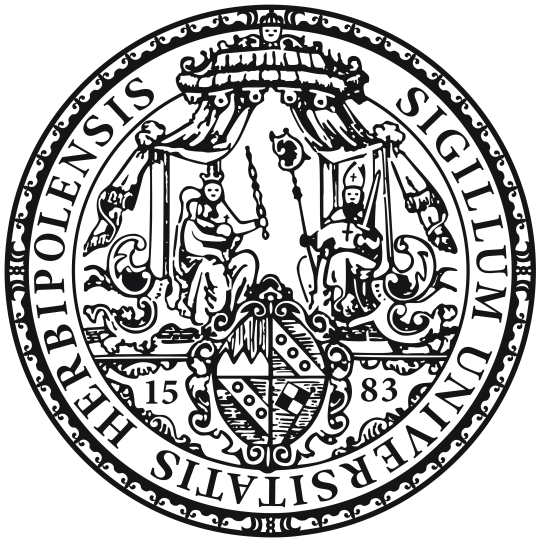
Ad Hoc Information Extraction in a Clinical Data Warehouse with Case Studies for Data Exploration and Consistency Checks

vorgelegt von

Georg Dietrich

Würzburg, 2019



Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades
der Bayerischen Julius-Maximilians-Universität Würzburg

Abstract

The importance of Clinical Data Warehouses (CDW) has increased significantly in recent years as they support or enable many applications such as clinical trials, data mining, and decision making. CDWs integrate Electronic Health Records which still contain a large amount of text data, such as discharge letters or reports on diagnostic findings in addition to structured and coded data like ICD-codes of diagnoses. Existing CDWs hardly support features to gain information covered in texts. Information extraction methods offer a solution for this problem but they have a high and long development effort, which can only be carried out by computer scientists. Moreover, such systems only exist for a few medical domains.

This paper presents a method empowering clinicians to extract information from texts on their own. Medical concepts can be extracted ad hoc from e.g. discharge letters, thus physicians can work promptly and autonomously. The proposed system achieves these improvements by efficient data storage, preprocessing, and with powerful query features. Negations in texts are recognized and automatically excluded, as well as the context of information is determined and undesired facts are filtered, such as historical events or references to other persons (family history). Context-sensitive queries ensure the semantic integrity of the concepts to be extracted. A new feature not available in other CDWs is to query numerical concepts in texts and even filter them (e.g. BMI $>$ 25). The retrieved values can be extracted and exported for further analysis.

This technique is implemented within the efficient architecture of the PaDaWaN CDW and evaluated with comprehensive and complex tests. The results outperform similar approaches reported in the literature. Ad hoc IE determines the results in a few (milli-) seconds and a user friendly GUI enables interactive working, allowing flexible adaptation of the extraction.

In addition, the applicability of this system is demonstrated in three real-world applications at the Würzburg University Hospital (UKW). Several drug trend studies are replicated: Findings of five studies on high blood pressure, atrial fibrillation and chronic renal failure can be partially or completely confirmed in the UKW. Another case study evaluates the prevalence of heart failure in inpatient hospitals using an algorithm that extracts information with ad hoc IE from discharge letters and echocardiogram report (e.g. “LVEF” $<$ 45) and other sources of the hospital information system. This study reveals that the use of ICD codes leads to a significant underestimation (31%) of the true prevalence of heart failure. The third case study evaluates the consistency of diagnoses by comparing structured ICD-10-coded diagnoses with the diagnoses described in the diagnostic section of the discharge letter. These diagnoses are extracted from texts with ad hoc IE, using synonyms generated with a novel method. The developed approach can extract diagnoses from the discharge letter with a high accuracy and furthermore it can prove the degree of consistency between the coded and reported diagnoses.

Zusammenfassung

Die Bedeutung von Clinical Data Warehouses (CDW) hat in den letzten Jahren stark zugenommen, da sie viele Anwendungen wie klinische Studien, Data Mining und Entscheidungsfindung unterstützen oder ermöglichen. CDWs integrieren elektronische Patientenakten, die neben strukturierten und kodierten Daten wie ICD-Codes von Diagnosen immer noch sehr vielen Textdaten enthalten, sowie Arztbriefe oder Befundberichte. Bestehende CDWs unterstützen kaum Funktionen, um die in den Texten enthaltenen Informationen zu nutzen. Informationsextraktionsmethoden bieten zwar eine Lösung für dieses Problem, erfordern aber einen hohen und langen Entwicklungsaufwand, der nur von Informatikern durchgeführt werden kann. Außerdem gibt es solche Systeme nur für wenige medizinische Bereiche.

Diese Arbeit stellt eine Methode vor, die es Ärzten ermöglicht, Informationen aus Texten selbstständig zu extrahieren. Medizinische Konzepte können ad hoc aus Texten (z. B. Arztbriefen) extrahiert werden, so dass Ärzte unverzüglich und autonom arbeiten können. Das vorgestellte System erreicht diese Verbesserungen durch effiziente Datenspeicherung, Vorverarbeitung und leistungsstarke Abfragefunktionen. Negationen in Texten werden erkannt und automatisch ausgeschlossen, ebenso wird der Kontext von Informationen bestimmt und unerwünschte Fakten gefiltert, wie z. B. historische Ereignisse oder ein Bezug zu anderen Personen (Familiengeschichte). Kontextsensitive Abfragen gewährleisten die semantische Integrität der zu extrahierenden Konzepte. Eine neue Funktion, die in anderen CDWs nicht verfügbar ist, ist die Abfrage numerischer Konzepte in Texten und sogar deren Filterung (z. B. BMI > 25). Die abgerufenen Werte können extrahiert und zur weiteren Analyse exportiert werden.

Diese Technik wird innerhalb der effizienten Architektur des PaDaWaN-CDW implementiert und mit umfangreichen und aufwendigen Tests evaluiert. Die Ergebnisse übertreffen ähnliche Ansätze, die in der Literatur beschrieben werden. Ad hoc IE ermittelt die Ergebnisse in wenigen (Milli-)Sekunden und die benutzerfreundliche Oberfläche ermöglicht interaktives Arbeiten und eine flexible Anpassung der Extraktion.

Darüber hinaus wird die Anwendbarkeit dieses Systems in drei realen Anwendungen am Universitätsklinikum Würzburg (UKW) demonstriert: Mehrere Medikationstrendstudien werden repliziert: Die Ergebnisse aus fünf Studien zu Bluthochdruck, Vorhofflimmern und chronischem Nierenversagen können in dem UKW teilweise oder vollständig bestätigt werden. Eine weitere Fallstudie bewertet die Prävalenz von Herzinsuffizienz in stationären Patienten in Krankenhäusern mit einem Algorithmus, der Informationen mit Ad-hoc-IE aus Arztbriefen, Echokardiogrammbericht und aus anderen Quellen des Krankenhausinformationssystems extrahiert (z. B. LVEF < 45). Diese Studie zeigt, dass die Verwendung von ICD-Codes zu einer signifikanten Unterschätzung (31%) der tatsächlichen Prävalenz von Herzinsuffizienz führt. Die dritte Fallstudie bewertet die Konsistenz von Diagnosen, indem sie strukturierte ICD-10-codierte Diagnosen mit den Diagnosen, die im Diagnoseabschnitt des Arztbriefes beschriebenen, vergleicht. Diese Diagnosen werden mit Ad-hoc-IE aus den Texten gewonnen, dabei werden Synonyme verwendet, die mit

einer neuartigen Methode generiert werden. Der verwendete Ansatz kann Diagnosen mit hoher Genauigkeit aus Arztbriefen extrahieren und darüber hinaus den Grad der Übereinstimmung zwischen den kodierten und beschriebenen Diagnosen bestimmen.

Acknowledgements / Danksagung

Viele Personen haben zum Gelingen dieser Arbeit beigetragen: Als erstes möchte ich meinem Doktorvater Frank Puppe für die sehr gute Betreuung, seine stets offene Tür und die angenehme Forschungsumgebung danken. Er nahm sich immer Zeit für spannende Diskussionen und Forschungsfragen. Außerdem danke ich meinem Zweitgutachter Thomas Tolxdorff für seine Arbeit und Unterstützung.

Weiterhin gilt der Dank meinen Kollegen und Freunden der Arbeitsgruppe, die mich über die vergangenen Jahre begleitet und mich bei meinen Projekten und verschiedensten Forschungsthemen unterstützt haben. Es war eine Freude mit euch zu arbeiten: Philip Beck, Martin Becker, Alexander Dallmann, Maximilian Ertl, Björn Eyselein, Friedrich Fell, Georg Fette, Alexander Gehrke, Felix Herrmann, Lena Hettinger, Andreas Hotho, Marianus Ifland, Mathias Kaspar, Peter Klügl, Markus Krug, Jonathan Krebs, Florian Lemmerich, Leon Liman, Thomas Niebler, Christian Reul, David Schmidt, Christoph Wick, Albin Zehe und Daniel Zoller.

Ein ganz herzlicher Dank geht an meine Familie und meine Freundin Cornelia Kolb, die mich auf diesem Weg begleitet, mir den Rücken frei gehalten, mich immer wieder ermutigt und wunderbar unterstützt haben.

Würzburg, März 2019

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The *Information Age* and the *Digital Transformation* affect all industries including medicine and patient heath care. In the last years, more and more patients have been documented with Electronic Health Records (EHR). In the recent past, the secondary use of EHRs also has increased offering great perspectives for many use cases like support of clinical trials, knowledge discovery and decision making. A common way to gain the benefits is an integration of EHRs in a Clinical Data Warehouse (CDW).

CDWs store a wide range of information, such as structured data (e.g. diagnosis codes, laboratory values or drugs) and unstructured data like free-text discharge letters and reports on diagnostic findings. A lot of patient information in the EHR is still stored in free text. Jensen et al. retrieved on average 146 unstructured text documents for each patient from EHR in their hospital for their study [109]. A large amount of medical knowledge which is embedded in these texts, is unstructured, like interpretations, reasons or differential diagnoses, for instance, free-text reports can be required in 59–77% of all cases in order to resolve inclusion criteria for clinical studies [185].

CDWs can deal with structured data very well, but they poorly support features to gain information covered in texts. The query features provided in the user interfaces are very limited and hardly exceed keyword search. It is not possible with these tools to extract and to recover information hidden.

In particular, extracting information from medical texts is a non trivial task as there are some pitfalls: Many phrases are negated in clinical documents [28] and information can be historical, heuristical or they can relate to other persons (family history).

An alternative approach is to perform information extraction within the extract-transform-load (ETL) process by transferring data from the EHR into the CDW, but this has several disadvantages: It has a high and long development effort due to the requirement of expensive resources, such as ontologies and a high volume of handcrafted rules or a large amount of manually labeled training data in the respective language. Other drawbacks are the low promptness and a unavailable adaptability by users [55]. Furthermore, such systems only exist in a few medical domains.

For that reason a solution is worthwhile, which enables physicians to autonomously work and immediately extract the desired information.

1.2 Goal

The main objective of this work is to extract medical concepts from text documents ad hoc. *Ad hoc information extraction* describes the technical concept of extracting the existence of any concept (e.g. chronic kidney disease) or any number (e.g. the LVEF value) from a source in real-time thus allowing the application of the usual query operations (e.g. counting the number of patient cases with LVEF < 45) on the extracted concepts.

To achieve this goal, it is required to develop a pipeline for *ad hoc information extraction* being able to reliably count Boolean and constrained numerical values in clinical textual documents. Because many findings are negated, historical or relate to other persons in clinical texts, this includes three subtasks: (a) Determining the context of information: recognizing and excluding negations, temporal and heuristic context and their scopes as well as information referring to other persons in text documents (b) extracting Boolean (e.g. moderate mitral insufficiency) and (c) numeric values fulfilling constraints (e.g. LVEF < 45) with context sensitive search queries.

In addition to counting the occurrences, extracting and further processing the actual values are desired. In particular, they should be made available for application specific tasks. For example, if a task requires the extraction of drugs, an extension of the framework which extracts the daily dose taken by the patient, should be possible as well.

This ad hoc IE shall not be considered as a replacement for conventional IE, but rather a supplement allowing quick shallow data aggregation to potentially answer any question in the first approximation without the complex pre-defined specifications required for standard IE.

This work aims to develop and integrate the approach into an existing clinical data warehouse. Both evaluations of accuracy and efficiency are part of scientific work. Furthermore, this work seeks to prove the usability and applicability in several real world use cases.

1.3 Contribution

The main contribution of this work is the introduction of *ad hoc Information Extraction*. This novel technique allows the extraction of concepts and their values from texts on the fly, as defined above. This is achieved by creating a system using novel methods, integrating existing tools and significantly rebuilding established approaches or porting them to the German language. The system design was created, developed, evaluated, and is still in productive use. The available text fragmentation methods are adapted and extended for different medical domains. The negation detection is rebuilt from previous approaches reported in the literature and greatly expanded in order to be able identify negated scopes, furthermore, the context algorithm was ported to German. Simultaneously, new query methods have been created that have a large expressiveness.

The architecture of ad hoc IE is not limited to the medical domain. The presented solution can be seen as use case to extract concepts from texts on the fly. Hence, the presented approach is able to fit in many other domains. The general feasibility of this approach is demonstrated by evaluating it in various experiments and case studies.

A further contribution is the implementation of ad hoc IE in the PaDaWaN CDW. This comprises several tasks: A search engine with a full text index was added to the architecture as an additional data store and the request process was consequently changed: The full text index facilitates a vast extension of the text search capabilities [54]. The document-centered data schema enables the storage of all patient data in a single document. As a result, all queries can be sent directly to the query-optimized search engine, which speeds up the query process by a huge factor. This new document-centered storage of patient data in combination with huge capabilities to retrieve information from texts, raises the query process to a new level. PaDaWaN is the first CDW with this kind of architecture using a search engine as a data store that contains all patient data.

Finally, three real-world case-studies are presented that made use of the proposed novel mechanisms. They prove the usability and applicability of ad hoc IE and deliver new significant medical knowledge. (1) Several medical trend studies are replicated using ad hoc IE and the medical findings are matched with the original publications. (2) The prevalence of heart failure in hospital inpatients is assessed via ad hoc IE from discharge letters and an underestimation is ascertained in contrast to the encoded ICD-10 diagnoses. (3) The consistency of diagnoses (ICD-10 encoded and mentioned in discharge letter) are checked with ad hoc IE. A method is developed to extract any diagnosis from discharge letters by generating synonyms.

1.4 Structure of this Work

The remainder of this work is divided in eight chapters:

Chapter 2 presents the state of the art and the background of this thesis. After an overview of the data in clinical information system, the structure of Clinical Data Warehouses is explained in more detail, including benefits, architecture, processes, protocols, query languages and medical ontologies. Afterwards, well-known CDWs will be presented, paying special attention to their text query features. A literature review on information extraction and relevant algorithms in natural language processing complete this chapter.

Chapter 3 introduces the PaDaWaN system: an efficient CDW architecture. The framework structure is outlined including, inter alia, important processes, the data model and the query language. The user interface focusing on the query surface is presented as implementation detail.

Chapter 4 presents the main theoretical contribution of this work: The novel methods for ad hoc information extraction are introduced, starting with essential data structures for processing and storing text documents and followed by various algorithms and functions enabling ad hoc information extraction. This includes, text sectioning and segmentation

procedures, a negation detection to recognize negated scopes in texts, a context detection to filter historical information or non patient-related information and sophisticated, powerful query features to extract information ad hoc.

Chapter 5 describes the implementation of these algorithms in the PaDaWaN CDW.

The accuracy and the efficiency of ad hoc information extraction is evaluated with comprehensive experiments in Chapter 6. The entire system is tested in total, as well as sub modules as negation detection.

The results are discussed and compared to related work in Chapter 7. Ad hoc IE is compared to similar approaches and systems reported in literature and discussed in general. The chapter ends with with a recap of the benefits and the limitations.

Three case studies are conducted at the Würzburg University Hospital in Chapter 8: Medical trend studies are replicated, the prevalence of heart failure in hospital inpatients is assessed and the consistency of diagnoses (ICD-10 encoded and mentioned in discharge letter) is analyzed.

Finally, Chapter 9 summarizes this works and concludes this thesis with an outlook on future work.

2 Background and State of the Art

This chapter provides a general overview on *Clinical Data Warehouses* and techniques for *Information Extraction*. Section 2.1 starts with a characterization of data in a clinical information system. Section 2.2 is dedicated to the topic of clinical data warehouses focusing on basics and concepts, such as use cases, architecture, processes, and clinical ontologies. The most important CDWs are presented in Section 2.3 paying special attention to their text query features. Section 2.4 concludes this chapter with a state of the art review on relevant techniques in process natural language.

2.1 Data in a Clinical Information System

A *hospital information systems* (HIS) covers administrative and medical modules [34]. Administrative systems contain e.g. billing and accounting data. The medical subsystem of a HIS containing all medical patient data is defined as *clinical information system* (CIS) [236]. It only serves patient care comprising inpatients and outpatients [149].

The *electronic health record* (EHR) is defined as "a longitudinal electronic record of patient health information generated by one or more encounters in any care delivery setting. Included in this information are patient demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data, and radiology reports" [86]. A CIS stores EHR data and provides access for physicians via user interfaces and for clinical systems via application programming interface (API).

