Technical report, Technical report 105, University of Massachusetts, Massachusetts, USA.
- Quercini, G. & Reynaud, C. (2013), Entity discovery and annotation in tables, *in* 'Proceedings of the 16th International Conference on Extending Database Technology', ACM, pp. 693–704.
- Ramel, J.-Y., Crucianu, M., Vincent, N. & Faure, C. (2003), Detection, extraction and representation of tables, *in* 'Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on', IEEE, pp. 374–378.
- Rastan, R., Paik, H.-Y. & Shepherd, J. (2015), Texus: a task-based approach for table extraction and understanding, *in* 'Proceedings of the 2015 ACM Symposium on Document Engineering', ACM, pp. 25–34.
- Reeve, L. & Han, H. (2005), Survey of semantic annotation platforms, *in* 'Proceedings of the 2005 ACM symposium on Applied computing', ACM, pp. 1634–1638.
- Rus, D. & Summers, K. (1994), Using white space for automated document structuring, Technical report, Cornell University.
- Schmidhuber, J. (2015), 'Deep learning in neural networks: An overview', *Neural networks* **61**, 85–117.
- Severson, E. & Bingham, H. (1995), 'Table interoperability: Issues for the cals table model', *OASIS Technical Research Paper 9501:1995*.

- Shigarov, A. O. (2015), 'Table understanding using a rule engine', *Expert Systems with Applications* **42**(2), 929–937.
- Shiloach, M., Frencher, S. K., Steeger, J. E., Rowell, K. S., Bartzokis, K., Tomeh, M. G., Richards, K. E., Ko, C. Y. & Hall, B. L. (2010), 'Toward robust information: data quality and inter-rater reliability in the american college of surgeons national surgical quality improvement program', *Journal of the American College of Surgeons* **210**(1), 6–16.
- Shmanina, T., Zukerman, I., Cheam, A. L., Bochynek, T. & Cavedon, L. (2016), 'A corpus of tables in full-text biomedical research publications', *BioTxtM 2016* p. 70.
- Silva, A. (2010), Parts that add up to a whole: a framework for the analysis of tables, PhD thesis, University of Edinburgh.
- Silva, A. C., Jorge, A. & Torgo, L. (2003), Automatic selection of table areas in documents for information extraction, in 'Progress in Artificial Intelligence', Springer, pp. 460–465.
- Sim, I., Tu, S. W., Carini, S., Lehmann, H. P., Pollock, B. H., Peleg, M. & Wittkowski, K. M. (2014), 'The ontology of clinical research (ocre): an informatics foundation for the science of clinical research', *Journal of biomedical informatics* **52**, 78–91.
- Smith, D. E. & Ginsburg, J. (1937), 'Numbers and numerals.'
- Son, J.-W., Lee, J.-A., Park, S.-B., Song, H.-J., Lee, S.-J. & 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', Vol. 1, IEEE, pp. 368–371.
- Stoffel, A., Spretke, D., Kinnemann, H. & Keim, D. A. (2010), Enhancing document structure analysis using visual analytics, in 'Proceedings of the 2010 ACM Symposium on Applied Computing', ACM, pp. 8–12.
- Su, R. (2016), A web-based dashboard for visualisation and curation of data extracted from tables, PhD thesis, University of Manchester, UK.
- Sun, H., Ma, H., He, X., Yih, W.-t., Su, Y. & Yan, X. (2016), Table cell search for question answering, in 'Proceedings of the 25th International Conference on World Wide Web', International World Wide Web Conferences Steering Committee, pp. 771–782.

- Swain, M. C. & Cole, J. M. (2016), 'Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature', *Journal of Chemical Information and Modeling*.
- Tanaka, M. & Ishida, T. (2006), Ontology extraction from tables on the web, in 'Applications and the Internet, 2006. SAINT 2006. International Symposium on', IEEE, pp. 7–pp.
- Tang, Y. (2016), A model and web-based dashboard for querying data extracted from tables in clinical literature, PhD thesis, University of Manchester, UK.
- Tengli, A., Yang, Y. & Ma, N. L. (2004), Learning table extraction from examples, in 'Proceedings of the 20th international conference on Computational Linguistics', Association for Computational Linguistics, p. 987.
- Thompson, M. (1996), A tables manifesto, in 'Proceedings of SGML Europe', pp. 151–153.
- Tijerino, Y. A., Embley, D., Lonsdale, D. W. & Nagy, G. (2003), Ontology generation from tables, in 'Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on', IEEE, pp. 242–249.
- US Food and Drug Administration (2014), 'Cfr-code of federal regulations title 21', *Current good manufacturing practice for finished pharmaceuticals Part 211*.
- Van Assem, M., Rijgersberg, H., Wigham, M. & Top, J. (2010), Converting and annotating quantitative data tables, in 'The Semantic Web–ISWC 2010', Springer, pp. 16–31.
- Vasilescu, F., Langlais, P. & Lapalme, G. (2004), Evaluating variants of the lesk approach for disambiguating words., in 'Lrec'.
- Vos, T., Allen, C., Arora, M., Barber, R. M., Bhutta, Z. A., Brown, A., Carter, A., Casey, D. C., Charlson, F. J., Chen, A. Z. et al. (2016), 'Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990-2015: a systematic analysis for the global burden of disease study 2015', *The Lancet* **388**(10053), 1545–1602.
- Walsh, N. (1999), 'Xml exchange table model document', *OASIS Technical Memorandum TR 9901:1999*.

- Wang, H., Naghavi, M., Allen, C., Barber, R. M., Bhutta, Z. A., Carter, A., Casey, D. C., Charlson, F. J., Chen, A. Z., Coates, M. M. et al. (2016), 'Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the global burden of disease study 2015', *The Lancet* **388**(10053), 1459–1544.
- Wang, X. F. (2013), 'Research on information extraction based on web table structure and ontology', *Applied Mechanics and Materials* **321**, 2254–2259.
- Wang, X. & Wood, D. (1993), Tabular abstraction for tabular editing and formatting, in 'Proceedings of 3rd International Conference for Young Computer Scientists', pp. 17–29.
- Wang, X. & Wood, D. (1995), Tabular abstraction, editing, and formatting, PhD thesis.
- Wang, Y. & Hu, J. (2002), A machine learning based approach for table detection on the web, in 'Proceedings of the 11th international conference on World Wide Web', ACM, pp. 242–250.
- Warner, J. L., Anick, P. & Drews, R. E. (2013), 'Physician inter-annotator agreement in the quality oncology practice initiative manual abstraction task', *Journal of Oncology Practice* **9**(3), e96–e102.
- Wei, X., Croft, B. & McCallum, A. (2006), 'Table extraction for answer retrieval', *Information retrieval* **9**(5), 589–611.
- Welte, T. & Groneberg, D. A. (2006), 'Asthma and copd', *Experimental and Toxicologic Pathology* **57**, 35–40.
- Whittaker, H. (2013), 'Social and symbolic aspects of minoan writing', *European journal of archaeology*.
- Wong, W., Martinez, D. & 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', Association for Computational Linguistics, pp. 46–54.
- Wright, P. (1968), 'Using tabulated information', *Ergonomics* **11**(4), 331–343.
- Wright, P. (1977), 'Decision making as a factor in the ease of using numerical tables', *Ergonomics* **20**(1), 91–96.

- Wright, P. & Fox, K. (1970), 'Presenting information in tables', *Applied Ergonomics* **1**(4), 234–242.
- Xu, H., Stenner, S. P., Doan, S., Johnson, K. B., Waitman, L. R. & Denny, J. C. (2010), 'Medex: a medication information extraction system for clinical narratives', *Journal of the American Medical Informatics Association* **17**(1), 19–24.
- Xu, R. & Wang, Q. (2015a), 'Combining automatic table classification and relationship extraction in extracting anticancer drug–side effect pairs from full-text articles', *Journal of biomedical informatics* **53**, 128–135.
- Xu, R. & Wang, Q. (2015b), 'Large-scale automatic extraction of side effects associated with targeted anticancer drugs from full-text oncological articles', *Journal of biomedical informatics* **55**, 64–72.
- Yakel, E. (2007), 'Digital curation', *OCLC Systems & Services: International digital library perspectives* **23**(4), 335–340.
- Yesilada, Y., Stevens, R., Goble, C. & Hussein, S. (2004), Rendering tables in audio: the interaction of structure and reading styles, in 'ACM SIGACCESS Accessibility and Computing', number 77-78, ACM, pp. 16–23.
- Yildiz, B., Kaiser, K. & Miksch, S. (2005), pdf2table: A method to extract table information from pdf files., in 'IICAI', pp. 1773–1785.
- Zanibbi, R., Blostein, D. & Cordy, J. R. (2004), 'A survey of table recognition', *Document Analysis and Recognition* **7**(1), 1–16.
- Zeldes, A., Zipser, F. & Neumann, A. (2013), 'Paula xml documentation'.

