TABLE MINING AND DATA CURATION FROM BIOMEDICAL LITERATURE

AN END OF YEAR REPORT SUBMITTED TO THE UNIVERSITY OF MANCHESTER
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2014

By
Nikola Milošević
Supervised by
Dr Goran Nenadić, Robert Hernandez, Cassie Gregson
School of Computer Science

Contents

Abstract

TABLE MINING AND DATA CURATION FROM BIOMEDICAL LITERATURE
An end of year report submitted to the University of Manchester.
Nikola Milošević
Supervisors: Dr Goran Nenadić, Robert Hernandez, Cassie Gregson

Current text mining efforts are mostly focused on extracting information from the main body of research and scholarly articles. However, tables contain important information such as key characteristics of clinical trials, clinical outcomes, interaction between drugs or proteins. Processing of information from tables is often limited to the textual caption and data presented in tables are typically ignored.

The aim of this project is to examine information extraction and table mining from biomedical scientific publications. We aim to provide support for semi-automated data curation of the data stored in tables and its integration with other information presented in the article.

Table processing usually includes three steps: table recognition (locating tables in the document), functional analysis of the tables (recognizing cell's function, i.e. whether the cell is part of the header, stub, sub-header or body of the table) and table understanding (semantic processing, analysis and understanding of data in the table). In our method we split table understanding in a set of tasks which include header and stub processing, finding navigational path for each data cell, normalization of cell's value, pattern analysis, pattern linking, information extraction and knowledge integration.

As a pilot project, we present a case study of the information extraction of body mass index, participant group names and patient weights from tables in clinical trial publications from PubMed Central (PMC). The study showed that it is possible to successfully extract information from table, although some classes are more challenging because of the layout and the way their data is presented. Certain amount of domain knowledge seems to be inevitable in order to correctly select the right piece of information. Preliminary evaluation of our method showed F-measure of 85% for body

mass index extraction, 71.3% for participant group name extraction and 57.7% for participant weight extraction.

Chapter 1

Introduction, Aims and Objectives

1.1 Introduction

The amount of published scientific research is accelerating: the number of published papers is growing almost exponentially. This is especially true for the biomedical research.

The exponential growth can be better viewed through the cumulative number of citations in Medline, which can be seen in Figure 1.2.

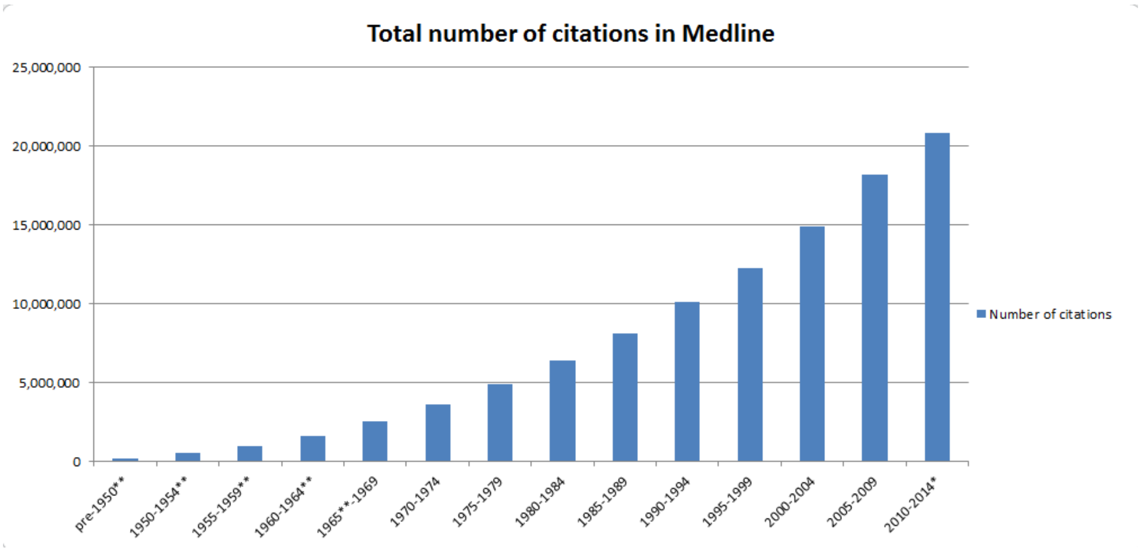


Figure 1.1: Cumulative number of Medline citations [1]

MEDLINE contains over 23 million references from approximately 5600 journals in 30 languages. Only in 2013, over 734 000 new citation were added to the database

[1] http://www.nlm.nih.gov/bsd/medline_lang_distr.html

(United States National Library of Medicine, 2014). On average, there are over 2000 scientific papers published every day in biomedical domain. It is impossible for the scientists and professionals, who need to be on track with the state-of-the-art in the field, to cope with this amount of published research.

Text mining can provide a tools to cope with this enormous amount of articles. In recent years, a notables progress have been made in biomedical text mining (Cohen and Hersh, 2005). However, these efforts have been focused mainly on processing of unstructured text and most of them ignored lists, tables and figures. Lists, tables and figures are the main means of presenting data in articles and these data are currently lost while using text mining tools.

Tables are used as an appropriate format for storing large amount of factual or statistical data in a structured and compact way. They also provide a framework for storing multidimensional data. Various table layout provide flexibility for structuring data and for storing large amount of information in compact way. If the body of the article is referred as unstructured text, tables may be called semi-structured textual parts of the article. Tables have two main uses: to present data and to present short parallel description in compact and structured way that otherwise would have to be expanded and listed in the text (Alley, 1996). Example of table presenting parallel descriptions can be seen in Figure 1.2. If the authors wanted to present data in text he would have to expand the description and use large amount of repetition.

Huge amount of relevant data in scientific publication is stored in tables. In the biomedical field, the results of clinical trials, interactions between substances, drug side effects, information about arms and patients are usually stored in tables. Experimental settings and results are often presented in the tables. In PMC database, more than 72% of research articles contain tables. However, we manually found that not all of the documents stored in the database are full text documents (for example some may contain only abstract and list of references). Having this in mind, the percentage of biomedical research publications with tables might be even higher. Also, we calculated that the articles contain at average 2.72 tables in PMC database. Mean number of cells in these tables are over 80. This average takes in account only documents that contain at least one table. The presented statistics heavily support the argument that tables are used frequently in the biomedical field. Also, it supports the argument that a significant portion of the information is presented in tables.

In the textual part 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.

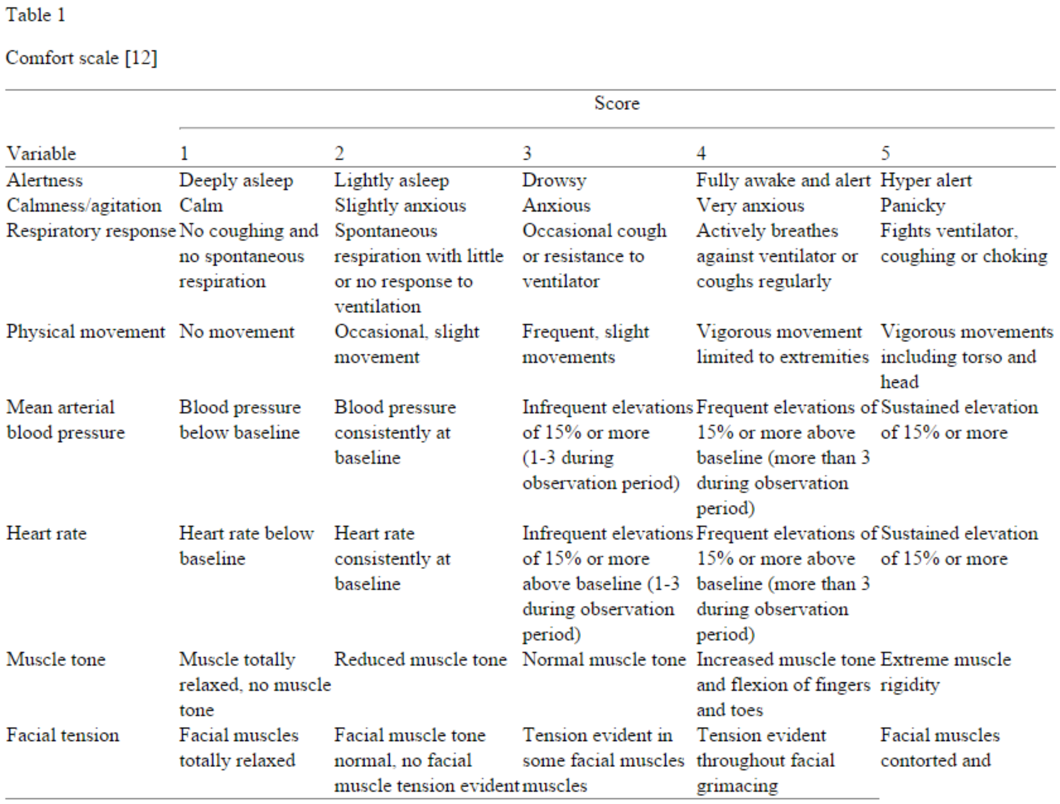


Table 1

Comfort scale [12]

	Score				
Variable	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 grimacing	Facial muscles contorted and

Figure 1.2: Example of table showing parallel description in structured and compact way (PMC31582)

In Figure 1.3 could be seen 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)”(Thomachot et al., 2001). The article highlights significant difference in table’s data and discusses the data, but do not mention the numerical values. Presenting and discussing data in this way is usual in scientific literature.

Tables may contain various layouts, but also text in the data cell may contain acronyms, abbreviations and ungrammatical phrases. Authors may use various means and patterns of presenting data in the tables. Because of its flexibility in structure and the means of data presentation, table mining is a challenging task. Tables, as special case of visually structured text, are hard to read for both people and machines. It has been shown, that the readers are prone to mistakes while reading tables without previous knowledge of the field or the experience on how to read certain types of tables

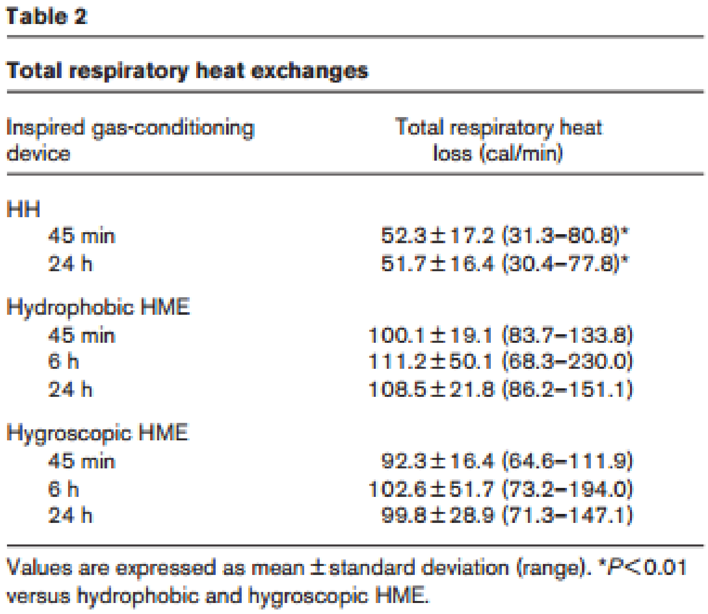


Table 2

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 (PMC29053)

(Wright, 1977).

The information stored in tables are often key for understanding the methodology and reproducing the experiment. They need to be integrated with other information extracted from the literature.

It is necessary for text mining systems to process all the elements of the article in order to be able to acquire knowledge that the human reader would acquire by reading the same document. Almost all the elements of the scientific document might contain important data and should be processed. Important data and information could be in a visually structured parts of the documents, such as lists, tables and even figures. In this work we will focus on tables. Solving the problem of table understanding and curating data from tables in scientific literature will be able to facilitate work for researchers and to speed-up the future research in the field.

1.2 Aims and Objectives

The aim of this research is to examine table mining and data curation from biomedical scientific literature. We aim to create a method for information extraction from tables in the biomedical scientific literature and allow efficient curation of data that has been extracted. We will use domain of clinical trials to motivate and validate the proposed method.

Specifically , the objectives of the research are:

1. Investigate types and common table structures in the biomedical literature, including the means and patterns of presenting values and dimensions.

2. Design, implement and evaluate a method for structured table decomposition and data extraction by identifying information about relationships between cells.

3. Design, implement and evaluate system for semi-automated curation and querying of data extracted from tables, in which user would be able to interact with the extracted data.

4. Perform a case studies on clinical trial and drug-drug interaction articles.

The code and data related to this research will be shared with scientific community under open source licences and will be freely available for further development.

1.3 Scope

The general scope of this research is to investigate table mining in the biomedical domain. However, we will start conducting our research on clinical trial documents as a sub-domain of biomedical literature. The clinical trial literature is one of the most relevant sub-fields for drug discovery and disease treatment research. Tables in the clinical trial documents may give an valuable insight about general characteristics of a clinical trial such as demographics of the participants, trial arms, names and the side effects of the tested drugs. Often clinical trial documents may contain information about interactions between biomedical substances, such as drug-drug or protein-protein interactions.

We are also aiming to explore and generalize our approach to other types of biomedical literature in the later phase of the research.

Biomedical documents are published in number of formats, including HTML, XML and PDF. Most of the well-known databases in the biomedical domain, such as MEDLINE and PMC, keep their documents in XML format. The focus of this research will be on processing of biomedical documents in XML format. For other common formats in literature, such as PDF, there are a number of tools that are capable of converting documents into the XML format such as pdf2xml, pdftohtml, pdfextract, SectLabel and PDFX (Constantin, 2014). These tools might be used more or less successfully in preparatory step for the PDF format. We will explore the role of PDF conversion on table mining and we will try to identify potential shortcomings of current approaches for the further table mining processing.

1.4 Report structure

This report contains five chapters. In the first chapter, we presented motivation, aim and objectives of the project. In the second chapter literature background is given, starting with general definitions from the text mining field, followed by review of relevant research in biomedical text mining and table mining. In the third chapter we presented an overview of our method for table mining and data curation from biomedical articles. The fourth chapter presents work done in the first year of the project, which includes analysis of the tables in clinical articles, table decomposition method, information extraction case study and data curation interface. The last chapter of the report concludes the report and describes the work that will be done in the following two years of the project, outlining the tasks and their timelines.

Chapter 2

Literature background

Table mining from biomedical literature joins together fields of text mining, its applications to biomedical documents and table mining. We will first give basic definitions from the text mining field. The second part of the review will examine relevant work in text mining of the biomedical literature. In the last part of this review we will give an overview of work in table processing, table information extraction and table mining.

2.1 Text mining introduction

”Text Mining” is a relatively new area of computer science that has a strong connection with natural language processing (NLP), data mining, machine learning, information retrieval and knowledge management. It uses information extraction and is defined as the process of discovering and extracting knowledge from unstructured data (Meystre et al., 2008; Hearst, 1999). Text mining typically comprises three steps: information retrieval (to gather relevant texts), information extraction (to extract specific types of information from texts of interest), and data mining (to find associations among the extracted pieces of information). Text Mining is able to help humans in many fields where professionals are dealing with a large amounts of unstructured textual data. One of the main application of text mining is extracting knowledge from scientific literature and helping scientists coping with information overload. Text mining is different from information retrieval and text summarization because they focus on the whole documents, while text mining typically operates on a finer level of granularity examining relationships between information both within and between documents (Cohen and Hersh, 2005).

Tasks of text mining can be separated into set of smaller subtasks. The subtasks can be divided broadly into three classes preparatory processing, general purpose NLP tasks, and problem-dependent tasks (Feldman and Sanger, 2007). A hierarchy of text mining subtasks is shown in Figure 2.1.

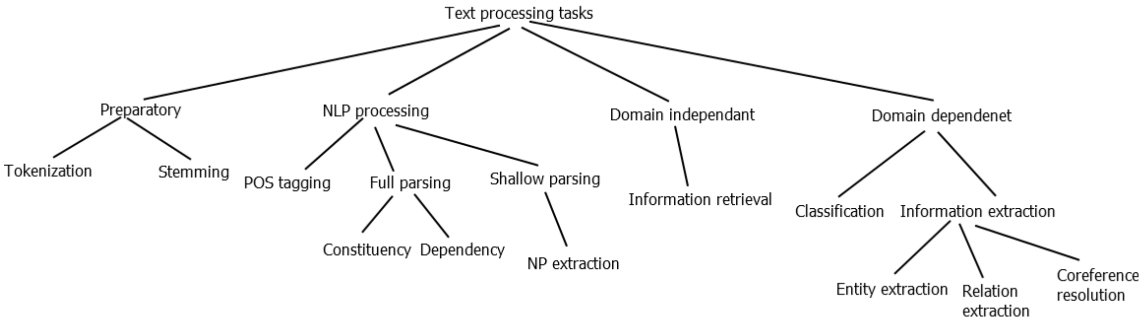


Figure 2.1: A taxonomy of text processing tasks

Preparatory processing converts the raw representation into a structure suitable for further linguistic processing. For example, the raw input can be a PDF document, scanned page or recorded speech, which needs to be converter to text.

Stemming is a computational procedure which reduces all words with the same root (or, if prefixes are left untouched, the same stem) to a common form, usually by stripping each word of its derivational and inflectional suffixes (Lovins, 1968). Stemmers are reducing all words to the same form using set of rules. **Lemmatisation** is the process of grouping together the different inflected forms of a word so they can be analysed as a single item (Brown, 1993). Lemmatization uses a dictionary of all word forms to group them together and create lemma (most often grammatical root of the word).

Sentence splitting is the process of breaking text into sentences.

Tokenization is the process of breaking the text into words (tokens).

Phrase recognition is the process of recognizing common phrases in text.

Part-of-speech (POS) tagging is the annotation of words with the appropriate POS tags based on the context in which they appear. POS tags divide words into categories based on the role they play in the sentence and provide information about the semantic content of a word. Example of the POS tagged sentence can be seen in the Figure 2.2.

Syntactical parsing performs a syntactical analysis of sentences according to a certain grammar theory. *Constituency grammars* describe the syntactical structure of sentences in terms of recursively built phrase sequences of syntactically grouped elements (noun phrases, verb phrases, prepositional phrases, adjective phrases, clauses). Example of the parsed sentence using constituency grammar can be seen in Figure 2.3.

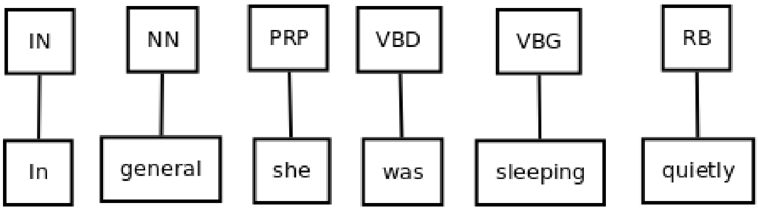


Figure 2.2: Part-of-speech tagged sentence: In general she was sleeping quietly. (IN - Preposition; NN - Noun; PRP - Preposition; VDB - Verb in past tense; VBG – Verb, gerund or present participle; RB - Adverb. Stanford parser (Toutanova et al., 2003) was used to POS tag sentence)

Dependency grammars do not recognize the constituents as separate linguistic units, but focus instead on the direct relations between words (Feldman and Sanger, 2007). Example of sentence parsed with dependency grammar can be seen in Figure 2.4. Instead of producing complete analysis of sentence, shallow parsers produce only partial non-recursive analysis. Shallow parser produces chunks of noun, verb or prepositional phrases (Li and Roth, 2001; Abney, 1992).

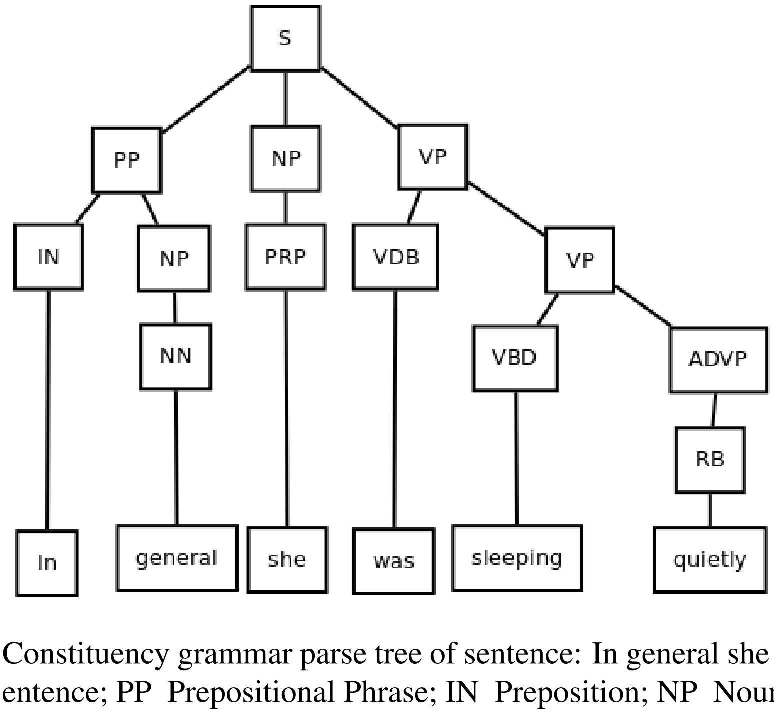


Figure 2.3: Constituency grammar parse tree of sentence: In general she was sleeping quietly (S Sentence; PP Prepositional Phrase; IN Preposition; NP Noun Phrase; NN Noun; PRP Preposition; VP Verb Phrase; VDB Verb in past tense; ADVP Adverb Phrase; RB Adverb. Stanford parser was used to parse sentence)

Categorization or classification is process where text is clustered and classified

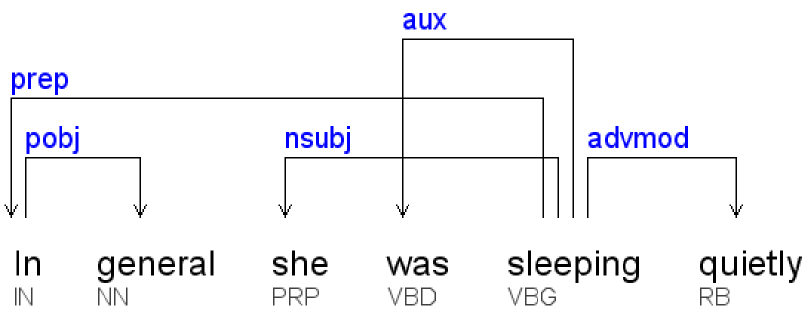


Figure 2.4: Dependency parse of sentence: In general she was sleeping quietly. Part-of-speech tags: IN - Preposition; NN - Noun, singular or mass; PRP – Personal pronoun; VBD – Verb in past tense; VBG – Verb, gerund or present participle; RB - Adverb; Link types: prep – prepositional modifier; pobj – object of a preposition; nsubj – nominal subject; aux – auxiliary; advmod – adverbial modifier. Parsed by Stanford parser (De Marneffe and Manning, 2008). Image created using DependenSee[3].

into particular document or sentence classes (classes can be topics, sentiments, subjectivity classes of article, etc.).

Information extraction (IE) involves extracting predefined types of information from text. In contrast, **information retrieval (IR)** is focused on finding documents and has some very popular examples such as Google or PubMed search engines. IR returns documents whereas IE returns information or facts (Meystre et al., 2008). Hobbs (1993) defined information extraction system as a cascade of transducers or modules that at each step add structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically. The information extraction task can be divided into several subtasks:

- *Named Entity Recognition (NER)* is a sub-field of information extraction and refers to the task of recognizing expressions denoting entities (i.e. Named Entities), such as diseases, drugs, or people's names, in free text documents (Meystre et al., 2008).

[3] `https://github.com/awaisathar/dependensee`

- *Relation recognition* is a task of recognizing relationships between extracted entities and making some relationship structure (usually graphs or trees)

- *Co-reference resolution* is the process of determining whether two expressions in natural language refer to the same entity in the world (Soon et al., 2001)

Trigger words are the words with high probability to find piece of information or some entity in its vicinity (Talukdar et al., 2006). They are usually used to trigger some rule or pattern in rule-based information extraction systems.

The usual flow of the text mining systems is shown in Figure 2.5.

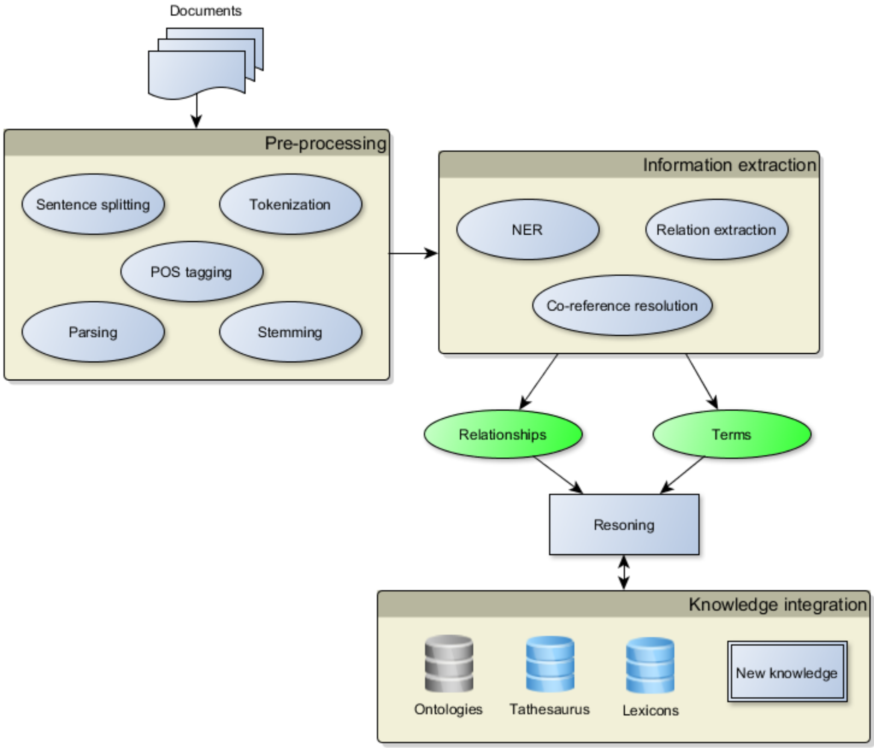


Figure 2.5: Text mining work flow. Selected documents are firstly preprocessed. Using information extraction techniques terms and relationships of interest are extracted. In the final step, using already integrated knowledge sources, reasoning is performed and potential new knowledge is discovered. When the new knowledge is discovered, knowledge sources may be updated

Three basic types of approaches to text mining have been prevalent:

- A co-occurrence-based approaches look for the concepts that occur in the same unit of text (sentence or paragraph) and make relationship between them.

- A rule-based approaches make use of the knowledge about language and common linguistic structures in some domain, transformed to rules by human experts.

- A statistical or machine learning-based approaches make advantage of machine learning algorithms to learn rules and patterns and extract relationships. Machine learning can be supervised, unsupervised or semi-supervised (Cohen and Hunter, 2008).