2.1.1 Data Sources

Potential data sources for a CDW can be found in the HIS and the CIS. Core data (e.g. name, age, sex) of patients is stored in the HIS. Medical data is stored in the CIS, which usually consists of several subsystems that provide various information. This can comprise a system of laboratory values (e.g. Swisslab), a system that stores and manages image data (e.g. PACS), a special system for intensive care unit (e.g. Copra), a system to maintain all medication, a system containing all discharge letters and much more systems that administrate a special kind of data like medical report on diagnostic findings [63].

2.1.2 Structured Data

Structured data is well organized data, formatted in the same way and typically stored in a database. It is often gathered using input forms that are filled in by clinicians (e.g.

patient core data, ICD diagnoses, operation codes) or the data is generated by computers (e.g. laboratory values).

2.1.3 Text Data

A lot of the patient information in the electronic health record (EHR) is still stored in free text. E.g. Jensen et al. (2017) retrieved on average 146 unstructured text documents for each patient from EHR of their hospital for their study [109]. Text documents originated from many different medical departments.

Examination reports or diagnostic findings reports are usually recorded in text, like sonography, medical history examination, physical examination, diagnosis section within the discharge letter, medication, cardiac catheterization, electrocardiogram, echocardiography, radiological examination, etc. [63].

Much documentation is still in free text form and not as structured data for several reasons: Structured data requires a high development effort, a high documentation effort for physicians or medical staff, or insufficient structured background knowledge, like missing ontologies [63].

Text documents can differ in their structure: If they are written by persons or are generated by examination devices using templates and predefined text blocks, the structure can range from natural language and no inner structure to semi-structured text data.

Another characteristic of text is the sentence structure that varies as well, mainly depending on the document type or the medical domain. A narrative style with long and complex sentences is used in sections such as anamnesis (medical history), physical examination and therapy. A rather telegraphic style with short sentences is mainly common for report of findings. Pure enumerations may also occur, e.g. in the discharge letter section *Laboratory*, but often a mixture of enumerations, key words and complete sentences is used [55].

2.1.3.1 Discharge Letter

The most important text and medical relevant document in the CIS is the discharge letter. Other names are doctor's letter, physician's letter or discharge summary. In the hospital, the letter is usually finalized at the discharge of a patient.

The discharge letter is an important communication document between physicians and summarizes important information. It should include these aspects [234]:

1. Addressees
2. Patient data including residence time
3. Diagnoses, procedures, operations
4. Epicrisis
5. Therapy recommendation
6. Findings in the appendix

Exemplary discharge letter. An example for a discharge letter in German is given below. Discharge letter is taken from Wikipedia.[1]

Männlicher Patient geb 1945, stationär 25.10.2011 – 07.11.2011

Diagnosen

- Akuter Hinterwandmyokardinfarkt mit ST-Hebung (sogenannter "STEMI")
- Koronare 2-Gefäßerkrankung
  - RCX: Langstreckiger Verschluß (alt) des RMS mit relativ guter Kollateralenbildung vom apikalen RIVA her
  - ACD = RCA = Rechte Kranzarterie (dominantes Gefäß): ostiumnaher Verschluß
    - Thrombusabsaugung und Implantation eines medikamentenfreisetzenden Stents (DES)
- Mitralklappeninsuffizienz Grad 1
- Kardiovaskuläre Risikofaktoren:
  - Benigne essentielle Hypertonie: Ohne Angabe einer hypertensiven Krise [I10.00]
  - Gestörte Glucosetolereanz,
  - Hyperlipidämie

Anamnese

Am Aufnahmetag bekam Herr Sowieso plötzlich starke Luftnot und Thoraxschmerzen bei leichter Belastung. Bereits seit ca. 6 Monaten habe er immer wieder leichte thorakale Schmerzen, die vom Hausarzt als vertebragen klassifiziert worden seien. Vom Notarzt hatte er Fentanyl und Nitro Spray bekommen. In der Familie keine Herzerkrankungen, kein Schlaganfall; fraglich Diabetes.

Labor

Natrium: 144 [135 - 145] mmol/l, Kalium: 4.7 [3.5 - 5] mmol/l, Creatinin: 1.1 [0.5 - 0.9] mg/dl, GOT (ASAT): 7.5 [0 - 15] U/L, GPT (ALAT): 15.0 [0 - 17] U/L, GGT: 20.0 [0 - 18] U/L, Thromboplastinzeit n. Quick: 112 [70 - 130] %, Leukozyten: 6.9 [5 - 10] n*1000/μl, Erythrozyten: 4.87 [4 - 5] n*10E6/μl, Hämoglobin: 16.2 [12 - 16] g/dl, Thrombozyten: 183 [150 - 450] n*1000/μl, C-reaktives Protein: 0.12 [0 - 0.5] mg/dl, TSH: 0.56 [0,3 - 4] mIU/L, T3 gesamt: 1.25 [1,1 - 2,6] nmol/L, freies T4:

[1] `https://de.wikibooks.org/wiki/Innere_Medizin_kk:_STEMI#Fall_1_Standardfall_Verschlu.C3.9F_rechte_Kranzarterie`, accessed: January 2019

23.1 [11 - 23] pmol/L

Transthorakales Echocardiogramm:

Linker Ventrikel normal weit und normal kontraktil, keine regionalen Wandbewegungsstörungen, auch keine erkennbare HW-Narbe, Auswurffraktion normal, EF planimetrisch 64 %. grenzwertige LV-Hypertrophie, Septum edd 10 mm, diastolische Relaxationsstörung Grad II (DT 284 ms, IVRT 57 ms, LA-Vol.-Index 35,1 ml/m^2). Vorhöfe leicht dilatiert (LA 23 cm^2, RA 17 cm^2) und rechter Ventrikel normal weit, gute RV-Funktion, TAPSE 24 mm. Aortenklappe leicht sklerosiert, keine Stenose, leichtgradige Insuffizienz (PHT nicht messbar) Leichtgradige Mitralinsuffizienz, V. contracta 4 mm. Trikuspidalis unauffällig. Pulmonalis mit leichtgradiger Insuffizienz. Normfrequenter Sinusrhythmus, kein Perikarderguss. Normal großer LV mit guter systolischer Pumpfunktion, grenzwertige LV-Hypertrophie mit diastolischer Relaxationsstörung, leichtgradige AI, MI und PI, gute RV-Funktion.

Herzkatheteruntersuchung

Beurteilung: Es zeigt sich als Ursache des akuten Hinterwandinfarktes ein proximaler Verschluss der dominanten RCA. In gleicher Sitzung erfolgt eine PCI. Nach Vorführen des Führungsdrahtes Thrombussaugextraktion, dann Gabe von insgesamt 8ml Aggrastat intracoronar, direkte Stentimplantation. Abschließend gutes Ergebnis, guter Fluss im Gefäß, Patient sofort beschwerdefrei, ST-Hebungen im EKG bilden sich vollständig zurück.

Epikrise:

Der Patient wurde wegen instabiler Angina pectoris eingewiesen. Im EKG zeigten sich diskrete Hebungen in der Ableitung II und III und ST-Senkungen V1 bis V3. In der Herzkatheteruntersuchung zeigte sich ein Verschluss des RCA, so dass in gleicher Sitzung eine Thrombusabsaugung und Implantation eines medikamentenfreisetzenden Stent (DES) erfolgte. Nach der Herzkatheteruntersuchung ist Hr. Sowieso beschwerdefrei. Im weiteren Verlauf war der Patient stabil, bei Krankengymnastik kooperativ.

Therapieempfehlung / Medikation:

Simva (Simvastatin) 20 0-0-1
Plavix (Clopidogrel) 75 1-0-0 für weitere 12 Monate
ASS (Acetylsalicylsäure) 100 0-1-0 dauerhaft
Beloc zok mite (Metoprolol) 1-0-0
Delix (Ramipril) 5 1-0-0

Beside the previously mentioned obligatory sections, a discharge letter can contain more sections. Table 2.1 lists all currently automatically identified sections and their occurrence

per discharge letter in the University Hospital Hospital of Würzburg in the PaDaWaN CDW.

In the UKW, discharge letter are created and edited with Microsoft Word and stored in the .doc or .docx format.

2.1.3.2 Medical Report on Diagnostic Findings

Other text documents that contain important information are medical reports on diagnostic findings. The result of examinations are reported, summarized and assessed by experts. These reports are sometimes partly embedded as sections in the doctor's letter.

Common requested reports are physical examination, chest radiography, computed tomography (CT), echocardiography, electrocardiography, magnetic resonance imaging (MRI) or pulmonary function.

The language of such reports is characterized by a rather telegraphic style with many short noun phrases and numerous negations that exclude symptoms and findings.

Two examples of chest radiography reports in German are listed below: The texts were created artificially by merging sentences of various reports of the UKW.

> Röntgen-Thorax: Degenerative Wirbelsäulenveränderungen. Kranialraffung des rechten Hilus bei bullösen Lungenumbauten rechts apikal. Linke Lunge unauffällig. Kein Anhalt auf intrapulmonale Rundherde. Aortensklerose. Ausgeprägte Verschwielung rechts pleural. Links pleural Kuppenschwiele.

> Röntgen-Thorax vom : Kein Hinweis auf cardio-pulmonalen Stau, Erguss oder frisches entzündliches Infiltrat, gesondert kein Hinweis auf metastasenverdächtige Lungenrundherde. Degenerative BWS-Veränderungen mit Spondylosis deformans und Zeichen der Osteoporose. Herzschrittmachergerät in situ.

In the UKW, findings reports are created, edited and stored in the CIS as plain text.

2.1.4 Other Data

In addition to structured data and text data, other data such as genomics or images exist. An important part of current diagnostics are imaging techniques. Applications, like digital X-ray or MRI, produce high-resolution images with a massive amount of data, which can be stored in a *picture archiving and communication system* (PACS). Processing medical images is a very challenging task with many fields. A good overview of algorithms, systems and applications is given by Tolxdorff et al. [229, 230]

Table 2.1: **List of section of a discharge letter and their occurrence per letter.** *UKW:* Würzburg University Hospital, *Med1:* Department of internal medicine I focusing on diseases of circulatory system.

Name	**UKW**	**Med1**
Allergies	0%	1%
Anamnesis (medical history)	50%	64%
Bronchology	0%	1%
Cardiac catheterization	2%	10%
Chest radiography	9%	19%
Computed Tomography (CT)	8%	8%
Cytology	9%	2%
Diagnoses	87%	95%
Discharge	2%	1%
Echocardiography	8%	20%
Electrocardiography	15%	44%
Electroencephalography	2%	0%
Electrophysiology	2%	1%
Endocrinology	1%	6%
Ergometrie	1%	3%
Garbdoppler	1%	3%
Gastro	1%	2%
Introduction	1%	1%
Laboratory findings	46%	70%
Long-term blood pressure	0%	1%
Magnetic resonance imaging (MRI)	5%	3%
Medication	39%	65%
Microbiology	2%	2%
Nephrology	0%	1%
Neurology	9%	3%
Pathology	2%	2%
PET	1%	0%
Physical examination	27%	53%
Procederes	88%	86%
Pulmonary function	2%	6%
Radiography	1%	1%
Radiology	1%	2%
Scintigraphy	2%	2%
Sonography	15%	10%
Surgery	1%	1%
Transplantation	1%	1%
Virology	2%	2%

2.2 Clinical Data Warehouses: Basics and Concepts

There exist a variety of definitions for a *Data Warehouse* (DW). Zeh described it as a physical database that provides an integrated view of the underlying data sources [251]. Kimball further specifies that data is "specifically structured for querying and reporting" [117]. A commonly used and much cited definition is given by Inmon in 1996 [107]: "A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process." The DWH must concern a specific domain (subject-oriented), the data must be harmonized (integrated), e.g. same units, updates must not change existing data (time-variant) and data must be stored permanently (nonvolatile).

A *Clinical Data Warehouses* (CDW) extends that definition adds that the imported data must be reorganized to support retrospective analysis [133]. Pedersen & Jensen compare a CDW with a conventional DW, they state that a CDW need to have a more advanced temporal support and a more complex data model than conventional DWs to handle very complex data [171]. *Data marts* are copies of smaller parts of a CDW: They are subject-oriented parts of a CDW in order to support user populations [133].

2.2.1 Use Cases, Goals, Benefits

An important purpose of CDWs is the support of clinical trials. However, CDWs can be used for many other use cases and offer manifold benefits:

Patient recruitment for clinical trials. The inclusion criteria and exclusion criteria of clinical trials can be mapped in a CDW query. The data warehouse serves as data pool and contains all integrated patients of a department or an entire hospital. The system retrieves potential candidates by executing the query [231].

Cohort analysis. Patients groups, such as participants of clinical trials, can be divided into cohorts. The characteristics of these cohorts can be analyzed, furthermore hypotheses can be tested [26].

Explorative data analysis. The process for knowledge discovery in databases starts with the *business understanding*, followed by the *data understanding* according to the ***C**ross **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining*-Modell (CRISP-DM) [27]. Fayyad also puts these steps at the beginning of his model [66]. A CDW can be used for explorative data analysis to get a feeling for the data.

Knowledge discovery. A main reason for building a CDW certainly is the discovery of new knowledge, including the relationship between medical concepts (disease, diagnoses, treatments, procedures, drugs, genomics data, demographic data) and temporal trend analysis [189].

Enhance data quality in the CIS. In a CDW, data from different sources is integrated and new data is derived, extracted or aggregated [195]. Discrepancies can be revealed by comparing redundant data of different sources or with constraint driven methods [53]. Such methods can be used to measure data quality and can help to improve it [231].

Enhancement patient care quality. A more common purpose of a CDW is the enhancement of patient care [99]. On the one hand, this can be achieved by clinical research (using a CDW). On the other hand, the patient care in hospital can be improved, by e.g. a visualization and an optimization of patient trails in a hospital.

Decision making. A CDW can support decision making by providing an empirical data basis for various issues [136, 180, 207, 250]. The problem setting and the action alternatives can be modeled and queried in the CDW and success probabilities can be evaluated retrospectively.

Data source for further analysis. Many of the above mentioned analyzes can be performed with built-in functions of a CDW, for further and deeper analysis, the CDW can also be used as a data source. For example, a medical study can pull additional data from a CDW, which greatly reduces the time it takes to collect the necessary data [189].

2.2.2 Architecture

The architecture of CDWs depend on their purpose. The architecture is mainly influenced by the data model and how data is linked, stored and queried. The structure of most systems can be generalized to a common architecture (see Figure 2.1) [115]: Structured and unstructured data in the hospital information system is extracted from various sources. This data is put on a *staging area* (also called *data lake*) and stored temporarily. After some transformation, it is imported into a permanent data storage: A warehouse stores all data, smaller *data marts* store subsets for specific use cases [115]. The warehouse can be requested or queried by applications, such as graphical user interfaces, mobile applications, reporting tools and export functions.

2.2.2.1 Data Models

The data models of most CDWs are based on an *entity–attribute–value* (EAV) model. Two familiar extensions are commonly used [115]: *star schema* and *snowflake schema*.

Entity–Attribute–Value Model: The EAV model is a basic way to represent and store data. Information is represented as triples: An entity has a value for an attribute. For example: “Patient X” (entity) has a “hemoglobin” (attribute) observation with “15 g/dl” (value).

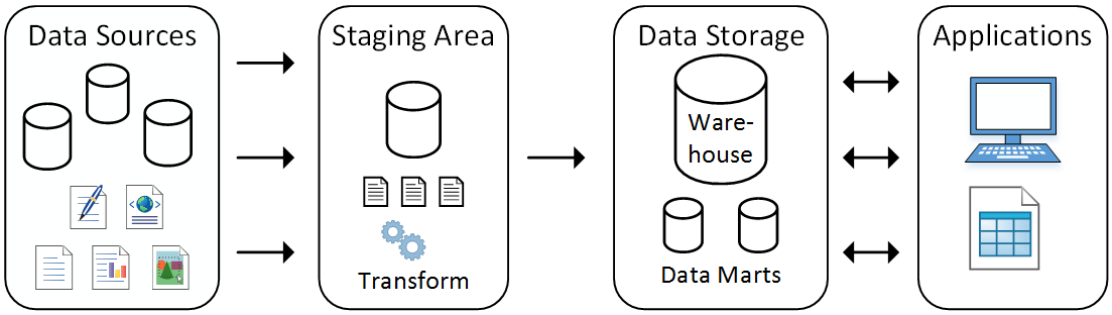


Figure 2.1: **General architecture of a CDW.** Data is extracted of various data sources, transformed in a staging area and imported into a warehouse or smaller data marts that store the data persistently. Applications like user interfaces request the warehouse and display and export data.

Star Schema: A star schema can represent multidimensional data. The core of this schema is a fact table with an EAV model. Facts can be attributed with dimensions. Each dimension (e.g. location, time) is represented as a discrete table and is linked with the fact table. Each attribute of a dimension is modeled of as a column of the dimension table. A value of a dimension is stored as row in that table and can be referenced by the fact table [31].

Snowflake Schema: The snowflake schema refines the star schema by enabling hierarchical dimension tables. A dimension table can be split up in sub dimension tables [31]. For example, the dimension "time" is divided in the sub dimension "year", "month" and "date". In other words: The star schema is not only applied to the fact table, but also to the dimension tables.

2.2.2.2 Data Storage

Most CDWs use relational databases as persistent data storage. The most common DB systems are Microsoft SQL[2], Oracle DB[3], PostgreSQL[4] and MySQL[5].