Appendix A

Case study on extraction of BMI, weight and number of patients

A.1 INTRODUCTION

The amount of published scientific research is accelerating: the number of published papers is growing at a double-exponential pace (Hunter & Cohen 2006). MEDLINE contains over 25 million references¹ 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 the 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.

¹<http://www.ncbi.nlm.nih.gov/pubmed>

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.

A.2 BACKGROUND

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 ASCII documents using Optical Character Recognition (Kieninger & 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 & 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 is 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.

A.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 tables 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 from 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 a 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 a structured manner and extract information.

A.3.1 Table Decomposition

Table decomposition contains five steps.

In the first step, the algorithm is locating a 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 a different type (e.g. text) from the following rows (e.g. numeric), the first cell is labelled as part of the header. If all five cells have values of the same type, the table has no header. Types of cells could be empty, numeric (integer or floating

²<http://www.ncbi.nlm.nih.gov/pmc/>

point number), partially numeric (number with special characters and punctuations) and string.

In the third step, spanning cells (recognised by the 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 A.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 A.1). Super-rows are usually presented as a row inside the data part of the table that is spanning through all columns or a row with a value only in one cell.

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 A.1: Example of the table (PMC 29053) and the decomposition XML output for one cell from that table

In the last step, our method is iterating through all data cells and trying to find the correct navigation 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 the 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 the 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 A.1).

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

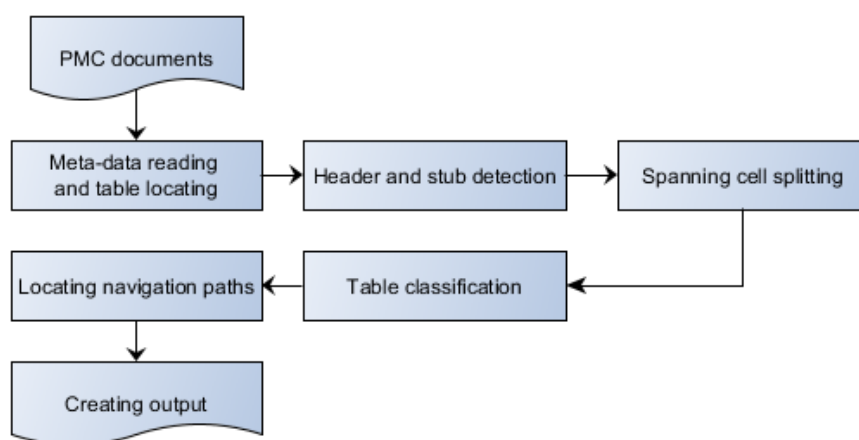


Figure A.2: Workflow of table decomposition method

A.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, the table shown in Figure A.3 has two participant groups. Extracted information will be: [Absolute Risk (n = 232): BMI: 27.4 (4.5)] and [NNT (n = 225): BMI: 27.0 (4.3)].

A.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 a 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 A.3).

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 $\geq 10\%$	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 A.3: Example of a clinical trial demographic table that contains information about patients BMI (PMC 58836)

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.

A.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 than 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.

A.4 RESULTS

A.4.1 Table Decomposition Results

We have processed all 2,517 PMC clinical trial documents. Our method extracted data from 3,573 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.

In Table A.1, we present the results of our evaluation.

Class	Tables in dataset	N. Eval.	Accuracy
Matrix tables	1,974 (55.24%)	100	89%
Super-row tables	1,517 (42.46%)	100	81%
List tables	27 (0.76%)	27	77.7%
Multi-table tables	55 (1.54%)	55	49.1%
Total	3,573	282	84.9%

Table A.1: Accuracy of table decomposition system

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%.

A.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 A.2.

Precision	73.53%
Recall	96.15%
F-measure	83.3%

Table A.2: Performance of extracting total number of patients

A.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 A.3.

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%

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

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.

A.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 (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. (2007) created generic information extraction system, but they reported F measure of 52%. 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 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.

Appendix B

Database schema for functional and structural table disentangling

Detected functions and relationships of cells can be stored in a MySQL database according to our model. The database schema is presented in Figure B.1

The database schema is a representation of the data model containing three layers – article, table, and cell. Table Article contains basic information about an article, such as title, PMCID, journal where the article was published, etc. Since an article usually has multiple authors, we store authors, their affiliations, and emails in separate tables. ArtTable (short for Article Table) stores general information about the table, such as table caption, the order in the article and footer. Here, we also store some table level annotations, such as the structural type of the table or pragmatic class of the table. A database table called Cell contains information about cells. For each cell, it contains its content, positions (row and column number), references to related navigational cells (headers, stubs, and super-rows). For easier processing, we created attributes in this table that contain content of all stubs, headers, and super-rows that are related to the particular cell. Since a cell can have multiple functions (e.g. header+stub, stub+super-row, data+super-row), functions are stored in a separate table. Annotations are also stored in the separate table with reference to the cell that they are annotating. This table stores annotation concept id, annotations description, the span of the annotation inside the cell and some provenance information about annotation source and the system that was used for annotation. Tables stored in this manner can be easily queried using SQL.

In addition to storing disentangled tables and their metadata in the database, we created a web application that can be used for exploring processed tables. Using our web application user can explore tables and cells. For tables, user can select tables

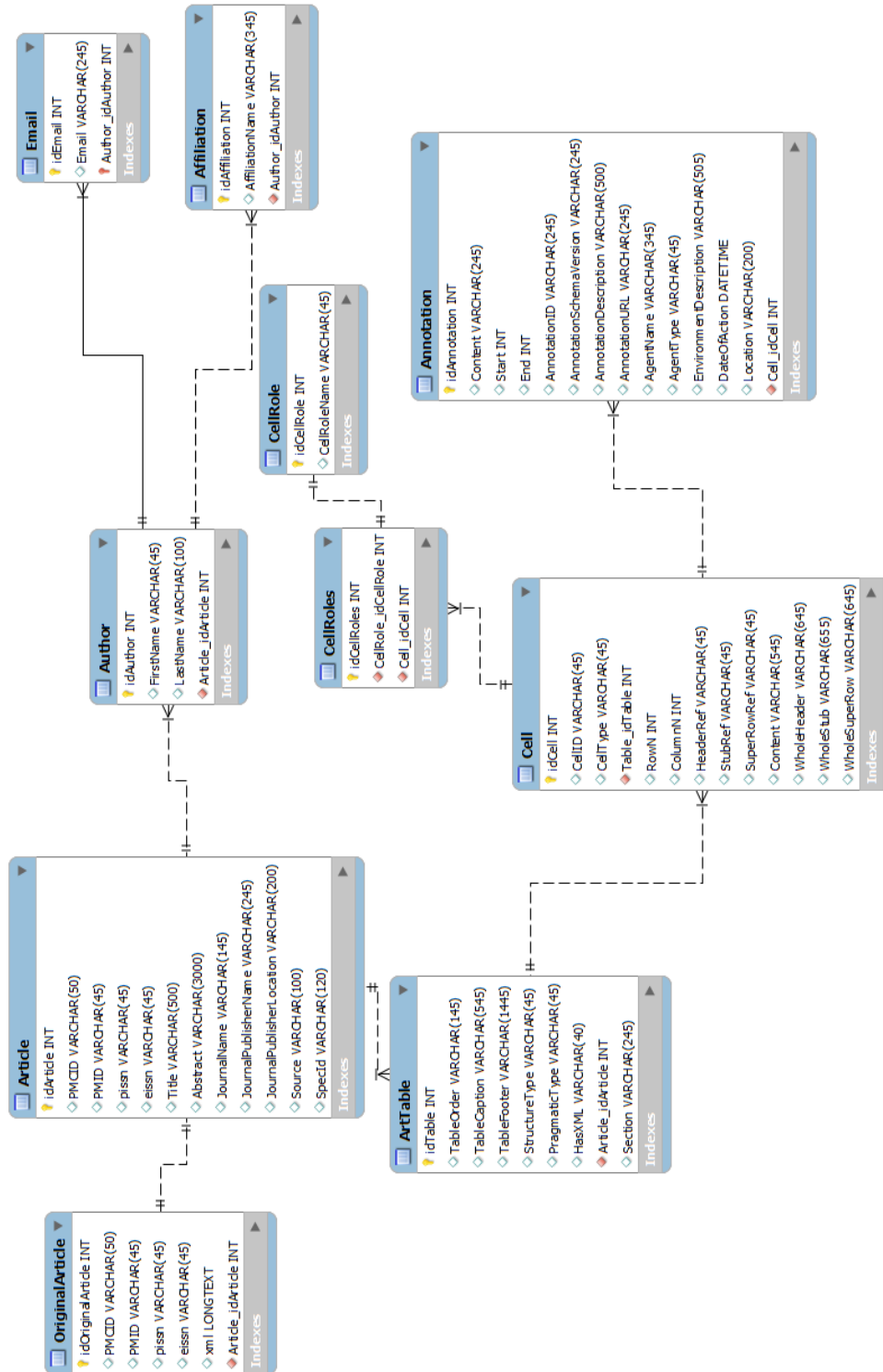


Figure B.1: Database schema used for string information about tables, cells, cell functions and inter-cell relationships with annotations

that contain certain string in their caption, footer, header, stub, super-row or data cells. When exploring cells, application is able to find cells having given string in related navigational areas (related header, stub or super-row cells). The web application is available at http://gnteam.cs.manchester.ac.uk/demos/table_explorer.