For evaluation of text mining systems, typical measures include precision, recall and F-measure. Items retrieved from a text mining system can be true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN). The total number of relevant items is TP + FN, while the total number of retrieved items is TP + FP. Recall is the proportion of relevant items retrieved over the all relevant documents. Precision is the proportion of relevant items retrieved over the all retrieved items.

$$Recall = \frac{TP}{TP+FN} \tag{2.1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2.2}$$

$$F = \frac{(1+\beta)^2 * Recall * Precision}{(\beta^2 * Precision) + Recall} \tag{2.3}$$

The variable beta indicates the relative value of precision. A value of beta = 1, which is usually used, indicates the equal value of recall and precision, whereas lower values indicate more emphasis on precision and higher values indicate more emphasis on recall (Hersh, 2005).

2.2 Text Mining of the Biomedical literature

For our purposes, we define biomedical text to be the kind of text that appears in biomedical books, articles, literature abstracts, posters, and so forth. Clinical texts, on the other hand, are texts written by clinicians in the clinical setting. These texts describe patients, their pathologies, their personal, social, and medical histories, findings made during interviews or during procedures, and so forth (Meystre et al., 2008).

Text mining of biomedical and clinical documents is one of the most researched domain specific applications of text mining. There is a growing need for aiding scientists in the biomedical field to be informed about current research and state-of-the-art practices.

2.2.1 Resources for biomedical text mining

MEDLINE contains over 21 million references to articles from approximately 5,600 journals in 30 languages, dating back to the 1960s. In 2013, over 734,000 new citations were added to the database, and it continues to grow at a steady pace (United States National Library of Medicine, 2014). Articles from MEDLINE could be accessed using PubMed. PubMed is a sophisticated boolean search engine that allows users to query not only on abstract text, but also on metadata fields such as MeSH terms. MeSH (Medical Subject Headings) is the US National Library of Medicine's controlled vocabulary thesaurus used for indexing articles for PubMed (The National Center for Biotechnology Information, 2014). PubMed Central (PMC) is a free full-text archive of biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine (The National Center for Biotechnology Information, 2014). MEDLINE database, PubMed and PubMed Central are the most commonly used as resources for articles in the field of biomedical and clinical text mining.

Over the time, several lexicons, thesauruses, ontologies and semantic networks are created for the biomedical field. The Unified Medical Language System is a repository of biomedical vocabularies developed by the US National Library of Medicine. The UMLS integrates over 2 million names for some 900 000 concepts from more than 60 families of biomedical vocabularies, as well as 12 million relations among these concepts. The major component of the UMLS is the Metathesaurus, a repository of inter-related biomedical concepts. The two other knowledge sources in the UMLS are the Semantic Network, providing high-level categories used to categorize every Metathesaurus concept, and lexical resources including the SPECIALIST lexicon and programs for generating the lexical variants of biomedical terms (Bodenreider, 2004).

The BioLexicon is a lexical resource that brings together terminologies from large public bioinformatics resources. It is representing terms in conjuction with lexical and statistical information. It is available in XML interchangeable format and it should be also available as a dump of relational database in the future.

The Read Codes are a hierarchically-arranged controlled clinical vocabulary introduced in the early 1980s and now consisting of three maintained versions. They are updated every three month for clinical terms and monthly for drugs and appliances(Robinson et al., 1997). Read is mainly used in the United Kingdom for coding of automated and clinical data.

The International Classification of Diseases (ICD) is a classification of diseases and other health problems recored on many types of health and vital records including death certificates and health records. It is developed by World Health Organization (WHO). ICD-10 is a current release of the classification and it came into use in WHO Memeber States as from 1994. The 11th version is under development and it is expected to be released in 2017. Most common use of the ICD-10 is to code diseases in clinical records, notes and death reports (World Health Organization , 2014).

Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) is one of the most comprehensive, multilingual clinical healthcare terminology in the world. SNOMED CT comprehensively covers clinical findings, symptoms, diagnoses, procedures, body structures, organisms and other etiologies, substances, pharmaceuticals, devices and specimen. It is managed and maintained by IHSDO (International health terminology standards development organization), an international non-profit organization from Denmark (IHTSDO , 2014).

The GO (Gene Ontology) project has developed three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner (Gene Ontology Consortium, 2014a).

The Ontology of Clinical Research (OCRe) is a formal ontology for describing human studies that provides methods for binding to external information standards (e.g. BRIDG) and clinical terminologies (e.g. SNOMED CT). OCRe is a set of modular components related by their import relationship. The core modules are clinical (containing shared upper-level entities), study design (containing descriptors of study design and a categorization of studies by their design descriptors) and research (containing terms and relationships that characterize a study) (Tu et al., 2009).

The Medical Entities Dictionary (MED) developed at Columbia Presbyterian Medical Center (CPMC) is a knowledge base of medical concepts that consist of taxonomic relations in addition to other relevant semantic relations. MED uses a semantic network model that includes a classification hierarchy. One concept, called Medical Entity, serves as the topmost node in the classification. All other concepts are nodes in this graph, as immediate descendants of at least one other node. These parent-child relationships between nodes correspond to the classification of the concepts. Each concept may have several parents; however, these relationships are acyclic and directed describing relationship. The MED was initially populated with the 131 semantic classes of the second version of the UMLS knowledge sources and UMLS Metathesaurus was used for terms (Cimino et al., 1994).

There are a lot of efforts already done in biomedical text mining. Efforts in this field include work on part-of-speech tagging, parsing, term variations, word sense and acronyms disambiguation, negation analysis, classification and sentiment analysis, information retrieval, information extraction and question answering.

2.2.2 Part-of-speech tagging

Accurate and reliable part-of-speech (POS) tagging is useful for many Natural Language Processing (NLP) tasks such as syntactic parsing, feature extraction for classification, semantic representation and forms the foundation of NLP-based approaches to information retrieval and data mining (Coden et al., 2005). Despite the fact that there are several high precision statistical POS taggers for general English language, they are not good enough for specific domains such as medical domain.

Coden et al. (2005) showed that a POS tagger trained on annotated domain-specific corpus in addition to a general English corpus boosts the tagger performance by 5-10% on medical documents. Also, adding small 500-word domain-specific lexicon additionally boosts performance of POS tagger by about 2%.

Hahn and Wermter (2004) compared a rule-based POS-tagger (Brill tagger) and statistical tagger (TnT) trained on general German language on clinical data. The statistical tagger, TnT trained on general texts performed almost as good as state of the art in the medical domain, which contradicts findings of Coden et al. (2005).

Lexical analysis of clinical texts is much harder then general biomedical literature. Clinical texts are ungrammatical and composed of short, telegraphic phrases (Meystre et al., 2008).

Tsuruoka et al. (2005) developed tagger that was trained on Wall Street Journal, GENIA corpus and PenBioIE corpora. Their tagger is based on cyclic dependency network. They used maximum entropy modelling as the local probabilistic classifier which gave the probability distribution for the tags on each token. The problem of high computational cost of the cyclic dependency network is solved by generating the candidate tags on each word using zero-th order probability given by the local classifier trained without the information about the adjacent tags. This method gave them considerable speed-up and little loss of tagging accuracy. The experiments showed that adding training data from different domain does not negatively affect the performance of the tagger. Their tagger archived very high precision (97% to 98%) on all the mentioned corpora. This tagger is one of the best POS taggers for the biomedical literature and it is commonly known as GENIA Tagger.

2.2.3 Parsing

Campbell and Johnson (2002) argues that the syntactic relationships are more well-defined and have less variation in scientific languages (sublanguages), such as the ones used in medical texts. Identifying word classes using syntactic relationships should be simpler and potentially more useful in these types of languages. Clinical text is frequently include telegraphic omissions, run-on structures, improper use of conjunctions, left attaching noun modifiers, etc. In many cases, many traditional phrase structures are absent or altered, making a phrase structure parse using traditional production rules difficult. A dependency grammar may still capture useful syntactic relationships when an accurate phrase grammar parse is not possible. Dependency grammars generate parses where words in a sentence are related directly to the word, which is its syntactic head. Dependency grammars may allow capitalization on the relative syntactic simplicity of medical language without the overhead of generating and identifying structures which will not be used. Authors applied a Transformational Based Learning algorithm to learn a dependency grammar for medical texts (algorithm was, also, used in Brill tagger). They presented 77% accuracy with only 830 sentences in training data.

SymText system combines a syntactic parser using augmented transition networks and transformational grammars with a model of semantics based on the Bayesian network statistical formalism (Haug et al., 1995). Leaf nodes in the Bayesian network provide placeholders for individual words or phrases from the x-ray report. The intermediate and root nodes are associated with slots for higher level concepts. Bayesian network structure is used to indicate the relationship between the words from the sentence with the concepts associated with these words. The Bayesian networks are a probabilistic inference engine. The probability of each possible value of a node is conditioned on the probabilities of neighbouring nodes (Christensen et al., 2002). Christensen et al. (2002) developed a successor of SymText, system called MPLUS, which differs from SymText in the size and modularity of its semantic Bayesian networks.

Taira and Soderland (1999) created natural language processor based on resonance probabilities between word pairs for parsing and semantic analysis of radiology domain medical reports. The parser has no hand coded rules, but rather gathers word affinity knowledge from training sentences.

Recall and precision reached a mid 80s percentage from a little over one hundred training sentences and reached recall 90% at precision 89% by one thousand training sentences.

2.2.4 Named entity recoginition

Medical name entity recognition is the identification of medical related concepts and name entities in unstructured clinical data such as Electronic Health Records (EHRs), patient discharged summaries and clinical notes and their classification into predefined categories (Cohen and Hersh, 2005) (Suakkaphong et al., 2011).

The approaches to the medical entity recognition could be grouped into three categories:

- **Knowledge based approaches** include rule and dictionary-based strategies to detect text features in order to be able to predict semantic categories of a given term candidate.

- **Machine learning approaches** include the use of machine learning algorithms in order to predict semantic classes of the term candidates. For medical named entity recognition are reported uses of supervised and semi-supervised algorithms.

- **Hybrid approaches** include some combination of the previous two approaches. These approaches are rare in medical NER. Often these system consist of machine learning approach and a rule-based post processing.

Dictionary-based method for named entity recognition consist of strict and lenient textual string matching. Dictionaries and domain knowledge resources contribute with domain specific knowledge in form of lexical and terminological coverage. The resources used as dictionaries for biomedical NER are SNOMED, ICD-10, UMLS Methatheasaurus(Aronson, 2001) or MED (Friedman et al., 1994).

Friedman et al. (1994) created a system called MedLEE, which was able to detect and map medical terms from clinical narratives. Their first application was in radiology field, but system was capable of extension to other fields (Friedman et al., 1995). To be able to detect and map clinical terms to a controlled vocabulary, MedLEE used a three component architecture. It contained a **lexicon**, which identifies and categorizes single-word and multi-word phrases that occur in the text, a **parser**, that used extended context-free semantic grammar to delineate semantic relations and structures and the **encoder**, which establishes correspondences between textual terms and controlled vocabulary terms. The system was also regularizing found phrases by maintaining a set of mappings representing their compositional structures. If the compositional structure was found it was replaced by target phrase. As a knowledge base system used the Medical Entities Dictionary (MED).

Gerner et al. (2010) described an open-source species name recognition and normalization software system, LINNAEUS. Biodiversity-oriented methods typically use rule-based approaches that rely on the structure of binomial nomenclature for species names adopted by Carl Linnaeus by whom system got its name. Using the NCBI (National Center for Biotechnology Information) taxonomy and a custom set of species synonyms, they created species dictionaries optimized for time-effective document tagging. There can be observed ambiguities in their dictionary, where the same term can refer to several different species. Also, acronyms listed for species in the NCBI taxonomy are not always exact and unambiguous. To overcome this problem, they used Acromine to query all listed acronyms in the NCBI taxonomy in order to retrieve frequency counts for the various expanded forms that the acronyms appear within MEDLINE. For mentions that remain ambiguous and where one of the possible candidate species is mentioned explicitly elsewhere in the text, all occurrences of the ambiguous term are resolved to refer to the explicitly mentioned species.

Pure rule based methods include the adoption of contextual and linguistic features to identify entity occurrences in text. They are quite rare and predominantly concerned on medication name extraction.

Spasić et al. (2010) described a system that in addition to manually curated dictionaries of medication terms, used a morphological features of medication names such as affixes (-cycline, -nazole, -sulfa, -statin) that are an indication of medication names. Similarly, Yang (2010) used a number of crafted lexicon including medication, dosage, frequency, duration, etc., which were used as a part of term-based rule matching strategy. In addition he applied a number of heuristics to cope with abbreviations, synonyms and spelling variations. Spasic's et al. and Yung archived F-measure of 83.8% and 85.8% respectively. For the comparison, a state-of-the-art machine learning approach for medication extraction archived 89.8% (Halgrim et al., 2011).

Supervised machine-learning approaches use labelled data to train statistical models to identify term occurences. Machine learning approach to named entity recognition typically model task as a label sequence problem. Conditional random fields

(CRF)is the state-of-the-art sequence labelling algorithm, which have been successfully applied to the biomedical named entity recognition. Typically CRF for named entity recognition uses lexical (tokens, n-grams, bag-of-words), morphological (lemma, suffices, prefixes), orthographic (word-class, is word alphanumeric, numeric or word), syntactic (part-of-speech tags, shallow chunks), semantic (class in terminological resources)(Leaman et al., 2008) and feature windows (Settles, 2004).

More recent medical NER approaches have focused of semi-supervised strategies who use both labelled and unlabelled data. This enables deriving successful models while relying on less data. de Bruijn et al. (2011) presented a method that used semi-supervised approach consisted of generating hierarchical word clusters based on contextual similarity. Similarly, Jonnalagadda et al. (2012) applied semi-supervised strategy for clinical concept extraction, which was generating word features derived from words that appear in similar context. Both of these approaches could be used to compensate for limited vocabulary observed in a smaller set of annotated data. Suakkaphong et al. (2011) combined conditional random fields and two supervised learning algorithms, self-training (bootstrapping) and co-training (feature sampling) to recognize disease names from biomedical literature. Firstly, he trained a classifier with a traditional CRF with a set of common features (lexical, syntactic and semantic) and used it to compare with other two semi-supervised methods. Bootstrapping is an iterative learning process in which a model is trained on a small initial label set. Then, the classifier is used on unlabelled data data to get self-labelled data. Most confident self-labelled data points are added to the set of labelled data and the classifier is retrained. This procedure is repeated until optimal outcome is reached. Feature sampling is an iterative learning process in which randomly generated sets of features, called views are generated. Co-training first learns a separate classifier for each view using any labelled examples. Than they classify unlabelled data with each classifier and using voting system that is based on the most confident guess select self-labelled data that are used for retraining.

2.2.5 Negation detection

Research focused explicitly on negation detection in biomedical literature started only a few years ago with NegEx (Meystre et al., 2008). Chapman et al. (2001) developed simple algorithm using regular expressions that can detect negated findings in the narrative reports. They identified 35 negation phrases that was divided into two groups.

1. Pseudo-negation phrases, consists of phrases that appear to indicate negation, but instead identify double negatives (not ruled out), modified meanings (gram-negative), and ambiguous phrasing (unremarkable)

2. Second group consists of phrases they believed are used to deny findings and diseases when used in one of two regular expressions. In the first regular expression the negation phrase precedes the UMLS term. In the second the negation phrase follows the UMLS term.

More formally, they may be represented as:

$$<negationterm> * <UMLSterm> \tag{2.4}$$

$$<UMLSterm> * <negationterm> \tag{2.5}$$

In both expressions asterisk indicates that up to five tokens may be placed between the negation phrase and the UMLS term.

NegEx had a specificity of 94.5%, a precision of 84.5% while maintaining a reasonable recall of 77.8%.

Chapman et al. (2007) made an extension to NegEx algorithm that was able to detect negations (negated, affirmed classes), temporality (with recent, historical and hypothetical classes) and experiencer (patient or other). Each of these three contextual features has a unique set of trigger and pseudo-trigger terms used for detection using regular expressions.

Mutalik et al. (2001) built a program Negfinder, which also used indexed concepts using UMLS and regular expression, but added a lexical scanner (lexer) and parser to their processing. The lexer identifies a very large number of negation signals and classifies them on the basis of properties such as whether they generally precede or succeed the concept they negate and whether they can negate multiple concepts. Each class generates a single token that is passed to the parser. The Negfinder lexer often returns a single token that corresponds to several possible combinations of words. Thus, combinations of is/was/were/are/been followed by an optional adverb followed by denied/refused/omitted/lacking/excluded will generate a single token that is passed to the parser. For this purpose, the lexer makes extensive use of regular expressions in addition to using limited part-of-speech information. Parser is used to assemble concepts into a list, to associate a concept or a list of concepts with a negative phrase that precedes or follows it, and to accurately determine where the negation starts and ends.

Sarafraz and Nenadic (2010) experimented with SVM with the command relation features to identify negated events in biomedical literature. Command concept was introduced to determine the scope within a sentence affected by an element. If the lowest ancestor of one element of the constituency parse tree is also an ancestor of the other element with some label X, than the first element X-commands the second one. They assumed that the events have been identified. They proposed a set of features for SVM that were mainly engineered from a sentence parse tree with lexical cues (negation cue, POS tags of negation cue and event trigger, the parse node type of the lowest common ancestor, whether negation cue commands event trigger or the participants, etc.). They reported F-measure of 63% for the detection of molecular events from BioNLP'09 Shared Task data.

2.2.6 Word sense disambiguation and term variation resolution

Many terms in the biomedical domain can be ambiguous, especially acronyms. There can be same acronym for more than one term. For tasks such as information extraction, entity recognition and question answering it is important to achieve unambiguous understanding of word senses.

Word Sense Disambiguation (WSD) is the process of understanding which sense of a word (from a set of candidates) is being used in a particular context (Meystre et al., 2008).

Liu et al. (2001) presented explanatory background review on general English and biomedical word sense disambiguation. Their approach was to use UMLS Metathesaurus to create lists of concepts with same meaning (using unique concept identifier CUI). Using this list, they annotated MEDLINE abstracts and The New York Presbyterian Hospital Clinical Data Repository entries with a unique concept on each occurrence of words from previously created lists of words with same concept meaning. At the end they used supervised machine learning with different window size on these sentences to train disambiguation classifiers. They used Naive Bayes and Decision List algorithms and reported the best accuracy for the Naive Bayes algorithm which worked on stemmed words and window size of 10 (accuracy 97%).

Nenadić et al. (2004) considered five types of term variation (orthographic, morphological, lexical, structural and acronyms and abbreviations). Their approach for automatic term recognition is to recognize term candidates by a set of linguistic filters and termhood assignment based on joint frequency of occurrence of all term variants.

They showed that automatic term recognition can be improved by incorporating treatment of term variation.

Also, Liu et al. (2004) showed that supervised machine learning approaches to word sense disambiguation is suitable only if there are enough training data with at least few dozens of sense-tagged instances for each sense. They also implemented four machine learning algorithms (Naive Bayes, two types of Decision Lists and hybrid approach using Naive Bayes and Instance-based learning combined) using different window sizes. They reported that Naive Bayes and their mixed machine learning algorithm performed the best for all the data sets.

Pakhomov et al. (2005) tried to focus on clinical discourse found in the clinical notes repository of Mayo Clinic. They tried to apply both supervised (Maximum Entropy and C5.0 Decision Trees) and semi-supervised approaches (sense inventory and context vector generation with C5.0 Decision Trees or cosine similarity of vectors). For unsupervised learning, they gathered data for each acronym sense context by matching the context with resources such as World Wide Web, MEDLINE abstracts and Mayo Clinic corpus and recording the surrounding lexical items within specified window. Their results are encouraging as they suggest that there is a potential in leveraging very large amounts of publicly available data for disambiguating acronyms found in clinical discourse.

2.2.7 Information Extraction

Information extraction is one of the most complex tasks in natural language processing using all the mentioned task as its parts. Because of its complexity, current approaches are focused on narrowly restricted domains. Similarly to other complex text mining tasks, approaches may be knowledge-driven, rule-based, machine learning and hybrid. Applications of information extraction could be extracting demographical information about patients, information about clinical trial or extracting interaction between proteins, genes and drugs.

Since there is no standardized annotation guidelines, Chapman and Dowling (2006) presented an annotation schema to manually annotate clinical conditions by applying the sociological tradition of grounded theory. The schema was developed based on 40 emergency department reports and tested on 20 such reports. Their schema was created in iterative process of annotating clinical reports by authors and extensive discussion about variables that should be included in annotation schema.

Cimino et al. (2007) created a system that is extracting both coded and narrative records from New York Presbyterian Hospital databases about patients. The system is extracting information from twelve sources from two clinical information systems and assembling them into a chronological sequence of medication history, plans, and orders that correspond to periods before, during and after a hospital admission. The system is using MedLEE for parsing and extracting medical terms in unstructured narratives and to return concept unique identifiers from the UMLS.

Hansen et al. (2008) built an information extraction system that extracted the number of trail participants from abstracts of Randomized Controlled Trials (RCT). They noticed that number of trial participants might be reduced during the progress of the study. Some of the participants might be excluded or rendered ineligible through specific criteria. The participants are allocated to one or more arms of the study in which they are treated with one or more interventions. In many cases the number of participants that are evaluable at the end of the study will be a subset of those who were allocated. Hansen et al. approach was to process only sentences that contained number. Also number should not be followed by unit that was defined in a dictionary. They used the SVM algorithm for classification and set of features including POS tags of previous and following words, words before or after, scale of integer etc. When all the numbers are classified, the number of trial participants is chosen as the one with the maximum number among all the positively classified numbers. Since during the trial some participants could be excluded, this number would not always give the correct result. However, they claimed accuracy 97% and 84% F-measure. Their corpus was quite small containing 148 abstracts for training data and 75 abstracts as test data.

Clinical trials are the most important sources of evidence for guiding evidence-based practice and the design of new trials. Kiritchenko et al. (2010) built an information extraction system called ExaCT that extracts key trial characteristics (e.g., eligibility criteria, sample size, drug dosage, primary outcomes) from free text reports. ExaCT consists of two parts: an information extraction (IE) engine that searches the article for text fragments that best describe the trial characteristics, and a web browser-based user interface that allows human reviewers to assess and modify the suggested selections. Information extraction engine uses a statistical text classifier to identify sentences containing some key trial characteristic. Algorithm used for statistical text classifier is SVM, which was trained on annotated data. Since most of the key trial characteristics are just segments, weak regular expression rules were manually crafted to extract various types of information (e.g. Dates, drug dosage, sample size...). For eligibility criteria whole sentence is considered. The system was returning five top rated items for each key trial characteristic. These items was shown to human curators using web browser-based user interface, so they can select the correct item or make some manual changes.

Katrenko and Adriaans (2007) were extracting protein-protein interactions (PPI) using dependency parsing and several machine learning algorithms (Naive Bayes, BayesNet and K nearest neighbors classifiers). For syntactic analysis, three parsers were used – LinkParser, Minipar and Charniak parser. They reported highest F-measure of 72.7% with the combination of classifiers on AImed corpus.

Segura-Bedmar et al. (2011) experimented on extracting drug-drug interaction (DDI) using SVM with shallow linguistic kernel. In their preprocessing phase they used MetaMap to recognize names of the drugs. Since there was no annotated corpus for DDI, they have created DrugDDI corpus. They modelled the task as classification task. As the features they used n-grams and neighbouring words, varying these parameter (in range from 1 to 3).The highest F-measure of 59.64% they reported with n-gram size 3 and a window-size of either 1 or 3.

GATE[4] and Minor Third[5]http://sourceforge.net/projects/minorthird/ are examples of systems that provide an information extraction framework. GATE is an open source rule-based framework and graphical development environment that enables information extraction by allowing users to develop and deploy language engineering components and resources in a robust fashion. More importantly, it enables user to write rules for a rule or knowledge driven information extraction system (Cunningham et al., 2002). Minor third is a machine learning-based framework for information extraction from text with a range of algorithms included and integrated with the visualization tools for manual and automatic annotation of text.

2.2.7.1 Question-answering

Question answering systems are computational systems that utilizes information extraction systems to answer questions posed by humans in natural language. Niu and Hirst (2004) created a question answering system that is able to answer question about clinical-evidence texts. Their system accepts keyword queries in the **PICO** format. In this format, a clinical question is represented by set of four fields that correspond to the basic element of the question:

P - a description of the patient (or the problem)

[4] `https://gate.ac.uk/overview.html`

[5] `\unskip\penalty\@M\vrulewidth\z@height\z@depth\dpff`

I - an intervention

C - a comparison or control intervention (may be omitted)

O - the clinical outcome

PICO elements correspond to three semantic classes: DISEASE (medical problem of the patient), INTERVENTION (medication applied to the disease) and the CLINICAL OUTCOME. They together constitute a SCENARIO of treatment. Similarly, a diagnosis scenario often includes SYMPTOMS, TESTING PROCEDURE, and HYPOTHESIZED DISEASES.

Niu and Hirst (2004) used MetaMap to separate input sentences into phrases, to identify the medical concepts embedded in the phrases, and to assign proper semantic categories to them according to the knowledge in UMLS. More than one semantic category in UMLS may correspond to MEDICATION or DISEASE. For example, either a PHARMACOLOGIC SUBSTANCE or a THERAPEUTIC OR PREVENTIVE PROCEDURE can be a MEDICATION. Also, DISEASE OR SYNDROME or a PATHOLOGIC FUNCTION can be a DISEASE. They used some training text to map UMLS categories into the either DISEASE or MEDICATION classes. For processing they again used MetaMap. For outcome detection they identified words that are often parts of an outcome. Sentences with that word was tagged for part-of-speech with Apple Pie parser. They identified that for the noun cues, the noun phrase that contains the noun will be part of the outcome; for the verb cue words, the verb and its object together constitute one portion of the outcome; for the adjective cue words, often the corresponding adjective phrase or the noun phrase belongs to the outcome. A limitation of this approach is that some connections between different portions of an outcome may be missing. To be able to answer questions system needs to know how different entities in the same semantic class are connected and what relations hold between different classes. Again, they identified group of cue words and symbols that suggest particular relationship between classes inside one sentence. Six different relationships were identified (comparison, alternative, combination, specification, substitute, preference) for relationships within classes, and it is identified that there is only cause-effect relationship between classes. Since most clinical outcome are either positive or negative, Nui and Hirst build a binary SVM sentiment classifier using SVMling package. Unfortunately, Nui and Hirst have not included overall evaluation of question answering system. They claim that they will continue work on identification of relationships within and between semantic classes.

Demner-Fushman and Lin (2007) built a more complex tool that was using several classifiers and UMLS knowledge base to detect same semantic classes as Niu and Hirst (2004). They also used as input PICO format, and for detecting patient and disease UMLS knowledge. For detecting outcome, they used combined method of cue words and several Naive Bayes classifier (returning probabilities that in different parts of abstract could be found outcome). To rank their answers they scored strength of evidence based on MEDLINE metadata and publication type. However, they built system that is able to retrieve all documents from PubMed search and score it based on PICO query and strength of evidence. This approach results in high quality information retrieval and it is might be moving towards question answering in future work.

2.3 Table mining

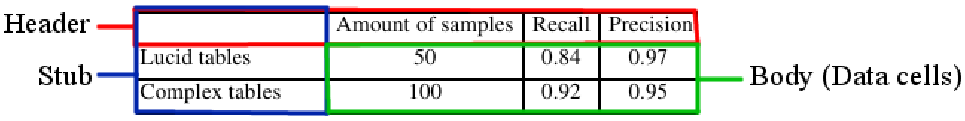
2.3.1 Introduction

Most of current text mining efforts are focused on the extraction of information from the main body of scientific articles. However, a significant part of information presented in literature is shown in figures and tables.