Some systems use NoSQL (not only SQL) systems like MongoDB[6] as storage engine. MongoDB is a document database that stores JSON-like documents with any structure. Thus, the system is much more flexible than a relational DB with fixed schemas.

Further data storage systems are index libraries like Apache Lucence[7], commonly used for full-text search. Lucene is similar to NoSQL DB and document stores, because it also

[2] `https://www.microsoft.com/de-de/sql-server/`, accessed: January 2019
[3] `https://www.oracle.com/de/database/`, accessed: January 2019
[4] `https://www.postgresql.org/`, accessed: January 2019
[5] `https://www.mysql.com/`, accessed: January 2019
[6] `https://www.mongodb.com/`, accessed: January 2019
[7] `http://lucene.apache.org/`, accessed: January 2019

represents its information in a document centered manner. Lucene is extended by the search engine servers Apache Solr[8] and Elastic[9], which are used by some CDW to index their text data.

2.2.2.3 User Interfaces

The data and information contained in a CDW are accessible via graphical user interfaces (GUI). GUIs are most commonly used for (1) the identification of patient groups, (2) the review and analysis of individual patient data and (3) statistical evaluations and data mining.

GUIs widely vary in their appearance and their functions. In some systems, only the size of a patient group is available; in other systems, any data of a patient or a patient group can be displayed in tabular form. Depending on the orientation of the system, the GUI of the CDW may also provide functions for static analysis

The selection of a patient group works similarly in all systems. The selection criteria are defined as filters constraining the patient set in the CDW. The number of criteria that can be modeled as a filter and the expressiveness of the filters depend on the query language, data model and data storage.

Users have different skills and backgrounds ranging from developers and experts to clinicians with no computer science background. There are easy to use interfaces with drag and drop support and expert interfaces with text input areas where queries have to be manually written in SQL syntax style.

2.2.3 Processes

Based on the purpose of a CDW, various processes are required. However, commonly used processes handle the data anonymization and the data integration.

2.2.3.1 Extract, Transform, Load

The *Extract, Transform, Load Process* (ETL) integrates the data from various sources into the Data Warehouse. Commonly used ETL tools are SQL Server Integration Services (SSIS)[10], Oracle Data Integrator (ODI)[11], i2b2, Talend[12], Kettle[13], Java based applications [115].

[8] `http://lucene.apache.org/solr/`, accessed: January 2019
[9] `https://www.elastic.co/`, accessed: January 2019
[10] `https://docs.microsoft.com/de-de/sql/integration-services/sql-server-integration-services`, accessed: January 2019
[11] `https://docs.oracle.com/cd/E14571_01/integrate.1111/e12643/intro.htm`, accessed: January 2019
[12] `https://de.talend.com/products/data-integration/`, accessed: January 2019
[13] `https://www.openhub.net/p/kettle`, accessed: January 2019

Extraction Process. The extraction process exports data of heterogeneous data sources with distinct characteristics that must be treated in different ways and with different tools [159].

Transformation Process. The extracted data is transformed to a common data model by applying a set of rules and functions [8]. This takes place in a so called *staging area*. The data manipulation methods comprises data aggregation, derivation of new values, standardization and harmonization of data [159]. Data cleaning and data validation is also part of this stage. Errors, inconsistencies and "dirty data" are detected and removed, enforcing a high data quality. Finally, the data is transformed into the target data schema, e.g. a star or an EAV schema.

Loading Process. The resulting data in the staging area is imported in the target system, such as a relational database with EAV, star or snowflake schema. Load tools or import functions of the target data store are used for this task.

2.2.3.2 De-Identification

De-identification removes private health information (PHI) from medical texts, such as discharge letter or report of findings. State of the art are pattern-matching approaches, including lexical matching, regular expressions and simple heuristics performing context checks [163]. A second approach are non-dictionary-based de-identification methods, like rule based or machine learning systems [156, 235].

2.2.4 Medical Query Languages

Users create queries in interfaces of CDWs in order to get answers for their questions. Queries are modeled in a textual representation by a system in respect to a query language, before they are passed to the query engine that executes queries and returns the results. This guarantees the independence between the user interface and the query engine requesting a data store. Furthermore queries can be stored, loaded and exchanged. The syntax of the medical query languages defines the structure of the document. The operators defined in the language specify the possible query operations. Medical query languages are often closely linked to a data model. The following section gives an overview of the most important and most popular ones.

Arden Syntax. The Arden Syntax was first published in 1990 and is used as markup language to represent medical knowledge [102]. The current version (2.10)[14] is maintained by the Health Level 7 (HL7) organization, whose standardized query language is the

[14] `http://www.hl7.org/implement/standards/product_brief.cfm?product_id=372`, accessed: January 2019

Guideline Expression Language Object-Oriented (GELLO) [196]. Hence, Arden Syntax is not a conventional query language, its rather a clinical and scientific knowledge representation in an executable format for clinical decision support systems [196]. The language contains rules with conditions and actions and is compiled by an Arden Syntax compiler and can be executed with the Arden Syntaxrule engine. Arden syntax can be used to supervise medical values (e.g. laboratory values) and trigger alert messages [196].

Archetype Query Language. Archetype Query Language (AQL)[15] is the query language of the openEHR project. AQL is a declarative query language and is similar to the SQL syntax and contains `SELECT`, `FROM`, `WHERE` and `ORDER BY` clauses as well [193]. It references the openEHR *archetypes*, which are clinical content specifications for e.g. blood pressure or smoker habits [135]. Archetypes can be nested, enabling the definition of strong hierarchical data structures. Archetypes can be addressed in AQL and variables contained in these archetypes can be filtered (e.g. systolic blood pressure > 120) [146, 193].

Guideline Expression Language Object-Oriented. The Guideline Expression Language Object-Oriented (GELLO)[16] is an object-oriented query language that was initially published in 2002 [50] and adopted by of the Health Level 7 organization in 2005 [214]. It aims to be usable to support clinical decision applications. The class-based, object-oriented approach seeks to increase the flexibility and extensibility [213, 214].

i2b2 Query Language. The i2b2 (Informatics for Integrating Biology & the Bedside) query language is defined in XML representation and strongly designed for the i2b2 data model, consisting of an extended EAV table [57]. Hence, it does not support complex hierarchical queries like AQL. However, several sub-queries concerning different concepts can be logically combined and temporal relations can be modeled as well [141].

Structured Query Language. An easy way to represent a user query in a textual way is the Structured Query Language (SQL) for relational databases proposed in 1970 [33]. SQL cannot be considered as a medical query language, especially since it is not independent of the data storage: It addresses a concrete database schema and does not show a generalizing character. It does not represent a meta-level, that facilitates the exchange of queries to other systems. Besides that, queries can be modeled in SQL using a complex logical structure and expressive content operators.

Other query languages. The National Quality Forum (NQF)[17] developed a Quality Data Model (QDM) and the Quality Data Language (QDL) to model and query medical

[15] `https://specifications.openehr.org/releases/QUERY/latest/AQL.html`, accessed: January 2019
[16] `http://www.openclinical.org/gmm_gello.html`, accessed: January 2019
[17] `https://www.qualityforum.org/home.aspx`, accessed: January 2019

data. Relationships between patients and medical concepts can be modeled in a standardized form in order to facilitate quantity performance measurements.[18] Other medical query languages are ASBRU [205], PROforma [225], Chronus II [168] and EliXR [244].

2.2.5 Clinical Ontologies

Standard terms, codes, terminologies and ontologies for clinical concepts promote structure and interoperability [242]. A (core) ontology is a tuple consisting of a set of concept and a partial order called concept hierarchy or taxonomy [101]. This section presents the most popular ones.

2.2.5.1 International Classification of Diseases

The *International Classification of Diseases 10* (ICD-10) is the tenth and current revision of the very important and well accepted classification system of medical diagnoses, published by the World Health Organization (WHO) [20, 170]. Its full name is *International Statistical Classification of Diseases and Related Health Problems*. It is publicly accessible and can be used via browser.[19]

In Germany, the physicians and other facilities involved in patient health care are obliged to encode diagnoses in accordance with ICD-10-GM (German Modification), according to § 295 1st paragraph 2end sentence of the fifth Social Code (germ. *§ 295 Absatz 1 Satz 2 des fünften Sozialgesetzbuchs [Abrechnung ärztlicher Leistungen]*).[20]

It is organized in 24 chapters (see Table 2.2) with 261 disease groups (e.g. "I20-I25 Ischaemic heart diseases") containing more than 2,000 disease categories (e.g "I21 Acute myocardial infarction") and consisting more than 14,000 diseases classes/subcategories (e.g. " I21.1 Acute transmural myocardial infarction of inferior wall").[21] Figure 2.2 shows exemplary the structure of ICD-10.

2.2.5.2 SNOMED CT

Systematized Nomenclature of Medicine, Clinical Terms (SNOMED CT) is considered as the most comprehensive clinical health care terminology [16, 58, 95], including clinical findings, symptoms, diagnoses, procedures, body structures, organisms and other etiologies, substances, pharmaceuticals, devices and specimens [108]. SNOP, the successor of SNOMED, was started in 1965, the first version of SNOMED was released in 1974, several versions of SNOMED were published over years and merged to SNOMED CT in 2002 [38, 41, 136, 219]. A literature review showed that the use of SNOMED CT was

[18] `https://ecqi.healthit.gov/qdm-quality-data-model`, accessed: January 2019
[19] `https://www.who.int/classifications/icd/icdonlineversions/en/`, accessed: February 2019
[20] `https://www.sozialgesetzbuch-sgb.de/sgbv/295.html`, accessed: February 2019
[21] `https://www.who.int/classifications/help/icdfaq/en/`, accessed: February 2019

Table 2.2: **Chapters of ICD-10.** The International Statistical Classification of Diseases and Related Health Problems 10th Revision is organized in 22 chapters.

Chapter	Blocks	Title
I	A00–B99	Certain infectious and parasitic diseases
II	C00–D48	Neoplasms
III	D50–D89	Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
IV	E00–E90	Endocrine, nutritional and metabolic diseases
V	F00–F99	Mental and behavioural disorders
VI	G00–G99	Diseases of the nervous system
VII	H00–H59	Diseases of the eye and adnexa
VIII	H60–H95	Diseases of the ear and mastoid process
IX	I00–I99	Diseases of the circulatory system
X	J00–J99	Diseases of the respiratory system
XI	K00–K93	Diseases of the digestive system
XII	L00–L99	Diseases of the skin and subcutaneous tissue
XIII	M00–M99	Diseases of the musculoskeletal system and connective tissue
XIV	N00–N99	Diseases of the genitourinary system
XV	O00–O99	Pregnancy, childbirth and the puerperium
XVI	P00–P96	Certain conditions originating in the perinatal period
XVII	Q00–Q99	Congenital malformations, deformations and chromosomal abnormalities
XVIII	R00–R99	Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX	S00–T98	Injury, poisoning and certain other consequences of external causes
XX	V01–Y98	External causes of morbidity and mortality
XXI	Z00–Z99	Factors influencing health status and contact with health services
XXII	U00–U99	Codes for special purposes

```
IX Diseases of the circulatory system
├── I00-I02 Acute rheumatic fever
├── I05-I09 Chronic rheumatic heart diseases
├── I10-I15 Hypertensive diseases
└── I20-I25 Ischaemic heart diseases
    ├── I20 Angina pectoris
    └── I21 Acute myocardial infarction
        ├── I21.0 Acute transmural myocardial infarction of anterior wall
        ├── I21.1 Acute transmural myocardial infarction of inferior wall
        ├── I21.2 Acute transmural myocardial infarction of other sites
        ├── I21.3 Acute transmural myocardial infarction of unspecified
        │   site
        ├── I21.4 Acute subendocardial myocardial infarction
        └── I21.9 Acute myocardial infarction, unspecified
```

Figure 2.2: **Hierachical structure of ICD-10.** Example of "Acute myocardial infarction"

reported in 488 papers between 2001 and 2012 [134]. SNOMED CT is a multinational terminology and available in English, Spanish, Danish and Swedish, but not in German.[22]

The fundamental structure of SNOMED CT is the logical model consisting of *concepts*, *descriptions* and *relationships*. Concepts represent clinical meanings and are encoded with unique, numeric identifier (e.g. "22298006"). Each concept has a set of textual descriptions in a human readable manner (e.g. "myocardial infarction (disorder)").

A relation type defines a *fully specified name* (FSN), e.g. "myocardial infarction (disorder)", and synonyms, such as "infarction of heard", "cardiac infarction", "heart attack". Furthermore, the descriptions are tagged as *preferred* or *acceptable* for each language model. An association between two concepts is represents with a *relationship*. A relationship type (attribute) specifies the semantic. For example: "diabetes mellitus type 2" (concept), "is a" (attribute), "diabetes mellitus" (concept). The most common form of relationship are *subtypes* defining "is a" relation between one or many concepts. For example: "cellulitis of foot" *is a* "cellulitis" and *is a* "disorder of foot".[23] SNOMED CT contains more than 340,659 (June 2018) concepts.[24]

Concepts are organized in a hierarchy with nine top-level concepts: Clinical finding concepts, procedure concepts, evaluation procedure concepts, specimen concepts, body structure concepts, pharmaceutical/biologic product concepts, situation with explicit context concepts, event concepts, physical object concepts. Each category of concepts is defined with various attribute. For example, clinical findings are described with 16

[22] `https://confluence.ihtsdotools.org/display/TRAN/Translations+Home`, accessed: February 2019
[23] `https://confluence.ihtsdotools.org/display/DOCSTART/5.+SNOMED+CT+Logical+Model`, accessed: February 2019
[24] `https://www.snomed.org/snomed-ct/five-step-briefing`, accessed: January 2019

```
53057004 |hand pain|  :
     363698007 |finding site|  = ( 76505004 |thumb structure|  :
         272741003 |laterality|  =  7771000 |left|  )
```

Listing 1: **Expressions in SNOMED CT.** The cinical meaning "pain in the left thumb" represented with SNOMED CT a expression.

attributes, such as *finding site* (specifications of the body site), *associated morphology* (morphologic changes), *after* (sequence of events happened before), *due to* (clinical finding or a procedure causing the current finding), *severity*, *finding method* (e.g. blood test, X-ray), *finding informer* (e.g. nurse, physician).[25]

Expressions are used to represent clinical meanings in the SNOMED CT syntax using concepts and relations. For example, "pain in the left thumb" can be seen in Listing 1.[26]

2.2.5.3 RadLex

RadLex[27] is a radiological lexicon, maintained by the Radiological Society of North America (RSNA) [132]. It provides imaging -related terms including imaging technologies, imaging findings, anatomy, and pathology [242]. RadLex is translated to German [150], the current version (4th version, released January 2019) contains 46636 classes (concepts) and is publicly available as OWL, CSV, RDF, XML format.[28]

Each concept is identified with a unique ID (RID) and contains preferred names for each language. The concepts are hierarchically structured to topics, such as *clinical findings*, *imaging modality*, *procedure* or *imaging observation*. Figure 2.3 shows a part of the *imaging observation* section. Concepts are represented with their preferred name and their RadLex ID.

2.2.5.4 LOINC

The Logical Observation Identifiers Names and Codes (LOINC) database is developed by the Regenstrief Institute of the Indiana University School of Medicine and was initially published in 1996. It covers 98% of the average laboratory's tests [72]. The current version (version 2.65, released in December 2018) contains 89,271 terms and is publicly available.[29] LOINC has been adopted in many countries, such as Switzerland, Australia, Canada, and by the German national standards organization (germ. *Deutsches Instituts*

[25] `https://confluence.ihtsdotools.org/display/DOCSTART/6.+SNOMED+CT+Concept+Model`, accessed: February 2019

[26] `https://confluence.ihtsdotools.org/display/DOCANLYT/4.5+Expressions`, accessed: February 2019

[27] `http://www.radlex.org/`, accessed: February 2019

[28] `https://bioportal.bioontology.org/ontologies/RADLEX`, accessed: February 2019

[29] `https://loinc.org/`, accessed: February 2019

```
Imaging observation (RID5)
├── Assessment (RID36026)
├── Asymmetry (RID34265)
├── Breast implant observation (RID49716)
├── Cortical bone observation (RID45758)
├── Density-related entity (RID45721)
├── Destruction pattern (RID6299)
├── ...
└── Lesion characteristic (RID43356)
    └── Effect of mass on surrounding tissue (RID34379)
        ├── Entering airway (RID46035)
        └── Lesion effect on liver (RID45677)
            ├── Abuts capsule of liver (RID43327)
            ├── Abuts hepatic vein (RID43329)
            ├── Abuts portal vein (RID43331)
            └── Bulges capsule of liver (RID43328)
```

Figure 2.3: **Hierachical structure RadLex.** Example of "Bulges capsule of liver".

Table 2.3: **Six axes of LOINC.** A LOINC name must contain values of the six major axes.

#	Name	Example
1	component	hemoglobin
2	measured property	mass concentration, enzyme activity
3	time information	24-h urine
4	type of sample or organ examined	urine, blood, chest
5	type of scale	quantitative, ordinal, nominal, or narrative
6	used method	

für Normung) and considered as universal standard [152]. LOINC also is used by other standards like IHE or HL7.[30]

Loinc aims to encode all medical laboratory observations with unique IDs, in order to assist in the electronic exchange and gathering of clinical results. A LONIC code refers to a LOINC name that consists of six values, each referencing a property of an observation. These properties are organized in six major groups (called axes). Table 2.3 lists this axis and give examples [152].