The same data model that was implemented as described database schema was also published as **Table Mining Ontology** available at <https://github.com/nikolamilosevic86/TableDisentangler/blob/master/TableMiningOntology.owl>.

Appendix C

TableInOut: Implementation details

C.1 Introduction

In this part, we describe TableInOut (Table Information Out) software. TableInOut is a tool for crafting rules for information extraction from the tables that were preprocessed using table disentangling tool, which we call TableDisentangler. The TableInOut tool follows the methodology described in Section 6.3.2. The tool is designed in a way that it allows the user to specify the extraction task by specifying variable, pragmatic table class, units of measure, lexical, semantic and syntactic cues. Once defined, it extracts information according to the defined extraction rules, specifications and stores extracted information in a relational database table containing columns as our defined template from Section 3.3.4.

In the following sections, we give an overview of the architecture of TableInOut software, as well as its relationship with other tools used in this project. The chapter that follows, explains in more details implementation of the system. The final section in this chapter discusses and evaluates several case studies of information extraction performed with TableInOut software.

C.2 TableInOut architecture overview

The work-flow of information extraction using TableInOut is presented in Figure C.1.

The information extraction pipeline consists of two tools that were developed during this project: TableDisentangler and TableInOut. TableDisentangler firstly disentangles the structure of the table, annotates functional areas, relationships between cells and the content of the cells using semantic resources. The output of TableDisentangler

is stored in the relational database as it was presented in Appendix B. Data is enriched and normalised using the annotation tool that we developed called Marvin. Once the data is in the database, a user of the TableInOut software can define the extraction task and extraction rules. The rule building process is usually an iterative process. Once the task specification and rule set are developed, TableInOut extracts the information from the table and stores them in a database table.

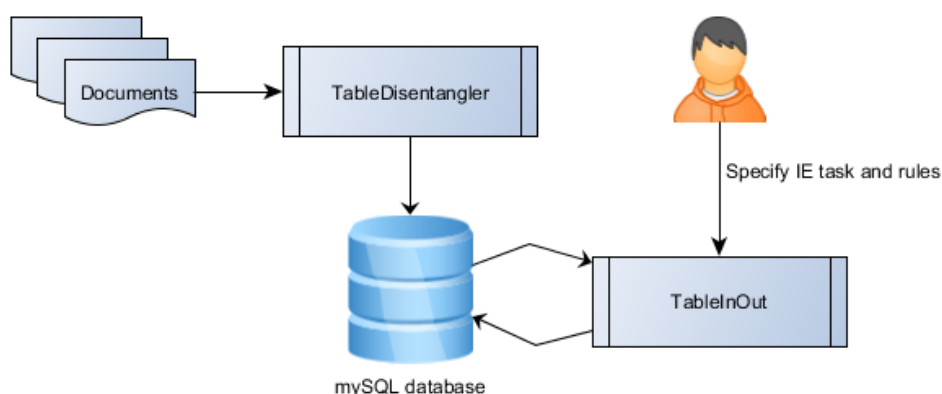


Figure C.1: Workflow of information extraction from tables using TableInOut

C.3 Marvin annotation tool

Marvin can annotate text using four knowledge sources (UMLS, WordNet, DBpedia, SKOS vocabulary), or any combination of them, which can be configured in its configuration file. Marvin firstly tokenizes the text. Tokenization is performed using OpenNLP (Baldrige, 2005) and the trained maximum entropy model provided by OpenNLP.

After the tokenization, annotation over the tokens is performed. However, for each knowledge source, the annotation is performed in a slightly different way.

Annotating using DBpedia. When annotating using DBpedia, our approach is to generate unigrams, bigram, and trigrams from the supplied text. The rationale is that there are a number of definitions on Wikipedia and DBpedia for concepts that are one, two or three words long. After unigram, bigrams and trigrams are generated; we capitalize the first letter since labels of DBpedia items are always with the first capital letter. Also, our approach puts the rest of the text in lowercase. We query DBpedia for

the generated strings. Querying is performed over the SPARQL interface. For testing, we used the public DBpedia interface (<http://dbpedia.org/sparql>). However, this interface has a certain restriction on the number of queries that can be submitted. For larger texts, it is advisable to have a local instance of DBpedia and its SPARQL interface.

Annotating using WordNet. While doing annotation using WordNet, Marvin is performing also part-of-speech tagging over the inputted text. This is done using OpenNLP part-of-speech tagger based on maximum entropy model for English downloaded from OpenNLP website. Part-of-speech tagging and tokenization are done in that way that for each token, there is also a part-of-speech tag. Using tokens and part-of-speech tags WordNet database is queried. The query returns all the possible senses of the word with a given part-of-speech.

Results from the query of WordNet contain senses that are not what text is about. Only one sense of the word is the actual sense in that context. With too many annotations for the senses, the annotations are not too useful. In order to retrieve only the right sense or a small number of the most probable senses, we applied word sense disambiguation.

In order to perform word sense disambiguation, we modified basic version of Lesk's algorithm (Lesk 1986). The basic idea of Lesk's algorithm is to count the number of words in the surroundings of the analyzed word and the words that appear in the dictionary definition of that term. The idea is very simple and there have been, over the years, attempts to improve the algorithm (Banerjee & Pedersen 2002, Vasilescu et al. 2004, Agirre & Edmonds 2007). The issue with the algorithm is that for different words, the size of definition can be different. Also, the size of the context window can be different. The ranking should not be the same if the number of matching terms are the same for two definitions, but one definition has more words than the other. Cases like this have to be weighted properly. In order to calculate weights for choosing the right definition, we took 15 words left and right of the current word in the text, if they exist, as a context. The algorithm is calculating for each definition how many words from the definition are appearing in the context of the annotated word. The sum of words appearing in both the context and the definition is divided by the number of the words in the definition. The definition with the largest result is chosen as the meaning of the word. If multiple definitions have the same result, they are all presented as possible definitions of the word.

Annotating using UMLS. Marvin is capable of annotating text using UMLS (Bodenreider 2004) with the aid of MetaMap (Aronson 2001). Marvin can send requests for annotations to the MetaMap server in case it has the location of the MetaMap server. Annotations with UMLS concepts are completely handled by MetaMap and Marvin only enriches these annotations with prevalence information and indexes of the word. Metamap and UMLS provide annotations for wide variety of concepts and semantic types in biomedical domains because UMLS encapsulates almost 200 biomedical controlled vocabularies and classification systems, including ICD-10, MeSH, SNOMED-CT and Gene Ontology¹. Since we were working with biomedical data, UMLS annotations proved to be the most useful.

Annotating using SKOS vocabularies. Previously we described annotation with WordNet, DBPedia, and MetaMap. These methods are using certain well-established vocabularies and they cannot be changed (apart from vendor's updates). However, when performing tasks such as information extraction, sometimes it is necessary to use custom made dictionaries. We have provided a method for users to supply a number of custom vocabularies, which our system will load and use to annotate text. For the vocabulary input format, we decided to use Simple Knowledge Organization System (SKOS) format.