Documents often contain tables in order to communicate densely packed, multi-dimensional, detailed information. Tables do this by employing visual layout patterns to efficiently indicate fields and records in two-dimensional form. Their rich combination of formatting and content presents difficulties for traditional text mining techniques. The ability to find tables and extract information from them is a necessary component of many information retrieval and information extraction systems (Wei et al., 2006).

Tables are the prevalent means of representing and communicating structured data. They may contain words, numbers, formulae, and even graphics. Developed originally in the days of printed or handwritten documents, they have been adapted to word processors and mark-up languages, and form the underlying paradigm for spreadsheets and relational database systems (Embley et al., 2006).

To be able to efficiently deal with the tables we will need first to introduce what actually tables are and some basic terminology used in tables. 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 convenience of study, reference, or calculation. Table in scientific literature often have following elements: title, caption, cells, column, row, header, sub-header, body and stub.


Header
Stub
Body (Data cells)

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

Figure 2.6: Table elements: Header, stub and body

Title describes what the table is about and usually is placed above the table (in some cases below the table).

Caption gives more detailed information about table and usually is placed below the table. In caption can be given the legend for symbols used in table or some observations about table data.

Cell is the basic grouping within a table. One cell usually contains only one value, word, phrase or concept. Often, cells 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.

Header is usually top-most row (or set of several top-most rows) of a table and defines what the column data are.

Dimension is a structure that describes or categorize the data.

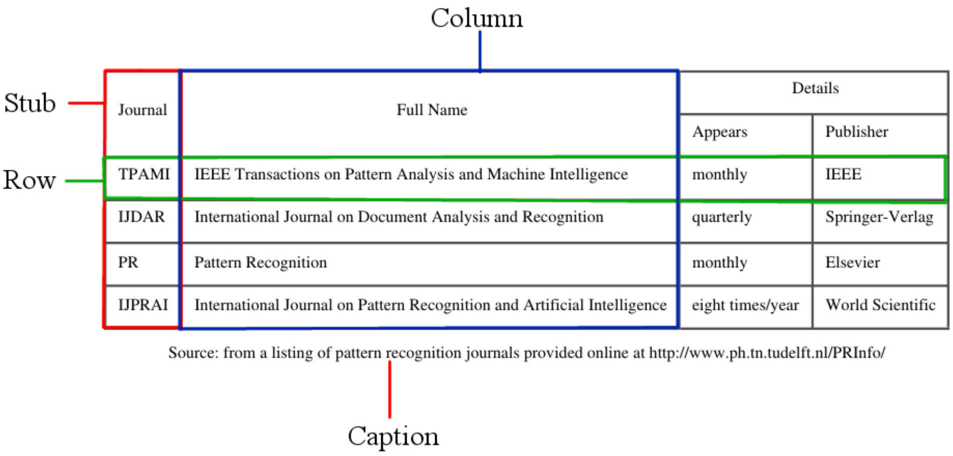
Sub-header creates an additional dimension of the table. Sub-header row is usually placed between data rows, separating them by some dimension or concept.

The stub is the left-most column of the table, usually containing the list of subjects or instances to which the values in the table body apply. The stub column is the only column that may not require a column head.

Body of the table contains actual table's data. Data cells are placed in the body of the table. Cells in the body represent value of things or value of relationship defined in headers, sub-headers and stub.

Example of tables with marked mentioned elements can be seen in Figures 2.6,2.7 and 2.8

Automated system that processes tables in documents have to solve several challenges. The first challenge is to detect tables. In many types of documents this task can be challenging (free text or HTML). The recognition of tables in documents is either a research goal in its own right, or the first step in an integrated system such as those created for IE or IR (Hurst, 2000). The other challenge is to determine which cells are



Journal	Full Name	Details	
		Appears	Publisher
TPAMI	IEEE Transactions on Pattern Analysis and Machine Intelligence	monthly	IEEE
IJDAR	International Journal on Document Analysis and Recognition	quarterly	Springer-Verlag
PR	Pattern Recognition	monthly	Elsevier
IJPRAI	International Journal on Pattern Recognition and Artificial Intelligence	eight times/year	World Scientific

Source: from a listing of pattern recognition journals provided online at http://www.ph.tn.tudelft.nl/PRInfo/

Figure 2.7: Table elements: Stub, Row, Column and Caption

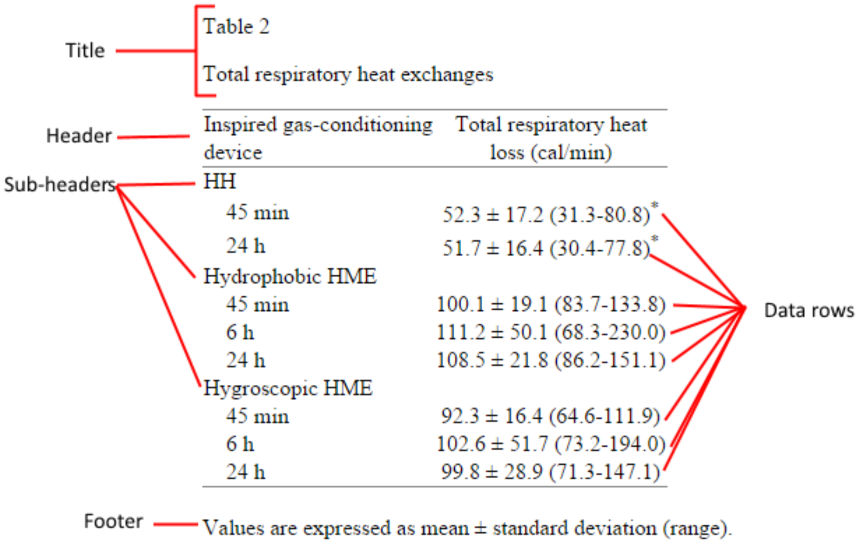


Table 2

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).

Figure 2.8: Elements of complex table

parts of header and which cells contain data. This task could be especially challenging in complex tables where some cells spans over several rows or columns. The third task in the table processing is table understanding. In this task, system should be able to correctly extract information by taking in account structure and semantic of the processed table. In this part we will examine related work in the field trying to solve these challenges.

2.3.2 Ergonomics of tables

Taber's Cyclopedic Medical Dictionary defines ergonomics as ”the science concerned with fitting a job to a person's anatomical, physiological and psychological characteristics in a way that enhances human efficiency and well being” (Venes, 2013). It is an interdisciplinary field that joins together psychology, physiology, medicine and engineering. Ergonomics examines the ease of use of things and devices humans are using.

Patricia Wright did a notable work on ergonomics of tables. She was examining ease of use of different types and layouts of tables. She showed that tables can be quite hard for humans to read and that easiest tables are explicit list tables, containing explicitly all the data one is looking for. List tables she defined as one dimensional tables. The easiest ones were key-value list tables. If tested person had to look for more values and somehow co-relate them (for example sum the values), mistakes were more probable and reading was slower. If the values are not explicit and person has to do additional cognitive processes to find the value, speed and probability of mistakes increase. She showed that by adding dimension possibility of mistakes and the speed of reading increase. Also, many tested people was not able to understand how to use two or multi-dimensional tablesWright (1977). However, it decreases with training and learning (Wright, 1968).

She defines explicit tables as a table in which all the information is presented in an explicit way. She defines implicit table as a table in which user has to do something more then just to look for an item. In term of ease of use, she shows that explicit tables are much easier to read. Her experiments showed that making columns within a table readably distinguishable does aid the user, but it should not be made too much so it becomes unreadable. One set of her experiments showed that abbreviations and acronyms in cells of table are hindrance and they make table harder to read. They may be helpful for item identification in case where there is no other distinguishing clues between columns (i.e. tables without header). One of the conducted experiments try to examine if it is easier to read horizontally or vertically arranged table. The results of this experiment showed that reading was faster and errors were fewer with the vertical table. One experiment also showed that it is easier and faster to read the table which has item that is looked for in the left column, while the unknown searched value on the right than vice versa (Wright and Fox, 1970). While reading tables user have to make some decision on how to find the searched value.

Generally, Wirght's research shows that tables are quite hard data structures for humans and the more cognitive operation user has to perform while reading it, he will be slower and more error prone.

2.3.3 Table models

Hurst (2000) did probably one of the first extensive studies of tables, table understanding and information extraction from tables in his PhD thesis. Hurst argue that tables have not received much attention from information extraction and information retrieval communities, despite the fact that there has been considerable body of work in image analysis field, psychological and educational research, document markup and formatting research. This is possibly due to lack of art and model, no available corpora and confusing markup. He also claims that the domain independent processing of tables can be taken as far as identifying the linguistic objects (e.g. CITY, Houston) which stand in some relation to each other (e.g. instance_of, price_of, etc.), but cannot necessarily identify what those relationships might be. However, domain specific processing could use some available resource for that domain that can enable identification of relationship. Hurst proposed a model of table, which has five components: Graphical (a basic graphical representation of the table, e.g. bitmap), Physical (a description of the table in terms of the physical relationships between its basic elements when rendered on page), Functional (the purpose of areas of the table with respect to the use of the table by the reader), Structural (the organization of cell as an indication of the relationships between them), Semantic (the meaning of meta text in the cell, object text in the cell, the relationship between the interpretations of cell contents, the meaning of structure in the table). In the functional component of the table model, he defined two types of cells - access cells (navigational cells such as headers and stubs) and data cells (containing table data). On the physical level can be observed some of the heuristics which can distinguish data cells from the access cells, such as font type, font color, cell background color, cell spanning, alignment, line art, etc. Hurst defines **reading path** as a path which reader takes through the array of cells when using the table to locate or read a particular piece of information. To be able to create an information extraction system, problem of human-like finding reading path have to be solved. In semantic component of the model, relationships between cells on the reading path are considered. This information cannot be read from the table explicitly. Some assumptions can be made, but there could be many possible combinations of possible relationships, and some can be ambiguous. Hurst proposed to build maximum dependency sets of dependent cells read from the table.

Hurst describes his table mining tool called TabPro, which is able to mine tables from text (ASCII), HTML and LaTeX documents. Input document is preprocessed using tokenization, chunking (discovering noun groups and verb groups), crystal detection (outputs sub-strings which are marked as of a certain semantic type, such as dates, units of measure, etc.), detection of sentences referring to tables either explicitly or implicitly. It is not clear from the thesis, how these sentences were used. The functional module determines, for each cell in the table, if it is a member of access area or data area or neither using a Naive Bayes classifier. There is also a functional module that uses pattern recognition and heuristics. The structural module for each cell in the table determines the set of cells which can be reached from it via the navigation of the table. It uses functional, physical cues, heuristics and WordNet to suggest relationships between cells arranged in the table columns. The semantic module extracts cell content and calculates relationships between each two cells. For the structural analysis for unseen data, Hurst claims result of 54% precision and recall. The combined task of functional and structural analysis scored 89.42% precision and 82.15% recall, though for that evaluation, the data was not unseen. The main contribution of Hurst's work were defined table model and components of table analysis.

Wang and Wood (1995) tried to define an abstract table model for editing and formatting using mathematical and logical representation and operators. Their model is presentation independent because it was constructed by keeping the logical structure of multi-dimensional tables and excluding any topological and typographic attributes. They defined table item in his model as an a node in the tree of a labels, identified by the sequence of labels on the path from the root to the node. By this definition data items would always be on leaf nodes of tree, while the other nodes would be labels (Wang and Wood, 1993). This model was primarily built for table formatting and editing of tables manually, but it can also facilitate table mining as it contains semantics of a table in its representation.

Zanibbi et al. (2004) wrote a survey of table recognition and systematized table detection and structure recognition as sequence of three basic operations: observations, transformations and inferences. Observations include feature measurement and data lookup, transformations are operation that alter or restructure data, and inferences generate and test hypotheses. They describe techniques used in these steps such as geometry, layout, histograms, textures, edit distance, cell cohesion measures, probabilities (observations), compression, re-sampling, binarization, mathematical morphology, tree and graph transformation used to merge or split regions, filtering small objects, histogram smoothing, tokenization, stemming (transformations), classification, segmentation, parsing (inferences).

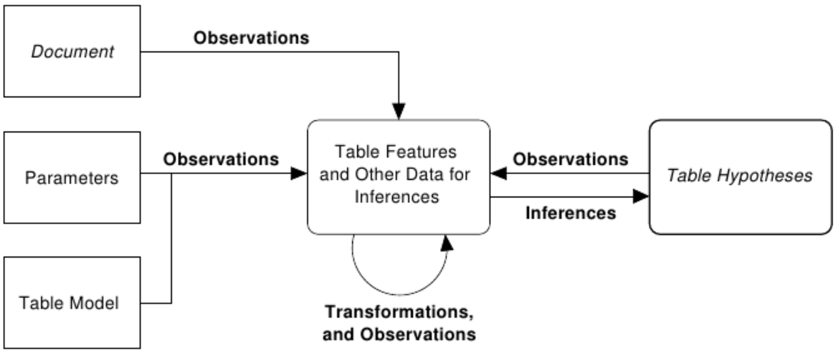


Figure 2.9: Table recognition process. Model proposed by Zanibbi et al. (2004)

2.3.4 Table detection

In table processing, the first challenge is to detect tables in documents. PDF documents are hard even for reading text, and most commonly used technique for retrieving data from PDF files is Optical Character Recognition (OCR). The HTLM document is easier to read and even detect tables because of the <table>tag. Unfortunately, not all tables that start with <table>tags, are genuine tables that contain the structured data. Some of these tables are used for HTML page formatting. In free text documents, tables are structured by using empty spaces or set of special characters. In this kind of document doesn't exist clear indication on what part of the document is table and what part of it is text.

2.3.4.1 Table detection in PDF documents

Kieninger and Strieder (1999) worked on recognition of tables using Optical Character Recognition (OCR). Their system T-Recs was able to read both ASCII or paper scanned documents. Their approach was to detect word-block clusters and to look for the words that belongs to the same logical unit. They look for the horizontally overlapping words and determines if they form rows of the tabular structure. There are more works that uses OCR to recognize tables using white space density graphs or by detecting line structures in documents. However, optical recognition of table is out of scope of this work, and would not be investigated in more detail.

Yildiz et al. (2005) built a tool named pdf2table. This tool is using pdf2html tool, which returns PDF elements in XML with position of these elements, and extracts tables. Extracting tables from PDF documents is done by using heuristic about content positions. Because of the complexity of the task and the used heuristics, which cannot cover all possible table structures, one cannot assume that the approach always returns correct results. To overcome these limitations the authors created graphical user interface which gives the user the ability of making adjustments on the extracted data. Pdf2table is freely available on author's website.

Similarly, Constantin et al. (2013) build a rule-based system designed to reconstruct the logical structure of scholarly articles in PDF form. The system output is an XML document that describes logical structure in terms of title, sections, figures, tables, references, etc. The system carries two-stage process in which, firstly, system constructs a geometrical model of the article's content to determine spatial organization of textual and graphical elements and then identify logical units of discourse based on their discriminative features (font styles, text positioning, lists of cue words, contextual features, etc.) (Constantin, 2014). Their system is quite successful in recognizing tables, but the transformation of the table into the XML format faces a number of challenges. They reported F-measure of 74.03% for extracting individual bibliographical items. However, for table extraction and transformation they reported F-measure of 57% on Luong et al. (2010), 28.78% on Elsevier and 13.27% on PMC dataset.

2.3.4.2 Table detection in text (ASCII) documents

Ng et al. (1999) built a system that was capable of recognizing tables in free text (ASCII documents). They used 2 machine learning algorithms C4,5 decision trees and back-propagation algorithm for artificial neural networks with 9 features for boundaries detection. Each feature has its representation for previous, current and next horizontal line. So, each training sentence consist of a set of 27 feature values. Their system could also detect if some vertical line is first line of a column, within column, last line of column, or outside any column. Similarly for the horizontal line and rows. Their features include number of white spaces in line, number of leading spaces, number of segments with more contiguous space characters, whether cell contains special characters, alphanumeric characters, etc. They claimed accuracy of 85% – 95% for the learning algorithms where no one learning algorithm clearly outperforms the other. C4.5 was giving higher accuracy on recognizing table boundaries and column, and back-propagation performing better at recognizing table rows.

Pande (2002) described an approach of using statistical cross-correlation in combination with heuristics to detect tables in plain text documents (ASCII). Cross-correlation is a standard statistical signal detection procedure that is useful for determining the similarity of two signals. He explored the cross-correlation concept as a way of computing similarity measures between lines of plain text or between aggregates of lines. His experiments showed that for perfectly aligned tables, the table entries have very high correlation values $(C^{f}_{fg}(0) > 0.7)$, but to minimize errors, he set threshold at 0.4. Although they integrated table detection system with the Odessa Digital library, there is no mention about quality of results archived with this approach.

e Silva et al. (2003) described a system for detecting tables in PDF and ASCII documents. In this work, e Silva et al. build a form of system described by Ng et al. (1999). They analyzed a set of Portuguese companies' financial statements using statistical and data mining techniques. Their system was labelling lines of text if they belong to a table. PDF files were converted to ASCII using pdftotext Linux utility. They used number of each line's inner spaces as feature for C4.5 decision tree. They used data mining suite Weka, and its J48 (which is C4.5 decision tree). One extra characteristic has to be taken into account: no matter how many inner spaces a line has, it is not in a table if the lines around it are plain text; and a line with no inner spaces is not plain text, if table lines surround it. They had to set threshold of percentage of empty spaces in a line. They reported recall of 99,4%, but precision or F-measure were not reported.

2.3.4.3 Table detection in XML and HTML documents

Wang and Hu (2002) used SVM (Support Vector Machines) and decision trees machine learning algorithm to detect genuine tables in the HTML documents. Genuine tables are tables that contain useful information. The other type of tables used in HTML are layout tables, which are used for document formatting. The features Wang and Hu used were the average number of columns, computed as the average number of cells per row, standard deviation of number of columns, average number of rows, standard deviation of number of rows, average cell length, standard deviation of cell length, average length consistency. They, also applied content type features. They defined 7 content types: Image, Form, Hiperlink, Alphabetical, Digit, Empty and Others. Their features for content type included histogram of content type for a given table and average content type consistency. The third group of features they applied are word group features where they calculated TF-IDF measure with particular adjustments. For SVM classifier they used SVMlight and two types of kernel function - linear and radial basis function (RBF). They showed that SVM with radial base function kernel in their case outperformed SVM with linear kernel. Also, SVM with radial base kernel performed slightly better than decision tree classifier.

Gatterbauer et al. (2007) argued that it is hard to detect all the tables from web documents by analysing tree like HTML structure. They note that not all tables in HTML documents are inside <table>tags. They focused on analysing tables using visual representation similar to ones that browsers uses to represent tables (taking in account CSS and how people view the table) using CSS2 visual box. They made an assumption that table creates frames of visual element nodes (rectangle visual nodes). Since algorithm have to render page, it takes in average 5 seconds per page to detect frames that form table. Speed of this approach puts this algorithm at a significant disadvantage to the other approaches that work without rendering page. They reported 81% recall and 68% precision for table detection. For table interpretation they reported 57% recall and 48% precision.

Son et al. (2008) presented approach to distinguish meaningful (genuine) tables in web documents from decorative ones by using SVM algorithm that can learn both structural and content features. For structural features, they used parse tree kernel (case of convolutional kernel). For the content features, they used linear kernel. The classifiers were doing their classification in parallel. After the classification, the results were summed using importance factor, which determines relative importance between the classifiers. They got best results with factor 0.7, which means that structural features are more important. With importance factor of 0.7 they reported 97.51% F measure.

2.3.5 Functional table analysis

The second step towards the information extraction from tables is functional analysis of cells and table. During this step, the data area should be distinguished from the area of headers and access cells. This is quite hard task, especially when dealing with complex tables that can have several subheads and multi-row headers. Even in simple tables, determining the header can be ambiguous task.

Most of the approaches to this task consider using machine learning classification techniques and mainly C4.5 decision trees algorithm. However, there are also other machine learning approaches, like hierarchical clustering, and hybrid approaches utilizing machine learning and heuristics rules.

Tanaka and Ishida (2006) proposed a formal representation of generalized table structure based on adjacency of cells and iterative structures. Once the interpretations of table structures are given by humans, the table structures are automatically generalized.

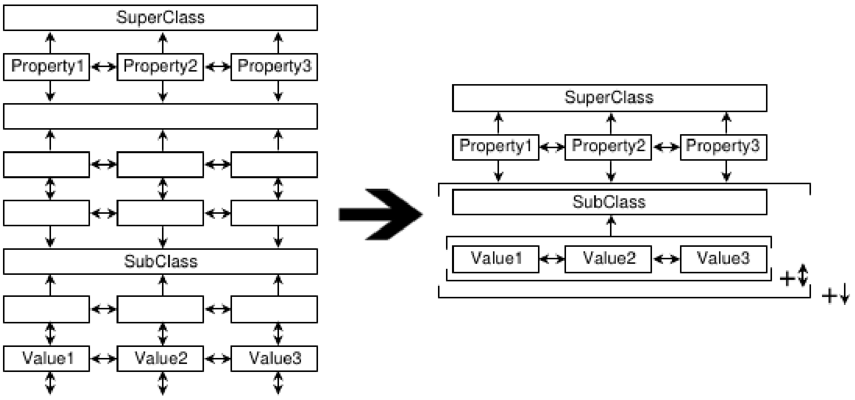


Figure 2.10: Generalization of table proposed by Tanaka and Ishida (2006)

Their approach is based on human interpretation of table structure, table generalization and relation extraction from the table. They use a set of RDF statements describing the relation between data in a structure as an interpretation of a structure. The RDF structure is then represented as a connected table and generalized by finding repetitive blocks and labelling them with '+' symbol (See Figure 2.10).

Hu et al. (2000a) and Hu et al. (2000b) presented a system for recognition of table structure from ASCII documents. They used hierarchical clustering to identify likely grouping of words and to built a binary tree representation of table. Their 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 some 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). The list of associated columns for each phrase is computed by searching for the maximal list of consecutive columns such that the span of each column in the list overlaps the span of the phrase. Based on detection of cell function they built a directed acyclic graph representation of the table, which they were able to visualize and edit in graphical user interface.

Tengli et al. (2004) presented a system that recognizes HTML web tables and to distinguish labels (”access” cells) from the data cells. To distinguish between genuine tables and layout tables they used a simplified version of the table detection algorithm described by Wang and Hu (2002). The algorithm uses decision trees to learn rules using features like cell similarity, number of cells and type of cells. They provided to system table examples with marked labels. The labels were extracted, indexed, if relative string edit distance is less then 0.09, they are merged together. At the end, a ranked list of labels, obtained by thresholding on term frequency, is used by the extraction algorithm. They used learned labels for label detection, but also, they used some heuristics and transformations for detecting sub-header rows. If the span tag is detected, the tags are split and assigned to multiple rows or columns. If row has empty cells, and before and after it there are non-empty data cells, or row contains just one non-empty cell – it is assumed to be sub-header label. Sub-header labels are concatenated with label cells below until another sub-header label is found. The system outputs results in XML format. They reported 91.4% of F-measure.

Jung and Kwon (2006) and Chavan and Shirgave (2011) described similar hybrid approach in which they were first filtering easy detectable non-genuine tables using a set of heuristics about empty rows, tables without rows or columns, one dimensional tables and fraction of hiperlinks and images. After this filtering, the C4.5 decision tree mining 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 <img>,<a>,<input>tags, fraction of cells containing text, symbols, numeric data, size of sentence, table shape, probability of presence of the head, etc. Chavan and Shirgave (2011) used also consistency features like standard deviation of number of columns, number of rows, length of columns and rows and assigned priority weights to these tags, so if some cells contain more of them, the probability of that cell belonging to head will be higher. Jung and Kwon (2006) argued that cell similarity can be an indication of head since heads are the origin of the columns’ similarity and sets the pattern of the body cells’ content that follow it in the column. Empty cell in a first row or column is indication of head to extract table HEADs. They reported 95% F-measure for distinguishing table types and 82.1% accuracy in extracting table HEAD from genuine tables.

Silva (2010) argued that no single algorithm is good enough to solve challenges in table mining. Table mining is a complex multi-layer problem and no one algorithm is capable of accurately treating all tables. Coordination between different table processing approaches, whether these are alternative or sequential to each other, is thus fundamental. She states that table interpretation is dependent on context knowledge and therefore must be done using domain ontologies and lexicons. However, analysing table structure or detecting tables are not dependent on domain knowledge and can be done without any domain specific resource. She was focusing on table detection, cell and column segmentation from tables in ASCII texts and PDF (converted to ACSII using pdftotext linux utility). Silva applied several algorithms on this tasks like SVM, heuristics, and some graphical approaches such as Markov Random Fields. She also argued that precision and recall are not the best measures for tables and proposes new measures:

- Completeness – 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 – proportion of purely detected elements with respect to the total number of detected elements; a pure element is one whose components belong to only one original element.

This might be unnecessary since measures like precision, recall and accuracy are established in the text mining field and they were applied on evaluation of table systems.

2.3.6 Information retrieval from tables

Hearst et al. (2007) and Divoli et al. (2010) presented their work on information retrieval system called BioText Search that is able to retrieve information from texts and abstracts, figure captions and tables from PubMedCentral database of biomedical literature. For article indexing they used Apache Lucane, an open-source information retrieval library for creating search engines. For different document elements (title, text, abstract, table captions, figure captions, table data), they assigned different weights. Their user interface allows search by text, images and tables. If a table search is selected different indexing weights are used, than if the text search is chosen. For table search maximal weights have table captions and table data, while article text and image caption have low weight. Indexes and complete table and images are stored locally. People behind BioText Search engine had also performed a lot of effort in usability testing and testing their system on users. They reported that users gave them useful feedback and were generally satisfied with the engine.

Xu et al. (2008) did similar work at the Yale University, but their system was search engine for images in PubMedCentral articles, called Yale image finder. They applied OCR and image transformation (turning image for 90) to read text content from the image. They, also applied Apache Lucane information retrieval library to make their IR index.

Liu (2009) created dedicated information retrieval system for tables. He argues that current search engines are in deficit in helping the end-users to find satisfactory table results. Lui created a system called TableSeer and proposed a ranking 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. TableSeer is focused on table indexing in PDF files using OCR techniques, but can be adopted to other media types. TableSeer system consists from 5 elements: a table crawler, a table metadata extractor, a table metadata indexer, a table ranking algorithm, and a table searching 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 (OCR method), extracts the metadata for each identified table, ranks the matched tables against the end-users 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 <query, table>pair by replacing the document vectors with the table vectors. Lui used classical TF-IDF approach, but replaced documents and terms with tables and terms. For matching query similarity TableRank is using cosine measure. However, parts of tables in table vector are differently weighted, so table title would have higher importance than table data during the search. TableSeer uses for indexing Lucane framework. Lui reported that the engine performed better than commercial IR systems that were not designed for table IR. Compared to BioText, TableSeer returns whole documents with relevant table, while BioText extracts and shows only tables.