For example, the LOINC-code `2951-2` references to the LOINC-name `2951-2 SODIUM:SCNC:PT:SER/PLAS:QN` and can be composed to the *component* = `SODIUM`, the *property* = `SCNC` (substance concentration), *time* = `PT` (point in time), *system* = `SER/PLAS` (serum or plasma) and *scale* = `QN` (quantitative).

[30] `https://wiki.hl7.de/index.php?title=Logical_Observation_Identifiers_Names_and_Codes_(LOINC)`, accessed: February 2019

2.2.5.5 Operation and Procedure Keys

The *Operationen- und Prozedurenschlüssel* (OPS)[31] (engl. *Operation and procedure keys*) is the German version of the International Classification of Health Interventions (ICHI) published by the German Institute for Medical Documentation and Information (DIMDI) on behalf of the Federal Ministry of Health (BMG) [83]. OPS is the official classification of operational procedures in medicine in Germany. OPS and ICD-10 code are used to generate the Diagnosis Related Groups (DRG) codes. It is free to use and online accessible.

The codes are organized in hierarchical structure, consisting of nine chapters with the subjects: diagnostic measures, imaging diagnostics, operations, drugs, non-surgical therapeutic measures and complementary measures. Chapters are divided in groups, categories and sub-categories. Figure 2.4 gives an example of the hierarchical structure. Each entry has a name and optional descriptions, inclusion criteria and exclusion criteria.

2.2.5.6 ATC

The Anatomical Therapeutic Chemical (ATC) Classification System is a classification system of drugs and their substances, organized and proposed as international standard since 1981 by the WHO aiming to monitor drug utilization over time.[32] A German version is provided by the German Institute for Medical Documentation and Information (DIMDI).[33]

The active substances of drugs are structured in a hierarchy with five levels with fourteen top-level groups representing anatomical or pharmacological areas. The second layer indicates a pharmacological or therapeutic use. The next two levels are chemical, pharmacological or therapeutic subgroups, followed by last (5th) level containing the chemical substance.[34] Table 2.4 illustrates the system and gives an example for the drug "Ramipril".

2.2.5.7 Further Classifications and Standards

Medical Subject Headings. The Medical Subject Headings (MeSH) is a thesaurus for medical vocabulary with the purpose to index medical databases to enable powerful queries [138, 142]. For example, the well-known database and search-engine for medical articles PubMed[35] uses MeSH for query expansion and to extract topics of publications [144, 145]. MeSH is free to use and online accessible in the English and German Language. It comprises various medical topics, such as anatomy, organisms, diseases, chemicals and

[31] `https://www.dimdi.de/dynamic/de/klassifikationen/ops/`, accessed: February 2019
[32] `https://www.whocc.no/atc_ddd_methodology/purpose_of_the_atc_ddd_system/`, accessed: February 2019
[33] `https://www.dimdi.de/dynamic/de/arzneimittel/atc-klassifikation/`, accessed: February 2019
[34] `https://www.whocc.no/atc/structure_and_principles/`, accessd: February 2019
[35] `https://www.ncbi.nlm.nih.gov/pubmed/`

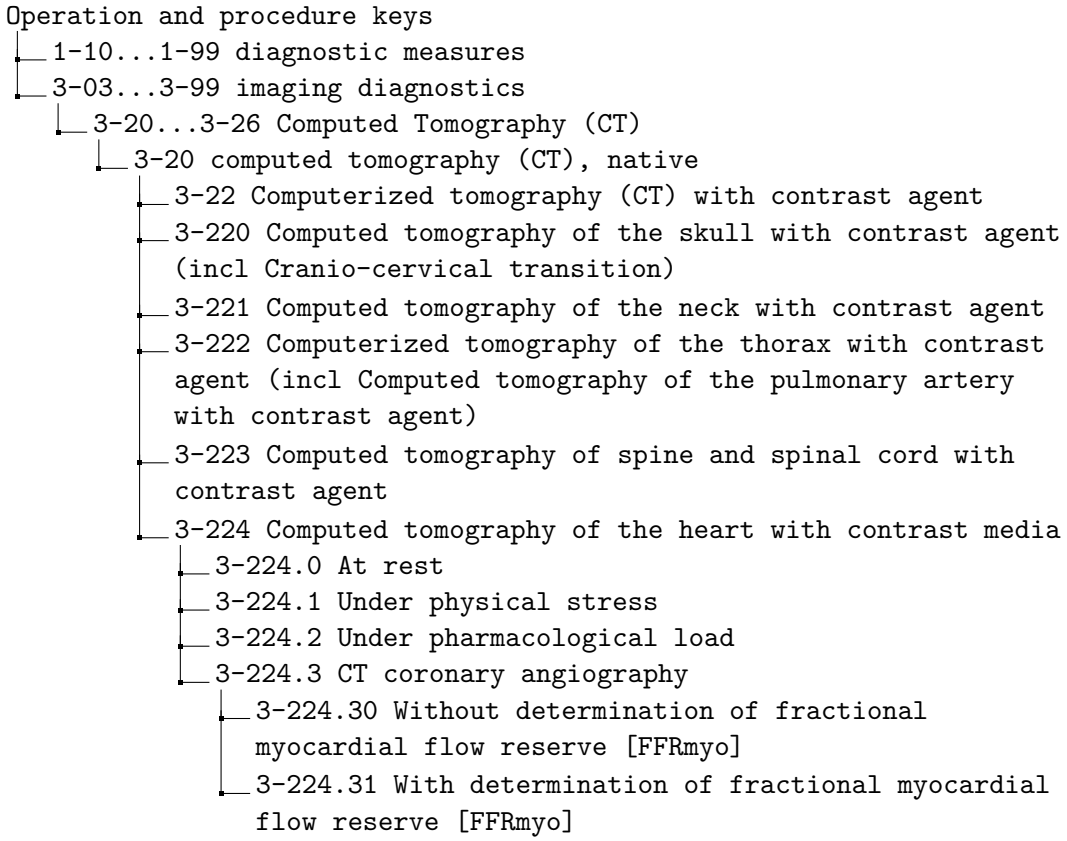


Figure 2.4: **Hierarchical structure of OPS.** Example of "Computed tomography of the heart (coronary angiography) with contrast media and determination of fractional myocardial flow reserve".

Table 2.4: **Hierarchical structure of the ATC classification system.** The system consists of five levels, each describing a particular characteristic of a drug. "Ramipril" is given as example.

Level	Content	ID	Name
1	anatomical main group	C	Cardiovascular system
2	therapeutic subgroup	C09	Agents acting on the renin-angiotensin system
3	pharmacological subgroup	C09A	ACE inhibitors, plain
4	chemical subgroup	C09AA	ACE inhibitors, plain
5	chemical substance	C09AA05	Ramipril

Table 2.5: **Sample of Alpha-IDs for I64: Stroke.** Alpha-ID codes are mapped to ICD-10 codes. Table shows the first lines for the ICD-10 diagnoses "I64 Stroke" of all 39 entries.

Alpha-ID	ICD-10	Synonym
I24888	I64	Hirnstamminsult
I25725	I64	Apoplex
I25726	I64	Hirnschlag
I25727	I64	Gehirnschlag
I25728	I64	Zerebrovaskulärer Insult
I25729	I64	Schlaganfall
I25730	I64	Zerebraler Insult
I25731	I64	Hirninsult
I25732	I64	Angiospastischer Insult
I25733	I64	Zerebrale Apoplexie
I25735	I64	Akute zerebrale Lähmung
I25736	I64	Akute Zerebralparalyse

drugs, diagnostic and therapeutic techniques and equipment, technology and food and beverages, persons and health care. The current MeSH versions 2019 contains 29,351 main heading as well in English as in German, and 214,879 synonyms in English (68,789 in German).[36]

Alpha ID. The Alpha-ID is a German mapping of everyday language diagnosis names and ICD-10 diagnostic codes. Each entry has a stable, unique identification number: the Alpha ID code. Alpha-ID is maintained and provided by the German Institute for Medical Documentation and Information (DIMDI).[37] It is available as a CSV version and is currently revised annually.

Table 2.5 depicts a sample of the Alpha-ID list for ICD-10 diagnoses "I64 Stroke". Diagnoses names and their Alpha-ID codes are mapped to ICD-10 codes.

2.3 Clinical Data Warehouses and Text Query Features

This sections presents the most popular and the most relevant CDWs with a brief overview and lists their features to query text data. Most CDWs does not support textual queries very well. That revealed a research in literature and websites of CDWs, their extensions, patient recruitment and clinical systems. Most of them do not describe how textual data can be queried. A few systems provide information on how the data is stored, which allows conclusions on the possible text related query features.

[36] `https://www.dimdi.de/dynamic/de/klassifikationen/weitere-klassifikationen-und-standards/mesh/`, accessed: February 2019

[37] `https://www.dimdi.de/dynamic/de/klassifikationen/icd/alpha-id/`, accessed: February 2019

2.3.1 i2b2 & tranSMART

Informatics for Integrating Biology and the Bedside (i2b2) provides open source software tools for clinical investigators to collect and manage clinical research data. The i2b2-platform has been adopted by over 200 hospitals and research centers.[38]

tranSMART is built on top of i2b2 and focuses on clinical trials and omics data in order to investigate correlations between phenotypic and omics data [26]. The corresponding tranSMART Foundation and the i2b2 Foundation decided to merge in 2017.[38]

While groups of patients can be searched in i2b2 by defining filters and constraints to recruit patients for clinical trials, tranSMART is primarily used for the analysis of different cohorts within studies.

2.3.1.1 History

Informatics for Integrating Biology and the Bedside (i2b2) is a National Center for Biomedical Computing (NCBC) located at Partners HealthCare System in Boston [35]. It was founded by the National Institutes of Health (NIH) in 2004. One year later, the i2b2 Foundation was established at Harvard medical School. The i2b2 Center develops frameworks to support the translation of genomic findings and to enable the work with hypotheses in model systems relevant to human health. The first i2b2 workbench was published in 2007[39] with the goal to provide software tools for clinical investigators to collect and manage clinical research data such as genomics data [161].

The tranSMART Foundation was founded by the Pistoia Alliance, University of Michigan, Imperial College London among others in 2013, resulting in a collaborations between scientists in the United States and the European Union. More than 300 organizations joined the tranSMART Foundation. The first version of the tranSMART data management system was published in 2009 for pharmaceutical researchers [92]. More than 100 organizations joined the tranSMART community.[38]

The tranSMART Foundation and i2b2 Foundation merged to create a single organization in 2017. They are integrating their information and analysis platforms that are used in the clinical research and transnational.[38]

2.3.1.2 Architecture and Features

i2b2. i2b2 uses an extended entity attribute value (EAV) model for observations. The medical concepts are organized in a terminology. All data is stored in a relational database: supported are MS SQL, MySQL and PostgreSQL.

The i2b2 system has a cell structure called *i2b2 Hive*. Cells have different functions and communicate via defined interfaces. Figure 2.5 shows the structure and important cells.

[38] `https://transmartfoundation.org/our-history/`, accessed: January 2019
[39] `https://www.i2b2.org/software/archive.html`, accessed: January 2019

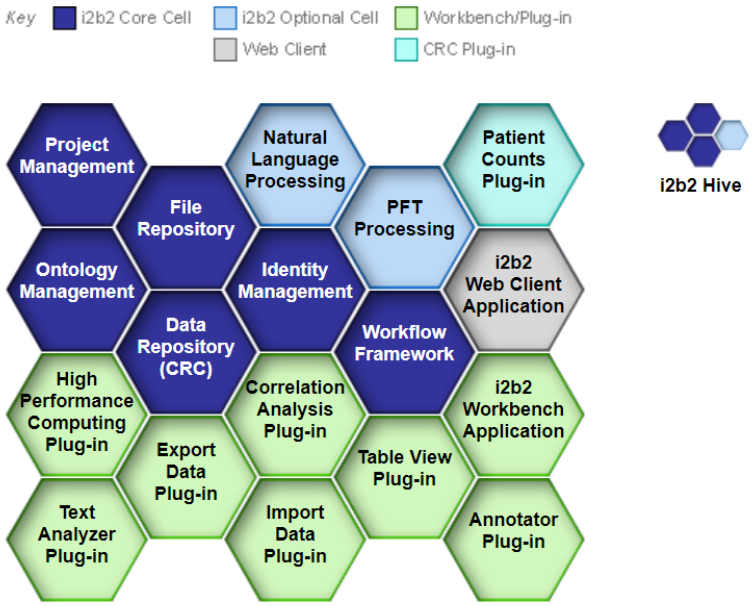


Figure 2.5: **Cell structure of the i2b2 Hive.** Core cells, workbench cells and others are connected with interfaces.[40]

In addition to basic functions as data import/export, identity management and repository management, advanced features exist, like the *Natural Language Processing* cell that can be used for interactive work in order to extract information from text with the conventional approach including gold standard creation among other things. A plug-in structure enables the usage of further components, such as *table view plugin* or a *correlation analysis plugin*.[41]

i2b2 offers a workbench and a web client as user surface. The web client can be seen in Figure 2.6.

tranSMART. tranSMART is built on top of the i2b2 system. In addition, it can handle high-content biomarker data such as gene expression profiles, genotypes, metabolomics and proteomics data [26]. Furthermore it provides analysis tools for advanced descriptive and analytics statistics.

Data management is based on i2b2 using a terminology to organize medical concepts. tranSMART uses index engines (Apache Solr) and NoSQL document stores (MongoDB) in addition to relational databases.

[40] Source: `https://www.i2b2.org/software/index.html`, accessed: January 2019
[41] `https://www.i2b2.org/events/slides/i2b2_AMIA_Tutorial_20100310.pdf`, accessed: January 2019
[42] Source: `https://www.i2b2.org/webclient/`, accessed: January 2019

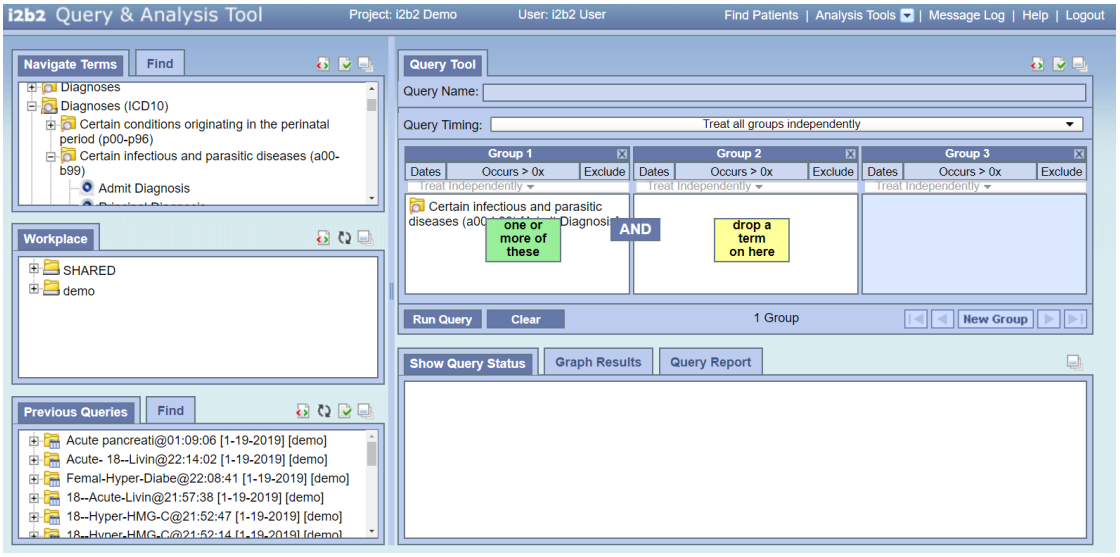


Figure 2.6: **User interface of the of i2b2 web client.** Screenshot taken from the i2b2 demo site[42].

tranSMART integrates functionality and surfaces of third party frameworks like R or omics analysis frameworks: Bioconductor [252] and GenePattern [187]. Figure 2.7 shows the genome browser in tranSMART.

2.3.1.3 Text Query Features

i2b2. i2b2 stores their data in SQL-DBs. A full text index can be built on text data. The `LIKE`-operator can be used to query these texts[44]. The generated SQL queries look like `WHERE column_xy LIKE '%token%'`, if the word "token" is searched in the column "column_xy".

In addition, a SQL full text index can be build on text columns as well.[45] The resulting SQL queries look like `WHERE CONTAINS(column_xy, 'token1 AND token2')`. The `CONTAINS`-predicate is much faster than the `LIKE`-operator because it uses a full text index with look-up operations instead of scanning all texts. Single or multiple words can be queried with the so generated SQL statement. A basic lexical analysis is applied on the queried word, like *stemming* and *loweracseing*, in order to get more matched results. Stemming removes the word endings to eliminate the inflections of words and return a stem of a word. The queried words must occur anywhere in the text. It is not specified, that they are next to each other or in a semantic relation. In addition to words, prefix-terms can be searched as well. A prefix-term is matched with any term in the text

[43] Source: `https://jira.transmartfoundation.org/`, accessed: January 2019
[44] `http://community.i2b2.org/wiki/display/DevForum/Text+search+in+i2b2`
[45] `https://community.i2b2.org/wiki/display/IGD/Demo+Data`, accessed: January 2019

Figure 2.7: **Genome browser of tranSMART.** Screenshot taken from the tranSMART demo site[43].

that start with the specified sequence. For example the query "ware*" would match the word "warehouse" [39].

tranSMART. tranSMART was built to manage many studies and to analyze the data of these studies. When using tranSMART, a study must be selected first. This is done with the *Browse*-features that lists all existing studies. A query tool facilitates the retrieval by providing a keyword search and a full text search to filter these studies. The queried tokens may be searched in some data sources: the study or analysis name, in genes, in chromosomes, in diseases and observations. tranSMART indexes all data (files and table records) in Apache Solr in order to facilitate this search [92]. However, patients cannot be searched here, only whole studies [232].