Simple Knowledge Organization System is a RDF vocabulary for expressing the basic structure and content of concept schemes, such as thesauri, classification schemes, taxonomies, terminologies, glossaries and other types of controlled vocabularies (Miles et al. 2005). It is designed and recommended by World Wide Web Consortium as a standard for representing controlled vocabularies (Miles & Bechhofer 2009). As a W3C standard for representing vocabularies in RDF format, we expect that the format is well developed and adopted in the community. For the reading of SKOS vocabulary files we used SKOS API that has been designed to work with SKOS models at a high level of abstraction (Jupp et al. 2009). We have tested the reading of SKOS files created as export from ThManager 2.0, an open source tool for creating and visualizing SKOS (Lacasta et al. 2007). The text which needs to be annotated is first transformed to lowercase and broken into the words using tokenizer. For each word, Marvin searches the hash map that maps words into concepts. If found, it annotates that part of the text with the associated concept. If the concept contains some broader concept, Marvin will look up for that concept as well. Annotation with broader concepts is continued

¹<https://www.nlm.nih.gov/research/umls/sourcereleasedocs/index.html>

until the top level is reached. Since annotations are kept separately, it is possible to annotate the same word in Marvin with multiple annotations.

C.4 TableInOut implementation details

TableInOut is developed in Python. The user can interact with the tool through Graphical User Interface (GUI). GUI is designed in a form of wizard, containing in total seven views. Each project, when defined, is described in a set of files in a defined file structure. Folder structure of TableInOut project is presented in Figure C.2.

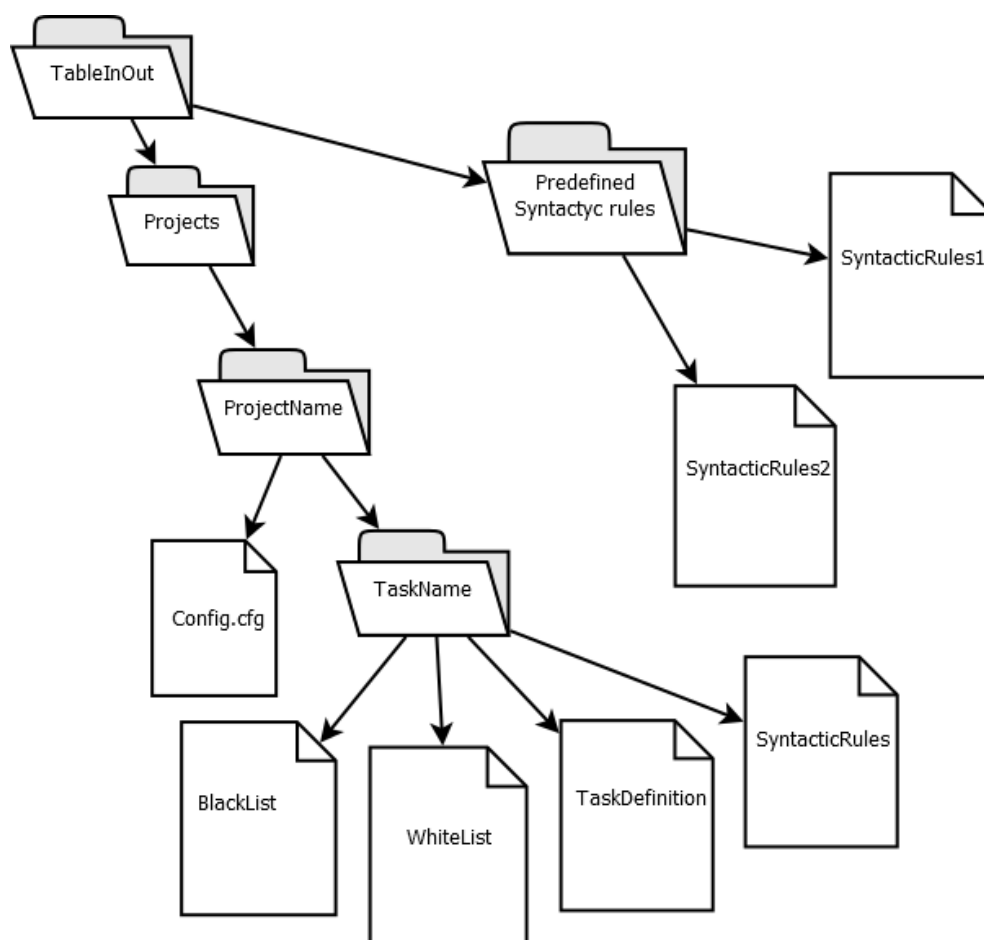


Figure C.2: Folder structure of a project in TableInOut

TableInOut root folder contains a folder named "Projects", where information about table information extraction projects are stored, and a folder where predefined syntactic rules are stored. From the set of predefined syntactic rules, the user can choose one, reuse it or modify it. Project folder can contain multiple projects. Each project

contains configuration file, containing information on the database to in which the TableDisentangler preprocessed data is stored. In the context of TableInOut, the task is an information extraction process with defined task specifications from Section 6.3.2 (information class name, pragmatic type of the table in which the information should be searched for, lexical, syntactic and semantic rules for extracting the information). One project can have multiple defined tasks that will execute sequentially, as ordered in the project. Tasks can be used to model cases of extracting the same variable or to extract multiple variables in one project. Task folder contains a definition of the task, white list, black list and syntactic rules specific to the task. This file and folder structure is read, edited or created during user's interaction with the wizard. In the execution phase, information from the file system is read, stored in the data structures in memory and according to the loaded data structures, information extraction process is executed.

Further we describe TableInOut wizard screens:

Project management screen

In project management screen a user can manage information extraction projects. The user can create new projects, delete or load existing projects. Example can be seen in Figure C.3.

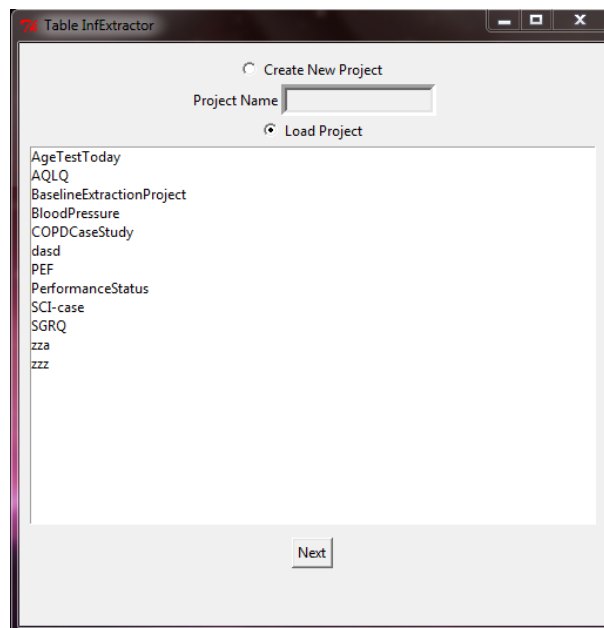


Figure C.3: TableInOut project management screen

Task management screen

Task management screen presents the list of tasks for the selected or created project. In this view, a task can be created, edited, deleted and reordered (task are executed in the order of presentation). From this screen database management screen can be accessed. Adding a task opens task definition screen, while task editing will lead to the same screen pre-populated with already defined values. From this screen, the user can also start the execution of a task (execution screen). Example of the screen can be seen in Figure C.4.

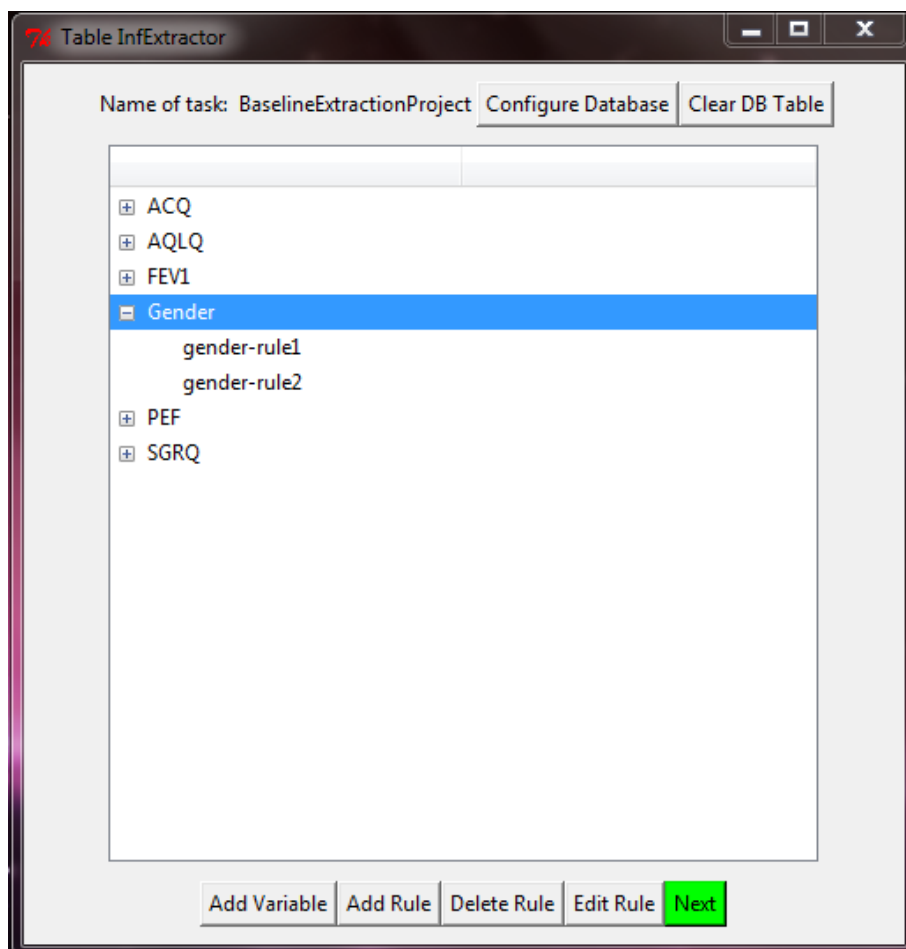


Figure C.4: Example of task management screen containing 6 variables and rules for each variable. This set-up was used for case study described in Section 7.1.