2.3.7 Information extraction and semantic processing of tables

The final step in table processing is extraction of data and its relationships from the table. This data should be represented in suitable form for querying. It still remains a hard challenge to understand data from table in a correct way. Relationships, especially in complex tables could be fuzzy and ambiguous.

Several concepts of information extraction from tables can be identified. The common concept is to map information from the table to the predefined database. The other concept is to create attribute-value pairs from the table, which would be similarly stored to the database, but the database would not have predefined fields. The third concept is to find new instances of the particular class of information (ie. new gene mutations).

Similarly to information extraction from text, there are three approaches to this problem, including machine learning, rule-based and hybrid. Among rule-based we may separately identify approaches that uses variety of domain knowledge resources such as domain databases, linked data and lexicons. We will refer to this third approach as knowledge-driven.

Rule-based approaches

Chen et al. (2000) were probably among the first researchers trying to extract data from tables. They proposed an algorithm based on string, named entity and number category similarity for detecting tables. When the tables were detected, they drop COLSPAN and ROWSPAN by duplicating required number of cells' copies in their proper positions. They proposed representing a table as attribute-value pairs. They observed that cell may be a value of more than one attribute and a cell may act as an attribute in one case and a value in another case. Because of this, multiple attribute-value pairs can be merged to represent actual meaning of a table. They reported only results for table detection, but not for IE task.

Dalvi et al. (2012) presented a system that was able to cluster data from web tables and to build concept-term relations using overlapping triplets from tables. By doing this, they showed that it is possible to make concept-term clusters in an unsupervised way. Their proposition is to take from each table triples of data, so each table have to have at least 3 rows. Data were read just by rows. Ambiguities are solved using this approach since all the terms have a context in triplets. Triplets can be clustered by concept since each triplet represent instances of the same concept. Tables are labelled with the help of Hyponym Concept dataset, which has a data about terms and concepts they occur in with counts for each concept. However, this approach can be used only for narrow set of tables that are exact as tables in relational databases (access cells only in the first row, rest is data). They reported that they had to filter out 93% of collected tables, since they were useless for their purposes.

Google have patented way to extract relational tables from lists on the web (Elmeleegy et al., 2014). In this patent they extracted lists from web pages. List lines are split and using filed quality measure, algorithm is trying to find the most probable fields from the list. Looking for the fields is done in several iterations. First candidates for the first field are made, by making list of first token without any other token or with 1 or more sequential tokens. For each item in this list field quality is calculated and item with maximum quality is chosen. Then same process is repeated for the token that sequels last token from chosen field. Unfortunately, in patent application is not disclosed how the field quality measure is calculated.

Knowledge-driven approaches

Embley et al. (2005) proposed a system that was able to extract table information as attribute-value pairs using ontologies. Based on recognizing expected attribute names and values specified in their ontology, they were able to find the tables of interest. Using the ontology, they also identified attribute (access) cells of the table. They associated value cells with their attribute and created attribute-value pairs. If the value cell is empty, they observed whether the cell is missing or cell has a value based on internal factoring by observing a pattern of empty cells in a column. Some attribute-values pairs have to be adjusted and converted to the right format for easy recognition by the extraction ontology. For example, icons are transformed to text and all boolean indicators (yes/no, true/false, 1/0, checked/empty) to the same form. Using the ontology and given table layout extraction patterns are recognized. The extraction ontology should be able to recognize that Model and Make attribute for table about car models, should be split into two parts, and that first part of value is Model and the second part is the Make. The information extraction step creates array of attribute-value pairs. They tested their system with cell-phone sales and car sales tables. They reported correctly located 90% of the tables in pages for these two applications. Then, from the located tables they inferred 93% of the appropriate mappings with a precision of 96% for car-ads application and inferred 91% of the appropriate mappings with a precision of 85% for cell-phone application. However, their approach of table understanding works only for top-level tables whose attributes are at the top of columns, but their approach is not able to deal with complex tables. Each application of this approach needs to have its own ontology.

Mulwad et al. (2010) presented an automated framework for interpreting data in a table using existing Linked Data Knowledge Bases like DBPedia, Wikitology, Freebase, WordNet and Yago. Using the interpretation of the table they generated linked RDF. Their approach comprises four steps. In the first step, ontology classes are associated with columns. In a typical, well formed table, each column contains data of a single syntactic type (e.g., strings) that represent entities or values of a common semantic type (e.g., people, yearly salary in US dollars). 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 perform this, an algorithm was querying Wikitology knowledge base. In next step algorithm is linking table cells to entities from the Linked Open Data cloud. In the third step, they tried to identify relation between table columns by generating set of candidate relations from the relations that exist between the concepts associated with the string in each row of two columns. To identify relations they queried Dbpedia using its public SPARQL endpoint. In the final step, they developed a template for annotating and representing tables as linked RDF. They reported that 66.12% of the table cell strings were correctly linked. Their 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 might be because of sparseness of data in the knowledge base about these types of entities.

Wang (2013) proposed an information extraction system for web tables that was using heuristics to detect tables and ontologies and thesaurus for data extraction. He proposed a table simplification method to standardize all the tables to the single standard type, with just one header row. If some cell contains col-span or row-span, that cell is appended to the next header cells (over which it spans) and deleted. However, his approach does not involve header detection, but just appending spanned cells to the following ones. This approach can have good performance on relatively simple tables, but would not be able to determine correctly access sells for some more complex types of tables that involve sub-headers and sub-classes. His information extraction algorithm reads the properties from a first table row and look up in the ontology. If the property is found in ontology, the values are stored in database. If it is not, it looks in the thesaurus for the synonyms. If synonym is found, data are saved. If property is not matched, then it makes new value in the ontology. If property can be matched with recognition rate greater than 0.6, then this property belongs to the set of objects in particular ontology body and is added there. Wang reported average accuracy of 93.6% for table information extraction.

Machine learning approaches

Wei et al. (2006) created a question-answering system that was answering a questions that contained answers in tables. Their system was analysing ASCII free text documents. Their approach was to tag table lines with TITLE, SUPERHEADER, TABLEHEADER, SUBHEADER, SECTIONHEADER, DATAROW, SECTIONDATAROW, TABLEFOOTNOTE, TABLECAPTION, NONTABLE, BLANKLINE, SEPARTATOR tags using conditional random fields algorithm (CRF). In this case, CRFs can be roughly understood as conditionally-trained hidden Markov models and globally-normalized extension to Maximum Entropy Markov Models that avoids label-bias problem. They used white space (number and length of white spaces and gaps), text (number of cells on a line, certain types of string are more usual on some parts of tables, type of character on a line) and separator (special characters, successive characters) as features. Also, CRF has the ability to take into account information from before and after the current label. Their system then was able to create an XML document for each cell with data, metadata and table captions related to that cell's information. For extracting data from tables, heuristics about head cells are used (cells in the first column are used as row headers, header rows in the middle of table indicate a new section and should apply only to the next section, if a cell in a header row can cover multiple cells in a data row, it is included in the headers for all covered columns). Given a query, cell documents are ranked using a language model framework for retrieval. The basic approach of using language models for IR assumes that the user generates a query as text that is representative of the ”ideal” document. The task of the system is then to estimate, for each of the documents in the database, which is most likely to be the ideal document.

Crestan and Pantel (2010) proposed encoding of extracted information from tables to semantic triples of the form <p, s, o>, where p is a predicate or relation, s is the subject of the predicate and o is its object. However, focus of their work was identification of subject (protagonist) in the attribute/value tables. In attribute-value tables, 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 could be found: within the table (occasionally found in the table with a generic attribute such as name or model), within the document or the html <title>tag and anchor text pointing to the page. They used N-gram based approach (1 to 12-grams) on documents and anchor text (obtained from a commercial search engine's web link graph) in combination with Gradient Boosted Decision tree

(GBDT) regression model. They reported 40% precision, which they consider as a good starting point, as they, also report 97% chance to find the correct protagonist in the top-100 ranked candidates.

Extraction of Named Entities from Tables in Gene Mutation Literature (Wong et al., 2009) is probably the only paper on table IE in biomedicine. The paper describes a system that was able to extract gene mutation names from Medline. Since tables in Medline are on different HTML document than the article and are referenced by link, authors of this paper didn't have to deal with table detection. Their extraction task is grounded in the specific content of the Mismatch Repair (MMR) Database - a database of known genetic mutations related to hereditary non polyposis colorectal cancer (HNPCC), along with links to the research papers from which the database has been constructed. From the database and its links to papers, they were able to construct a collection of tables related to HNPCC mutations, and then use the MMR database records themselves as a gold standard for evaluating their techniques. They obtained tables by scraping PubMed interface with automatic crawler. Once tables were obtained, column headers were detected using <hr>tag, and row headers were detected by checking if the top-left cell is empty (pattern in most row-major tables). They performed classification of full columns/rows (depending on whether the table was row- or column-oriented) into relevant entities. The following entities or classes were identified as relevant: Gene, Exon, Mutation, Codon, Statistic and Other. Several approaches for table vector classification were introduced. They used heuristic for classifying headers by matching the header string to the names of the classes in a case-insensitive manner. The second approach was to build a more informed classifier for the class ”Mutation” using the point mutation NER system MutationFinder. They applied this tool to the text in the table cells and identified which table-vector contained at least one mutation mention. Algorithms used in this approach were Naive Bayes (NB), Support Vector Machines (SVM), Propositional Rule Learner (Jrip) and Decision Trees (J48) with different sets of features (cell bag-of-words, header bag-of-words, cell+header bag-of-words, basic features like header string+average median cell length+is data numeric). They concluded that SVM is the best algorithm for this dataset, with JRip and J48 following, and NB performing the worst of the four in most cases. Table vector classification allowed them to extract lists of candidate mutation names from tables to be added to the database.

2.4 Summary

In the past 15 years the field text mining of biomedical research documents have established itself as one of the most active research sub-fields in text mining and natural language processing. A lot of tools have been built, which help biomedical researchers and clinicians in their day to day work and research, such as text preprocessing tools, named entity recognition, information retrieval and information extraction systems. However, almost all these systems are ignoring tables.

In the past years, there have been attempts on processing tables. Table processing is divided into three subtasks, all of which are research fields by its own right: table detection, functional table analysis and table understanding. Most of the work, done so far, is focused on table detection from different media types (ASCII, PDF, HTML, XML) and functional table analysis. There are very few systems trying to solve table understanding as a whole. Also, several IR systems have been created, mainly by applying techniques of text based IR to tables. Some researchers proposed even some Information Extraction systems, mainly based on heuristics and some external knowledge source about the domain. Understanding of table data relationship is a domain-dependent problem, and and is not likely to be solved without domain knowledge resources. The mentioned systems were able to solve the problem for a narrow set of tables or they didn't have results high enough for the real usage scenario. The only systems that are able to process tables in biomedicine are BioText search engine (Hearst et al., 2007) and gene mutation named entity recognition system (Wong et al., 2009). Text mining scientists gained interest in table data mining of biomedical literature just couple of years ago,. So far, no information extraction system for biomedical and clinical domain that is able to work with tables.

We believe that table mining is an important and still not enough researched field, that can improve machine's understanding of documents by taking in account data in a structured and semi-structured parts of the documents. Biomedical research documents contain a vast amount of important data in tables. Because of this fact, we are not able to say, that some system is able to extract the most important data without looking at the tables. Our future research efforts will focus on addressing table information extraction problem for biomedical research field, and then, if possible, to generalize solution for the other fields.

Chapter 3

Method Overview

3.1 Introduction

This project will focus on information extraction and data curation from tables in biomedical documents. Generally, semi-automated data curation systems consist of four parts: information extraction engine, data store, data curation interface and query interface (Alex et al., 2008). Each four parts of the system may be supported by various knowledge sources such as lexicons, thesaurus, databases or linked data sources.

An architecture overview of the proposed approach is shown in Figure 3.1. The documents are firstly processed by automated information extraction system. After the data is stored, human expert curators may check and edit the extracted data using a data curation interface. This should assure the validity of the extracted data. Users may access the data using a query interface. In the following sections we will describe in more details all components of the system.

3.2 Table information extraction

The major part of our system is the table information extraction engine. This part of the system should be able to extract information from the source documents automatically. The workflow of the proposed table information extraction engine can be seen in Figure 3.2.

In the first part of the method, tables are decomposed into the cell objects containing a cell's value and the data from its navigational paths (header, sub-headers and stub). In the second step, cell values are normalized and semantically analysed. In the final step, inference on the table level is performed. Relationships between cells are

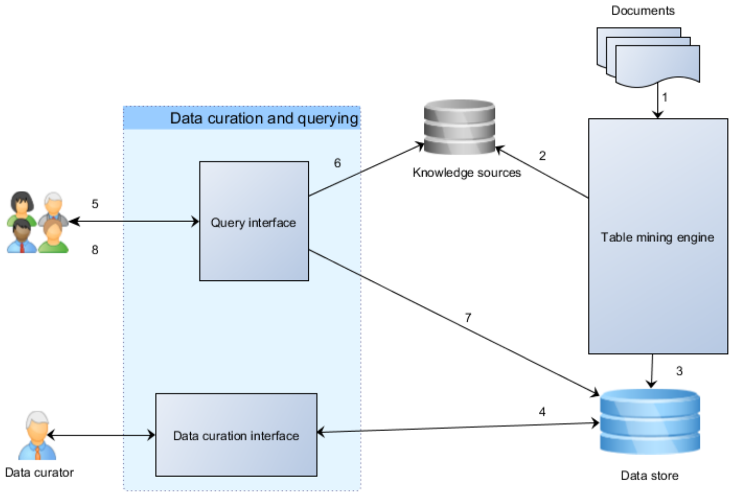


Figure 3.1: High level architecture of the proposed method
1. Retrieved documents are sent to the table mining engine. 2. The table mining engine uses knowledge sources to extract information from the table. 3. Extracted information is stored to a data store. 4. Data curators review and correct extracted information. 5. Users may submit queries to the query interface in natural language. 6. Queries are processed and normalized using knowledge sources. 7. Using normalized queries, data store is queried. 8. Relevant information is presented to the user.

recognized and information may be extracted. There is also a possibility for a fourth step in which the method will look for the information in the body of the article and link this information with the data presented in the table. In the following subsections we will give a more detailed description of these steps.

3.2.1 Table decomposition

3.2.1.1 Table detection

In the table mining, firstly, it is necessary to identify table mentions in the documents. This task in some types of documents might be trivial. For example, in many XML formats it is possible to identify tables by extracting content of a specific XML tag. However, in some types of XML documents and in HTML it is a much harder task. The research on identification of tables from HTML was described in Section 2.3.4. Our system will follow Son et al. (2008) where identification of genuine tables is not trivial (i.e. not achievable through XML tags).

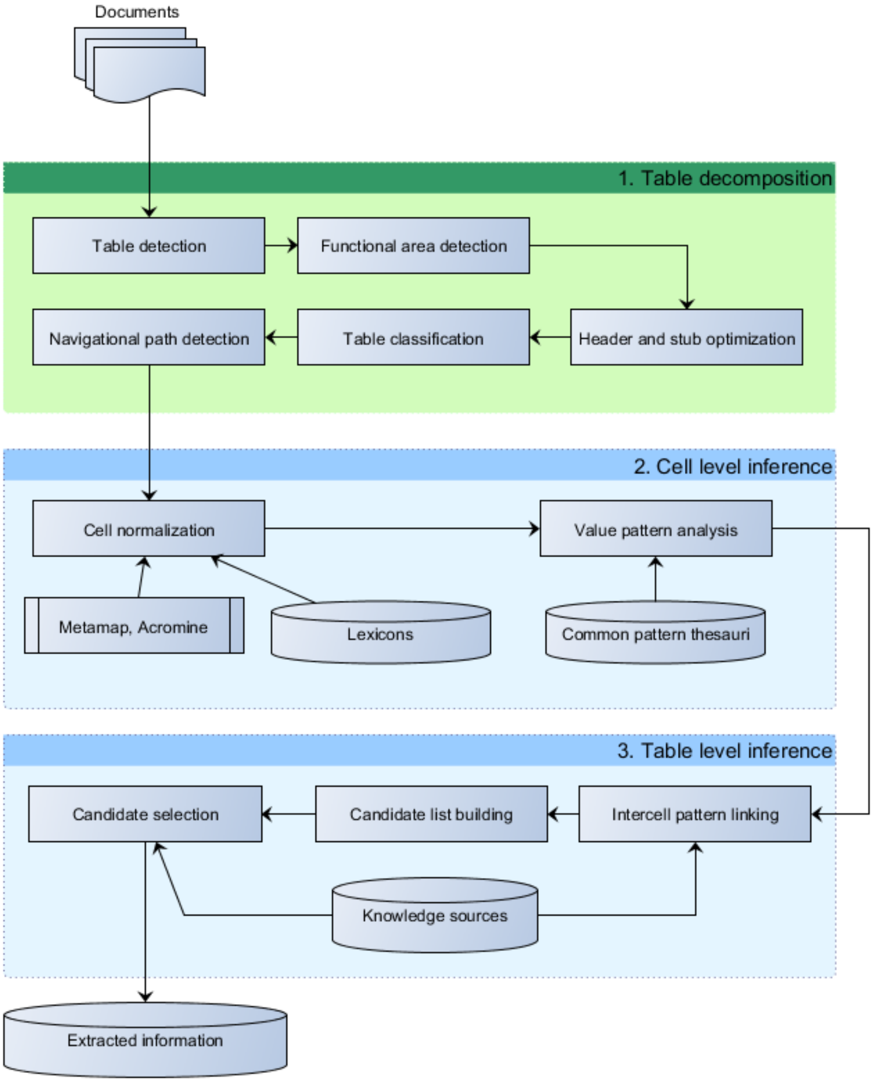


Figure 3.2: Workflow of the table information extraction system.

3.2.1.2 Functional area detection

The second step in the table information extraction is detection of navigational and data areas. This task is called in literature functional area detection, since it determines the function of cells in a table. In this task, it is important to distinguish header, stub and sub-header cells from data cells. Research that addressed this task was already discussed in Section 2.3.5. Many authors used machine learning to approach this problem. However, the big challenge is that there is no publicly available annotated dataset. Because of this we decided to follow approaches based on heuristics. Our approach takes into account cell's characteristics (<thead>,<tbody>tags, whether cell is spanning, font typeface, whether cell is empty or has starting space characters) and cell's position. Also, our approach is taking into account context of the cell in row or column (how many empty cells are in a row or column). The performance of our approach can be improved by learning typical labels for navigational cells. We may assume that many tables in the same domain have similar labels in navigational areas (Wong et al., 2009). By detecting a number of learned labels in some table row or column we may assume that the whole row or column is part of the navigational part of the table. Another way to improve the performance is to measure the similarity of the cell's content through a row or column. The similarity of cell's content can be measured in a number of ways, including, for example to check the type of the content: navigational cells are almost always textual, but data fields often contain numerical values.

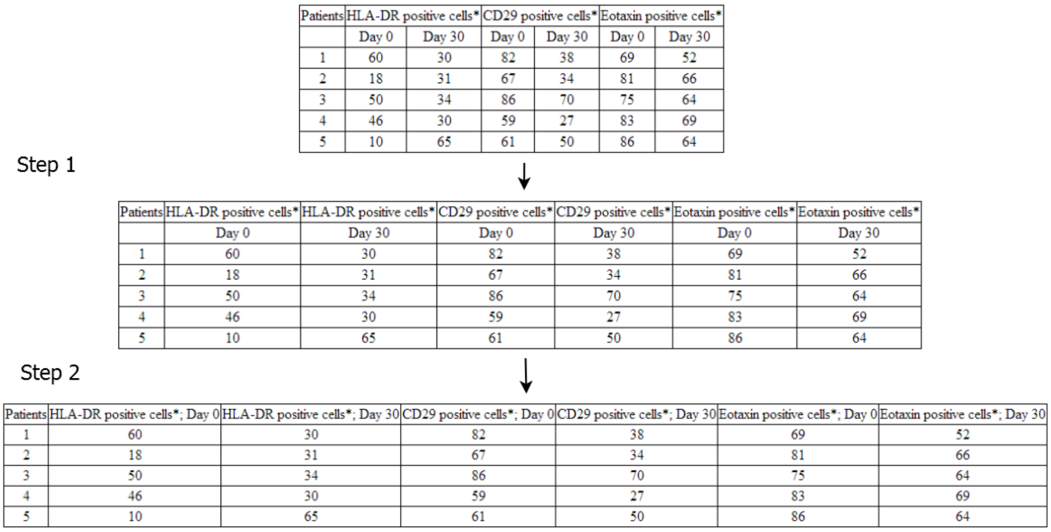
3.2.1.3 Header and stub processing

Many headers and stubs in scientific and especially in biomedical documents are complex. Headers may contains multiple rows and column-spanning cells. Similarly, columns might contain multiple columns and row-spanning cells. The aim of this task is to "simplify" these areas, so the header would have only one row and stub would have only one column. However, our system should keep track of the original structural organization of the table. In order to store this information, we will propose two structures. The first one will keep the original structural organization of the table, while the other will store the "simplified" version of the table. The simplified table will contain merged headers and stubs. In order to merge headers, the system is prepending higher order cell's value to the lower order cell's value, while deleting higher order rows, until there is only one row in the header. The example of header processing is given in Figure 3.3.

If table contains multiple columns in a stub, it is processed similarly. The row-spanning cells are split and copied to the newly created cells by splitting. In the second step, the content of the left-most cells is copied to its right neighbour. When content of all the cells is copied, left-most cell is deleted.

3.2.1.4 Table classification

There are two reasons for table classification. The first reason is to identify tables that contain data of interest. The second reason are different types of tables which need to be addressed differently during the reading phase.



Patients	HLA-DR positive cells*		CD29 positive cells*		Eotaxin positive cells*	
	Day 0	Day 30	Day 0	Day 30	Day 0	Day 30
1	60	30	82	38	69	52
2	18	31	67	34	81	66
3	50	34	86	70	75	64
4	46	30	59	27	83	69
5	10	65	61	50	86	64

Step 1

Patients	HLA-DR positive cells*	HLA-DR positive cells*	CD29 positive cells*	CD29 positive cells*	Eotaxin positive cells*	Eotaxin positive cells*
	Day 0	Day 30	Day 0	Day 30	Day 0	Day 30
1	60	30	82	38	69	52
2	18	31	67	34	81	66
3	50	34	86	70	75	64
4	46	30	59	27	83	69
5	10	65	61	50	86	64

Step 2

Patients	HLA-DR positive cells*; Day 0	HLA-DR positive cells*; Day 30	CD29 positive cells*; Day 0	CD29 positive cells*; Day 30	Eotaxin positive cells*; Day 0	Eotaxin positive cells*; Day 30
1	60	30	82	38	69	52
2	18	31	67	34	81	66
3	50	34	86	70	75	64
4	46	30	59	27	83	69
5	10	65	61	50	86	64

Figure 3.3: Header processing. In the first step, spanning cells are split and copied. In the second step, higher order cell's content is merged with content of the lower cells in header. (PMC140320)

There are different types of tables that need to be approached differently while reading. We distinguish three types of tables: list tables, matrix tables and sub-header tables. List tables are one dimensional tables, which contain list of items. Matrix tables are simple tables containing header, stub and data part arranged as a matrix. Sub-header tables are similar to matrix tables, but they contain sub-headers, which may add more description to the table's data. A more detailed analysis of the table types can be found in Section 4.2.5.

3.2.1.5 Navigational path detection

The aim of navigational path detection is to find navigational cells (i.e. header, stub and sub-header) related to each data cell in the table. Also, this task has to find and maintain structural relationships between them. This task is highly dependant on functional analysis of the table. If the functional analysis is performed successfully, this step is straightforward. Header related to the data cell is the cell in the same column on top of the table and stub cell is the left-most cell in the same row. Header and stub cells have to be in a part of table labelled as navigational parts in the functional analysis step since there might be some tables which do not have headers or stub.

Identification of sub-headers is a more complex task, since tables might have a tree like structures of sub-headers (see example in Figure 3.4). The "size" of white space at the beginning of the cell might help in determining how many sub-headers are related to the processed data cell. However, some tables might have confusing marking of sub-headers and remains a challenge (i.e. spacing rules are not followed). The related sub-headers are typically above the processed data cell, so our method will iterate stub column, looking for all related sub-header, until it reaches the root of sub-header structure.

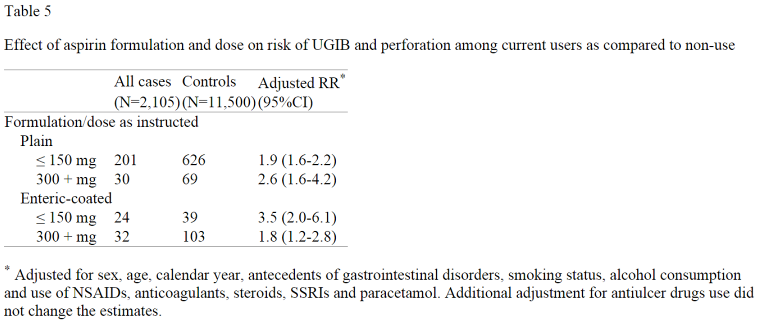


Table 5

Effect of aspirin formulation and dose on risk of UGIB and perforation among current users as compared to non-use

	All cases (N=2,105)	Controls (N=11,500)	Adjusted RR* (95%CI)
Formulation/dose as instructed			
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)

* Adjusted for sex, age, calendar year, antecedents of gastrointestinal disorders, smoking status, alcohol consumption and use of NSAIDs, anticoagulants, steroids, SSRIs and paracetamol. Additional adjustment for antiulcer drugs use did not change the estimates.

Figure 3.4: Example of table with tree like sub-header structure. This table has two levels of sub-headers in its stub (PMC32172)

The value in the most left and most top cell, also plays a part in navigational path. It usually generalize stub column. If this cell is not empty, it has to be extracted as part of the navigational path.

3.2.2 Cell level inference

3.2.2.1 Cell value normalization

Data in table often use acronyms, abbreviations or ungrammatical constructions. Data normalization enables easier processing of data.

Firstly, textual cells can be normalized using standard text mining preprocessing methods such as stemming or lemmatization. Abbreviation and acronyms have to be expanded to the long form using dictionaries of acronyms and abbreviation or existing tools such as Acromine[6] (Okazaki and Ananiadou, 2006). The main reason for expanding short forms is that the short forms may be hard and ambiguous for semantic analysis. Because of semantic analysis it would be also useful to map various terms describing the same concept into one concept term. This would be possible to perform using MetaMap (Aronson, 2001) or something similar.

[6] http://www.nactem.ac.uk/software/acromine/

Numeric data cells also need to be normalized, since there could be various kinds of number representations, using various separators for decimal point and various separators between the figures. For further pattern analysis it is useful to have these various numeric representations to the one standardized form.

3.2.2.2 Cell pattern analysis

A multidimensional data elements are data objects being represented by a multidimensional vector (Lee et al., 2000). These data object in our case are table data, represented by descriptions in navigational areas. However, data cells may contain additional information, such as standard deviation of value, alternative value, description, etc.

Multiple information can be stored in single data cell using various patterns. Patterns may include ranges, formulas, different types of separated values or the combination of these mentioned types. The meaning of each value in the pattern usually could be either read from the navigational data of that particular cell, or could be derived from the domain knowledge (i.e. usage of the ± sign to separate mean value and standard deviation). Firstly, it is necessary to recognize patterns of value presentation.