The *Analyze* function can be used to compare two different cohorts that can be specified by the researcher on its own. The analyzer give a deeper insight into the data and hypotheses can be tested. The cohorts are defined by constraints. Only Boolean and numeric concepts (medical entities) can be used to select the patients. Text data cannot be queried [232].

2.3.2 openEHR

openEHR is a community with the goal to convert health data from a physical form into an electronic one by ensuring universal interoperability among all forms of electronic

data. Their main focus in on electronic health records (EHR) and related systems.[46]

The approach of openEHR is a *multi-level, single source modelling* within a service-oriented software architecture. Domain experts built models (archetypes) that represent such level layers, such as "Tobacco smoking summary".[46]

The openEHR foundation was founded 2003 as nonprofit organization [111]. openEHR is used by European research projects as the current *MY AIR COACH*[47], by open source organizations, governments and academic research projects of ten countries[48].

openEHR uses the *Archetype Query Language* (AQL) to query information. In addition to `EXIST` and `NOT EXIST` that assess the existence of a Boolean concept, numerical operators `greater than`, `greater than or equal to`, `smaller than` and `smaller than or equal to` in order to constraint numerical concepts. String/text values can be queried with the operator support `equals to`, `not equals to` and `matches`. The operator `equal` checks if a string value is identical with a queried string. The operator `matches` is similar, but instead of a single word, a list of allowed search terms is passed. A query looks like "`code_string matches{'18919-1', '18961-3', '19000-9'}` " [169]. Figure 2.8 shows a query in the user interface.

AQL queries the data model and not the data schema in the database. openEHR does not specify the data store engine. Vendors of openEHR implementations can use different data stores, e.g. SQL, NoSQL, document stores or others. The fact that AQL offers almost no feature to search in texts, certainly is related with the fact that various openEHR providers could hardly provide them.

2.3.3 STRIDE

The *Stanford Translational Research Integrated Database Environment* (STRIDE)[50] has three components: (1) a clinical data warehouse, (2) a framework to develop research data management applications and (3) biospecimen data management system [143]. STRIDE was developed at the Stanford University and published in 2009. The data, modeled in an EAV-data schema, is stored in an Oracle relational database. It can be enriched with semantic linkage to medical ontologies, such as ICD9-CM (International classification of diseases) [209], ICDO (International classification of diseases for oncology) [174], CPT (Current procedural terminology) [98], RxNorm (standardized nomenclature for clinical drugs) [139] and SNOMED (Systematized Nomenclature of Medicine) [41]. The data is imported into the CDW by receiving HL7 messages from numerous clinical systems. Structured and unstructured data from various domains are supported such as diagnoses, procedures, laboratory values, pharmacy orders (structured) and radiology reports, pathology reports and clinical documents (unstructured text documents) [143].

[46] `https://www.openehr.org/what_is_openehr`, accessed: January 2019
[47] `http://www.myaircoach.eu/`, accessed: January 2019
[48] `https://www.openehr.org/who_is_using_openehr/`, accessed: January 2019
[49] Source: `https://www.youtube.com/watch?v=pC6mUtqqK9U`, accessed: January 2019
[50] `https://med.stanford.edu/researchit/infrastructure/clinical-data-warehouse.html`, accessed: January 2019

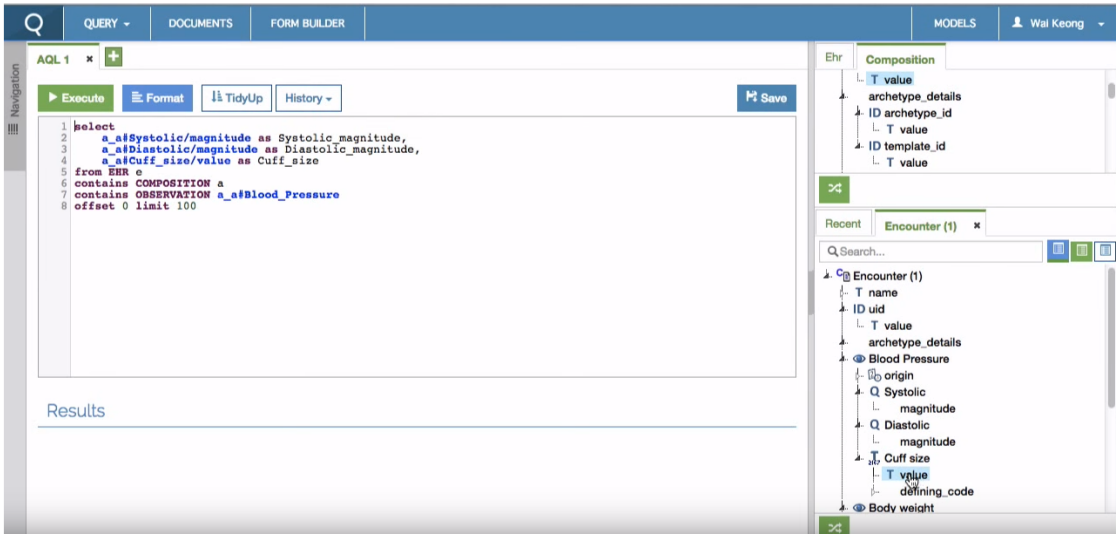


Figure 2.8: **AQL query surface in openEHR.** AQL query is specified on the left side and data elements that can be queried on the right side. Screenshot taken from the openEHR demo showcase.[49]

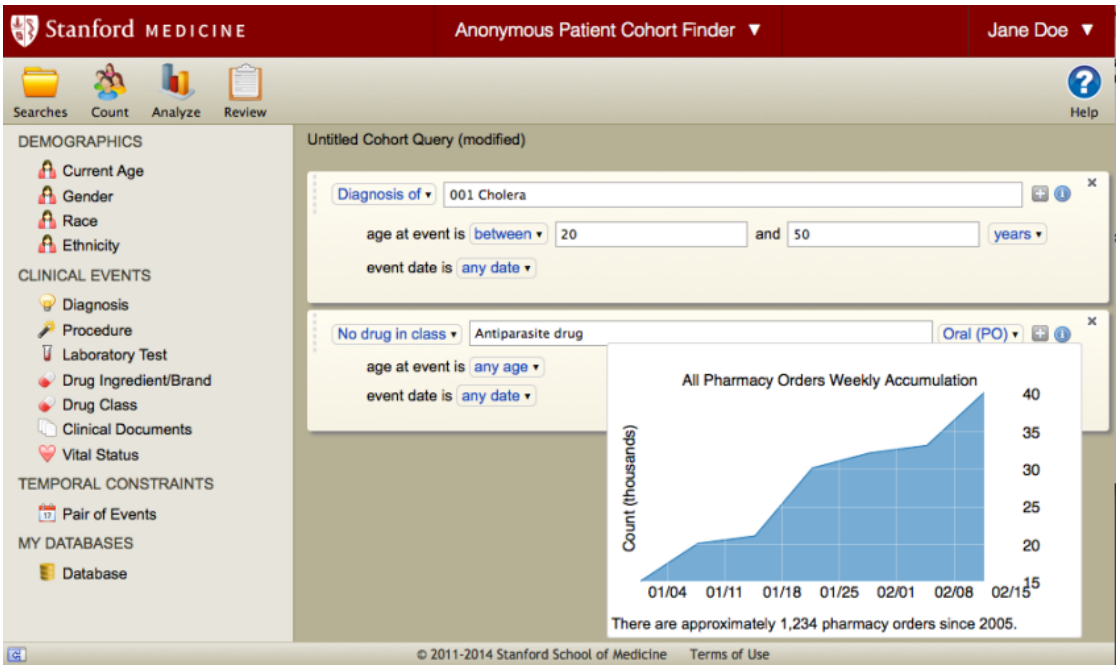


Figure 2.9: **Patient cohort finder GUI of STRIDE.** Cohorts can be identified and analyzed anonymously with this user interface. Image is taken from STRIDE website.[51]

Users can query patients in the cohort discovery tool via graphical interface. (See Figure 2.9.) All clinical text documents and reports are full-text indexed and searchable using Oracle Text [143].

2.3.4 Roogle

Roogle is a project of a document oriented CDW with the aim to offer a search engine that combines semantic search and full text search. Roogle is not available on the web, there is no version to download and no visible code repository. It is a feasibility study to demonstrate the benefits of CDW with full-text search capabilities. The system is not well documented and all information are gained by one paper [44].

Roogle uses an Oracle database to store patient information in a *star* schema. The patient data is semantically enriched by extracting medical concepts form reports using NOMINDEX [90]. The concepts are restricted to the MeSH thesaurus. All patient data, including structured, text data and extracted data, is indexed with the Apache Lucene library [94]. This data can be queried with a search engine. Structured data, like sex, age and medical department, can the retrieved and full-texts searches can be performed on text data.

2.3.5 Dr. Warehouse

Dr.Warehouse[52] (DrWH) an open-source document-oriented data warehouse focused toward clinicians [81]. The main features of this CDW is screening patients by searching in text data for cohort recruitment and phenotying. That system is dedicated for the use by clinicians and is built for French texts [80].

Dr. Warehouse was deployed in the first hospital (Necker Enfants Malades Hospital in Paris) in January 2017 [81]. The code is open source since September 2017.[53] It is also installed at the European Hospital George Pompidou (HEGP) in Paris, which was the first hospital in French that used i2b2.

In the center of the database model are narrative reports. Further tables containing structured data, such as demographic data, diagnosis codes, and movements, are related to the narrative reports [81]. The centerpiece of the architecture is a search engine that retrieves relent document. Displaying these documents is a main principle of the CDW. Dr. Warehouse deals with negations and the context of information. Texts are split up into sentences and propositions with regular expressions. It is determined, for each proposition whether it is negated or not and whether it is related to the patient or to the family history. The triples (proposition, negation [yes/no], context [patient, family history]) are stored in a table in the DB [80].

[51] Source: `https://med.stanford.edu/researchit/infrastructure/clinical-data-warehouse/cohort-tool.html`, accessed: January 2019

[52] `http://www.drwarehouse.org/`, accessed: January 2019

[53] `https://github.com/imagine-bdd/DRWH`, accessed: January 2019

All text data is indexed with the Oracle Text Module[54] and can be queried and displayed in an anonymized mode or with full identity of the patients. However, all texts are de-identified by removing names, birth date, addresses and phone numbers from the documents [81]. Structured data is linked to that text data, facilitating queries on full-text and coded data.

Query expansion for phenotypes using synonyms of the Unified Medical Language System Metathesaurus (UMLS) [137] is mentioned in [81]. However, it is described in a different way: Phenotypes terms in French are gained from the UMLS Metathesaurus in order to identify these phenotype concepts in the searchable texts. These UMLS concepts and their synonyms are added to the document [81]. Thus, the query is not expanded, but the documents are semantically enriched.

Users can query Dr. Warehous in a "Google like" manner by entering some keyword (symptoms, diagnoses) and get relevant documents displayed [81]. The system searches by default in positives statements: Phrases that are not negated and relate to the patient, not to the family history. An additional query feature is the misspelling correction. The system displays the most similar queries if a query return less than ten patients [81].

The user interface of Dr. Warehouse can be seen in Figure 2.10. The system retrieves patients and documents for the queried token "infection% and eczema and thrombocytopenia". The tokens are highlighted in the documents found and appear at any position of the test. Further query features concerning the semantic queries with related words are not reported.

2.3.6 Other Clinical Data Warehouses

ArchiMed. A Medical Information- and Retrieval System (ArchiMed) was published in 1999. It was developed and deployed at the Vienna General Hospital-University Hospital. ArchiMed is no classic CDW, is is described as medical data storage and retrieval system [59]. However, it is able to completely manage patient data including storing, retrieving, filtering and displaying patient data [59]. Text query features are not reported.

BigQ. BigQ is a NoSQL based framework to handle genomic variants in i2b2. BigQ extends i2b2 with a data store for genomicx data. It uses the NoSQL document store CouchDB[55], which is a project of the Apache Software Foundation. The data is imported with a specially developed tool and can be queried from the i2b2 data warehouse with the BigQ-plugin. The query focuses on the genetic data such as interval queries with start and end positions for genomes and does not contain features to query narrative texts [78].

[54] `https://www.oracle.com/technetwork/testcontent/index-098492.html`, accessed: January 2018
[55] `https://couchdb.apache.org/`, accessed: January 2018

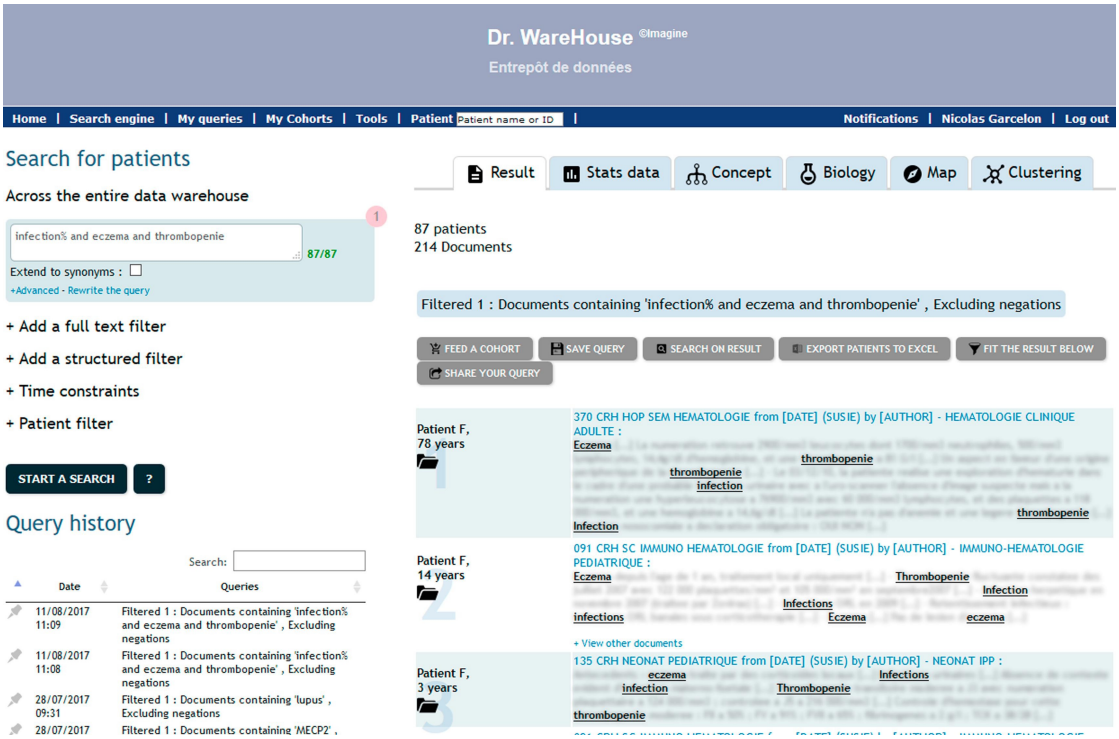


Figure 2.10: **Dr. Warehouse user interface with search engine.** On the left side, the tokens "infection% and eczema and thrombocytopenia" are queried. The retrieved, relevant patients and their documents are displayed on the right side. Image is taken from [81].

ConSoRe. ConSoRe is a project that supports the implementation of big data in oncology. It is developed by the teams of the 4 French pilot centers (the Georges-François Leclerc center in Dijon, the Léon-Bérard center in Lyon, the Montpellier cancer institute and the Curie institute in Paris) in collaboration with and external company (SWORD). The system contains structured and unstructured data that can be retrieved, but further detail are not described in the short article written in French [97].

DW4TR. The Data Warehouse for Translational Research (DW4TR) has been developed as generalizable system to combine data of different research domains such as the clinical and laboratory domains.

It has two data models: a patient centered clinical data model and a specimen centered molecular data model. They include temporal relationships within the data. Medical images like mammograms are de-identified with DICOM [12] and linked with the patients in the CDW. An Oracle relational database is used as persistence layer.

The system provides two user interfaces: An *Aggregated Biomedical-Information Browser* (ABB) and an *Individual Subject Information Viewer* (ISIV). ABB performs ROLAP operations and calculates the queried information at the time of use. ISIV does not perform queries, it displays all information of patients for a given ID or a pre-defined cohort.

It is not mentioned wether text data can be integrated in the system or how it can be queried [103].

EMERSE EMERSE[56] (The Electronic Medical Record Search Engine) is a search engine for free-text documents in the electronic medical record, developed by the University of Michigan [89]. EMERSE is no clinical data warehouse, but rather an information retrieval tool based on Apache Solr.[56] It aims to support operational use cases, such as quality management, risk management, health information management, registries and coding compliance.