Database management screen

The purpose of this screen is to provide information about the database, including host address of database server, port number, database name, username, and password.

Example of this screen can be seen in Figure C.5.

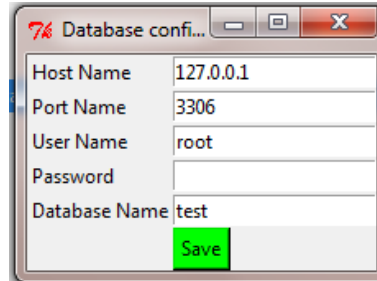


Figure C.5: TableInOut database management screen

Task definition screen

Task definition screen defines information extraction task on a high level. In this screen user can define the variable name for the information that will be extracted, pragmatic type of the table in which the information should be searched for, possible and default unit and in which functional area of the table (header, stub, super-row, stub) the information should be searched for. Task definition screen leads to lexical and semantic rule definition screen. This screen is used for creating and editing task cues. When the user is creating a new task, he/she needs to populate information about the task. In the case of editing, a user is presented with the information defined in the past and he/she can edit them. The example of the task definition screen can be seen in Figure C.6.

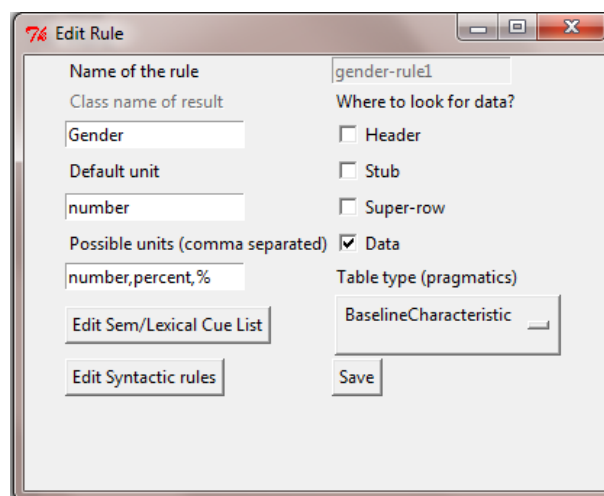


Figure C.6: TableInOut task definition screen

Lexical and semantic rule definition screen

This screen allows a user to define lexical and semantic rules and in which functional area of the table they should be searched for. The screen contains two text boxes, one of defining the white list, while the other for defining the black list. For both lists, the user can define in which functional areas of the table list items should be searched. The user can define lexical cues by adding words or phrases to the lists. The algorithm will perform later matching of the given cues in defined table areas. Also, semantic cues can be defined by stating semantic concept id or annotation type. Previously, the table was annotated with some vocabulary (e.g. UMLS). These annotations will be searched, and in case they are in the stated functional areas, the selected cell will be flagged for extraction. In the case of the UMLS, we store its annotations concept id (as annotation ID) and semantic types (as annotation types). Lexical and semantic rule definition screen can be seen in Figure C.7, with example rule definition.

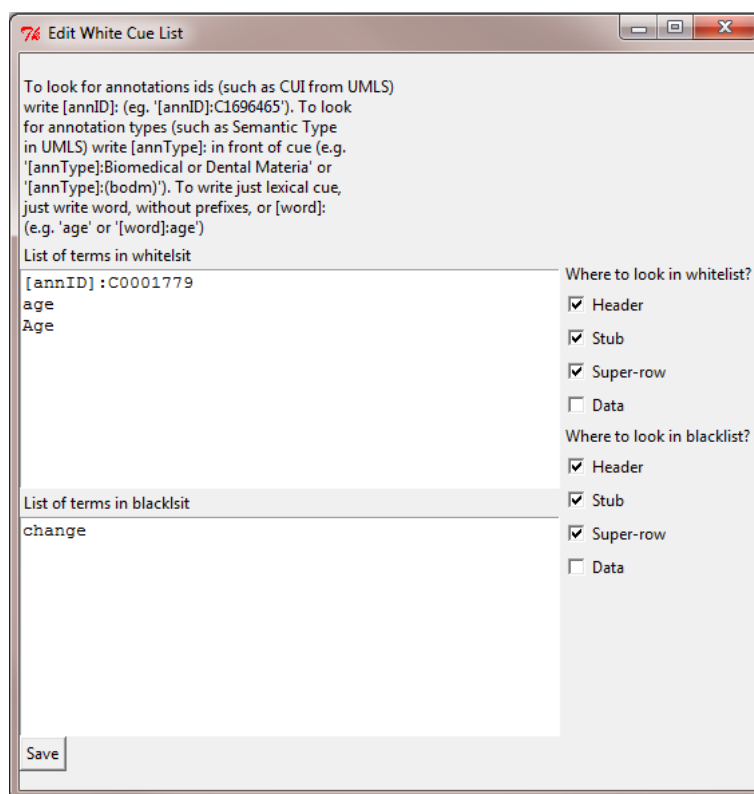


Figure C.7: Lexical and semantic rule definition screen. User can define lexical cues just by stating them or by stating [word] as a prefix; semantic cues can be stated as annotations ids, usually referring to concept ids in certain vocabulary, using [annID] prefix or as annotation types, referring, for example, to UMLS semantic types of annotation, using [annType] prefix.

Syntactic rule definition screens

Using syntactic rule definition screen user can define syntactic rules for extracting information from the table. Syntactic rules are supposed to parse the content in cells and extract the relevant information of interest (e.g. values of the given variable). During the rule creation process, the user is prompt to select one of the pre-created rule sets. These pre-created rule sets contain rules for extracting, for example, statistical values (mean, standard deviation, range), alternative values (two separated numerical values), or just one numeric integer or floating point value. After choosing one of the pre-created rule sets, the user can add new or edit existing rules. Also, the user has a choice to create a totally new rule set. Once created, rule sets can be reused in the future. More details about creating syntactic rules and their syntax will be provided in Section 6.3.3. Example can be seen in Figure C.8.

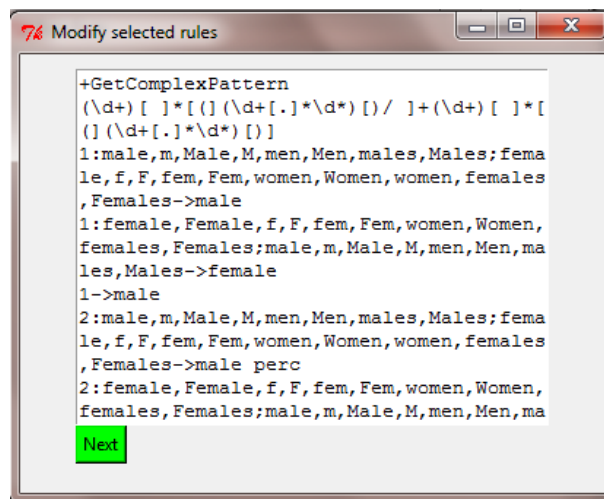


Figure C.8: TableInOut syntactic rule definition screen

Execution screen

Execution screen is the last screen of the wizard, that is presented while execution of information extraction methodology is taking place. The extraction methodology is selecting cells containing any of the defined white list keywords or annotations in the specified areas. Selected cells are checked against the black list. In case they contain cue from the black list the cell is not taken into account. Once cells are selected, syntactic rules are applied to their content, information is extracted, and stored in the database.