We will analyse the main patterns of cell data representation patterns in biomedical literature. Our algorithm matches the common numeric structures, like single value, ranges, mean ± standard deviation or multiple separated values. Also, algorithm looks for the particular separators in textual cells that semantically splits the text into the multiple information segments (i.e. /,[,],),(, etc.). The list of values could be created for each cell, which would be later linked with its meaning using domain knowledge or data from the navigational path of the cell.

3.2.3 Table level inference

3.2.3.1 Cell pattern linking

Values selected through patterns retrieved in the previous step need to be linked with their meaning. The meaning of these values are usually described in the header or stub. If the meaning of the values are described explicitly, it is possible to match them since they are usually using same separators in both data cells and navigational cells. However, it is possible that some values are not described explicitly. However, based on pattern and domain knowledge, we will determine the meaning based on specific cases. For example ranges or mean value with standard deviation could be recognized without explicit label in stub or header saying what the value is. Figure 3.5 shows an example of implicit and explicit cell pattern linking.

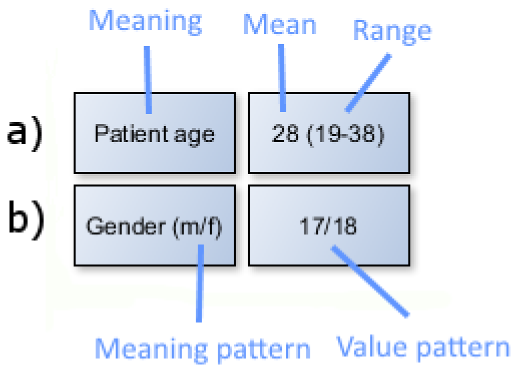


Figure 3.5: Example of pattern linking. In a), patient age pattern is not presented explicitly. However, knowledge from the field helps to determine that the first value is mean age, while the range in brackets is the age range of the participants. In b), gender is presented explicitly. Knowledge is also required for linking of acronyms (m and f) into the right term (male, female)

3.2.3.2 Candidate selection

In biomedical documents there might be multiple cells in tables that matches some rule about one information class. In this situation, the system needs to build a candidate list, that will be further semantically analysed. If there are multiple possible values, the system is looking at surrounding tokens in labels and domain knowledge in order to determine the most probable value. Dictionaries of tokens and patterns may be created from the domain knowledge that will help determine information that should be extracted. Also, field knowledge like the expected range of values for some class or what is the type of value (text, numeric, range, etc.) may be used in order to select the right value and discard false positives. After the right information is selected it can be extracted and stored.

3.3 Data store

Several different approaches for storing data have been developed, including relational databases, document stores and triple stores. Our method requires flexibility from the data store, so the information classes may be added with minimal effort. Also, data should be semantically linked. Therefore, the logical choice for the data store is linked data triple store. It allows new classes to be added with minimal effort and the data are linked. Data may be even linked with external data sources in order to obtain additional information and knowledge about the concept. Information from different sources can easily be combined by merging sets of triples into the single data graph (Heath and Bizer, 2011). Also, this data store would allow us to integrate data extracted from tables with the data extracted from the text of the article using different tools. Linked data allows standardized data querying using SPARQL query language (Bizer et al., 2009).

3.4 Data curation interface

The information extraction engine is the first step in our semi-automated data curation pipeline. However, all text mining and information extraction system make mistakes. Only human experts are able to evaluate whether the extracted data are completely correct. Because of this, we will propose a data curation interface that will allow human experts to check the extracted data and, if necessary, to correct, add or delete information. The main purpose of the data curation engine is to provide user friendly interface for human experts for reviewing and editing previously extracted information. We will propose a web interface that will enable expert users to explore and modify data extracted from tables.

3.5 Query interface

When the data are extracted and reviewed by human experts, they may be used by a broader user base. Users then can query our data store in order to find relevant information for them. Since the data was assessed by expert curators, users will get from the system correct information.

3.6 Summary

In this chapter we described the general methodology for the information extraction and data curation from tables in biomedical literature. The methodology consists of four main parts: information extraction engine, data store, data curation interface and query interface. The methodology for information extraction engine contains the first two steps (table recognition and functional analysis) defined by Hurst (2000). However, the third step he defined, table understanding, we have split to a number of smaller tasks such as value normalization, pattern analysis, pattern linking, semantic analysis and information extraction. In order to archive the goal of table understanding, it is necessary to use and develop tools with a good performance for these steps. The system will allow data curator to review and correct extracted data. In order to use the data, users need a query engine from which they will be able to search for the most relevant information. There are a number of unsolved challenges in table mining. Tables may not be correctly tagged and there is a variety of visual layouts and data presentation patterns that may be used. Also, semantic linking between navigational cells and data cells can be ambiguous. More detailed analysis of challenges is given in Section 4.2.6. We will address these challenges in our work.

Chapter 4

Current progress

4.1 Introduction

So far, a number of experiments have been performed on clinical trial documents. Clinical trial documents have been chosen, because of the potential impact on treatment and drug discovery.

The current work done in this project include four components:

- We have performed table analysis and identified the major table types in clinical trial documents.
- We have developed a method for decomposing tables and finding cell's navigational paths.
- We have performed a case study on information extraction of some of patient demographic data, such as body mass index, participant's weight and number of participants of the trial.
- We have developed an user friendly data curation interface for reviewing and correcting extracted information from tables.

Dataset

Our dataset was retrieved from the PMC database. In order to retrieve clinical trial documents we have filtered PMC documents by publication type. The publication types that were of our interest were ”Clinical Trial”, ”Clinical Trial, Phase I”, ”Clinical Trial, Phase II”, ”Clinical Trial, Phase III” and ”Clinical Trial, Phase IV”. Documents that did not contain any table were also filtered out. Our dataset had 1275 documents containing at least one table, with total of 4132 tables.

4.2 Manual analysis of tables in PMC documents

4.2.1 Aim

In this stage of our research we aim to analyse randomly selected sample of tables from PMC database to learn about structure and types of tables and cell value presentation patterns. Also, we wanted to identify the main challenges in table mining from biomedical literature.

4.2.2 Methodology overview

We have manually analysed tables and their data in PMC database. We have conducted analysis on 50 randomly selected PMC documents, with total 155 different tables. Our analysis can be split into three parts:

- Analysis of data presentation in cells of the tables. We recorded how in different tables values are presented, what are the semantics of a different patterns and how these semantics can be recognised.
- Structural table analysis. In this part we looked at structural building blocks of the table, such as cells, rows, columns, headers, captions, etc. We analysed them both visually and by looking into XML presentation.
- Analysis of dimensions of the tables and classifying tables. We analysed how many dimensions could table have and how that affects the way of table reading. Then, we classified tables according to the number of dimensions and difference in way how table is read.

4.2.3 Data in the cells

Tables in clinical trial articles contain various types of information raging from term definitions, measurement scale definitions, patient information to information about arms of clinical trial and therapies used during the trials. These data can be represented in the form of text, numbers, dates, formulas etc.

Text is used for listing items and defining terms. Usually, table header and stub contain more text then the data part of the table. The role of the headers and stubs is to define and to describe the data stored in the data cells. Data in clinical trials are often measurement values, but also there are tables containing text in data cells, defining terms, listing items or describing clinical trial outcomes and drug side effects.

Numeric data are used frequently to describe measurements or information about patients in arms of the trial. Number of participants or age of the patients are always represented by integer. Measurement data are usually presented by real numbers. Real numbers may be challenging, since they may have various representations. While both dot (.) and comma (,) could be used in different notations, dot is more often used to represent decimal dot and comma is used to separate multiple numbers. We noticed several tables that for the values between 0 and 1, used notation starting with dot (.). These different value presentation pattern are due to cultural or regional differences.

Ranges of numbers are usually separated using dash (–) or comma (,) symbol between values. Authors also use greater (>) or less (<) symbols prior to value.

Simple formulas are often used to describe data with its standard deviation. These data are typically in the following form:10 ± 3.

In biomedicine, it is often that some value are not present or evaluable. In the case when the value is not evaluable, authors may use NE (Not evaluable) or NA (Not applicable) acronym. For missing value dash (–) symbol is often used or the cell might be left empty.

Tables may have columns with binary values, showing presence or absence of some item or substance. Presence of substance is usually presented with a word yes or some symbol (X, T). Absence can be denoted by empty cell, words no, none or symbol (-,N,F).

In order to present multiple values in one cell, authors may use combination of data presentation types. They may use numbers together with ranges, names of the units or textual comments. Similarly can be done with the binary values, combined for example with population size or text. Combining data helps in presenting multiple information in a single cell, but creates space for a huge variety of value presentation patterns.

The variety of data representation formats and a number of dimensions in biomedical documents are making semantic analysis and value-meaning linking a challenging task. However, cell's content is not the only feature that can make tables complex.

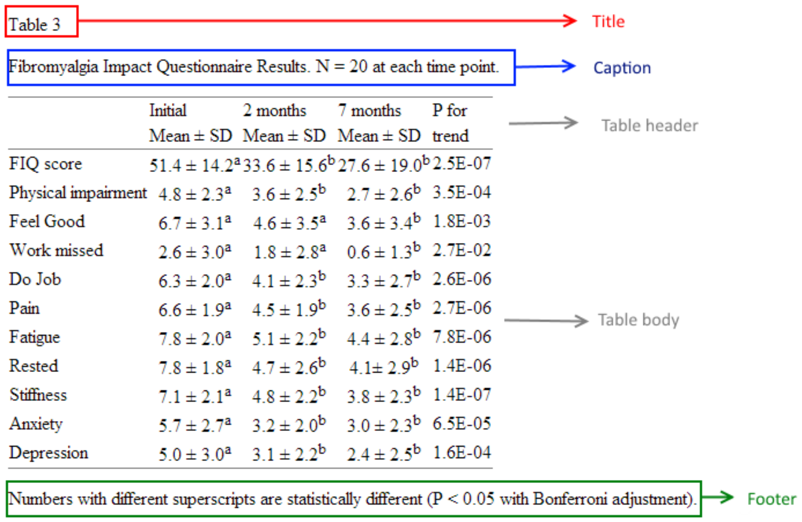


Table 3

Fibromyalgia Impact Questionnaire Results. N = 20 at each time point.

	Initial Mean ± SD	2 months Mean ± SD	7 months Mean ± SD	P for trend
FIQ score	51.4 ± 14.2^{a}	33.6 ± 15.6^{b}	27.6 ± 19.0^{b}	2.5E-07
Physical impairment	4.8 ± 2.3^{a}	3.6 ± 2.5^{b}	2.7 ± 2.6^{b}	3.5E-04
Feel Good	6.7 ± 3.1^{a}	4.6 ± 3.5^{a}	3.6 ± 3.4^{b}	1.8E-03
Work missed	2.6 ± 3.0^{a}	1.8 ± 2.8^{a}	0.6 ± 1.3^{b}	2.7E-02
Do Job	6.3 ± 2.0^{a}	4.1 ± 2.3^{b}	3.3 ± 2.7^{b}	2.6E-06
Pain	6.6 ± 1.9^{a}	4.5 ± 1.9^{b}	3.6 ± 2.5^{b}	2.7E-06
Fatigue	7.8 ± 2.0^{a}	5.1 ± 2.2^{b}	4.4 ± 2.8^{b}	7.8E-06
Rested	7.8 ± 1.8^{a}	4.7 ± 2.6^{b}	4.1 ± 2.9^{b}	1.4E-06
Stiffness	7.1 ± 2.1^{a}	4.8 ± 2.2^{b}	3.8 ± 2.3^{b}	1.4E-07
Anxiety	5.7 ± 2.7^{a}	3.2 ± 2.0^{b}	3.0 ± 2.3^{b}	6.5E-05
Depression	5.0 ± 3.0^{a}	3.1 ± 2.2^{b}	2.4 ± 2.5^{b}	1.6E-04

Numbers with different superscripts are statistically different ($P < 0.05$ with Bonferroni adjustment).

Figure 4.1: Elements of the table in PMC documents (PMC57816)

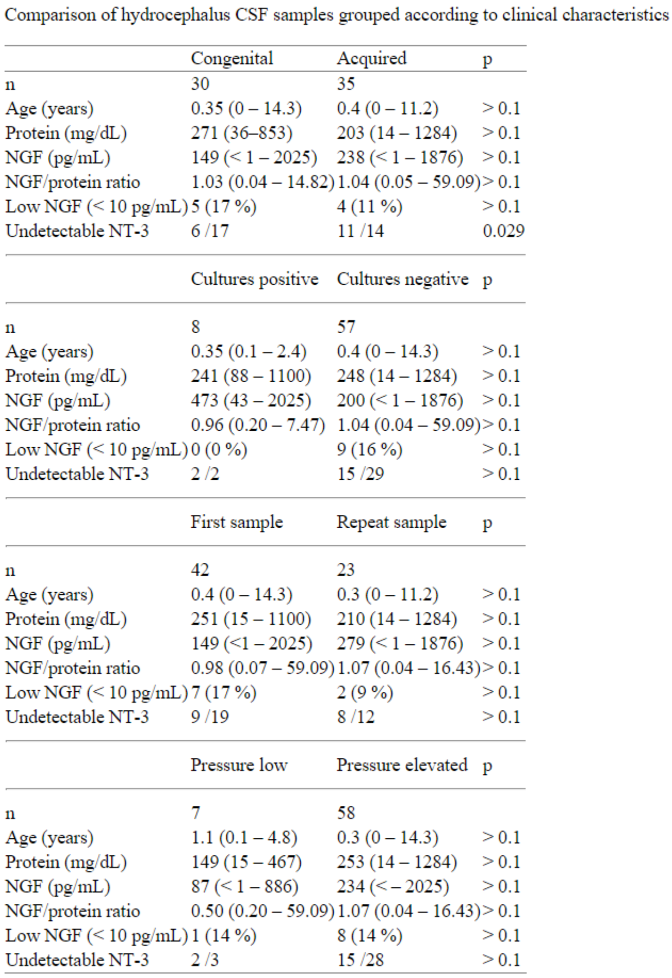
Table structure and visual layout are the other means of presenting complex and multidimensional data in a compact way.

4.2.4 Structure of the tables

Documents in PMC and Medline are stored in the XML format (United States National Library of Medicine , 2014c). XMLs in PMC has a standardized format, and table data are usually inside particular tags. Inside these tags, there are also tags containing table's title, caption, footer and the actual table.

Table titles in PMC documents usually consist from the word ”Table” and ordinal number of the table in the documents (i.e. ”Table 1”, ”Table 2”, etc.). Content of table caption actually describes the table. In the footer, some additional information about table and legend of symbols used in tables can be found. The footer is not mandatory and many tables do not have it. Title, caption and footer are placed in particular XML tags in PMC documents and they are trivial to identify. An example of the table with marked elements can be seen in the Figure 4.1. Example of the table with its XML presentation can be seen in Appendix A.

Tables are placed inside table tags. However, one table may have multiple XML table tags, containing different parts of the presented table. Most of these tables might be viewed and processed as separate tables, since the authors might just merged them because they carry similar data. Other way of storing complex tables consisting from more than one table is to store it in a single table tags and section (sub-tables) may



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 4.2: Example of multi-table (PMC57003).

be separated using horizontal lines. These lines might be used to construct a number of headers for the merged tables. We would refer to these tables as *multi-tables* and examples of these tables might be seen on Figure 4.2

Header and body of the table are usually marked with a set of particular tags. However, there was a small number of tables that did not have labelled headers. Also, we noticed that some tables are not labelled correctly. In multi-tables, only the header of the first table may be labelled.

Stub may contain sub-headers or super classes. Cells containing these types of data are either spanned through all the columns of the table, or have significant amount of empty cells. Not all cells in the sub-header row have to be empty, since some cells might have mean value for the whole class of elements. Subclasses are often formatted with starting empty space or dash (–) symbol.

4.2.5 Table types and dimensions

In our data, there were three major common types of tables excluding multi-tables. These types are list, matrix (regular) and sub-header tables.

List tables are tables that contains a list of items of some type. These are usually one dimensional tables. Because they contain only one dimension we may also call them vector tables. Tables of this type may contain a header, which describes type of items, and then in one or more columns are listed items. List items in this type of tables are most often portions of text. Usually, header in this type of tables spans over all the columns or is repeated in each header cell.

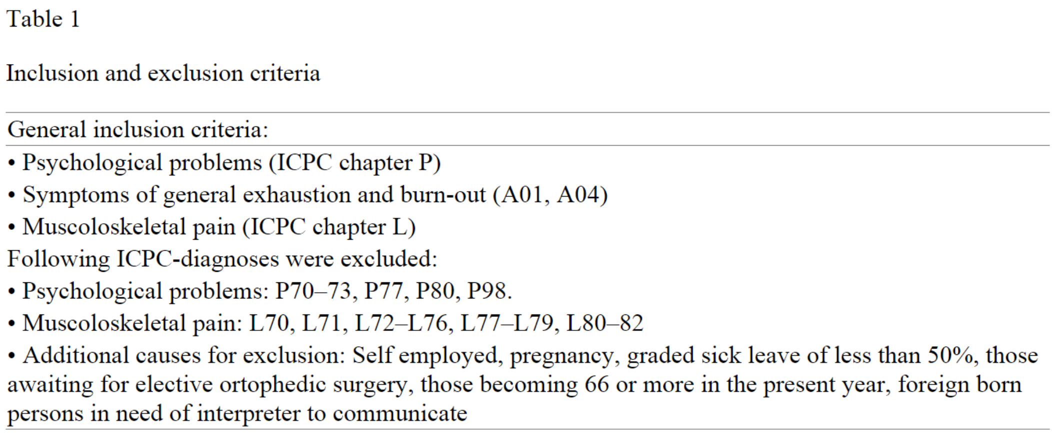


Table 1

Inclusion and exclusion criteria

General inclusion criteria:
• Psychological problems (ICPC chapter P)
• Symptoms of general exhaustion and burn-out (A01, A04)
• Muscoloskeletal pain (ICPC chapter L)
Following ICPC-diagnoses were excluded:
• Psychological problems: P70–73, P77, P80, P98.
• Muscoloskeletal pain: L70, L71, L72–L76, L77–L79, L80–82
• Additional causes for exclusion: Self employed, pregnancy, graded sick leave of less than 50%, those awaiting for elective ortophedic surgery, those becoming 66 or more in the present year, foreign born persons in need of interpreter to communicate

Figure 4.3: Example of list table (PMC161814)

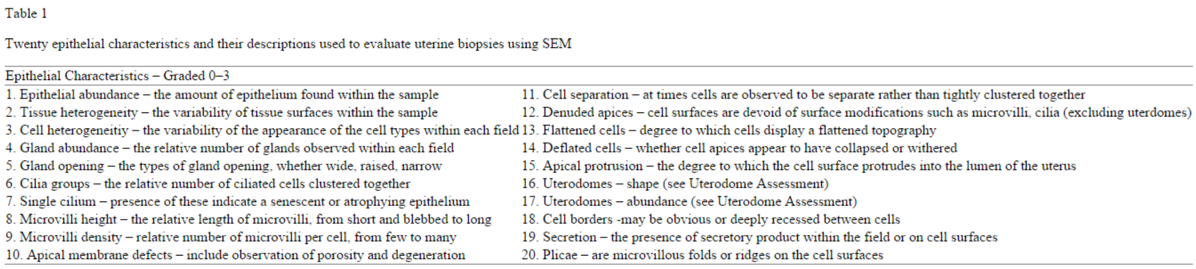


Table 1

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 uterdomes)
3. Cell heterogeneitiy – 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 4.4: Example of list table with multiple columns (PMC161814)

If there is only one column, table is a list table. However, if there are multiple columns, it becomes much harder. It may be determined from the spanning header through all the columns or repeating cells in header. However, if there is no header and table has more than one column, it is hard to distinguish this type of table from matrix or sub-header table. This may be possible through semantic similarity of cells values. An example of the list table can be seen on Figure 4.3. Example of list table with multiple columns can be seen on Figure 4.4.

Tables, in which cell values can be read as a relationship between the values of a header cell in the same column as the value cell and stub cell in the same row as the value cell, are regular tables. Wright (Wright and Fox, 1970) argued that these tables can be called matrix tables. These tables are most suitable for representing numeric values or ranges. These tables may have complex headers or stubs, but the body of the table is arranged in a simple matrix. Headers could contain multiple rows with some cells spanning over several columns. Usually, higher cells are spanning more than the lower ones. Although, the structure of body in this table type is simple, cells may contain multiple values. An example of a matrix table can be seen in Figure 4.5.

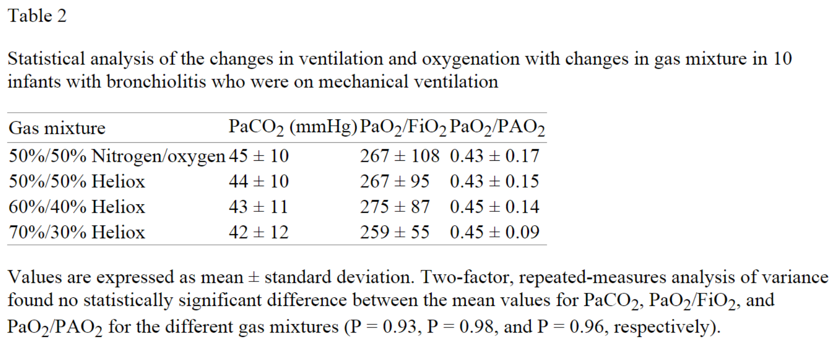


Table 2

Statistical analysis of the changes in ventilation and oxygenation with changes in gas mixture in 10 infants with bronchiolitis who were on mechanical ventilation

Gas mixture	$PaCO_2$ (mmHg)	PaO_2/FiO_2	PaO_2/PAO_2
50%/50% Nitrogen/oxygen	45 ± 10	267 ± 108	0.43 ± 0.17
50%/50% Heliox	44 ± 10	267 ± 95	0.43 ± 0.15
60%/40% Heliox	43 ± 11	275 ± 87	0.45 ± 0.14
70%/30% Heliox	42 ± 12	259 ± 55	0.45 ± 0.09

Values are expressed as mean ± standard deviation. Two-factor, repeated-measures analysis of variance found no statistically significant difference between the mean values for $PaCO_2$, PaO_2/FiO_2, and PaO_2/PAO_2 for the different gas mixtures (P = 0.93, P = 0.98, and P = 0.96, respectively).

Figure 4.5: Example of matrix table (PMC29042)

More than half of the tables in our dataset had sub-headers. Generally these tables are similar to matrix tables, but presence of sub-headers changes the semantics of these tables. Sub-headers are rows or cells, which define superclass of the values below, until next sub-header. As a header defines class of row's cells, sub-header groups items in the rows and defines subclasses of items. Sub-headers are usually defined in a stub or in a row that is spanned over the whole table. When the sub-header is defined in a stub, often row in which is sub-header contains only empty cells. In many cases sub-headers and the items under the sub-header form a tree like structure, where items are starting with one or more empty characters. The sub-header structure in a table may have a number of levels. However, usually this type of table has one or two levels of sub-headers. Generally, sub-headers add dimensions to the table. Figure 4.6 shows an example of sub-header table.

Multi-table type is a special type of the table, that is composed from more than one table from the previously mentioned types.

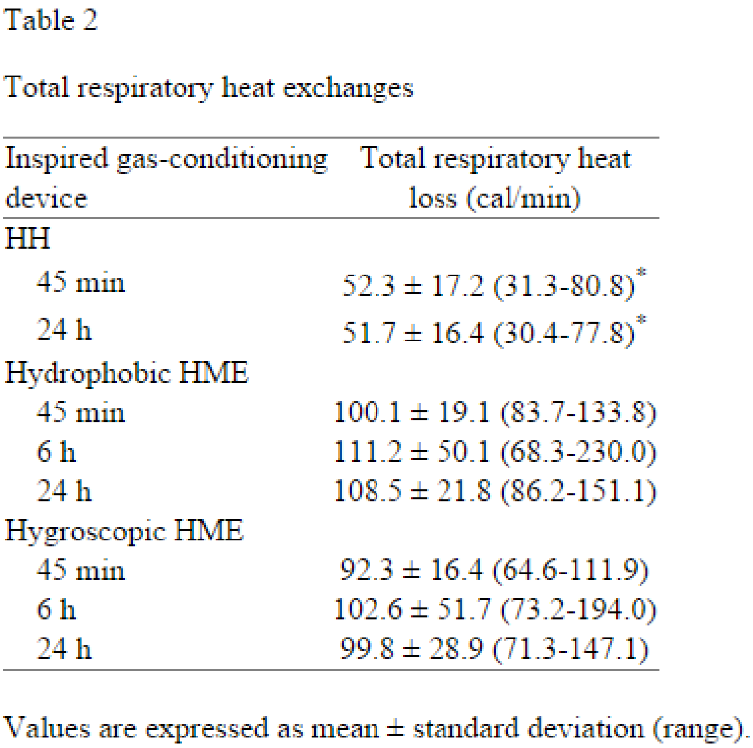


Table 2

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).

Figure 4.6: Example of sub-header table (PMC29053)

It is interesting to observe that a table with sub-headers may be transformed into matrix table with multi-column stub and vice versa without any information loss. The role of the sub-header is to group data and add some additional information. The same could be achieved with multiple levels of stub columns. Example of the transformation from the sub-header table into the matrix table with two column stub is shown in Figure 4.7.

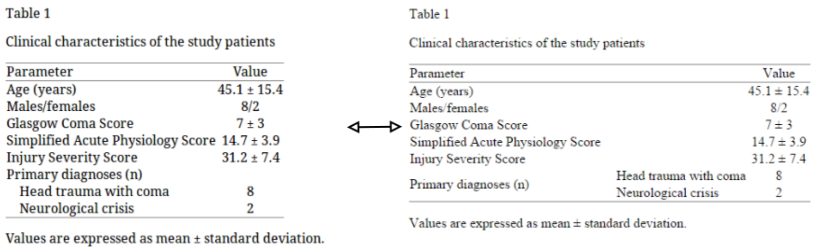


Table 1

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 4.7: Example of transformation from sub-header table into the matrix table with multi-level stub (PMC29053)

4.2.6 Challenges

During our analysis of tables in PMC database we have encountered four main challenges in processing tables. The challenges are:

1. **Variety of structural table layouts**
   In Section 4.2.5 we presented major structural table layout types and generalized reading paths for these layout types. Most of the tables can be approached in this way. However, these layout types can have their subtypes and on some tables these generalised way of reading cannot be applied. Also, some tables require a combination of different reading approaches (i.e. multi-tables). We have encountered two tables in analysed dataset which sub-header structure was complex and unclear, due to uncommon use of spacing. Tables that cannot be read using generalised reading approach are relatively rare, but very challenging and should be approached.

2. **Incorrect XML tagging in PMC database**
   In PMC database, not all XML tags are placed correctly. We have encountered two common types of incorrect tagging. One is incorrect tagging of header, while the other is incorrect tagging of spanning cells. Incorrectly tagged tables looks visually correct, but tags incorrectly placed. For example, some rows that are part of a header are not labelled as header rows or rows that are not part of the header may be labelled as a header rows. Also, instead of spanning cells can be just one cell with content (usually the first or central), while the other cells are empty. Since the table processing system mainly relays on tags, this issue may lead to misinterpretation of the table.