The data that can be integrated are just medical text documents, patient demographics (name, sex, age) and study memberships.[57] The user interface looks very similar to Google and consists of one input box. This easy-to-use surface is built to facilitate query operations for clinicians and other users with no computer science background.

The system provides a simple negation handling: Users can manually add negated formulations of the queried word to be excluded of the search. For example, if a user queries the word “fever”, he can add the terms “no fever” to his search and specify, that these terms must not appear (exclusion criteria).

[56] `http://project-emerse.org/`, accessed: January 2019

[57] `http://project-emerse.org/documentation_archive/emerse_documentation_version_03_5/data_guide.html`, acccessed: January 2019

Harvest. Harvest[58] is a platform consisting of an open-source framework with modular components for developing web-based biomedical data discovery and reporting applications [173]. However, it is no clinical data warehouse, but rather a data application framework for biomedical purpose. It was deployed at the Children's Hospital of Philadelphia (CHOP) and the system was demonstrated with infectious disease data, published by the OpenMRS open-source electronic health record (EHR) project [103]. A main component of Harvest is the data abstraction layer Avocado, which indexes text data for "subsequent search". Avocado is used to reveal inherent characteristics of the raw data, such as determination of categorical versus continuous type [103]. The text query features are not further described in detail.

METEOR. METEOR is described as an enterprise health informatics environment to support evidence-based medicine. It is developed at the Houston Methodist Hospital and consists of an enterprise data warehouse and of a software intelligence and analytics layer. It shall be used for clinical decision support, such as hypothesis testing, cohort identification, data mining, risk prediction, and clinical research training. The system contains a free text information retrieval module with a standard vector space model to retrieve relevant (fuzzy matching) documents like patients' medical reports, laboratory results, and any other records. Information can be extracted from these texts by manually reading and screening [179]. The system is not available in the web.

radBank. radBank is a data warehouse for integrating radiologic and pathologic data developed by the Stanford Medical Informatics Department. It uses a relational Microsoft SQL database to store demographic data, ICD-10 codes and radiology reports and pathology reports. A parsing module recognizes headings of sections in these reports with regular expressions and stores these sections separately enabling a keyword search in individual sections. The user interface of radBank is an SQL query interface [191].

SMEYEDAT. The smart eye database (SMEYEDAT) is an ophthalmologic DW at the University Eye Hospital of Munich, Germany.

Its patient oriented data model stores EHR data in a Microsoft SQL database. The records are linked to the *picture archiving and communication system* (PACS) with the *digital imaging and communications in medicine* (DICOM) standard using the Health Level 7 (HL7) standard.

A developed discovery tool enables the exploration and visualization of patient data. Patients can be queried by defining filters like number of interventions, medical therapies, and diagnoses or by a free text search for keywords. However, no natural language processing is implemented [123].

[58] `http://harvest.research.chop.edu/`, accessed: January 2019

Starmaker. Starmaker is a data warehouse tool used in pathology with studies mainly concerning laboratory values. Starmaker includes some data mining functions and is used for scholarly or quality improvement projects. Simple queries can be created with a visual interface, more complex queries can be entered with SQL. A database is used as persistence layer [124]. Text query features are not described, but they would be restricted to the DB-server capabilities.

The Enterprise Data Trust. The Enterprise Data Trust is a collection of data from patient care, education and research including genomics data. Its aims are to support information retrieval, business intelligence, and high-level decision making especially for clinical and biomedical research. The unique characteristic is the focus on data governance including substantial resource investment in consensus information models [32]. The abilities to handle or query text are not stated. The system has no web appearance and does not offer any resources as a demo site, a system version to download or a code repository.

Vanderbilt CDW. The research data warehouse framework of the Vanderbilt University (Nashville, Tennessee, USA) consists of two repositories for identified and de-identified clinical data.

The fully identified research data warehouse called the *Research Derivative* (RD) contains data of many domains and subsystems and can be considered as research optimized mirror of the clinical information system. The de-identified version called of RD *Synthetic Derivate* (SD) contains links to an anonymized DNA biobank [47].

The systems contain structured and unstructured data such as medication prescriptions. Information extraction is performed on these texts during data import in order to e.g. extract drugs using MedEx [249].

The data can be queried via two web-based user interfaces. Both provide the same query functionality, but the *Record Counter* user interface just displays aggregated counts of patient groups and the *Synthetic Derivate* user interface allows the review of the patient data including DNA data. Requests are built with the query generator by selecting specific inclusion/exclusion criteria for diagnoses, medication or procedure codes [47]. It is not mentioned that text data can be queried.

2.4 Processing Natural Language

This section presents algorithms and procedures to process natural languages that are relevant to this work. Starting with basic steps of lexical analysis, followed by approaches of conventional information extraction and concluding with mechanisms to identify the context of information in texts.

2.4.1 Lexical Analysis

Linguistic Preprocessing is an important step in dealing with natural language. It is important to clearly define letters, words and phrases, for further linguistic analysis in machine language processing or information retrieval [100]. In order to define these units, various problems have to be solved.

The following section will introduce the text preprocessing techniques relevant to this work.

Tokenizing. *Tokenizing* is the division of the input text into small units called *tokens*, where a token is a word or something similar, such as a number, or a punctuation mark [148]. The definition of a word is controversial in linguistics. The *graphic word*, a alphanumeric string covered by whitespaces, proposed 1967 [130], does not fit to modern words, such as "C++", "T.V." or "info@uni-wuerzburg.de". Other issues are compound words like the German "Lebensversicherungsgesellschaftsangestellter" in contrast to the English translation "life insurance company employee". There is no absolute definition for the process of tokenizing, which defines a word. In most cases, the application determines the procedure [106]. Space separated languages can be treated with simple tokenizers such as a whitespace tokenizer. In many cases, advanced tokenizers are used with rule-based or regular-expression techniques [106, 148].

Stemming. Words can appear in texts in various grammatical forms due to declination or conjugation, such as "organize", "organizing" and "organized". Moreover, similar words can be summarized to word families, e.g. "browser" and "browse". The goal of stemming is a reduction of diffraction forms and the return of related words to a common basic form, as the following example shows:

relate, related, relating, relation, relational → relate

Stemming is usually a rather crude, heuristic process that cuts off the endings of words. Stemmer work with manually defined language-specific rules. However, they do not always deliver the correct result, but they are fast and robust, and unlike lemmatization (see below), they do not require a vocabulary.

The most common algorithm for English texts is the very effective *Porter Algorithm* [147].[59] Five word reduction operations are sequentially applied, each is represented with action rules consisting of word suffixes as condition and replacements as action. A well-known implementation is the Porter Snowball stemmer [176].

Lemmatization. An alternative approach to stemming is *lemmatization*, which uses a morphological analysis to remove the diffraction word from a word and return the word to its basic form (*lemma*). Basic approaches use a dictionary to lookup the lemma for each word [147].

[59] `http://tartarus.org/~martin/PorterStemmer/` accessed: February 2019

Stop Words Removal. Very common words that are not relevant to the content of the text are called *stop words* [147, p. 27]. Common stop words are articles, conjunctions, and frequent prepositions. The general strategy for determining stop words is to sort the words by their frequency in the corpus and to index the most common words on a *stop word list*. Using stop-word lists significantly reduces the amount of time a system must process and store [147, p. 27]. Stop word are removed in many NLP applications in order to speed up the process. However, the trend in *Information Retrieval (IR)* systems goes from quite large stop word lists (200-300 terms) to very small lists (7-12 terms), up to no stop word lists [147].

Part of Speech Tagging. Natural language processing usually follows a particular sequence of steps: Beginning with a phoneme and morpheme-based analysis and leading to a semantics or discourse analysis [106, p. 205]. One of the first steps in this pipeline is the so-called *part-of-speech* (POS) *tagging*. This is the assignment of words and punctuation marks of a text to the individual parts of speech.

The assignment of the tokens to the parts of speech tags requires a set of tags called *tagsets*. The Stuttgart-Tübingen-Tagset (STTS)[60] is commonly used for the German language. The sentence "The human has 206 bones." would be tagged in German with the STTS as shown below: (The tag is appended to the corresponding word after a delimiter "/")

Der/ART Mensch/NN besitzt/VVFIN 206/CARD Knochen/NN ./$.

The POS-tag "ART" indicates an article, "NN" a normal noun, "VVFIN" a full finite verb, "CARD" a cardinal and "$" a punctation.

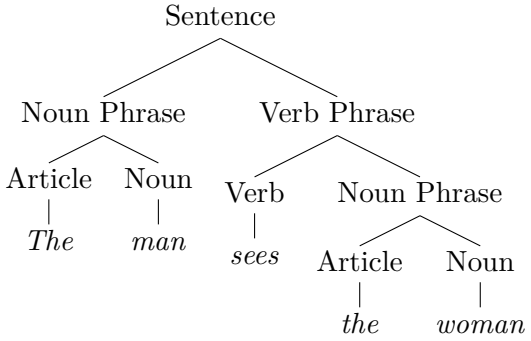
There are different approaches to POS tagging: Generative methods using rule-based system [19, 96] or *hidden Markov models* (HMM) [61, 67, 93], but they are outperformed by discriminative approaches like maximum entropy models [186], since they are more flexible and can better make use of contextual information [106].

Good results for German texts shows the Trigrams'n Tags (TnT) statistical POS tagger [22].

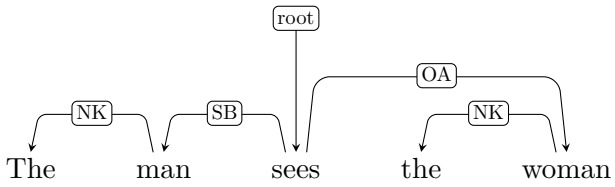
2.4.2 Parsing of Natural Language

The literature mainly contains techniques for constituent parsing and dependency parsing. Both can be distinguished into two different approaches, a) grammar-driven approaches and b) data-driven techniques. Constituent parsing analyses a sentences and builds hierarchical models of phrases, such as noun phrases or verb phrases. For example the phrase "The man sees the woman" is parsed as:

[60] `http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-table.html`



Dependency parsing seeks to identify relationships of words: A dependency consists of a *head* that is modified by another word. For example, the arrow between "man" and "sees" indicates that "man" affects "sees". The dependency graph of the above mentioned examples looks like:



The best known representatives for grammar-driven dependency parsing in English are Pro3Gres [201] and ParZu [204] for German language, both rule-based methods implemented in Prolog. ParZu achieves a labelled F1-score of 89.5% on TüBa-D/Z [48], the best known treebank in German. A treebank is parsed text corpus with annotated syntactic or semantic sentence structure. ParZu is based on the comprehensive constraint-dependency grammar of German language introduced by Foth [73].

Stanford University offers two constituent parsers: A basic parser that works with Probabilistic context-free grammar (PCFG) ,[119] and a lexicalized PCFG parser, that uses as well executing rules and in addition the head of a phrase is included to each rule [118]. The German model for this was supplemented by Rafferty in 2008 [184] and delivers 88.8% F1-score on Tüba-D/Z and 74% on the TIGER corpus [21] in the best setting.

Diverse approaches exist for data-driven language parsing: Constituent parsing uses techniques such as Parse Reranking [36], Tree-CRFs [70], and neural CRF parsing [60]. The basic idea of Parse Reranking is to determine all valid parse-trees with a grammar in the first step and then to rate them in a second step using a machine learning scoring function, so that the tree with the highest score is evaluated as output. The CRF evaluates the applied rules during decoding and determines the best tree directly.

Data-driven dependency parsing has been the more dominant technique for about 10 years, as the task is easier to model. Instead of building hierarchical trees, the challenging task for dependency parsing is to find the right head for each word. The result can be

heuristically merged relatively well, an algorithm can be found in [129]. This dominance was reinforced by CoNLL challenges in 2006 and 2007, which translated dependency parsing to different languages with results (labeled attachment scores) ranging from 76.31 (Greek) and 87.3 (German) to 91.7 (Japanese) [24, 166].

Recently, the evaluation and annotation guidelines have changed from language specific setups to a large universal project that has been launched in order to use a single set of tags, so-called Universal Dependencies. The Shared Tasks of 2006 and 2007 have been renewed and 2017[61] and 2018[62].

The techniques for dependency parsing can be divided into two categories: a) transition-based parsing and b) graph-based parsing. Multitude of parsers have emerged from both technologies: a) e.g. Malt-Parser [167] or Parsey Mc Parseface (now Syntaxnet [9]) and b) e.g. MSt Parser [153]. Bernd Bohnnet combined the strengths of both methods for German and designed the Mate Parser [18].

2.4.3 Conventional Information Extraction (IE)

IE turns unstructured information embedded in texts into structured data [110]. More precisely, it is the automatic extraction of concepts, entities and events, as well as their relations and associated attributes [243]. It consists of subtasks, i.e. entity recognition, relation extraction, event extraction (including time and date), and template filling [110]. Two major approaches exist in order to solve this this problem: rule based system and machine learning based systems.

2.4.3.1 Rule Based Information Extraction

Rule bases techniques have an earlier origin than statistical methods for extraction information. However, many reasons justify their existence still nowadays: explainable results, the lack of annotated training data, or unclear and frequently changing specifications. Rules can be adjusted faster than re-labelling the entire training data set [121].

Rule based IE requires extraction rules and an interpreter that applies the rules on texts to extract information. Rules consist of conditions and actions: A condition can be a set of constraints that must be fulfilled in order to apply the action. An action creates or modifies an annotation on text by assigning a type to a word or to any text span [121].

Listing 2 illustrates an annotation rule: The condition on the left side matches to the word "dog" and "cat". The action on the right side creates an annotation of type "Animal" to this word. Rules can be built on each other with a complex logic. They can have annotations as conditions and can derive further, high-class annotations [121].

[61] `http://universaldependencies.org/conll17/`, accessed: February 2019
[62] `http://universaldependencies.org/conll18/`, accessed: February 2019
[63] Source: `https://uima.apache.org/d/ruta-current/tools.ruta.book.html`, accessed: February 2019

```
1      // creation of an annotation
2      "dog|cat" -> Animal;
3
4      // creation of a relation
5      DECLARE Annotation EmplRelation
6         (Employee employeeRef, Employer employerRef);
7      Sentence{CONTAINS(EmploymentIndicator) -> CREATE(EmplRelation,
8         "employeeRef" = Employee, "employerRef" = Employer)};
```

Listing 2: **Annotation rules defined in UIMA Ruta.** The condition "dog|cat" applies the annotation rule "Animal". Examples are taken from UIMA documentation.[63]

Well known rule languages are UIMA Ruta [120], the common pattern specification language (CPSL) [10], Java Annotation Patterns Engine (JAPE) [45] and the annotation-based finite-state transducers [17].

2.4.3.2 Machine Learning Based Information Extraction

In the literature mainly machine learning methods are presented. The reasons are manifold: Labeled training data is available in public machine learning repositories, statistical models can deal with numerous features and they achieve better results for many tasks [106, 110].

Machine learning algorithms require annotated training data. In a first step, features are extracted from the training documents. These feature are the input for the model of the algorithm. The model is trained with the labeled examples by tuning the parameters of the model. These supervised machine learning approaches require a huge amount of training data in order to achieve high results.

In recent years, several models in machine learning approaches have been developed to extract information. Most popular methods are hidden Markov models (HMM) [76], maximum entropy Markov models [151], conditional random fields (CRF) [131] and recurrent neuronal networks, e.g. long short term memory networks (LSTMS) [105].

2.4.4 The Context of Information

The context of information is an important topic in medicine. Information embedded in a discharge letter, report of findings or clinical notes of a patient is not always a characterization of the patient itself. Many pieces of information are negated [28] (e.g. "no fever", "dizziness is denied") or they relate to other persons, such as information within the section of family history in the discharge letter (e.g. "father died due to myocardial infarction"). This problem setting in medical context can be divided in two task: The identification of negations and the identification of context of information. In both areas to major approaches exist: rule based systems and machine learning

setups. Medicine suffers from lack of public available annotated data sets, an important reason certainly is privacy protection. Therefore, the training data itself must be created manually for machine learning methods, which increases the obstacle to use them. A review on information extraction from clinical texts showed, that rule-based system were used seven times more often than machine learning approaches [71].

2.4.4.1 Negation Detection

The best known approach for identifying negated findings in discharge summaries was introduced by Chapman et al. al in 2001: the NegEx algorithm [29]. It uses a list of terms indicating negations called *triggers*, such as "no", "denied" and "ruled out". Furthermore, phrases that contain negation triggers, but do not negate the meaning of phrase are defined as "pseudo negation triggers", e.g. "no further", "not certain if" and "without further". The rule based algorithm works quite simple [29]:

Input: A sentence with indexed clinical concepts and trigger a list.

Output: An assessment, whether the concepts is affirmed or negated.

---

1. Regular expressions are built for preceding and followed negation triggers matching all tokens in a range of 5 tokens.
2. If an indexed concepts is matched by an regular expression, it is negated, otherwise affirmed.

The trigger sets have been translated in multiple languages: Swedish [208], French [49], Spanish [40, 43], Dutch [4], Swedish, French and German [30]. The German triggers have been extended and the algorithm has been adapted. It showed good results in a small evaluation with eight discharge letters (F1-score: 0.91) and 175 clinical notes (F1-score 0.96) [42]. One evaluation on negation detection in German clinical text was made by Gros and Stede with their Netopus system. It achieved good results on finding the negation triggers, but could only determine the exact scope in 54% in German medical texts [84].