C.5 Summary

In this part we presented implementation details of TableInOut software for information extraction from tables. TableInOut uses the output of TableDisentangler, a tool that follows the methodology for functional and structural table analysis. The output of TableDisentangler is enriched using Marvin annotation tool that can annotate table content using multiple knowledge source. Once enriched, user can develop lexical, semantic and syntactic rules using TableInOut and perform information extraction.

TableInOut tool allows reuse of developed rules. It is possible to copy and modify previously developed rules for the other variables. We have included a library of the common numeric value presentation patterns in the TableInOut.

We have performed studies that were described as case studies previously. With the use of TableInOut it was possible to replicate the results of the case studies and in some cases we were able to improve the results.

Appendix D

Guide for writing syntactic rules

D.1 Overview

In this Appendix, we describe the syntactic analysis query language that was designed for the purposes of the work described in this thesis. We developed an engine and description language for defining syntactic rules within selected table cells of interest. It is used by TabInOut as the main engine and description language for defining syntactic rules.

The rules are saved in a single file for each table information extraction project. Then the rules are executed from top to bottom, therefore more specific rules should be on top of the file, while the more generic rules should be at the bottom of the file

D.2 Writing a simple syntactic rule

Each syntactic rule contains three elements:

1. Rule name
2. Rule regular expression
3. Semantic assignment descriptors

The example of syntactic rule looks in the following manner:

```
+SingleFloat1  
(/d+[\.,]{0,1}\d*)  
l->value
```

The first line, that always starts with + symbol contains name of the rule. Name of the rule starts after + symbol. Plus symbol determines the start of the new rule.

The second line is the regular expression of the rule. It is advised to use regular expression groupings (brackets around the value or sequence of interest).

The third line is semantic assignment descriptor of the rule. This rule has only one descriptor, for one regular expression group. However, rule can have many semantic assignment descriptors. However, each rule can have only one name and only one regular expression.

The presented rule is able to floating point values and integers.

D.3 Adding semantics to the syntactic rule

Here is presented a more complex rule:

```
+GetMean
(\d+\.{0,}\d*)[? ]{0,}[()(\d+\.{0,}\d*)[]]
1->mean
2->SD
```

This rule is able to detect values with standard deviation in the brackets. For example: "16.4 (2.3)". The rule contains again name in the first line. Regular expression contains two potentially floating point numbers. These numbers are in regex groups. Then we have two semantic assignment descriptors. First is saying that the first value is mean value, while the second value is standard deviation.

It is possible that for same representation, it is hard to tell what the value is, unless we look at the navigational areas of the table (headers, stubs). For example, value "16.4 (2.3)", can be ambiguous. The first value can be both mean or median. If we have such situation we can use the following rule:

```
+GetMean
(\d+\.{0,}\d*)[? ]{0,}[()(\d+\.{0,}\d*)[]]
1: median , Median->median
1->mean
2->SD
```

In this rule, name and regular expression are the same. However, we have additional semantic assignment descriptor. It again is for the first regex group (the first value), and it has additional part after column symbol (:). After column symbol we can

put a comma separated list of cues that will be searched in navigational areas (headers, stubs, super-rows). In case cue is found that rule will be evoked. Otherwise, the next (default descriptor will be used 1-¿mean). In this situation we assume that in case median is mentioned in header or stubs, the first value is median, otherwise the first value is mean.

Another case is when we have multiple named groups and whose order matter. In the previous cases we knew how to chose value, but we needed cues in order to assign semantics. However, if we have value such as 18:19 and it is related to gender, we cannot know whether there were 18 males or females participants.

For such cases, it is possible to use the following rule:

+GetMaleFemaleRule

(\d+)[/:\ \ , \]\{1,\}(\d+)

1: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F, fem ,Fem, women , Women, females , Females→male

1: female , Female , f ,F, fem ,Fem, women , Women, females , Females ; male ,m, Male ,M, Men, men , males , Males→female

1→male

2: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F, fem ,Fem, women , Women, Females , females→female

2: female , Female , f ,F, fem ,Fem, women , Women, females , Females ; male ,m, Male ,M, Men, men , males , males→male

2→female

Here we have a list after column of comma separated cues about males. After this comes semi-column symbol (;). After semi-column comes comma separated list of female cues. After that comes the assignment (-¿male).

In order to generalize, we can have a semi column separated list of comma separated lists of cues for the words that have to come in certain order. In the given example we are expecting to have one of the male cues appearing before some of the female cues in navigational area in the first semantic assignment descriptor. In the second it is the opposite. We expect some of the female cues to appear before any of the male cues. Third descriptor is the default descriptor for the first group (the first value, e.g. 18) that will invoke in case the cues are not present. Following 2 descriptors are same as the first two, just for the second regex group (it assigns the semantics for the second value (e.g. 19). The last descriptor is the default assignment for the second group.

D.4 Conclusion

Our syntactic analysis query language is suitable for creating rules and adding semantics for extracting particular parts of the cell content. The rules are reusable, especially given that authors present values in tables in relatively standardized way. In the default installation of the TabInOut are provided reusable files for extracting of cumulative statistical values, gender, integers, floating point numbers and alternatives (e.g. 19/21/12).

Appendix E

Examples of lexical and syntactic rules for TableInOut

E.1 Rules for extracting age of patients

E.1.1 Rule configuration

Class : Age
RuleType : Numeric
DefUnit : years
PosUnit : years , weeks , mounts , days , day , mounth
PragClass : BaselineCharacteristic
RuleCreationMech : Lexical
DataInHeader : 0
DataInStub : 0
DataInSuperRow : 0
DataInData : 1

E.1.2 White list

Type : WhiteList
Header : 1
Stub : 1
Super-row : 1
Data : 0
WordList :

[annID] : C0001779

age

Age

E.1.3 Black list

Type : BlackList

Header : 1

Stub : 1

Super-row : 1

Data : 0

WordList :

change

E.1.4 Syntactic rules

Some unicode character needed to be changed from the original file, since latex does not allow unicode character. There are multiple space characters or dash (–) character. Also, pm corresponds to plus-minus (\pm) symbol.

+GetMean1

(\d+\.*\d*)[?]*(\-----,)+(\d+\.*\d*)[]*(\d+\.*\d*)
[?]*(\pm)[?]*(\d+\.*\d*)[]*

1->range_min

2->range_max

3: median , Median->median

3->mean

4->SD

+GetMean2

(\d+\.*\d*)[?]*(\pm)[?]*(\d+\.*\d*)[]*(\d+\.*\d*)
[?]*\{0,\}(\-----,)+[]*(\d+\.*\d*)[]*

1: median , Median->median

1->mean

2->SD

3->range_min

4->range_max

+GetMean4

(\d+\.*\d*)[?]*(\d+\.*\d*)[?]*\{0,\}(\-----,)+

```

(\d+\.\.*\d*)(\s)*
1: median , Median->median
1->mean
2->range_min
3->range_max
+GetRange1
(\d+\.\.*\d*)([? ])*[(\s)(\d+\.\.*\d*)([? ]){0,}[\s]-----,]{1,}
[? ]{0,}(\d+\.\.*\d*)
1: median , Median->median
2->range_min
3->range_max
+GetRange2
[(\d+\.\.*\d*)([? ]){0,}[\s]-----,;]{1,}([? ]){0,}(\d+\.\.*\d*)
[)]([? ])*(\d+\.\.*\d*)
1->range_min
2->range_max
3: median , Median->median
3->mean
+GetRange21
(\d+\.\.*\d*)([? ]){0,}[\s]-----,;]{1,}([? ]){0,}(\d+\.\.*\d*)
[,? ]+(\d+\.\.*\d*)
1->range_min
2->range_max
3->mean
+GetRange3
(\d+\.\.*\d*)([? ]){0,1}[\s]-----,;to ]{1,}([? ]){0,}(\d+\.\.*\d*)
1->range_min
2->range_max
+GetMean6
(\d+\.\.*\d*)([? ]){0,}[\s]pm]{1,}([? ]){0,}(\d+\.\.*\d*)
1: median , Median->median
1->mean
2->SD
+GetMean7
(\d+\.\.*\d*)([? ]){0,}[(\d+\.\.*\d*)([? ]){0,}(\d+\.\.*\d*)]

```

```

1: median , Median->median
1->mean
2->SD
+GetMean8
(\ d+ \. * \ d *)
1: median , Median->median
1->mean

```

E.2 Rules for extracting gender

E.2.1 Rule configuration

```

Class : Gender
RuleType : Numeric
DefUnit : number
PosUnit : number , percent , %
PragClass : BaselineCharacteristic
RuleCreationMech : Lexical
DataInHeader : 0
DataInStub : 0
DataInSuperRow : 0
DataInData : 1