3. **Variety of value presentation patterns**
   In biomedical tables can be seen a vast variety of value presentation patterns. Some of these patterns are due to the cultural or regional differences (for example, whether someone would mark floating point with dot or comma symbol). However, most of the patterns are used as a means to present more data in a limited space. Since there is no standardisation on value presentation, same intention can be presented differently by different authors. For example, different authors may use different separator symbols. Some authors may use descriptive language, while the others will use formal and mathematical language. The way of presenting values in data cells is completely left on authors of the table. Because of this, a number of patterns that could be found in literature is large and

may be growing with newly published articles.

4. **Semantics of presented values**
   Semantics of processed text is one of the biggest challenges in text mining. This is also true for table mining. The meaning of value presented in cell is described in navigational path. However, it is challenging to link information from navigational path correctly, so no information is lost. Also, multiple values can be presented in one cell and it is challenging to determine meaning of each one. Some of these values may be explained explicitly in navigational area of the table, while the others are implicit and understanding of the field and common value presentation patterns is necessary. Even correct linking of explicitly presented values with its meaning described in navigational area of table may be challenging and may also require some knowledge from the field (For example, knowledge about acronyms, abbreviations and synonyms).

5. **Lack publicly available annotations**
   This challenge was spotted during literature review and also while experimenting. There are no publicly available annotated dataset for tables, although some of the authors used machine learning for table detection and functional analysis. These authors were building their datasets on their own and most of them have not released their data. Because of this, we would need to build our annotated dataset as well. Lack of annotated dataset have also led us to try rule based approach first.

Approaches to these challenges will be further examined during the project.

4.3 Table decomposition

4.3.1 Motivation

As we described in previous section, tables have various layouts and different means of presenting its values. For automated system it is hard to infer the meaning of the data in the table directly. We assume that if we decompose table into cell structures while retaining relationships with relevant navigational cells and table's caption and footer, it will be easier to perform further table mining. These cell structures have to contain all information that is needed to interpret table's data correctly in structured manner. Tables are already semi-structured part of the document, but the structure of the table is not standardised and it is hard to propose a single method that would successfully extract information from the tables. However, if we are able to transform table into the set of standardised structures, it would be easier to create a method that is able to perform various text mining tasks over table's data, including information extraction.

In order to extract data that is needed for the interpretation of the cell's value, we need to analyse how people interpret values in the table's cells and what information they need in order to interpret data correctly. Meta data for each cell are basically headers and stubs. However, if a table contains a sub-headers, all the levels of sub-headers need to be identified. Information about what the tables is about and how to interpret certain data types can be found in captions and footers of the table. Here we will present a method for decomposing a table into set of cell structures containing cell's value, headers, sub-headers, stubs, table's caption and footer.

4.3.2 Method

In order to build a method that will be able to process correctly tables from our dataset, we first conduct a manual analysis on a small sample of 50 PMC documents with 155 tables. During the development of our system we have also analysed additional 20 PMC documents. Based on this analysis we were able to create rules to identify structure and decompose tables in structured and standardised manner.

Our method is comprised of 6 steps. The first step is reading an article's metadata and then it is locating table in the article. The third step is to extract table's metadata, which is followed by a statistical analysis of the table's cells and table classification. The fifth step is merging of complex headers and complex stubs, while the last step is locating navigational paths and creating output. Work-flow diagram of our method can be seen in Figure 4.8.

4.3.2.1 Reading metadata

Our algorithm is reading the data about article and the tables at the first place. It finds tags in the documents that are containing document title, PMC number and author names. Similarly, it finds tables in the document. Also, it locates tables and its captions and footers. This information is stored in particular XML tags. Example of XML representation of table can be seen on Figure 4.9.

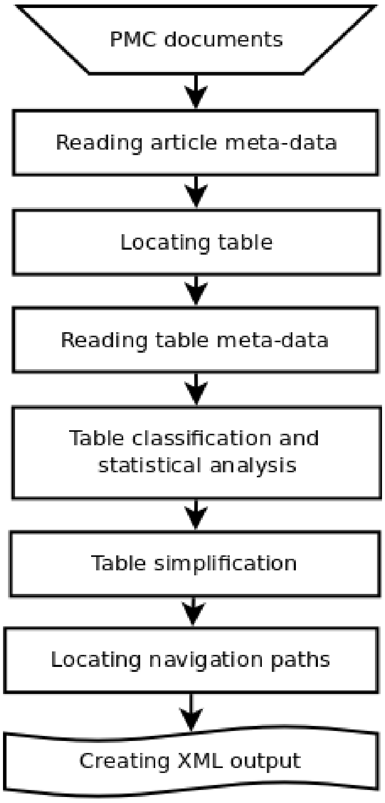

PMC documents
Reading article meta-data
Locating table
Reading table meta-data
Table classification and statistical analysis
Table simplification
Locating navigation paths
Creating XML output

Figure 4.8: Algorithm work-flow for data extraction of data from PMC documents

4.3.2.2 Table processing and statistical analysis

In this step, the algorithm is looking for the table XML tags inside table-wrap tags, which contain the actual table. However, we found some malformed tables that were created by joining two tables together. In this case, all the tables are read and labelled as they were inside the same actual table. Each cell is read and stored in a cell object with the relevant information that describes the cell's value, such as its header, stub, sub-header, table caption and footer. If the cell is spanning, the content of the cell is copied to all the cells it is spanning across (Chen et al., 2000). The order of the cell and the number of the cells that take part in spanning are also recorded. Cells that are inside the thead tags are labelled as header cells with probability 1. The most left column cells are labelled as the stub cells with 0.9 probability. This leaves chance that table is a list table, which does not have a stub.

Our algorithm needs to check whether tables that do not have *thead* tag really do not have the header. If the first row (or first couple of rows) have column spanning cells, the part of table until the first row without any column spanning cells is labelled as a potential header. If the table does not have spanning cells in the first row, the cell

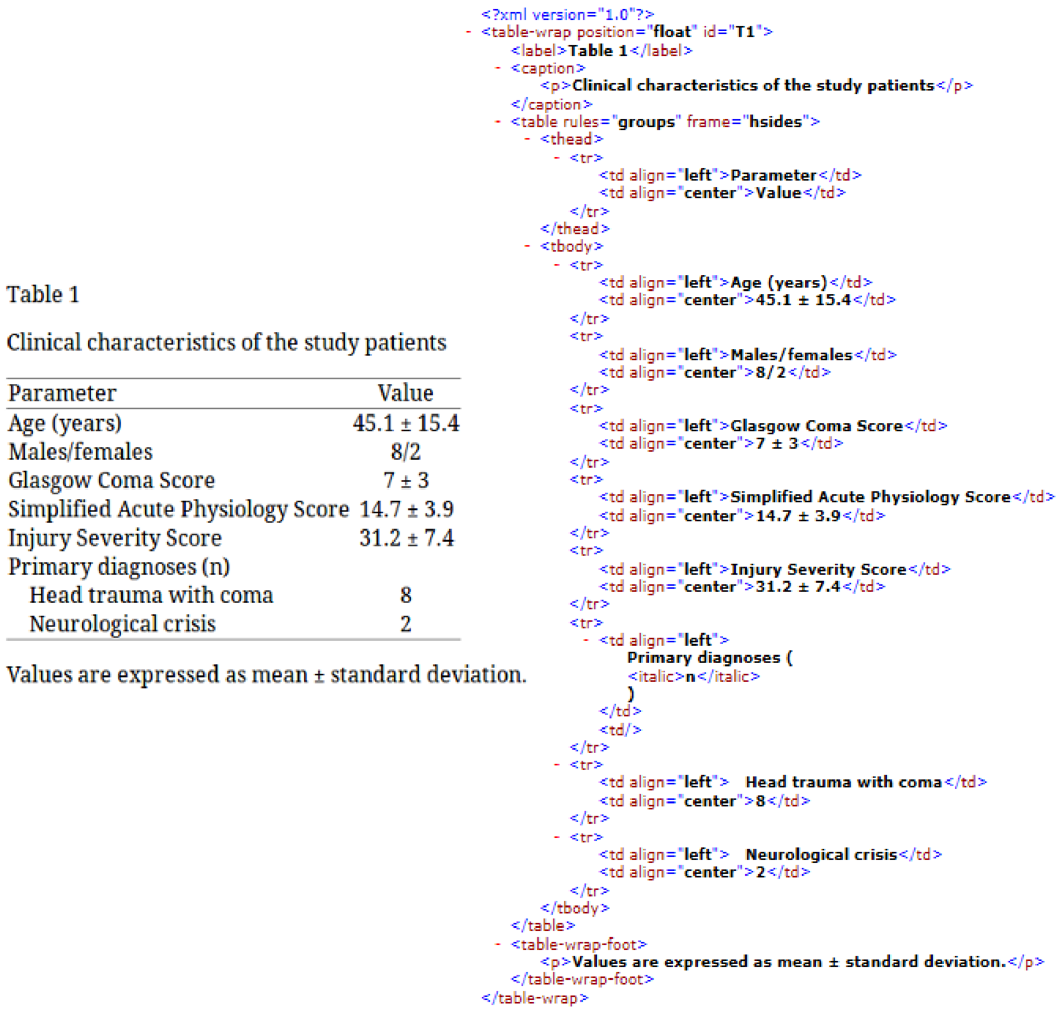


Table 1

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"?>
- <table-wrap position="float" id="T1">
    <label>Table 1</label>
  - <caption>
      <p>Clinical characteristics of the study patients</p>
    </caption>
  - <table rules="groups" frame="hsides">
    - <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>
        <tr>
        - <td align="left">
            Primary diagnoses (
            <italic>n</italic>
            )
          </td>
          <td/>
        </tr>
      - <tr>
          <td align="left">  Head trauma with coma</td>
          <td align="center">8</td>
        </tr>
      - <tr>
          <td align="left">  Neurological crisis</td>
          <td align="center">2</td>
        </tr>
      </tbody>
    </table>
  - <table-wrap-foot>
      <p>Values are expressed as mean ± standard deviation.</p>
    </table-wrap-foot>
  </table-wrap>
```

Figure 4.9: Example of table (PMC29053) and its XML representation

type similarity between first and the next four cells in the column is measured. If the first row cell in any of the column has different data type than other four cells in the same column, the first row is assumed to be the header. We assume that data cell will be consistent in type over the column and chose to use four column cells to test it.

During the reading of the table, statistical analysis is conducted over the tables. We calculate number of cells in the head, in the body of the table, number of empty cells and spanning cells. Also, statistics are conducted over the content of the cells. We have defined 4 types of the cells. Pure numerical cells are cells that contain one real number. Partially numeric cells are the cells that contains more than 50% of numeric characters. These cells usually present ranges, percentage values, value sequences, values with unit at the end or values with standard deviation. Textual cells are cells that contains more than 50% of non numeric characters. Empty cells are cells without any value or with only space characters.

Our algorithm, in this step, filters out tables that have reference to the image file. Tables in images are out of scope of our research since processing of such tables would require optical character recognition for cell detection and content extraction.

4.3.2.3 Header and stub merging

A step that is also needed is merging of headers and stubs. Many tables have headers that spread over several rows and have spanning cells. Our aim is to internally represent all the tables in order to have only one header row and one stub column. Our algorithm is copying the content of the cells in higher rows to the cells in lower cells, adding the content of the higher row cells before the content of the lower row cells. The copied higher rows of the header are deleted, so at the end of the process, there will be only one header row. This logically makes sense because higher row cells are just specialized with lower row cells inside the header (Wang, 2013). Process of merging headers on an example can be seen in Figure 3.3 Usually, stub is the left most column. However, in some cases, tables have in the left most column row spanning cells, that is specialized with the second column. In PMC files we looked at, often row spanning is not used, but there is the first row of the row spanning cell populated with the value, while other cells are empty. The second column is, however, filled with values, that are specializing the first column value. In order to deal with this situation, our application in such cases will merge first two columns. During the merge process, we copy the value from the first column cell that is not empty to the sequential cells that are empty, until next non-empty left-most cell. While merging stub, the most left column content is copied to the cell on the right before the value of the cell in the second column. After the values are generated in the second column, our algorithm deletes the first column.

4.3.2.4 Navigational path detection and linking

After the tables headers and stubs are merged, it is possible to extract information. Our algorithm makes XML document for each cell. This XML document contains the information and meta-data needed for semantic understanding of that cell. The cell document contains the title of the article from which was table extracted, ordinal number of the table in document, table's caption, table's footer, values in cell's navigation path (headers, sub-headers, stub values), cell value and PMCID. Also we recorded

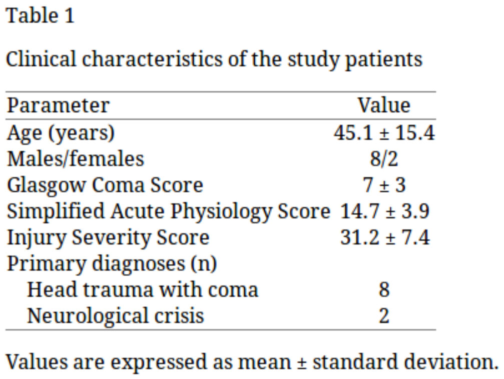


Table 1

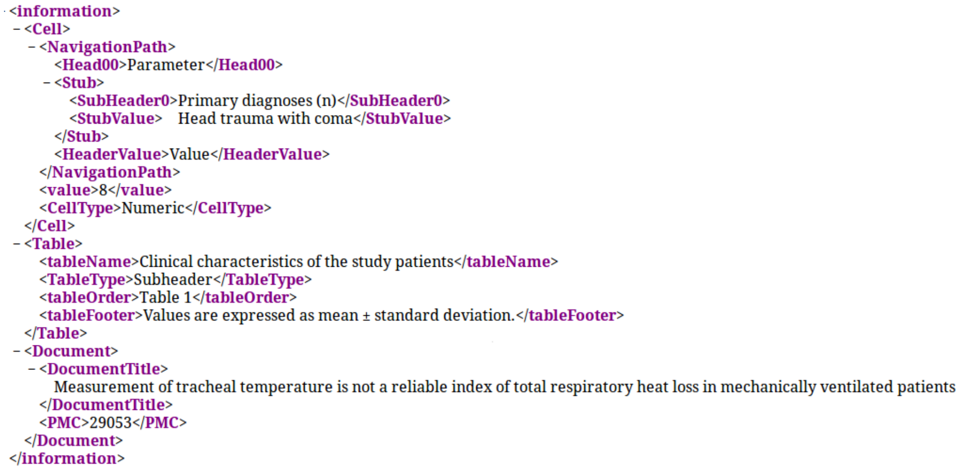
Clinical characteristics of the study patients

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: Example of the input table (PMC29053)

type of the table (matrix, sub-header, list) and type of the cell (pure numeric, partially numeric, text, empty).



```
<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 4.11: Application XML output for one table cell from table shown in Figure 4.10

While some of this information is easy to retrieve, it is much harder to retrieve data about navigation paths. Our algorithm is iterating through all the cells that are not part of the header and stub. While doing that, the algorithm is trying to find the correct navigation path to the cell. We have defined three types of tables that has to be approached differently.

List tables are tables that contain a list of items. They are usually not hard to read. They can be recognized if the header is spanning through all the columns or if the one value repeats in all header cells. In this type of table there is only header value that is in the navigation path. Multiple columns can be used in this kind of table for the space saving reasons (see example in Figures 4.3 and 4.4).

The second type of tables are matrix or regular tables. Matrix tables may have complex headers or stubs, but the body of the table is arranged in simple matrix of cells (see example in Figure 4.5). We may identify two sub-types. The first sub-type does not have any value in the most upper left cell. These tables are ”connecting” the values in the header and the stub. The second sub-type has a value in the most upper left cell. This value is mainly describing stub values, but in some rare cases it can also describe the header values. The algorithm has only to read the header cell in the same column as the given cell, the stub cell in the same row as the cell and, if exists, the most top left cell in order to get navigational paths.

The third type are tables with sub-headers. Generally these tables are similar to matrix tables, but the presence of sub-headers changes the way these tables are read (see example in Figure 4.6). Since the sub-header table may have a number of sub-header levels in a tree-like structure, we created a stack data structure that stores current sub-header path. When the algorithm is reading a table with sub-headers, firstly it reads a header value for the stub (which might or might not exist, similarly to matrix tables), then it should read all levels of sub-header above the item of interest, the stub value and cell's value.

Data retrieved from the tables are stored in the XML elements and stored in XML file. Example of the file is shown on Figure 4.11. This format is influenced by the format used by Wei et al. (Wei et al., 2006).

During the creation of the cell documents empty rows are ignored (XML cell documents for them are not created).

4.3.3 Results

We have run our application on all 1125 extracted PMC clinical trial documents. Our method was able to extract data from 3376 tables. Since each table has on average about 80 cells, it would be impossible to evaluate whole dataset. We have therefore chosen 30 random tables and evaluated their output manually. We also used this output to create a golden standard for the next evaluations.

Evaluation was performed manually by inspecting every table and its cell documents for correctness. We have evaluated correctness of each tag in outputted cell documents, but also output documents as a whole. In the first step of evaluation, each output tag is inspected if it is correctly extracted. If only in one XML cell document some tag is not read correctly, table is labelled as incorrectly decomposed. Based on this inspection we calculate accuracy for each tag. In the second step of evaluation we took more strict approach. If any of the output tags in some cell document is incorrect, whole table is labelled as incorrectly decomposed.

First we will examine results of the evaluation for each tag in the cell document.

There are 3 tag groups in our document. One tag group describes document in which table is. Information that this tag contains are document title and PMC id. This information is read from particular XML tag in the document and they always exist. Because of this property they are always correct.

The second tag group describes tables. It contains table caption, table's ordinal number in the document, footer and table type. Table's caption, ordinal number and footer are also extracted from the particular XML tags and they are always correct in our evaluation set. However, the table type is calculated by our algorithm. This tag's value was not correct in three tables in our evaluation set, so it had accuracy of 90% in our evaluation set.

The last tag group is a group that describes cells. This group contains header, stub and sub-header values. In our evaluation set we had two tables which were wrongly read. One because of its formatting (*thead* was labelling something that was not header in table), while the other was not recognized as a sub-header table (it was recognized as a matrix table). However, we had one more table, which was generally correctly read, but missed last two rows. In the last two rows there was no values in the stub, while in some of the cells there were values in the brackets. This table is shown in Figure 4.12.

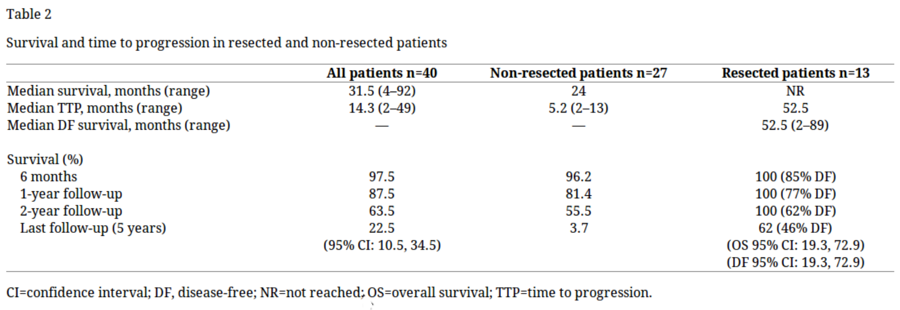


Table 2

Survival and time to progression in resected and non-resected patients

	All patients n=40	Non-resected patients n=27	Resected patients n=13
Median survival, months (range)	31.5 (4–92)	24	NR
Median TTP, months (range)	14.3 (2–49)	5.2 (2–13)	52.5
Median DF survival, months (range)	—	—	52.5 (2–89)
Survival (%)			
6 months	97.5	96.2	100 (85% DF)
1-year follow-up	87.5	81.4	100 (77% DF)
2-year follow-up	63.5	55.5	100 (62% DF)
Last follow-up (5 years)	22.5	3.7	62 (46% DF)
	(95% CI: 10.5, 34.5)		(OS 95% CI: 19.3, 72.9)
			(DF 95% CI: 19.3, 72.9)

CI=confidence interval; DF, disease-free; NR=not reached; OS=overall survival; TTP=time to progression.

Figure 4.12: Table example that was not read correctly - last two rows not read (PMC2360439)

In the cell tag group we had three tables that had wrongly read navigational paths. The accuracy of our method in this tag group is 90%.

Our final step of evaluation is also the most strict one. Here we have evaluated how

Mean number of cells per table	80.93309
Mean number of header cells per table	8.24846
Mean number of header rows per table	1.4758376
Mean number of columns in table	5.2184873
Mean number of rows in table	15.694118
Mean number of chars in cells	7.138295
Mean number of chars in partially numeric cells	8.499677
Mean number of chars in pure numeric cells	2.4207056
Mean number of chars in numeric cells	4.1323333
Mean number of chars in non empty cells	8.564969
Tables without head (thead tag)	197
Tables without body (tbody tag)	2
Tables with no XML representation (reference to image)	559 (16%)
Tables with colspans	1746 (51.7%)
Tables with rowspans	52 (1.5%)
Matrix tables	1636 (48%)
List tables	40 (1.1%)
Sub-header tables	1749 (51.8%)
Tables processed	3376

Table 4.1: Statistics about tables in our PMC clinical trial XML dataset

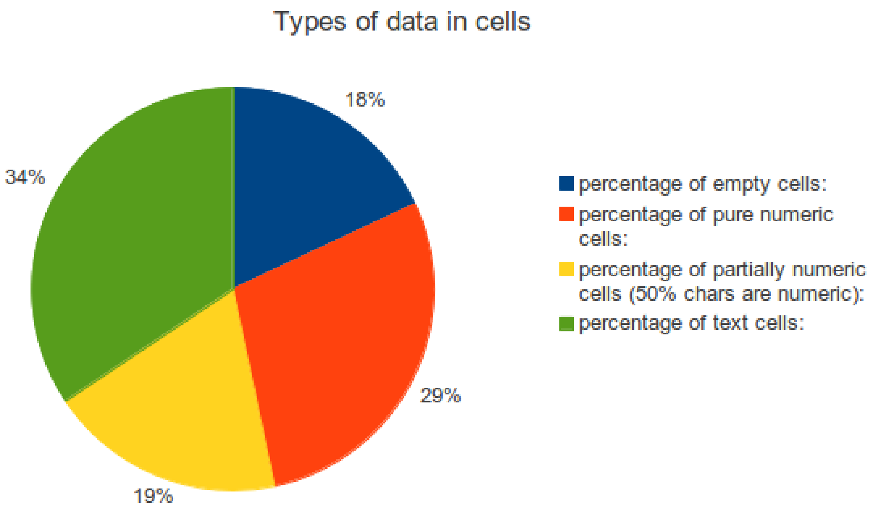


Figure 4.13: Statistics of cell data types

many tables had an error in any of the fields. There were four incorrectly processed tables in our evaluation data. Overall accuracy of our system is 86,67%.

We also calculated statistics for tables and data types that were presented in our tables. Our software calculated that there are 48% of matrix tables, 1% of list tables and 51% of sub-header tables. We may here make an assumption that sub-header type of tables is popular and suitable for presenting information in biomedical documents. They are especially suitable because they provide a mechanism to present high number of dimensions in a compact and systematic way. Detailed statistics about tables in our dataset can be seen in Table 4.1. There is a difference between number of tables processed and the total number of matrix, list and sub-header tables. In some cases multiple tables can be stored in the same table-wrap tag. There were 49 tables of this kind. Table processed statistic shows the number of table-wrap tags that can be processed, while the statistic about matrix, list and sub-header tables operates with data that were stored inside table tags.

4.3.4 Conclusion and future work

Here, we have presented system for decomposing tables to the cell level, while retaining the information about relationships between cells. This system is primarily build for the decomposition of biomedical tables from PMC database. However, it does not use any domain specific resource, in principle it can be applied to the tables in other domains.

This work is the first step on building a full scale table information extraction system. In this step we extracted the data for each cell that would be sufficient for its semantic understanding. However, semantic normalization and linking of the cells at this stage is not developed, but we are planning to perform further research in this area.

Currently, as input, our algorithm is expecting particular XML format, but this can be easily extended. PDF files could be supported using tools like PDFX (Constantin et al., 2013), which is able to convert PDF documents to XML and HTML.

This work was also heavily motivated by obtaining more knowledge about biomedical tables and building a system that will obtain statistics about biomedical tables. The system was able to extract statistics from the dataset and gave us an overview about tables in the biomedical domain. Based on this knowledge we were able to identify more efficiently challenges of table mining and extracting information from tables.

Currently our system does not support tables that do not have header and body tags defined. These type of tables were not common in our PMC dataset, so we decided to ignore them at the moment.

Accuracy of our system is promising and can be possibly improved with the use of additional heuristics and some domain specific resources.

4.4 Case study: Information extraction of BMI and number of patients per arms

4.4.1 Aim

To fully understand the data presented in the tables we need to translate data expressed in tables to a structured data. Information extraction from tables is challenging because tables may contain chunks of text, abbreviations, acronyms and various visual layouts. The visual layout of the table is often describing relationships between the items in the table. The algorithm for information extraction has to be aware of cell position and its navigation path. Complexity of table structure may play also an important role in the extraction of the information from tables.

The aim of this experiment was to examine information extraction from tables. Two case studies were performed. The first case study had an objective to extract the total number of patients and total number of male and female participants in a clinical trial. The second case study had an objective to extract body mass index and weight of patients participating in a clinical trial. In this task, the participant group names of clinical trials had to be extracted as well with the appropriate mean body mass index or weight of the participating patients. For example, table shown in Figure 4.14 has two participant groups. Information extracted from this table will be: Absolute Rist (n = 232): BMI: 27.4 (4.5) and NNT (n = 225): BMI: 27.0 (4.3).

The case studies were performed as a step towards information extraction of demographic data from the tables in clinical trial documents. Demographic tables are one of the most common in the clinical trial articles and they can provide important information about the settings of the experiment.