Other approaches as the popular token-based algorithm NegEx exist in particular for English texts. Rule-based systems use ontologies [62] or syntactic parsing [104]. Dependency parsing was used as well to enrich the negation detection and scope determination [212]. Even some machine learning approaches were made e.g. by trying to classify a negation with a support vector machine [246]. A good overview is given by Mehrabi [155]. Although many papers show good results, Wue et al. show that the negation detection problem is not solved yet. If no in-domain development or training-data is available the algorithms perform poor [246].

2.4.4.2 Context Detection

The most known approach to determine the context of information is the "ConText" algorithm by Harkema [91]. The basic idea is similar to the NegEx algorithm of Chapman, but it makes some modifications and extensions. It determines tree properties for information and assigns a value to each attribute: negation (affirmed, negated), temporality (recent,historical, hypothetical) and the experiencer (patient, other) [91]. To determine these properties, ConText uses *trigger terms*, such as "no" and "denies" for the negation attribute, "if", "should" and "history" for the temporality attribute and terms like "family history" or "father" for the experiencer [91]. *Pseudo-triggers* that contain other trigger tokens, but do not influence the context of an information (e.g., "poor history", "by his brother") are labeled in order to be aware of them. The algorithms works as shown below [91]:

Input: A sentence with annotated clinical concepts and a trigger list.

Output: Properties (negation, temporality, experiencer) of the input concepts.

---

1. Label all trigger terms, pseudo triggers an the end of the sentence.
2. Iterate through these triggers and perform actions depending on the type.
   a) Skip pseudo trigger terms.
   b) For regular triggers: Determine the scope of trigger and assign the corresponding annotation to all clinical input concepts within the scope.

ConText was implemented in English in 2009 and partly ported to Swedish as pyConTextSwe and achieved an overall F1-Scores of 0.81 [239].

Other systems only face sub-task such as the detection of historical content.

An overview how temporal relations are processed in clinical texts is given by [46, 158, 224, 238].

Rule-based system like the temporal tagger HeidelTime extract temporal event [222]. The modular system is also available in various languages [46]. Other rule based systems exist for Swedish [237], French [88] and Portugese [226]. A Machine learning approach using a SVM achieves F1 scores between 0.2 and 0.5 for English medical texts of various domains [223].

Furthermore, systems just address the determination of the degree of certainty also called *factuality detection* ranging from definitely affirmed to uncertain and not affirmed [46]. Machine learning approaches using SVMs and CRFs have been evaluated and obtain results (F1 score) between 0.8 and 0.9 [160].

3 PaDaWaN : An Efficient CDW Architecture

The *PaDaWaN* (Patient Data Warehouse Navigator) is a tool for building a clinical data Warehouse (CDW). It has been developed at the chair of Artificial Intelligence and Applied Informatics at the University of Würzburg and has been implemented at the Würzburg University Hospital (UKW), based on their *hospital information system* (HIS). It satisfies the data protection guidelines and is approved by the Data Protection Officer of UKW. It is in productive use since five years and contains currently[1] about 1,2 million patients with 5,6 million clinical cases and more than 660 million facts of 160 thousand various clinical concepts.

The development effort is based on the work with my colleagues: Georg Fette, Max Ertl, Mathias Kaspar, Jonathan Krebs, Leon Liman, Friedrich Fell, Philip Beck, and others. Several parts of the PaDaWaN CDW are significantly improved and extended in the context of this work, including the medical query language, permission management, REST interface, data export, index pipeline, and the above briefly mentioned query process composed of query parsing and result generation. Another big point was the creation of a new, appealing user interface with many usability features leveraging the usage of ad hoc IE.

This chapter starts with the presentation of the architecture of the PaDaWaN data warehouse in Section 3.1. A quick overview is given in Section 3.1.1 followed by the description of the individual modules in detail. The second part outlines the implementation of the user interface in Section 3.2. The query surface is described especially in Section 3.2.1 as well as the attribute analyzer in Section 3.2.2.

3.1 PaDaWaN DWH Architecture

The architecture of the PaDaWaN DWH contains an *extract, transform, load* (ETL) process, a query optimized storage and data interfaces like a web application for user interactions.

3.1.1 Overview

Figure 3.1 gives an overview of the PaDaWaN architecture. The data is extracted from the HIS and exported to the ETL platform. The anonymization of the data takes place

[1] Status: October 2018

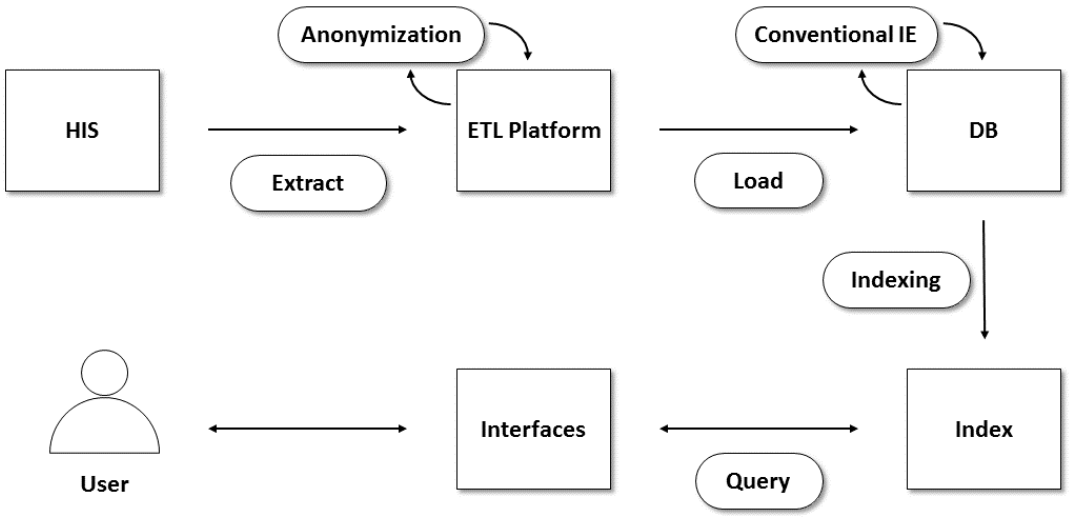


Figure 3.1: **System design PaDaWaN DWH architecture.** Data is extracted from the HIS, anonymized and transformed on the ETL platform. During the *Load* process new information is derived with the conventional IE. Patient information is indexed and can be queried via various interfaces, such as a web application for users.

during the export process. After several transformation steps, the data is imported in a database (DB). Some processes use the data from the DB and refine it or derive new information, like the conventional *information extraction* (IE) process. For performance reasons and to provide a comprehensive query functionality, the data is stored in a search engine index. Users can create queries in a web based graphical user interface. These queries are sent to REST-style interface, which in turn requests data from the index server. The returned information is summarized and various result formats are derived like diagrams, Excel or CSV.

3.1.2 Extract (Data Export)

The hospital information system consists of several sub-systems, such as a laboratory findings database, a storage of radiology reports, a sub-system for medication data and a special discharge letter database. For further information see Section 2.1. The first step of the ETL-process is the *Extract* task, which exports data from all these subsystems. This is done by several system specific scripts. In addition to patient data, meta data is exported as well, to provide the semantic knowledge of the data.

All exported data is pseudonymized with several procedures. This is described in the next Section. After that, the data is transferred to the ETL platform and stored as files. Figure 3.2 illustrates the extract process.

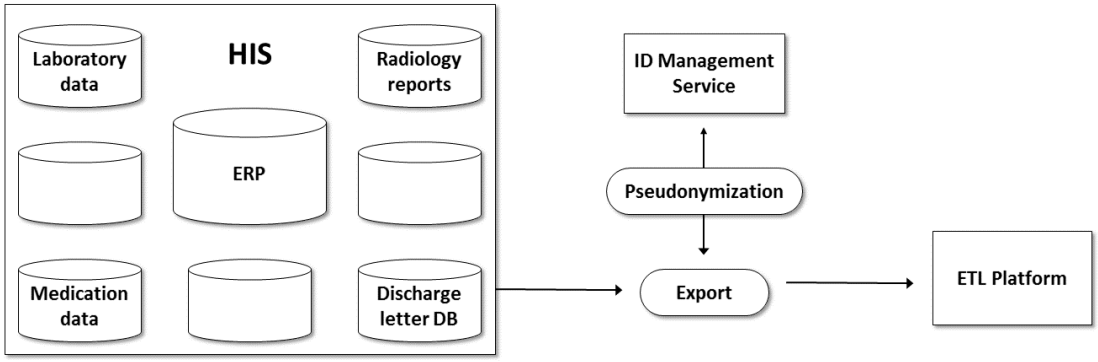


Figure 3.2: **Extract step in the ETL process.** Data of the several sub-systems of the HIS is extracted in SAP. During the export process, the data is pseudonymized by an external ID management service, before it is filed on the ETL platform.

3.1.3 De-Identification

The de-identification framework of the PaDaWaN system comprises several components to protect the privacy of patients: pseudonymization, anonymization of documents and k-anonymization of query results.

De-identification is defined by Ribaric et al. as "the process of concealing or removing personal identifiers, or replacing them with surrogate personal identifiers in multimedia content, in order to prevent the disclosure and use of data for purposes unrelated to the purpose for which the information was originally obtained" [188].

The pseudonymization and identity management system of PaDaWaN is according to the TMF (Technologie- und Methodenplattform für die vernetzte medizinische Forschung e.V.[2]) data protection guidelines [175] for clinical data warehouses.

3.1.3.1 Pseudonymization

A central point in the data protection system is the pseudonymization. Article 4 of the European Union (EU) General Data Protection Regulation (EU-GDPR) defines pseudonymisation that way [233]:

Definition 3.1.1. *Pseudonymisation* means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organizational measures to ensure that the personal data are not attributed to an identified or identifiable natural person.

[2] `http://www.tmf-ev.de/,accessedOctober2018`

This means in the context of the CDW, that each identifier that relates to a patient like a SAP universal patient identifier, is replaced with an alias. That is a generated ID that does not contain any information of a patient such as a part of his birthday. The management of these IDs is organized in an independent organization. This *identity management* service handles the translation of the different IDs with mapping tables.

3.1.3.2 Anonymization

Anonymization is the identification and the removing of data like names, addresses, and full post codes and any other information which could identify the patient [164, 192, 203]. It is a one-directional (irreversible) process [188]. Anonymization and pseudonymization are similar to each other. Both remove identifiers of persons in data. The difference is, that pseudonymization stores the mapping of the patients to the corresponding data, whereas anonymization does not keep such information.

All text documents or textual parts of a document are anonymized. This is done in two steps:

1. Heuristic identification and removing of person related information.
2. Removing of all known structured patient information.

The heuristic anonymization uses predefined patterns and rules to extract patient information. A simple pattern may be a salutation followed by a name like "Mr. | Mrs. `capitalized word`". That would match the name "Mustermann" in the example text "The patient Mr. Mustermann reports, that the pain ...". The detailed procedure is described in [125].

The second step is a rather simple keyword matching. Personal data that exists in a structured manner, such as name, address, postcode and phone numbers are searched and removed in the text.

This anonymization process is available in two versions:

- As a web service, which expects the text to be anonymized and the structured personal patient data. The anonymized text is returned.
- As a runnable library, which expects folder names as input parameters. These folders must contain the data to be anonymized and the structured personal patient data. The result is written in an output folder.

3.1.3.3 K-Anonymization

The default query mode of the PaDaWaN returns aggregated counts of patient groups. In order to prevent the identification of patients in small groups, the exact size of groups with fewer than k patients is not displayed, but censored with a wildcard. The default value for k is 10.

3.1.4 Load (Import)

The *load*-step of the ETL workflow imports data from the ETL platform into a DB. This is done via importers for each medical domain. They can be customized with a XML configuration file, which contains amongst others user credentials for the target CDW [69]. The importers can handle the common data formats (plain text, CSV and XML files). The target definition is a CDW: If that is the PaDaWaN, the data will be stored in a database: Microsoft SQL[3] and MySQL[4] are supported. An alternative target definition is the i2b2 CDW. The data will be stored in the i2b2 data schema in a Microsoft SQL DB[5].

3.1.5 Conventional Information Extraction

The conventional Information Extraction (IE) comprises two diverse topics. The first one is the terminology generation and the definition of the extraction patterns and extraction rules. The second part is the extraction application, which derives information from texts with extraction rules. Machine learning techniques can be used alternatively. A deeper insight is given in Section 2.4.3.

The terminology engineering takes place outside the pipeline as depicted in Figure 3.1. Since medical IE in German language suffers from the availability of fewer resources than IE in other languages or other domains, terminologies were built manually with domain experts [227]. A second approach is the development of a process to derive a terminology of domain specific texts [126]. A sample of clinical texts like radiology reports is reused to extract terms and to build a local terminology. A second step maps this terminology to a standard terminology or ontology. The value pattern and the extraction rules are gained in the first step as well. However, some terms have to be refined manually.

After engineering, testing and evaluating these IE methods, they can be used in the PaDaWaN pipeline. The actual extraction of the information is part of the ETL process. The input for this task are clinical texts like radiology reports, the corresponding terminology and the extraction knowledge. The texts are selected from the DB. The output is extracted information. They are added to the information of the corresponding patient case and stored in the DB.

3.1.6 Extended Data Models

The PaDaWaN CDW uses various data models depending on the task. An *Entity Attribute Value Model* is used in the DB, which serves as data sink and persistence layer. The index server has a document oriented data structures in order to provide fast response times for queries.

[3] `https://www.microsoft.com/de-de/sql-server/sql-server-downloads`, accessed: October 2018
[4] `https://www.mysql.com/`, accessed: October 2018
[5] `https://www.microsoft.com/de-de/sql-server/sql-server-downloads`, accessed: October 2018

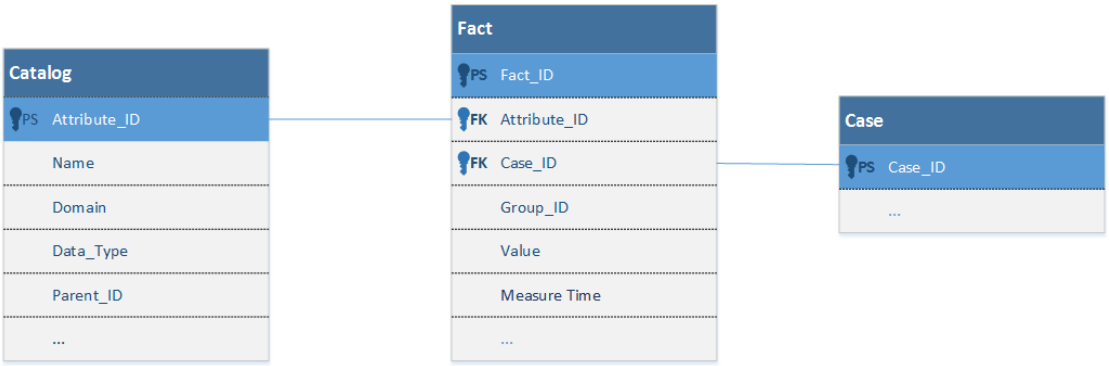


Figure 3.3: **Entity Attribute Value Model of the DB.** The catalog table contains attributes representing medical concepts. The fact table contains information of the patients.

3.1.6.1 Entity Attribute Value Model

The data model of the database of the PaDaWaN is an entity-attribute-value (EAV) model that is extended with some additional data. All attributes representing medical concepts are stored in a *catalog* table. The name, domain and the data type of the attribute are part of the table. The possible data types of an attribute are *number, date, text, Boolean* and *single choice*. Moreover, a hierarchy information is stored in a parent-child manner. Hence, the entire catalog is a terminology and can be seen as one big tree.

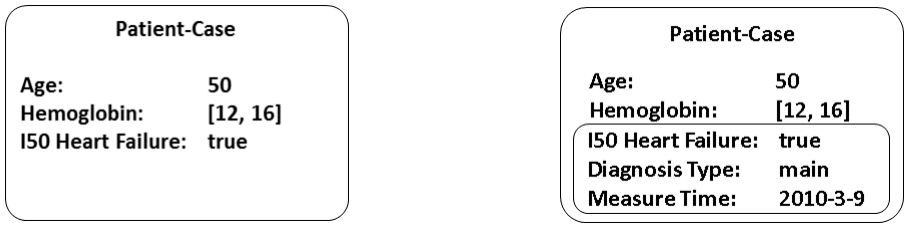
Facts are stored in the *fact* table in EAV style, whereas a patient (case) is the entity, the attribute is a reference to an attribute in the catalog table and the value is the value of the fact / observation. For example: “Patient X” (entity) has a “hemoglobin” (attribute) observation with “15 g/dl” (value) at 15.2.2018 12:15:00 (measure time). The measure time is added to the EAV model, to store a timestamp of the fact. If a patient has several facts of the same concepts, e.g. laboratory measurements, they can be chronologically ordered.

An additional *group-ID* indicates a semantic conjunction of different facts to each other. This group-ID is used in most cases to enrich existing facts with further details. The following example illustrates the use of the group-ID: A patient has several diagnosis and one of them is the discharge diagnosis. The discharge diagnosis is modeled with two pieces of information. A first fact defines the existence of a diagnosis and another fact indicates that this is the main diagnosis.

Figure 3.3 shows a section of the DB schema with the tables described above.