```

E.2.2 White list

```

Type : WhiteList
Header : 0
Stub : 1
Super-row : 1
Data : 0
WordList :
Gender
gender
sex
Sex
male

```

Male
males
Males
Female
female
females
Females
males / females
Male / Females
male / female
Male / Female
Male / Fem
Male : Female
male : female
m/ f
m: f
M/F
M: F
women
Women
Men
 Male
 Female

E.2.3 Black list

Type : BlackList
Header : 1
Stub : 1
Super-row : 0
Data : 0
WordList :
p value
p-value
P value
P-value

change

increase

decrease

p ;

P ;

E.2.4 Syntactic rules

+GetComplexPattern

(\d+)[]*[(](\d+[.]*\d*)[)]/ []+(\d+)[]*[(](\d+[.]*\d*)[)]

1: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F, fem ,Fem, women, Women, women, females , Females →male

1: female , Female , f ,F, fem ,Fem, women, Women, females , Females ; male ,m, Male ,M, men ,Men, males , Males →female

1→male

2: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F, fem ,Fem, women, Women, women, females , Females →male perc

2: female , Female , f ,F, fem ,Fem, women, Women, females , Females ; male ,m, Male ,M, men ,Men, males , Males →female perc

2→male perc

3: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F, fem ,Fem, women, Women, women, Females , females →female

3: female , Female , f ,F, fem ,Fem, women, Women, females , Females ; male ,m, Male ,M, men ,Men, males , Males →male

3→female

4: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F, fem ,Fem, women, Women, women, Females , females →female perc

4: female , Female , f ,F, fem ,Fem, women, Women, females , Females ; male ,m, Male ,M, men ,Men, males , Males →male perc

4→female perc

+GetWithSymbols

(\d+)[]*[Mm][]*[; ,: /][]*(\d+)[]*[Ff]

1→male

2→female

+GetWithSymbols2

(\d+)[]*[Ff][]*[; ,: /][]*(\d+)[]*[Mm]

1→female
 2→male
 +GetFemalePerc
 (\d+[.]*\d*)[]*[%]+[]*[(\d+)]
 1: male , Male , m,M, men , Men, males , Males→male perc
 1: Female , female , fem , F, f , Fem, women, Women, females , Females
 →female perc
 2: male , Male , m, men , Men, males , Males→male
 2: Female , female , fem , F, f , Fem, women, Women, females , Females
 →female
 +GetPercInBrackets
 (\d*[.]*\d+[.]*\d*)[]*[(\d+[.]*\d*)]
 1: male , m, Male , M, men , Men, males , Males→male
 1: female , f , F, fem , Fem, Female , women, Women, Females , females
 →female
 2: male , m, Male , M, men , Men, males , Males→male perc
 2: female , f , F, fem , Fem, Female , women, Women, females , Females
 →female perc
 +GetMaleFemale1Perc
 (\d+[.]\d+)[/:\ ,]{1,}(\d+[.]\d+)
 1: male , m, Male , M, men , Men, males , Males ; female , f , F, fem , Fem,
 women, Women, women, Females , females →male perc
 1: female , Female , f , F, fem , Fem, women, Women, females , Females ;
 male , m, Male , M, men , Men, males , Males→female perc
 1→male perc
 2: male , m, Male , M, men , Men, males , Males ; female , f , F, fem , Fem,
 women, Women, women, females , Females→female perc
 2: female , Female , f , F, fem , Fem, women, Women, females , Females ;
 male , m, Male , M, men , Men, males , males→male perc
 2→female perc
 +GetPercWOBrackets
 (\d+)[]{1,}(\d+[.]*\d*)[%]
 1: male , Male , m,M, men , Men, males , Males→male
 1: female , Female , f , F, fem , Fem, women, Women, females , Females
 →female

1→male

2: male , Male , m,M, men , Men, males , Males→male perc

2: female , Female , f , F, fem , Fem, women, Women, Females , females
→female perc

2→female perc

+GetMaleFemale1

(\d+)[/:\ \ ,]{1,}(\d+)

1: male , m, Male , M, men , Men, males , Males ; female , f , F, fem , Fem,
women, Women, females , Females→male

1: female , Female , f , F, fem , Fem, women, Women, females , Females ;
male , m, Male , M, Men, men , males , Males→female

1→male

2: male , m, Male , M, men , Men, males , Males ; female , f , F, fem , Fem,
women, Women, Females , females→female

2: female , Female , f , F, fem , Fem, women, Women, females , Females ;
male , m, Male , M, Men, men , males , males→male

2→female

+GetMaleFromTxt

Male : [](\d+)[]+(\d+)[%]

1→male

2→male perc

+GetFemaleFromTxt

Female [:] [](\d+)[]+(\d+)[%]

1→female

2→female perc

+GetMaleFemalePerc

(\d+[.]*\d*)([/:\ \ ,]{1,})(\d+[.]*\d*)

1: male , m, Male , M, Men, men , males , Males ; female , f , F, fem , Fem,
women, Women, females , Females→male perc

1: female , Female , f , F, fem , Fem, women, Women, females , Females ;
male , m, Male , M, Men, men , males , Males→female perc

1→male perc

2: male , m, Male , M, Men, men , males , Males ; female , f , F, fem , Fem,
women, Women, females , Females→female perc

2: female , Female , f , F, fem , Fem, women, Women, females , Females ;

```

male ,m, Male ,M, men ,Men, males , Males->male  perc
2->female  perc
+GetMaleFemale2
(\d+)[ ]{1,}(\d+[.]*\d*)[%]{1,}
1: male ,m, Male ,M, men ,Men, males , Males->male
1: female , f ,F, Fem, Female , women, Women, females , Females
->female
1->total
2: male ,m, Male ,M, men ,Men, males , Males->male  prec
2: female , f ,F, Fem, Female , women, Women, Females , females
->female  perc
2->female  perc
+GetPerc
(\d+)[ ]{1,}(\d+[.]*\d*){1,}
1: male ,m, Male ,M, Men, men , males , Males->male
1: female , f ,F, Fem, Female , women, Women, females , Females
->female
1->total
2: male ,m, Male ,M, Men, men , males , Males->male  perc
2: female , f ,F, Fem, Female , women, Women, females , Females
->female  perc
2->female  perc
+GetMale
Male: (\d+)[ ][(\d+)[%][ ])]
1->male
2->male  percent
+GetFemale
Female: (\d+)[ ][(\d+)[%][ ])]
1->female
2->female  percent
+GetNumPerc
(\d+[.]\d+)
1: male ,m, Male ,M, men ,Men, males , Males->male  perc
1: female , f ,Female ,F, women, Fem, fem , Women, women, females ,
Females->female  perc

```

+GetNum

(\d+)

1 : male , m , Male , M , men , Men , males , Males → male

1 : female , f , Female , F , women , Fem , fem , women , Women , females ,
Females → female

1 → male

Appendix F

Library of developed syntactic rules

Rules that were developed can be found in a row format at <https://github.com/nikolamilosevic86/TabInOut/tree/master/DefaultSyntacticRules>. In this Appendix, some of the special characters are replaced, therefore we recommend use of the rules from the stated link.

F.1 Syntactic rule for extracting single positive integer value

```
+SingleInteger1  
(/d+)  
l->value
```

F.2 Syntactic rule for extracting single positive floating point value

```
+SingleFloat1  
(/d+[\.]{0,1}\d*)  
l->value
```

F.3 Syntactic rules for extracting positive statistical values (mean, median, standard deviation, ranges, percentages)

+GetMean1

(\d+\.\d*)[?]*\[\-----,](\d+\.\d*)[]*(
(\d+\.\d*)[?]*\[\pm][?]*(\d+\.\d*)[])*

1->range_min

2->range_max

3: median , Median->median

3->mean

4->SD

+GetMean2

(\d+\.\d*)[?]*\[\pm][?]*(\d+\.\d*)[]*([-]
\d+\.\d)[?]*\{0,\}[\-----,]+[]*(\d+\.\d*)[])*

1: median , Median->median

1->mean

2->SD

3->range_min

4->range_max

+GetMean4

(\d+\.\d*)[?]*\[(\[([-]*\d+\.\d*)[?]*\{0,\}
[\-----,]+[]*\{0,\}([-]*\d+\.\d*)[\])\])*

1: median , Median->median

1->mean

2->range_min

3->range_max

+GetRange1

(\d+\.\d*)[?]*\[(\[([-]*\d+\.\d*)[?]*\{0,\}
[\-----,]\{1,\}[?]*\{0,\}(\d+\.\d*)