4.4.2 Methods

4.4.2.1 Extracting a total number of patients

Extracting number of patients from clinical trial may be considered relatively simple task. The number of patients is almost always a numerical value and there is a limited

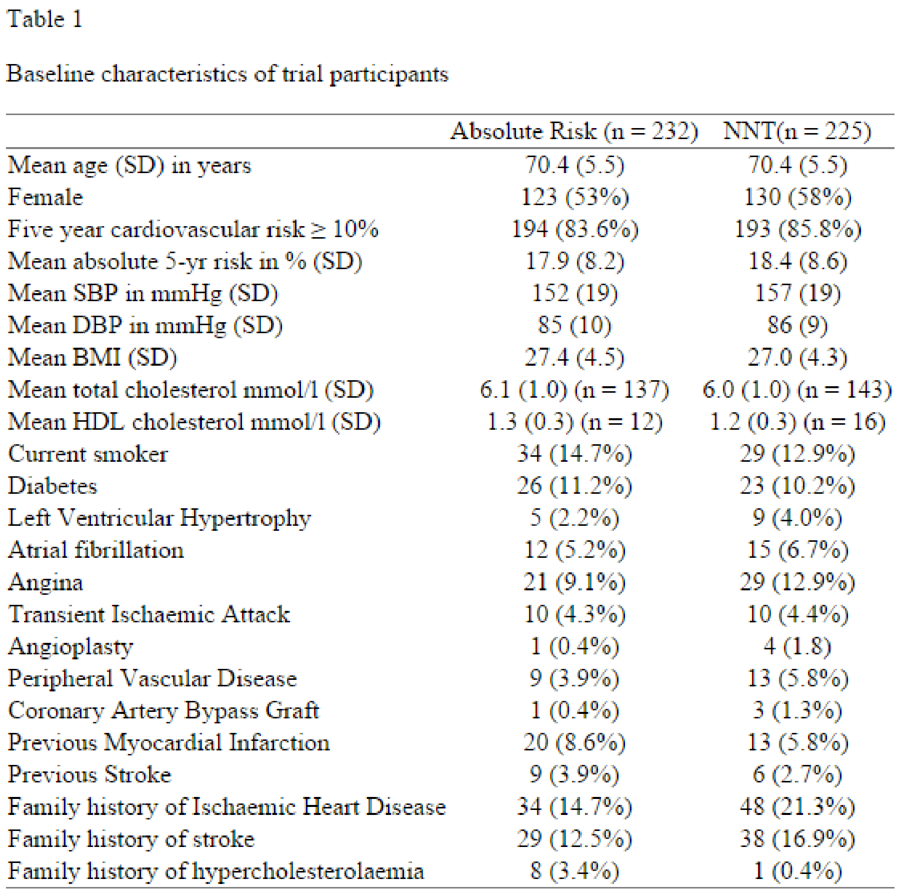


Table 1

Baseline characteristics of trial participants

	Absolute Risk (n = 232)	NNT(n = 225)
Mean age (SD) in years	70.4 (5.5)	70.4 (5.5)
Female	123 (53%)	130 (58%)
Five year cardiovascular risk ≥ 10%	194 (83.6%)	193 (85.8%)
Mean absolute 5-yr risk in % (SD)	17.9 (8.2)	18.4 (8.6)
Mean SBP in mmHg (SD)	152 (19)	157 (19)
Mean DBP in mmHg (SD)	85 (10)	86 (9)
Mean BMI (SD)	27.4 (4.5)	27.0 (4.3)
Mean total cholesterol mmol/l (SD)	6.1 (1.0) (n = 137)	6.0 (1.0) (n = 143)
Mean HDL cholesterol mmol/l (SD)	1.3 (0.3) (n = 12)	1.2 (0.3) (n = 16)
Current smoker	34 (14.7%)	29 (12.9%)
Diabetes	26 (11.2%)	23 (10.2%)
Left Ventricular Hypertrophy	5 (2.2%)	9 (4.0%)
Atrial fibrillation	12 (5.2%)	15 (6.7%)
Angina	21 (9.1%)	29 (12.9%)
Transient Ischaemic Attack	10 (4.3%)	10 (4.4%)
Angioplasty	1 (0.4%)	4 (1.8)
Peripheral Vascular Disease	9 (3.9%)	13 (5.8%)
Coronary Artery Bypass Graft	1 (0.4%)	3 (1.3%)
Previous Myocardial Infarction	20 (8.6%)	13 (5.8%)
Previous Stroke	9 (3.9%)	6 (2.7%)
Family history of Ischaemic Heart Disease	34 (14.7%)	48 (21.3%)
Family history of stroke	29 (12.5%)	38 (16.9%)
Family history of hypercholesterolaemia	8 (3.4%)	1 (0.4%)

Figure 4.14: Example of clinical trial demographic table that contains information about patients BMI (PMC58836)

number of trigger words.

The number of patients could be presented in different places in the table. They may be a part of table's caption, part of a header or stub value or a value in the data cell. Also, the total number of patients in a clinical trial may not be presented as a single (overall) number, but may be presented as a number of patients per arm of trial. The total number of patients may or may not be included.

Our system is looking in all of these places for the total number of patients. If it does not find a total number of patients, but it does find partial numbers (ie. per trial arm) it sums up these values. Our approach also used assumption that the demographic table is one of the first tables in the document. So when the value for number of patients is found, our algorithm would assume that once it found demographic table and would not process other tables in the document.

Total number of patients in the table caption

The table caption is usually one or more sentences of text with some basic information about what is the table about. Since the caption contains unstructured text, we applied a rule-based approach for extracting the total number of patients. There are only 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 first next word to the number, since in some cases authors want to more specifically identify the participants (e.g. 16 1-month-old infants, 1239 blood donors). However, a number, closely followed by some of the defined trigger words, usually determines the total number of patients in some clinical trial. The second rule is looking for a pattern consisting of letter n, the equals sign and a number (e.g. n=19). In the table caption is usually presented the total number of the clinical trial participants. If algorithm found the required value in the caption it may stop looking further.

Number of patients in the navigational cells

There are several ways to store number of clinical trial participants in a navigational cells of the table. One way is to store the total number of patients in a stub. Other ways include storing it in the header. However, header cells may describe name of clinical trial arms and the numbers in the headers may represent the number of participants in the each trial arm. Also, there might be a column with the total number of participants. In the most left and the most top cell may be also placed a total number of participants.

Usually, in stubs and headers number of patients are presented in the form of equation starting with letter n (e.g. n=19). In stubs, we are often expecting the total number of patients in one cell. However, header may have these values in multiple columns. All columns have to be taken in account in order to determine the total number of trial participants. For this we have created a list of candidates. Firstly, all the values are added to the list. Then the content is analysed. If content of some cell contained word ”overall”, ”total” or a phrase ”all patients”, the value in that candidate cell is considered as the total number of clinical trial participants. However, if the cell with described content does not exist, we check if the top-most and the most left cell has a value for number of patients. If it does, that value is considered the number of patient. If none of this is the case, the values from the header columns are summed, since the columns represent the arms of the trial (example of this can be seen in Figure 4.14). The sum is considered to be the total number of patients.

Number of patients in the data cells

Also, the number of patients may be placed in the body of the table. Similarly to headers, some tables may present the number of patients in parts (e.g. per arm), while the other may present the total number. Some tables may present both.

Since in the data cells are only the values, looking for trigger words and patterns has to be done in the appropriate stub cells. A number of trigger phrases have been defined, which are looked for in the stub. If found, values from the data cells that have the same pattern (containing a numeric value) are extracted and added to the list of candidates. Headers of the tables 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 total number of participants. If there is no such column, values are summed in order to calculate the total number of patients.

4.4.2.2 Extracting body mass index and weight

The second part of our case study is extracting information about body mass index and trial participant mean weight. This task also analyses demographic tables from clinical trial documents. However, this task is much more complex because we want to extract body mass indexes or weight, together with the participant group names in which these values were measured.

Extraction of information such as body mass index is more important than the total number of participants for the clinicians and drug discovery teams. It gives a relevant information about patient body on which the treatment or drug was tested.

For the body mass index extraction, we have experimented with several approaches, but the approach that worked the best were looking in the stub of the table for one of the two trigger phrases. One was ”body mass index” while the other was acronym ”bmi”. If table contains these trigger phrases, values from the table body are extracted. However, it is also checked if the value is in appropriate range. Body mass index should be in range between 15 and 40. If the value is not in this range, it does not represent mean body mass index value, but other value such as body mass index change, standard deviation, etc. Usually, mean value of body mass index is presented with the value of standard deviation separated by ± symbol. However, some tables might have standard deviation presented in a separate column. This column in header always have ”SD” or phrase ”standard deviation” which may be used as trigger words for extracting standard deviation. Standard deviation is usually the first column to the right of the column where the measurement mean values were presented. When the body mass index value is identified, algorithm looks at the header to identify the name of the arm. If there are more then one column with values of body mass index, the headers are probably the names of the participant groups. To identify header cells that are not representing participant group name values, list of trigger terms is created containing tokens such as ”range”,”p*”,” ”,”T”,”p-value”,”p* value”,”%”,”significance”. If there is only one column with body mass index, our algorithm assumes that the value in the column is the mean body mass index for all the patients.

Prior to extracting the names of participant group names using heuristic after finding the value for BMI, we tried to identify the participant group name using Metamap. Generally, it would be useful to have independent method for extracting trial group names, since various clinical trials may have different demographic data or measurements performed. Since the arms of medical trial are in the most cases names of treatments, we tried to used Metamap to extract phrases referring to semantic types that may indicate medical treatment. However, Metamap gave quite bad results in detecting trial arm names and treatments over the patients. It was already measured that Metamap and its variants, such as metamap+ are not very good at recognizing medical treatments, archiving only 56% F-score recognizing medical treatments (Abacha and Zweigenbaum, 2011). Because of the bad performance of Metamap, it have been decided to remain depend on demographic indicators and using heuristics to recognize patient groups names from the demographic tables. Using this heuristic it is not possible to obtain only arm names, but rather patients groups, since the authors may create demographic tables where they divide patient 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).

When the body mass index is identified, algorithm is able also to extract number of patients, number of male and female patients from the same table. Body mass indexes are presented in the same demographic tables as the number of patients are. Similar heuristics as described in 4.4.2.1 were used, just number of patients per group were not summed up, but extracted separately.

Similarly, weight of patients was also extracted. In this case trigger phrases were ”weight” and ”bodyweight”. Since, table could present a number of different measures related to weight, a stop list was introduced. Stop list has a role to discard entries if the stub contains some word from the list near the the trigger phrases. Stop list contained words like ”loss”, ”gain” and ”change”. Range of values were of great help while extracting the BMI, but range cannot be created for weight extraction. Weight can have values in different measurement units (g, kg, lb), so any value could be possible.

4.4.3 Results

4.4.3.1 Number of patients results

For the number of patients, we run our algorithm on all the documents in our clinical trial dataset. As we mentioned before, our dataset had 1275 documents, out of which from 758 were extracted some number as a total number of patients. For evaluation purposes we have randomly selected 50 documents. Our system performed with F-measure of 83.3%. More detailed statistics can be seen in table 4.2.

Precision	73.53%
Recall	96.15%
F-measure	83.3%

Table 4.2: Performance of extracting total number of patients from PMC clinical trial documents

4.4.3.2 Body mass index, weight and patient group name extracting results

For our second case study, which is able to extract body mass index, weight and patient group names we selected dataset that contain 113 documents. These documents contain in at least one of the tables some token related to body mass index or weight.

We have separately evaluated the performance of the patient group extraction, weight and body mass index. The results are shown in the table 4.3.

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

Table 4.3: Performance of extracting BMI, weight and patient groups from PMC clinical trial documents

From the table can be seen that we have quite promising results for extracting body mass indexes. These results could be further improved and are quite dependent on how well the patients groups were recognized because each extracted body mass index value has to be assigned to the extracted patient group. On the other hand patient groups could be quite hard to extract correctly. They may be formed from the wide range of concepts (location, drug, treatment, time, etc.) and may include acronyms. Complex tables with multiple levels of headers may create additional complexity, since it might be hard to determine where name of the group ends and where technical or statistical separation of the table's cells starts (ie. to mean and standard deviation columns).

Body mass index has a greater F-measure than a patient group extraction. This may look strange, because in order to extract body mass index, patient group has to be extracted as well correctly. However, defined body mass index range made a large contribution to discarding false patient group names in this task.

Our method for the weight extraction performed with hight recall but with very low precision. This is due to the fact that the method was matching trigger phrases, but did not have well enough crafted stop list and did not model enough knowledge that may help it to separate what is actual patient weight from the measurement of other weight related concepts. In our future work, we will try to improve current method, especially focusing on its precision.

This method after finding body mass index or weight of the patient could also extract patient ages, number of males and number of females. However, currently we have focused on the extraction of body mass index and weight and the data about these information classes were not evaluated.

4.4.4 Conclusion and discussion

Information extraction from table is the topic that is not extensively researched. However, in many fields such as biomedical it is very useful, because of the importance of the data presented in tables. Information extraction from tables is using some of the well established techniques of information extraction from unstructured text. However, because of the new challenges of coping with the visual layout, new approaches and heuristics should be developed.

This research, so far, was going towards information extraction by developing tools that will lead to a full scale clinical trial and biomedical table information extraction system. During the analysis of the clinical trial tables from PMC documents we identified some of the challenges in table mining. These challenges are variety of table visual layout as well as the variety of ways in representing dimensions or values in the tables. Afterwards we were able to build a tool that is able to read table data with associated navigational data. On these structures it is possible to perform more complex semantic analysis and extract knowledge and information. We performed a case study in which we extracted total number of patients, body mass indexes and names of the clinical arms in particular clinical trial. This research indicated that some of the information classes may be easier to extract, because for some information classes it is easier to craft rules and model expected values. However, some of the information classes remain a challenge.

The evaluated performances of these tools are quite promising, but they leave a space for advancement. Our case study results showed that our system is able to help data curation from clinical trial publication. Results are still not good enough for machine to perform all the work alone, so there is still need for the human curator to control the system and correct possible mistakes. However, we believe our system will reduce time needed for performing data curation from medical documents.

4.5 Data curation and evaluation system

4.5.1 Motivation

Most of the information extraction systems are not reliable enough to be used without human control and possibility of human correction. Only human experts are able to assure that the extracted data are of sufficiently quality to be entered into database (Alex et al., 2008). Natural language processing and especially information extraction can however help to reduce the time human curator needs to capture required data from the article. It is reported that these saving may range from 30% (Alex et al., 2008) to 70% (Donaldson et al., 2003). These are large enough saving to motivate research in information extraction, but also to motivate building of semi-automated curation interfaces.

Evaluation of information extraction is a similar process to semi-automated curation. However, in the evaluation, human should strictly evaluate whether system extracted information correctly. Performing this process manually, using database interface and scientific articles are quite time consuming process. This laborious task motivated us to propose an evaluation and data curation interface, which is able to speed up both processes.

4.5.2 System architecture

Data curation interface should provide curator or evaluator quick access to the extracted and original data. MedCurator is proposed software that contain an easy to use interface supplying user needed data for data curation from tables. The system is using the same database as the information extraction engine. It reads and presents extracted data together with the original ones to the user. User than may control, edit or delete extracted data. The workflow of the system is presented in Figure 4.15.

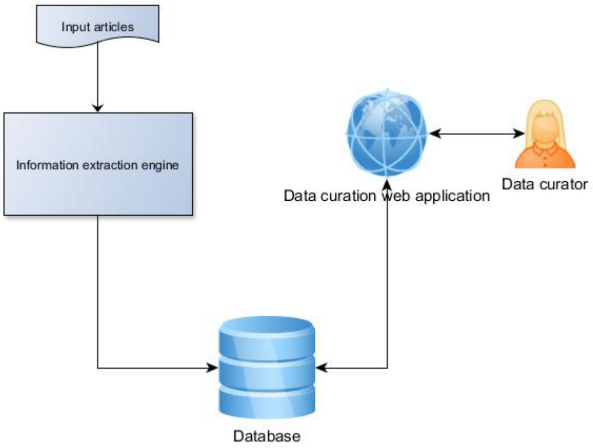
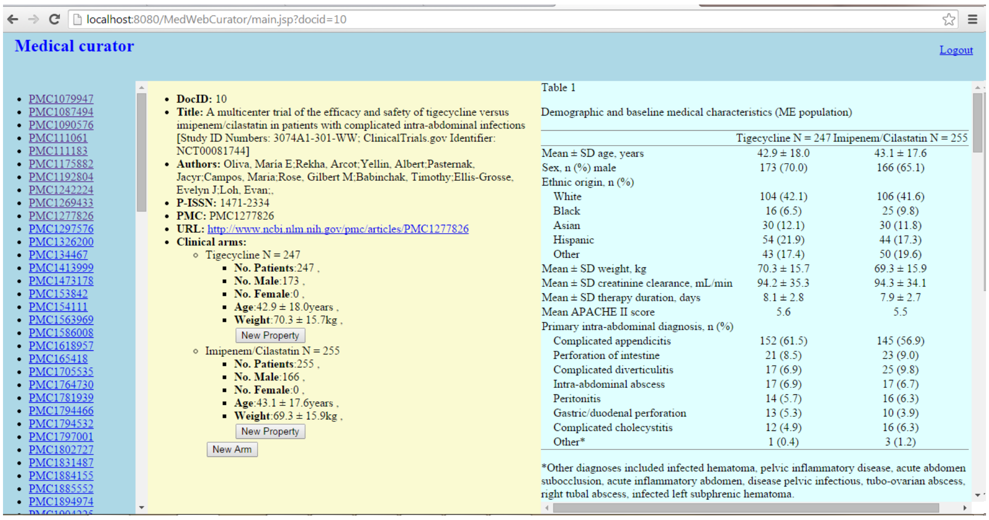


Figure 4.15: Workflow of semi-automated data curation system

System is designed to provide a user with view where he may see extracted information and the original tables on one page. The curation page is organized in three sections. In the first section are listed all the documents on which information extraction was run. User may click on a document and other two views will change with the data from that document. The second view shows extracted information. First it shows basic information about document such as document name, author names, PMC number, ISSN number etc., followed by extracted information from the participant groups. These information user may edit, delete or add new information to the database if it is missing. The last view shows all the tables that were identified in the analysed article. MedCurator data curation page can be seen in a Figure 4.16.



- PMC1079947
- PMC1087494
- PMC1090576
- PMC111061
- PMC111183
- PMC1175882
- PMC1192804
- PMC1242224
- PMC1269433
- PMC1277826
- PMC1297576
- PMC1326200
- PMC134467
- PMC1413999
- PMC1473178
- PMC153842
- PMC154111
- PMC1563969
- PMC1586008
- PMC1618957
- PMC165418
- PMC1705535
- PMC1764730
- PMC1781939
- PMC1794466
- PMC1794532
- PMC1797001
- PMC1802727
- PMC1831487
- PMC1884155
- PMC1885552
- PMC1894974

- **DocID:** 10
- **Title:** A multicenter trial of the efficacy and safety of tigecycline versus imipenem/cilastatin in patients with complicated intra-abdominal infections [Study ID Numbers: 3074A1-301-WW; ClinicalTrials.gov Identifier: NCT00081744]
- **Authors:** Oliva, María E;Rekha, Arcot;Yellin, Albert;Pasternak, Jacyr;Campos, Maria;Rose, Gilbert M;Babinchak, Timothy;Ellis-Grosse, Evelyn J;Loh, Evan;,
- **P-ISSN:** 1471-2334
- **PMC:** PMC1277826
- **URL:** http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1277826
- **Clinical arms:**
  - Tigecycline N = 247
    - **No. Patients**:247 ,
    - **No. Male**:173 ,
    - **No. Female**:0 ,
    - **Age**:42.9 ± 18.0years ,
    - **Weight**:70.3 ± 15.7kg ,
  - Imipenem/Cilastatin N = 255
    - **No. Patients**:255 ,
    - **No. Male**:166 ,
    - **No. Female**:0 ,
    - **Age**:43.1 ± 17.6years ,
    - **Weight**:69.3 ± 15.9kg ,

Table 1

Demographic and baseline medical characteristics (ME population)

	Tigecycline N = 247	Imipenem/Cilastatin N = 255
Mean ± SD age, years	42.9 ± 18.0	43.1 ± 17.6
Sex, n (%) male	173 (70.0)	166 (65.1)
Ethnic origin, n (%)		
White	104 (42.1)	106 (41.6)
Black	16 (6.5)	25 (9.8)
Asian	30 (12.1)	30 (11.8)
Hispanic	54 (21.9)	44 (17.3)
Other	43 (17.4)	50 (19.6)
Mean ± SD weight, kg	70.3 ± 15.7	69.3 ± 15.9
Mean ± SD creatinine clearance, mL/min	94.2 ± 35.3	94.3 ± 34.1
Mean ± SD therapy duration, days	8.1 ± 2.8	7.9 ± 2.7
Mean APACHE II score	5.6	5.5
Primary intra-abdominal diagnosis, n (%)		
Complicated appendicitis	152 (61.5)	145 (56.9)
Perforation of intestine	21 (8.5)	23 (9.0)
Complicated diverticulitis	17 (6.9)	25 (9.8)
Intra-abdominal abscess	17 (6.9)	17 (6.7)
Peritonitis	14 (5.7)	16 (6.3)
Gastric/duodenal perforation	13 (5.3)	10 (3.9)
Complicated cholecystitis	12 (4.9)	16 (6.3)
Other*	1 (0.4)	3 (1.2)

*Other diagnoses included infected hematoma, pelvic inflammatory disease, acute abdomen subocclusion, acute inflammatory abdomen, disease pelvic infectious, tubo-ovarian abscess, right tubal abscess, infected left subphrenic hematoma.

Figure 4.16: MedCurator data curation screen

Since MedCurator is an web application, for the security reasons, curator page is protected by login mechanism. Also, registration mechanism is currently enabled. However, for real life usage scenario of this application, only users that have been approved as data curators may use edit/delete/add option.

4.5.3 Discussion

Information extraction systems are getting better and better. However, until text mining system do not become totally correct in extracting information, the need for human curator will remain. They are needed to check extracted data and correct where it is necessary. The current state-of-the-art techniques do not give much insight that text mining system will be totally correct in the near future. There are still a lot of challenges for biomedical text mining, such as data ambiguity, data quality or semantic mapping and domain knowledge integration. These challenges are addressed by the text mining research. However, until complete solution could be proposed, human intervention in data curation systems is crucial. On the other hand, current quality of text mining systems are greatly improving the data curation process by speeding up the process and improving the quality of retrieved data.

Semi-automated data curation from tables, similarly to table information extraction have not received the needed attention. This led to motivation create MedCurator, which is able to give user the possibility to evaluate data mined from the tables, as well as correcting them. This system have also saved us a valuable time during the evaluation of our table information extraction system.

In the future we plan to get some feedback from the data curators about MedCurator and according to their suggestions to improve the system. System could be expanded not only to support data curation from tables, but also from the text. By supporting both text and tables, this system may become the main tool data curators need for extracting data from clinical (and biomedical in general) papers. Also, currently system is using mySQL database, but our whole system should work using linked data and triple store. In future MedCurator will be updated to use triple store to browse and update extracted data.

4.6 Summary

Our current work was focused on gaining more insight about tables in biomedical literature. We have performed an analysis of sample of tables from clinical literature in order to learn common structural and information presentation characteristics. The analysis also identified challenges in tables mining:

1. Incorrect XML tagging in PMC database

2. Variety of structural table layouts that may requite different approaches to information extraction.

3. A variety and ambiguity of value presentation patterns

4. Semantic understanding and linking of presented data to their meaning which require significant background knowledge

5. Lack of publicly available annotated data

We have also developed a rule based method for decomposing tables into the data cell structures containing all the relevant information needed for further semantic analysis. The method uses heuristics about the position, spacial features of the cells, content of the cell and its neighbouring cells. The method was evaluated and showed promising accuracy of 86,6%.

Further, we also performed an information extraction case study on clinical trial documents. In this case study, our method was extracting body mass indexes and weights of patients. We have created a rule-based system for extracting these information classes. Evaluation showed promising results for BMI extraction (F measure 83.7%), but also indicated that some classes are harder to extract (patient weight F measure was 57.7%).

For evaluation purposes we proposed a system that enables quick and intuitive comparison of original and extracted data. We believe that the proposed system also enables professional curators to evaluate and, if necessary, edit the extracted data.

Chapter 5

Summary and Future Directions of Research

5.1 Summary

The aim of this project is to develop methods for the information extraction from tables in biomedical literature and provide a data curation method for these data. Most of the current work in table mining is focused on table detection or structural and functional analysis of tables. However, semantic understanding and information extraction from tables so far had not attracted enough attention. We will aim to develop methods to process tables and transfer knowledge and information to a structured data format (database or linked data). This methods would assist semi-automated information extraction and data curation from clinical publications. We will initially focus on clinical trial documents, as an important source of information for the development of medical treatments and medications.

The method will be able to recognize tables in biomedical scientific publications, detect navigational paths for each cell, semantically analyse data from the cells and its navigational paths using UMLS resources, open biomedical databases and the open linked data cloud. A semantic analysis method should enable mapping between data in data cells and its navigational cells. At the end, inference about data in a table should be performed and information and knowledge should be extracted.

In the first year we have studied a number of topics:

- General text mining techniques
- Text mining techniques in the biomedical literature, including open biomedical

resources for semantic tagging and semantic analysis

- Work done on table processing from the web and scientific literature
- Natural language processing libraries such as OpenNLP, NLTK and Apache Lucene
- Machine learning tool kits and frameworks such as Weka

Furthermore, we have worker the following:

- **Performed analysis of tables** in clinical trial documents. The analysis was performed on 155 tables from 50 randomly selected documents. We have identified four main types of tables: list, matrix, sub-header and multi-table tables. We also analysed common value presentation patterns. In addition, the following challenges for table mining in biomedical literature were identified:

  1. Multidimensionality and variety of structural table layouts that may requite different approaches to information extraction.
  2. A variety and ambiguity of value presentation patterns
  3. Semantic understanding and linking of presented data to their meaning which require significant background knowledge
  4. Lack of publicly available annotated data

- **Developed a rule based method for decomposing tables** into the data cell documents containing relevant information needed for further semantic analysis. The method uses heuristics about position, spacial features of the cells, content of the cell and its neighbouring cells. The method showed accuracy of 86,6% evaluated on 30 randomly selected tables.

- **Performed a case study of information extraction** by creating rule-based system for extracting several information classes including body mass index (BMI), patient weight and number of patients. Evaluation showed promising results (e.g. BMI extraction showed F measure 83.7%), but also indicated that some classes are harder to extract (patient weight F measure was 57.7%), due to required background knowledge.

- **Proposed data evaluation interface** which provides an intuitive way of comparing original and extracted data. It also provides a mechanism to correct extracted data.

5.2 Future work

The current work done gave us some valuable insight on the information extraction and text mining of tables. In order to develop a method for the information extraction and knowledge management of the data stored in tables we have identified a number of steps:

- **Categorisation of value presentations in tables.** Values in the data cells of the table may be presented in the various different ways. Especially, if the cell contains multiple values, the way these values are presented may vary. Patterns of the value presentation may carry some important information about semantics of these values (e.g. two values separated with ± symbol are likely to be mean value and its standard deviation). We would like to categorise common data presentation patterns present in table. This would give us the list of common patterns and the overview of semantic knowledge these patterns carry. This information will be valuable in our pattern recognition building step.

- **Review and evaluation of open linked data resources, open biomedical databases and semantic resources in biomedicine.** In order to perform semantic analysis it is necessary to use resources with domain knowledge. There are a number of available resources in biomedicine that could be used. This part of research should give us an overview of the currently available open resources and how they could be used in semantic analysis of table data.

- **Cell normalization and semantic mapping.** Cells in the tables may contain abbreviations and acronyms, which have to be expanded. Also, it is easier to analyse data if the different synonyms of some concept is mapped to single name. These processing has to be done using NER techniques and knowledge sources such as lexicons, thesauruses, databases or linked data.

- **Pattern recognition and mapping.** As the result of this task we would like to map a number of information classes with the values in the navigational paths describing it. In order to do that, it is necessary to split values presented in the cell and map them with their meaning. This task should retain flexibility, so the users in future may add recognition and mapping of new patterns from the cells.

- **Storing and using information extracted from tables.** In this research phase we would like to examine methods for storing and querying the extracted data.

From the current perspective, triple stores and linked data seem to be the most appropriate choice, taking in account our flexibility requirements for the schema extensions. However, we may examine also relational databases and document stores (NoSQL databases) and its potential advantages for the parts of our system. After information is stored, users should be able to use it or even modify. For modification we would propose an user friendly interface in which only selected expert users can review extracted data and modify it. Also, we would propose an query interface for a broader group of users who would be only able to search for extracted information.