1: median , Median->median

2->range_min

3->range_max

+GetRange2

```

[ ( ( \ d + \ . * \ d * ) [ ?    ] { 0 , } [ \ - - - - , ; ] { 1 , } [ ?    ] { 0 , }
( \ d + \ . * \ d * ) [ ) ] [ ?    ] * ( \ d + \ . * \ d * )
1->range_min
2->range_max
3: median , Median->median
3->mean
+GetRange21
( \ d + \ . * \ d * ) [ ?    ] { 0 , } [ \ - - - - , ; ] { 1 , } [ ?    ] { 0 , }
( \ d + \ . * \ d * ) [ , ?    ] + ( \ d + \ . * \ d * )
1->range_min
2->range_max
3->mean
+GetRange3
( \ d + \ . * \ d * ) [ ?    ] { 0 , 1 } [ \ - - - - , ; to ] { 1 , } [ ?    ] { 0 , }
( \ d + \ . * \ d * )
1->range_min
2->range_max
+GetMean6
( \ d + \ . * \ d * ) [ ?    ] ( { 0 , } [ \ pm ] { 1 , } [ ?    ] { 0 , }
( \ d + \ . * \ d * )
1: median , Median->median
1->mean
2->SD
+GetMean7
( \ d + \ . { 0 , } \ d * ) [ ?    ] { 0 , } [ ( ( \ d + \ . { 0 , } \ d * ) [ ) ]
1: median , Median->median
1->mean
2->SD
+GetMean8
( \ d + \ . * \ d * )
1: median , Median->median
1->mean

```

F.4 Syntactic rules for extracting alternative values (case of gender extraction)

+GetComplexPattern

```
(\d+)[ ]*[(](\d+[.]*\d*)[)]/ [ ]+(\d+)[ ]*[(](\d+[.]*\d*)[)]
```

1: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F ,

fem ,Fem, women ,Women, women , females , Females→male

1: female , Female , f ,F, fem ,Fem, women ,Women, females ,

Females ; male ,m, Male ,M, men ,Men, males , Males→female

1→male

2: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F ,

fem ,Fem, women ,Women, women , females , Females→male perc

2: female , Female , f ,F, fem ,Fem, women ,Women, females ,

Females ; male ,m, Male ,M, men ,Men, males , Males→female perc

2→male perc

3: male ,m, Male ,M, men ,Men, males , Males ; female ,

f ,F, fem ,Fem. women ,Women, women , Females , females→female

3: female , Female , f ,F, fem ,Fem, women ,Women, females ,

Females ; male ,m, Male ,M, men ,Men, males , Males→male

3→female

4: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F ,

fem ,Fem. women ,Women, women , Females , females→female perc

4: female , Female , f ,F, fem ,Fem, women ,Women, females ,

Females ; male ,m, Male ,M, men ,Men, males , Males→male perc

4→female perc

+GetWithSymbols

```
(\d+)[ ]*[Mm][ ]*[,;:/][ ]*(\d+)[ ]*[Ff]
```

1→male

2→female

+GetWithSymbols2

```
(\d+)[ ]*[Ff][ ]*[,;:/][ ]*(\d+)[ ]*[Mm]
```

1→female

2→male

+GetFemalePerc

$(\backslash d + [.] * \backslash d *) [] * [\%] + [] * [() (\backslash d +) ()]$
 1: male , Male , m, M, men , Men, males , Males → male percent
 1: Female , female , fem , F, f , Fem, women , Women, females ,
 Females → female percent
 2: male , Male , m, men , Men, males , Males → male
 2: Female , female , fem , F, f , Fem, women , Women, females ,
 Females → female
 +GetPercInBrackets
 $(\backslash d * [.] * \backslash d + [.] * \backslash d *) [] * [() (\backslash d + [.] * \backslash d *) ()]$
 1: male , m, Male , M, men , Men, males , Males → male
 1: female , f , F, fem , Fem, Female , women , Women, Females ,
 females → female
 2: male , m, Male , M, men , Men, males , Males → male perc
 2: female , f , F, fem , Fem, Female , women , Women, females ,
 Females → female perc
 +GetMaleFemale1Perc
 $(\backslash d + [.] \backslash d +) [/ : \backslash ,] \{ 1 , \} (\backslash d + [.] \backslash d +)$
 1: male , m, Male , M, men , Men, males , Males ; female , f , F,
 fem , Fem . women , Women, women , Females , females → male perc
 1: female , Female , f , F, fem , Fem, women , Women, females ,
 Females ; male , m, Male , M, men , Men, males , Males → female perc
 1 → male perc
 2: male , m, Male , M, men , Men, males , Males ; female , f , F,
 fem , Fem, women , Women, women , females , Females → female perc
 2: female , Female , f , F, fem , Fem, women , Women, females ,
 Females ; male , m, Male , M, men , Men, males , males → male perc
 2 → female perc
 +GetPercWOBrackets
 $(\backslash d +) [] \{ 1 , \} (\backslash d + [.] * \backslash d *) [\%]$
 1: male , Male , m, M, men , Men, males , Males → male
 1: female , Female , f , F, fem , Fem, women , Women, females ,
 Females → female
 1 → male
 2: male , Male , m, M, men , Men, males , Males → male perc
 2: female , Female , f , F, fem , Fem, women , Women, Females ,


```

females->female perc
2->female perc
+GetMaleFemaleRule
(\d+)[/: , ]{1,}(\d+)
1: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F,
fem ,Fem, women ,Women, females , Females->male
1: female , Female , f ,F, fem ,Fem, women ,Women, females ,
Females ; male ,m, Male ,M, Men ,men , males , Males->female
1->male
2: male ,m, Male ,M, men ,Men, males , Males ; female , f ,F,
fem ,Fem, women ,Women, Females , females->female
2: female , Female , f ,F, fem ,Fem, women ,Women, females ,
Females ; male ,m, Male ,M, Men ,men , males , males->male
2->female
+GetMaleFromTxt
Male :[ ](\d+)[ ]+(\d+)[%]
1->male
2->male perc
+GetFemaleFromTxt
Female [:][ ](\d+)[ ]+(\d+)[%]
1->female
2->female perc
+GetMaleFemalePerc
(\d+[.]*\d*)[/: , ]{1,}(\d+[.]*\d*)
1: male ,m, Male ,M, Men ,men , males , Males ; female , f ,F,
fem ,Fem, women ,Women, females , Females->male perc
1: female , Female , f ,F, fem ,Fem, women ,Women, females ,
Females ; male ,m, Male ,M, Men ,men , males , Males->female perc
1->male perc
2: male ,m, Male ,M, Men ,men , males , Males ; female , f ,F,
fem ,Fem, women ,Women, females , Females->female perc
2: female , Female , f ,F, fem ,Fem, women ,Women, females ,
Females ; male ,m, Male ,M, men ,Men , males , Males->male perc
2->female perc
+GetMaleFemale2

```

$(\backslash d+)[\]\{1,\}(\backslash d+[.]*\backslash d*)[\%]\{1,\}$
 1: male ,m, Male ,M, men ,Men, males , Males→male
 1: female , f ,F, Fem, Female , women ,Women, females ,
 Females→female
 1→total
 2: male ,m, Male ,M, men ,Men, males , Males→male perc
 2: female , f ,F, Fem, Female , women ,Women, Females ,
 females→female perc
 2→female perc
 +GetPerc
 $(\backslash d+)[\]\{1,\}(\backslash d+[.]*\backslash d*)\{1,\}$
 1: male ,m, Male ,M, Men ,men , males , Males→male
 1: female , f ,F, Fem, Female , women ,Women, females ,
 Females→female
 1→total
 2: male ,m, Male ,M, Men ,men , males , Males→male perc
 2: female , f ,F, Fem, Female , women ,Women, females ,
 Females→female perc
 2→female perc
 +GetMale
 Male: $(\backslash d+)[\][(\backslash d+)[\%][\]]$
 1→male
 2→male percent
 +GetFemale
 Female: $(\backslash d+)[\][(\backslash d+)[\%][\]]$
 1→female
 2→female percent
 +GetNumPerc
 $(\backslash d+[.]*\backslash d+)$
 1: male ,m, Male ,M, men ,Men, males , Males→male perc
 1: female , f , Female ,F, women ,Fem, fem , Women, women ,
 females , Females→female perc
 +GetNum
 $(\backslash d+)$
 1: male ,m, Male ,M, men ,Men, males , Males→male

1 : female , f , Female , F , women , Fem , fem , women , Women ,
females , Females → female
1 → male