- **Linking information from the text with the information from the table.** In this phase we will examine linking of the data extracted using text mining techniques on the body of the article with the information extracted using table mining. This is necessary in order to obtain same amount of the information from the scientific article as the human reader would.
- **Expanding developed methods to other (non-clinical) biomedical literature.** The final step in this project will be to expand the scope from the clinical to more generalized category of biomedical documents. In this step we will research the possibility for adopting our methods to a wider range of biomedical literature and extract information and knowledge from them.

The proposed time plan is presented in Figure 5.1.
During the project we are planning to publish the following publications:

1. ”Decomposition of tables from PMC clinical trial documents”
   Target date: December 2014.

2. ”Analysis and categorisation of table and cell presentation patterns in biomedical literature”
   Target date: June 2015.

3. ”Automatic normalisation and analysis of cell values in the tables”
   Target date: October 2015.

4. ”Table information extraction from tables in biomedical documents”
   Target date: March 2016.

5. ”Data curation of information extracted from tables in biomedical documents”
   Target date: July 2016.

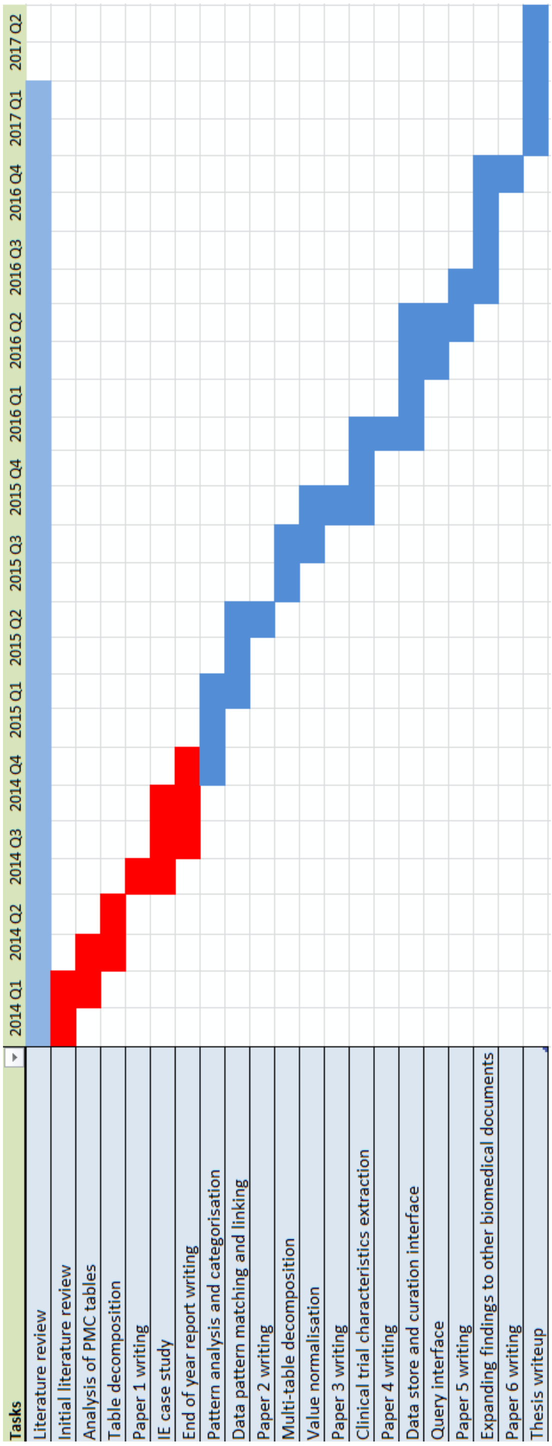


Figure 5.1: Gantt chart of a plan for this project.
Red bars represent the work done during the first year of the project. Blue bars represent an ongoing tasks or tasks that will be performed in the future.

6. ”Mining, curating and querying information from tables in biomedical literature”
   Target date: January 2017.

References

[Abacha and Zweigenbaum2011] Abacha, Asma Ben and Pierre Zweigenbaum. 2011. Medical entity recognition: A comparison of semantic and statistical methods. In *Proceedings of BioNLP 2011 Workshop*, pages 56–64. Association for Computational Linguistics.

[Abney1992] Abney, Steven P. 1992. *Parsing by chunks*. Springer.

[Aggarwal and Zhai2012] Aggarwal, Charu C and ChengXiang Zhai. 2012. *Mining text data*. Springer.

[Alex et al.2008] Alex, Beatrice, Claire Grover, Barry Haddow, Mijail Kabadjor, Ewan Klein, Michael Matthews, Stuart Roebuck, Richard Tobin, and Xinglong Wang. 2008. Assisted curation: Does text mining really help?. In *Pacific Symposium on Biocomputing*, volume 13, pages 556–567. Citeseer.

[Alley1996] Alley, Michael. 1996. *The craft of scientific writing*. Springer.

[Aronson2001] Aronson, Alan R. 2001. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

[Bizer et al.2009] Bizer, Christian, Tom Heath, and Tim Berners-Lee. 2009. Linked data-the story so far. *International journal on semantic web and information systems*, 5(3):1–22.

[Bodenreider2004] Bodenreider, Olivier. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

[Brown1993] Brown, Lesley. 1993. New shorter oxford english dictionary on historical principles.

[Campbell and Johnson2002] Campbell, David A and Stephen B Johnson. 2002. A transformational-based learner for dependency grammars in discharge summaries. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*, pages 37–44. Association for Computational Linguistics.

[Chapman et al.2001] Chapman, Wendy W, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

[Chapman et al.2007] Chapman, Wendy W, David Chu, and John N Dowling. 2007. Context: An algorithm for identifying contextual features from clinical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 81–88. Association for Computational Linguistics.

[Chapman and Dowling2006] Chapman, Wendy W and John N Dowling. 2006. Inductive creation of an annotation schema for manually indexing clinical conditions from emergency department reports. *Journal of biomedical informatics*, 39(2):196–208.

[Chavan and Shirgave2011] Chavan, Madhuri M and SK Shirgave. 2011. A methodology for extracting head contents from meaningful tables in web pages. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 272–277. IEEE.

[Chen et al.2000] Chen, Hsin-Hsi, Shih-Chung Tsai, and Jin-He Tsai. 2000. Mining tables from large scale html texts. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 166–172. Association for Computational Linguistics.

[Christensen et al.2002] Christensen, Lee M, Peter J Haug, and Marcelo Fiszman. 2002. Mplus: a probabilistic medical language understanding system. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*, pages 29–36. Association for Computational Linguistics.

[Cimino et al.2007] Cimino, James J, Tiffani J Bright, and Jianhau Li. 2007. Medication reconciliation using natural language processing and controlled terminologies. In *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, page 679. IOS Press.

[Cimino et al.1994] Cimino, James J, Paul D Clayton, George Hripcsak, and Stephen B

Johnson. 1994. Knowledge-based approaches to the maintenance of a large controlled medical terminology. *Journal of the American Medical Informatics Association*, 1(1):35–50.

[Coden et al.2005] Coden, Anni R, Serguei V Pakhomov, Rie K Ando, Patrick H Duffy, and Christopher G Chute. 2005. Domain-specific language models and lexicons for tagging. *Journal of biomedical informatics*, 38(6):422–430.

[Cohen and Hersh2005] Cohen, Aaron M and William R Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71.

[Cohen and Hunter2008] Cohen, K Bretonnel and Lawrence Hunter. 2008. Getting started in text mining. *PLoS computational biology*, 4(1):e20.

[Constantin2014] Constantin, Alexandru. 2014. *Automatic Structure and Keyphrase Analysis of Scientific Publications*. Ph.D. thesis.

[Constantin et al.2013] Constantin, Alexandru, Steve Pettifer, and Andrei Voronkov. 2013. Pdfx: fully-automated pdf-to-xml conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering*, pages 177–180. ACM.

[Crestan and Pantel2010] Crestan, Eric and Patrick Pantel. 2010. Web-scale knowledge extraction from semi-structured tables. In *Proceedings of the 19th international conference on World wide web*, pages 1081–1082. ACM.

[Cunningham et al.2002] Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. A framework and graphical development environment for robust nlp tools and applications. In *ACL*, pages 168–175.

[Dalvi et al.2012] Dalvi, Bhavana Bharat, William W Cohen, and Jamie Callan. 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*, pages 243–252. ACM.

[de Bruijn et al.2011] de Bruijn, Berry, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. 2011. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association*, 18(5):557–562.

[De Marneffe and Manning2008] De Marneffe, Marie-Catherine and Christopher D Manning. 2008. Stanford typed dependencies manual. *URL http://nlp. stanford. edu/software/dependencies manual. pdf.*

[Demner-Fushman and Lin2007] Demner-Fushman, Dina and Jimmy Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103.

[Divoli et al.2010] Divoli, Anna, Michael A Wooldridge, and Marti A Hearst. 2010. Full text and figure display improves bioscience literature search. *PloS one*, 5(4):e9619.

[Donaldson et al.2003] Donaldson, Ian, Joel Martin, Berry De Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D Bader, Katerina Michalickova, et al. 2003. Prebind and textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC bioinformatics*, 4(1):11.

[e Silva et al.2003] e Silva, Ana Costa, Alípio Jorge, and Luís Torgo. 2003. Automatic selection of table areas in documents for information extraction. In *Progress in Artificial Intelligence*. Springer, pages 460–465.

[Elmeleegy et al.2014] Elmeleegy, Hazem, Jayant Madhavan, and Alon Halevy. 2014. Harvesting relational tables from lists on the web, May 20. US Patent 8,732,116.

[Embley et al.2006] Embley, David W, Matthew Hurst, Daniel Lopresti, and George Nagy. 2006. Table-processing paradigms: a research survey. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(2-3):66–86.

[Embley et al.2005] Embley, David W, Cui Tao, and Stephen W Liddle. 2005. Automating the extraction of data from html tables with unknown structure. *Data & Knowledge Engineering*, 54(1):3–28.

[Feldman and Sanger2007] Feldman, Ronen and James Sanger. 2007. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press.

[Friedman et al.1994] Friedman, Carol, Philip O Alderson, John HM Austin, James J Cimino, and Stephen B Johnson. 1994. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174.

[Friedman et al.1995] Friedman, Carol, Stephen B Johnson, Bruce Forman, and Justin Starren. 1995. Architectural requirements for a multipurpose natural language processor in the clinical environment. In *ANNUAL SYMPOSIUM ON COMPUTER APPLICATIONS IN MEDICAL CARE*, volume 19, pages 347–351. IEEE COMPUTER SOCIETY PRESS.

[Gatterbauer et al.2007] Gatterbauer, Wolfgang, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl, and Bernhard Pollak. 2007. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, pages 71–80. ACM.

[Gene Ontology Consortium2014a] Gene Ontology Consortium. 2014a. Gene ontology documentation @ONLINE available at http://geneontology.org/page/documentation.

[Gerner et al.2010] Gerner, Martin, Goran Nenadic, and Casey M Bergman. 2010. Linnaeus: a species name identification system for biomedical literature. *BMC bioinformatics*, 11(1):85.

[Hahn and Wermter2004] Hahn, Udo and Joachim Wermter. 2004. High-performance tagging on medical texts. In *Proceedings of the 20th international conference on Computational Linguistics*, page 973. Association for Computational Linguistics.

[Halgrim et al.2011] Halgrim, Scott, Fei Xia, Imre Solti, Eithon Cadag, and Özlem Uzuner. 2011. A cascade of classifiers for extracting medication information from discharge summaries. *J. Biomedical Semantics*, 2(S-3):S2.

[Hansen et al.2008] Hansen, Marie J, NO Rasmussen, and Grace Chung. 2008. Extracting number of trial participants from abstracts of randomized controlled trials. In *Proceedings of Tromsø Telemedicine and eHealth Conference: 9-11 June 2008; Tromsø, Norway.*

[Haug et al.1995] Haug, Peter J, Spence Koehler, Lee Min Lau, Ping Wang, Roberto Rocha, and Stanley M Huff. 1995. Experience with a mixed semantic/syntactic parser. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 284. American Medical Informatics Association.

[Hearst1999] Hearst, Marti A. 1999. Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10. Association for Computational Linguistics.

[Hearst et al.2007] Hearst, Marti A, Anna Divoli, Harendra Guturu, Alex Ksikes, Preslav Nakov, Michael A Wooldridge, and Jerry Ye. 2007. Biotext search engine: beyond abstract search. *Bioinformatics*, 23(16):2196–2197.

[Heath and Bizer2011] Heath, Tom and Christian Bizer. 2011. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.

[Hersh2005] Hersh, William. 2005. Evaluation of biomedical text-mining systems: lessons learned from information retrieval. *Briefings in bioinformatics*, 6(4):344–356.

[Hobbs1993] Hobbs, Jerry R. 1993. The generic information extraction system. In *MUC*, pages 87–91.

[Hu et al.2000a] Hu, Jianying, Ramanujan Kashi, Daniel Lopresti, and Gordon Wilfong. 2000a. A system for understanding and reformulating tables. In *Proceedings of the Fourth IAPR International Workshop on Document Analysis Systems*, pages 361–372.

[Hu et al.2000b] Hu, Jianying, Ramanujan S Kashi, Daniel P Lopresti, and Gordon Wilfong. 2000b. Table structure recognition and its evaluation. In *Photonics West 2001-Electronic Imaging*, pages 44–55. International Society for Optics and Photonics.

[Hurst2000] Hurst, Matthew Francis. 2000. *The interpretation of tables in texts*. Ph.D. thesis.

[IHTSDO 2014] IHTSDO . 2014. Snomed ct @ONLINE available at http://www.ihtsdo.org/snomed-ct/.

[Jonnalagadda et al.2012] Jonnalagadda, Siddhartha, Trevor Cohen, Stephen Wu, and Graciela Gonzalez. 2012. Enhancing clinical concept extraction with distributional semantics. *Journal of biomedical informatics*, 45(1):129–140.

[Jung and Kwon2006] Jung, Sung-Won and Hyuk-Chul Kwon. 2006. A scalable hybrid approach for extracting head components from web tables. *Knowledge and Data Engineering, IEEE Transactions on*, 18(2):174–187.

[Katrenko and Adriaans2007] Katrenko, Sophia and Pieter Adriaans. 2007. Learning relations from biomedical corpora using dependency trees. In *Knowledge Discovery and Emergent Complexity in Bioinformatics*. Springer, pages 61–80.

[Kieninger and Strieder1999] Kieninger, Thomas G and Bernd Strieder. 1999. T-recs table recognition and validation approach. In *AAAI Fall Symposium on Using Layout for the Generation, Understanding and Retrieval of Documents*.

[Kiritchenko et al.2010] Kiritchenko, Svetlana, Berry de Bruijn, Simona Carini, Joel Martin, and Ida Sim. 2010. Exact: automatic extraction of clinical trial characteristics from journal publications. *BMC medical informatics and decision making*, 10(1):56.

[Leaman et al.2008] Leaman, Robert, Graciela Gonzalez, et al. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*, volume 13, pages 652–663.

[Lee et al.2000] Lee, Seok-Lyong, Seok-Ju Chun, Deok-Hwan Kim, Ju-Hong Lee, and Chin-Wan Chung. 2000. Similarity search for multidimensional data sequences. In *Data Engineering, 2000. Proceedings. 16th International Conference on*, pages 599–608. IEEE.

[Li and Roth2001] Li, Xin and Dan Roth. 2001. Exploring evidence for shallow parsing. In *Proceedings of the 2001 workshop on Computational Natural Language Learning-Volume 7*, page 6. Association for Computational Linguistics.

[Liu et al.2001] Liu, Hongfang, Yves A Lussier, and Carol Friedman. 2001. Disambiguating ambiguous biomedical terms in biomedical narrative text: an unsupervised method. *Journal of biomedical informatics*, 34(4):249–261.

[Liu et al.2004] Liu, Hongfang, Virginia Teller, and Carol Friedman. 2004. A multi-aspect comparison study of supervised word sense disambiguation. *Journal of the American Medical Informatics Association*, 11(4):320–331.

[Liu2009] Liu, Ying. 2009. *Tableseer: Automatic Table Extraction, Search, and Understanding*. Ph.D. thesis, The Pennsylvania State University.

[Lovins1968] Lovins, Julie B. 1968. Development of a stemming algorithm.

[Luong et al.2010] Luong, Minh-Thang, Thuy Dung Nguyen, and Min-Yen Kan. 2010. Logical structure recovery in scholarly articles with rich document features. *International Journal of Digital Library Systems (IJDLS)*, 1(4):1–23.

[Meystre et al.2008] Meystre, Stéphane M, Guergana K Savova, Karin C Kipper-Schuler, John F Hurdle, et al. 2008. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform*, 35:128–44.

[Mulwad et al.2010] Mulwad, Varish, Tim Finin, Zareen Syed, and Anupam Joshi. 2010. Using linked data to interpret tables. *COLD*, 665.

[Mutalik et al.2001] Mutalik, Pradeep G, Aniruddha Deshpande, and Prakash M Nadkarni. 2001. Use of general-purpose negation detection to augment concept indexing of medical documents a quantitative study using the umls. *Journal of the American Medical Informatics Association*, 8(6):598–609.

[Nenadić et al.2004] Nenadić, Goran, Sophia Ananiadou, and John McNaught. 2004. Enhancing automatic term recognition through recognition of variation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 604. Association for Computational Linguistics.

[Ng et al.1999] Ng, Hwee Tou, Chung Yong Lim, and Jessica Li Teng Koo. 1999. Learning to recognize tables in free text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 443–450. Association for Computational Linguistics.

[Niu and Hirst2004] Niu, Yun and Graeme Hirst. 2004. Analysis of semantic classes in medical text for question answering. In *Proceedings of the ACL 2004 Workshop on Question Answering in Restricted Domains*, pages 54–61.

[Okazaki and Ananiadou2006] Okazaki, Naoaki and Sophia Ananiadou. 2006. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, 22(24):3089–3095.

[Pakhomov et al.2005] Pakhomov, Serguei, Ted Pedersen, and Christopher G Chute. 2005. Abbreviation and acronym disambiguation in clinical discourse. In *AMIA*

Annual Symposium Proceedings, volume 2005, page 589. American Medical Informatics Association.

[Pande2002] Pande, Ashwini. 2002. Table understanding for information retrieval. Master's thesis, Citeseer.

[Robinson et al.1997] Robinson, David, Erich Schulz, Philip Brown, and Colin Price. 1997. Updating the read codes: user-interactive maintenance of a dynamic clinical vocabulary. *Journal of the American Medical Informatics Association*, 4(6):465–472.

[Sarafraz and Nenadic2010] Sarafraz, Farzaneh and Goran Nenadic. 2010. Using svms with the command relation features to identify negated events in biomedical literature. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 78–85. Association for Computational Linguistics.

[Segura-Bedmar et al.2011] Segura-Bedmar, Isabel, Paloma Martinez, and Cesar de Pablo-Sánchez. 2011. Using a shallow linguistic kernel for drug–drug interaction extraction. *Journal of biomedical informatics*, 44(5):789–804.

[Settles2004] Settles, Burr. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics.

[Silva2010] Silva, AC. 2010. *Parts that add up to a whole: a framework for the analysis of tables*. Ph.D. thesis, University of Edinburgh.

[Son et al.2008] Son, Jeong-Woo, Jae-An Lee, Seong-Bae Park, Hyun-Je Song, Sang-Jo Lee, and Se-Young Park. 2008. Discriminating meaningful web tables from decorative tables using a composite kernel. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, volume 1, pages 368–371. IEEE.

[Spasić et al.2010] Spasić, Irena, Farzaneh Sarafraz, John A Keane, and Goran Nenadić. 2010. Medication information extraction with linguistic pattern matching and semantic rules. *Journal of the American Medical Informatics Association*, 17(5):532–535.

[Suakkaphong et al.2011] Suakkaphong, Nichalin, Zhu Zhang, and Hsinchun Chen. 2011. Disease named entity recognition using semisupervised learning and conditional random fields. *Journal of the American Society for Information Science and Technology*, 62(4):727–737.

[Taira and Soderland1999] Taira, Ricky K and Stephen G Soderland. 1999. A statistical natural language processor for medical reports. In *Proceedings of the AMIA Symposium*, page 970. American Medical Informatics Association.

[Talukdar et al.2006] Talukdar, Partha Pratim, Thorsten Brants, Mark Liberman, and Fernando Pereira. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 141–148. Association for Computational Linguistics.

[Tanaka and Ishida2006] Tanaka, Masahiro and Toru Ishida. 2006. Ontology extraction from tables on the web. In *Applications and the Internet, 2006. SAINT 2006. International Symposium on*, pages 7–pp. IEEE.

[Tengli et al.2004] Tengli, Ashwin, Yiming Yang, and Nian Li Ma. 2004. Learning table extraction from examples. In *Proceedings of the 20th international conference on Computational Linguistics*, page 987. Association for Computational Linguistics.

[The National Center for Biotechnology Information2014] The National Center for Biotechnology Information. 2014. Mesh @ONLINE, available at http://www.ncbi.nlm.nih.gov/mesh.

[Thomachot et al.2001] Thomachot, Laurent, Xavier Viviand, Pierre Lagier, Jean Marc Dejode, Jacques Albanèse, and Claude Martin. 2001. Measurement of tracheal temperature is not a reliable index of total respiratory heat loss in mechanically ventilated patients. *CRITICAL CARE-LONDON-*, 5(1):24–30.

[Toutanova et al.2003] Toutanova, Kristina, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

[Tsuruoka et al.2005] Tsuruoka, Yoshimasa, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Junichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Advances in informatics*. Springer, pages 382–392.

[Tu et al.2009] Tu, Samson W, Simona Carini, Alan Rector, Peter Maccallum, Igor Toujilov, Steve Harris, and Ida Sim. 2009. Ocre: an ontology of clinical research. In *11th International Protege Conference*.

[United States National Library of Medicine 2014c] United States National Library of Medicine . 2014c. Medline @ONLINE available at http://www.nlm.nih.gov/news/medlinedata.html.

[United States National Library of Medicine2014] United States National Library of Medicine. 2014. Medline @ONLINE available at http://www.nlm.nih.gov/bsd/pmresources.html#statistics.

[Venes2013] Venes, Donald. 2013. *Taber's cyclopedic medical dictionary*. FA Davis.

[Wang2013] Wang, Xiao Feng. 2013. Research on information extraction based on web table structure and ontology. *Applied Mechanics and Materials*, 321:2254–2259.

[Wang and Wood1993] Wang, Xinxin and Derick Wood. 1993. Tabular abstraction for tabular editing and formatting. In *Proceedings of 3rd International Conference for Young Computer Scientists*, pages 17–29.

[Wang and Wood1995] Wang, Xinxin and Derick Wood. 1995. *Tabular abstraction, editing, and formatting*. Ph.D. thesis.

[Wang and Hu2002] Wang, Yalin and Jianying Hu. 2002. A machine learning based approach for table detection on the web. In *Proceedings of the 11th international conference on World Wide Web*, pages 242–250. ACM.

[Wei et al.2006] Wei, Xing, Bruce Croft, and Andrew McCallum. 2006. Table extraction for answer retrieval. *Information retrieval*, 9(5):589–611.

[Wong et al.2009] Wong, Wern, David Martinez, and Lawrence Cavedon. 2009. Extraction of named entities from tables in gene mutation literature. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 46–54. Association for Computational Linguistics.

[World Health Organization 2014] World Health Organization . 2014. International classification of diseases (icd) @ONLINE available at http://www.who.int/classifications/icd/en/.

[Wright1968] Wright, Patricia. 1968. Using tabulated information. *Ergonomics*, 11(4):331–343.

[Wright1977] Wright, Patricia. 1977. Decision making as a factor in the ease of using numerical tables. *Ergonomics*, 20(1):91–96.

[Wright and Fox1970] Wright, Patricia and Kathryn Fox. 1970. Presenting information in tables. *Applied Ergonomics*, 1(4):234–242.

[Xu et al.2008] Xu, Songhua, James McCusker, and Michael Krauthammer. 2008. Yale image finder (yif): a new search engine for retrieving biomedical images. *Bioinformatics*, 24(17):1968–1970.

[Yang2010] Yang, Hui. 2010. Automatic extraction of medication information from medical discharge summaries. *Journal of the American Medical Informatics Association*, 17(5):545–548.

[Yildiz et al.2005] Yildiz, Burcu, Katharina Kaiser, and Silvia Miksch. 2005. pdf2table: A method to extract table information from pdf files. In *IICAI*, pages 1773–1785.

[Zanibbi et al.2004] Zanibbi, Richard, Dorothea Blostein, and James R Cordy. 2004. A survey of table recognition. *Document Analysis and Recognition*, 7(1):1–16.

Appendices

Appendix A

Example of table

In Figure A.1 is given an example of the table. Below the figure, XML presentation from PMC database of the same table can be seen.

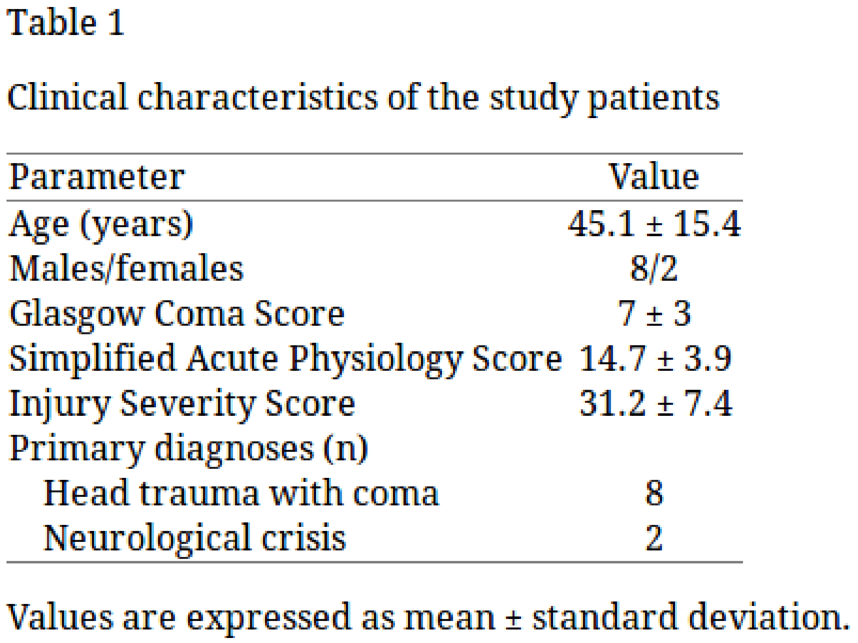


Table 1

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 A.1: Example of the table (PMC29053)

```
<table-wrap id="T1" position="float">
<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>
<tr><td align="left">Primary diagnoses (<italic>n</italic>)
</td><td/>
</tr><tr><td align="left"> Head trauma with coma</td>
<td align="center">8</td></tr>
<tr><td align="left"> Neurological crisis</td>
<td align="center">2</td></tr>
</tbody>
</table>
<table-wrap-foot><p>Values are expressed as mean
± standard deviation.</p></table-wrap-foot>
</table-wrap>
```