3.1.6.2 Document Structures

The database represents the persistence layer. The index is optimized for requests and provides extensive text query functionality. The DB is a relational database management



(a) Flat Document Structure (b) Hierarchical Document Structure

Figure 3.4: **Document structures of the index server.** A document represents a patient case. The flat structure stores the data in one field per attribute. The hierarchical structure has sub documents, which can contain multiple facts that are semantically related with each other.

system. In contrast to a search engine index, which is a document store. It can also be seen as a document-oriented database. All information of one patient case is stored in one document. That enables fast filter operation on multiple attributes. Two different data models can be used in the PaDaWaN . The first one is a flat data structure (see Figure 3.4a) and the second one is a hierarchical model (see Figure 3.4b).

Flat Structure. In the flat data structure, each attribute is represented by a *field*. A field is a storage of some information, it accepts various data types and contains a single value or a list of values [75]. If two facts have the same attribute, both are stored in the same field, like hemoglobin in Figure 3.4a. That data schema is simple, but saves storage and it can answer queries very fast.

Hierarchical Structure. The Hierarchical Structure allows in addition to the flat structure nested child documents. This has the advantage that information can be grouped together. In the example of Figure 3.4b is the diagnosis "I50 Heart Failure" grouped together with the fact "Diagnosis type: main". Additional information, such as details or modifiers, can be stored with this technique.

3.1.7 Indexing Process

The indexing process pulls the data of patient cases from the DB and stores it in a document oriented style in an index server. As mentioned above, this has two reasons: The query performance is improved and the query features are extended.

Figure 3.5 illustrates the index pipeline. For each patient case, all facts that belong to the case are selected from the database. They are passed to a *document creator*. In the first step, it prepares the document structure (flat or hierarchical). If a hierarchical structure is chosen, sub documents are created for grouped information.

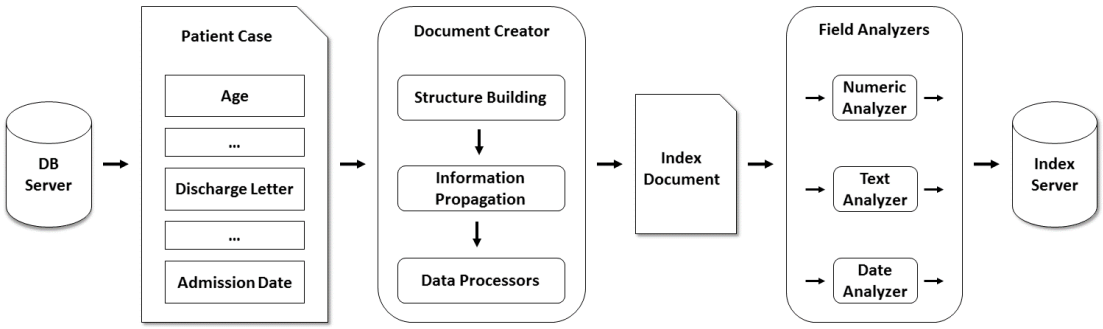


Figure 3.5: **Indexing Process of the PaDaWaN CDW.** The data for each patient is pulled from the DB. A document creator transforms this data in various steps into a index document. Its fields are analyzed before the document is stored in the index server.

A patient case consists of facts, which in turn are a tuple of an attribute and a value. Attributes are organized in the catalog, which has a tree structure. This hierarchy information is exploited in the next step. Facts are *propagated up* in the catalog tree. This means, that some pieces of information are passed through the tree in direction of the root node. For every successor attribute in the tree, a fact is created with the corresponding attribute. The value depends on the data type. In most cases, the name of the original attributes is stored. This has two benefits: (1) Queries work hierarchically. They also take information into account that is at a lower hierarchical level than the query attribute. (2) As a result value, not only the existence of an attribute can be returned, but the exact value of the attribute can be used and displayed in the result table. For example, if the diagnose “I50 heart failure” is queried, the more precise diagnosis “I50.813 Acute or chronic right heart failure” is found and returned. Figure 3.12c gives an example of such a use case, it shows the result of a high level diagnosis query.

This method was chosen in order to achieve a query speed optimization. An alternative way would be a query expansion, which would increase query processing time. However, the PaDaWaN system focuses on fast response times. The additional created data does not cause much extra disk space because index libraries have sophisticated storage mechanisms.

At the end of the document creator process, data processors analyze the data based on their types. For example, a numeric analyzer tags the first, last, minimum, and maximum value when a list of values is passed to it, like multiple measurements of laboratory values.

The created index document is passed to a data import handler of the index server. User defined field analyzers process the given data according to a predefined schema. At the end, the data is stored in an index.

The data source for the indexing process can be a PaDaWaN DB or an i2b2 DB.

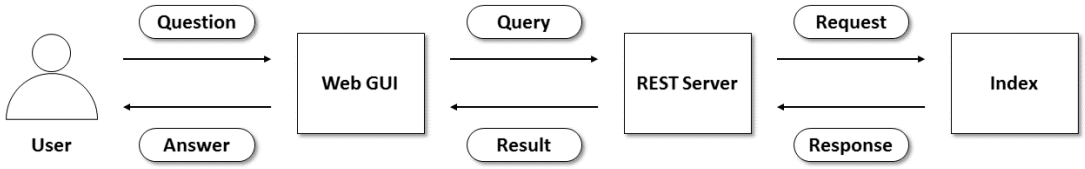


Figure 3.6: **Design of the query process in the PaDaWaN.** A user models his question into a query in the web application, which sends it to a REST server. The requested index server returns responses, which are transformed to a user result. That can be exported or used as input for further investigations.

3.1.8 Query Types

PaDaWaN offers two different query types: a *statistic query* and a *patient case query*.

Statistic Query. In many cases, this query mode is the entry point and first step for a detailed study. A statistic query returns aggregated results and can be used by many users, because it returns only anonymized data. (See Section 3.1.3 for further information.) The first column and the head row define constraints. Each cell in the table indicates the number of patients that meet the two conditions defined at the beginning of the row and the column. Filters constraining the result set can be defined additionally.

Patient Case Query. The patient case query shows the individual values of patients. Attributes can be defined in the query and they are mapped to columns in the result table. Each row of the result represents a patient and displays the values of the attributes. This query type can be used for a detailed analysis of patients. Use cases for that query mode are close looks at certain issues, refinements of queries or the acquisition of patients for a recruitment list for a medical trial. As explained in previous sections, the text data, like discharge letters, is pseudonymized. Every name, address or any other patient related information has been removed. In user-permission-management, it can be defined which user is allowed to use this query mode. For example, an approval of the data protection officer could be necessary for the permission.

3.1.9 Query Process

In the web based graphical user interface, users can create queries in an intuitive manner. The web application sends the defined query, represented in JSON, to a REST-server. This server receives PaDaWaN queries and translates them to index server queries. User results are build based on the response of the index server.

3.1.9.1 Query Parsing

The query parsing depends on the type of query.

Statistic Query. A statistic query is transformed in many subqueries for every result cell. These subqueries have the structure of a patient case query. They are further translated into index server requests like patient case queries. But in contrast to them, only the result count of patient cases is requested not the patient data itself. The subqueries are executed independently and the results are put together into a table.

Patient Case Query. The query parser analyzes the logical structure of a query and its attributes with their operators and arguments. Arguments are processed based on their data types. The output of a so called *data analyzer* is a valid index query part. These parts are combined preserving the query logic. The final query is sent to the index server, which selects the desired information and returns a response.

3.1.9.2 Result Creation

The result creation depends on the type of query as well.

Statistic Query. The responses of the index server to the subqueries of a statistic query are counts of patient case groups. They are put together into a result table. In order to prevent the identification of patients in small groups, the exact size of small groups may not be displayed. For this use case, a parameter can be configured. It ensures that the count of a group is not displayed if its size is less than k. This mechanism implements the concept of k-anonymity.

Patient Case Query. The result of a *patient case query* is not the count of a patient group, it is a list of patient information. After the request has been executed, the results lines will be streamed back. The table is generated of these result values with a result creator. A data type specific processor creates a user result according to the query attribute. This step includes filtering of optional query attributes, formatting and a possible discretization or generalization of values according to the data protection requirements.

3.1.10 Query Language

PaDaWaN queries are defined in a self-defined schema named *MXQL* (Medical XML Query Language). It defines (1) the patients cases being selected and (2) and the data that is returned.

3.1.10.1 Logical Structure

A query consists of structure elements and attributes. An attribute represents a medical concept that can be used as a filter. The available expressions for defining filter constraints

```
1  <Query limitResult="10" version="0.8">
2    <IDFilter filterIDType="CaseID">
3      <Attribute name="Alter" operator="LESS" argument="50" domain="alter" id="alter"/>
4      <Attribute name="Medizinische Klinik" domain="station" id="med1"/>
5      <Attribute name="Gestorben" operator="NOT_EXISTS" domain="status" id="gestorben" />
6      <Or>
7        <Attribute name="I50: Herzinsuffizienz" domain="diagnose" id="i50" />
8        <And>
9          <Attribute name="Arztbrief" operator="CONTAINS" argument="Herzschwäche"
           ↪  domain="arztbriefe" id="brieftext"/>
10         <Attribute name="LVEF" operator="LESS_OR_EQUAL" argument="50"
           ↪  filterUnknown="true" reductionOp="MAX" domain="labor" id="i11" />
11       </And>
12       <IDFilter filterIDType="GROUP">
13         <Attribute name="I11: Hypertensive Herzkrankheit" domain="diagnose" id="i11" />
14         <Attribute name="Entlass-Diagnose" domain="diagnose" id="entlaass" />
15       </IDFilter>
16     </Or>
17   </IDFilter>
18 </Query>
```

Listing 3: **Medical XML Query Language Example (MXQL).** Logical structure elements like `IDFilter`, `And` and `Or` define the composition of the `Attributes` that define the constraints of the query.

is described in detail in the next section 3.1.10.2. Attributes are combined with structure elements to create a query.

Listing 3 exemplifies the usage of this structure. The `Query` element defines a PaDaWaN query in the MXQL syntax. The `IDFilter` element in line 2 specifies with the argument "CaseID" that the query and especially the result values refer to patient cases. They are the basic reference set of a query. The child elements of `IDFilter` define filters. Each element is a *must-have* condition. Line 3 states, that "Alter" (engl. *age*) must be less than 50. Line 4 defines a further, additional constraint: The patient must be in the medical department. Line 5 describes that an entry "Gestorben" (engl. *died*) must not exist.

The elements `And` and `Or` are further important structure components. An `Or` element requires at least one child element to satisfy its condition. The `Or` element in line 6 has three children: an attribute (line 7), an `And` list (line 8-11) and an `IDFilter` (line 12-15). Unlike an `Or` list, every child of an `And` is a *must-have* constraint.

The `IDFilter` in line 12 with the attribute `filterIDType="GROUP"` defines that its children must occur in the same group. Facts in the same group have a semantic cohesion. (See Section 3.1.6.1.) In this example, the diagnosis ""I11: Hypertensive Herzkrankheit" (engl. *Hypertensive heart disease*) must have the additional tag "Entlass-Diagnose" (engl. *discharge diagnosis*).

This feature is only available if the hierarchical data structure is used (see Section 3.1.6.2).

```
1  <Query version="0.8">
2    <IDFilter filterIDType="CaseID">
3      <DistributionRow>
4        <Attribute name="I50: Herzinsuffizienz" domain="untersuchung" id="i50"/>
5        <Attribute name="I20-I25: Ischämische Herzkrankheite" argument="Nachfolger"
         ↪  domain="untersuchung" id="i20_i25"/>
6      </DistributionRow>
7      <DistributionColumn>
8        <Attribute name="Alter" operator="INTERVALS" argument="0;30;45;60;75;100"
         ↪  domain="alter" id="alter"/>
9      </DistributionColumn>
10     <DistributionFilter>
11       <Attribute name="Aufnahme" operator="MORE" argument="01.01.2010"
         ↪  domain="stammdaten" id="aufnahme"/>
12       <SubQuery name="Filter-Diagnosen" queryID="324"/>
13     </DistributionFilter>
14   </IDFilter>
15 </Query>
```

Listing 4: **Statistic Query in MXQL.** The elements `DistributionRow`, `DistributionColumn` and `DistributionFilter` contain the attributes or structure elements for the corresponding query part.

Statistic Queries. Statistic queries (see Section 3.1.8) can also be defined in MXQL. Listing 4 pictures a simple example. The elements `DistributionRow`, `DistributionColumn` and `DistributionFilter` represent the row, column and filter parts of the query. Attributes within the `DistributionRow` element correspond to a row in the result table like "I50: Herzinsuffizien" (engl. *heart failure*) in line 4. In order to make user input more comfortable and to keep the query compact, special constructs can be used as attribute definitions. One of these can bee seen in line 5: "I20-I25: Ischämische Herzkrankheiten" (engl. *Ischaemic heart diseases*) has the argument "Nachfolger"' (engl. *successors*). The successors of that attribute are the child attributes in the catalog: "I20 Angina pectoris", "I21 Acute myocardial infarction", ..., "I25 Chronic ischaemic heart disease". Line 5 is transformed to a list of these attributes. Another example of such a compressed notation is line 8. Intervals are defined for the attribute "Alter" (engl. *age*). The argument contains the boarders for the various intervals. This line is transformed in 5 lines with the content: age in range 0 to 30, age in range 31 to 45, etc.

The filter section contains a list of filters that constrain the result set. A filter is an attribute or a structure element like an `And` list, an `Or` list or an `IDFilter`. A further query element is a *subquery*, which references a stored query, like line 12. All queries can be stored in the PaDaWaN system and referenced in another query as a sub-query. It will be replaced with the original query at execution time. That allows an iterative refinement of complex queries.

Logical query parts like `And`-lists, `Or`-lists or sub queries can be nested arbitrarily deep.

Table 3.1: **Attribute operators for query attributes.** Query operators are provided for different data types. `PER_YEAR`, `PER_MONTH` and `PER_INTERVAL` are operators for statistical queries.

	Boolean	Numeric	Date	Text
EXISTS	✓	✓	✓	✓
NOT_EXISTS	✓	✓	✓	✓
EQUALS		✓	✓	✓
LESS		✓	✓	
LESS_OR_EQUAL		✓	✓	
MORE		✓	✓	
MORE_OR_EQUAL		✓	✓	
BETWEEN		✓	✓	
CONTAINS				✓
CONTAINS_NOT				✓
PER_YEAR			✓	
PER_MONTH			✓	
PER_INTERVAL		✓	✓	

3.1.10.2 Operators for Attributes

Various operators are applicable depending on the data type of the attribute. Table 3.1 lists the existing operators and their availability to the data types.

The following description explains the operators.

EXIST The patient case must contain a fact with this attribute or any attribute on a hierarchical lower lever.

NOT_EXIST The patient case must not contain a fact with this attribute or any attribute on a hierarchical lower lever.

EQUALS The numeric, date or text value of the attribute must be identical with the given argument.

LESS The numeric value must be lower than the given argument, the date value must be earlier than the given argument.

LESS_OR_EQUAL The values must be lower / earlier as or equal to the argument.

MORE The numeric value must be higher than the given argument, the date value must be later than the given argument.

MORE_OR_EQUAL The values must be higher / later as or equal to the argument.

BETWEEN The argument contains two boundaries. The numeric or date value must be between this boundaries.

CONTAINS The value of the attribute must contain the token or the tokens of the given argument.

`CONTAINS_NOT` The value of the attribute must not contain the token or the tokens of the given argument.

`PER_YEAR` This is a generator function for other attributes in statistic query mode. The arguments are two numbers indicating two years. The attribute is transformed in to several attributes, each covering a period of one year of the given interval.

`PER_MONTH` This is also a generator function in the statistic query mode. The argument is one number indicating a year. The attribute is transformed to several attributes, each covering a period of one month of the given year.

`PER_INTERVAL` This is also a proxy for other attributes. The arguments are various numbers indication boundaries of intervals. The attribute is transformed in to several attributes, each covering one interval of the given boundaries.

3.1.10.3 Other Features

The PaDaWaN query language contains several further features. Some of them appear in the Listings 3 and 4 and are explained here:

Filter unknown. Not every patient case contains all attributes. Line 10 in Listing 3 filters patient cases based on their "LVEF" value. Cases that have a value that is less or equals than 50 pass the filter, cases with a higher value are rejected. But what about the cases, which do not have an "LVEF" value. Per default they are excluded as well, but this setting can be defined explicitly with the parameter `filterUnknown`.

Reduction operator. An attribute can have multiple values. For example, when a laboratory value is measured several times during a hospital stay. The reduction operator can be used to define which one should be selected. These options are possible: *min*, *max*, *earliest* and *latest*. If none is defined, all values are used to check if the constraint is satisfied. At least one value must match the filter.

The `limitResult` tag in line 1 in Listing 3 defines the amount of rows that are displayed as preview in the GUI. Figure 3.12c shows an example of the result preview.

3.1.11 Permissions Management

User permissions in the PaDaWaN are administrated with groups. Every user is in any number of groups, but a least in one. The attributes in the catalog and the data in the form of patient cases can be managed with rights management. The management of the catalog is governed by black & white listing. Catalog entries that should or should not be visible to certain users are specified in a white or black list. The inclusion or exclusion rule applies to this catalog entry and all entries that are on a hierarchical lower level. All entries which are hierarchically higher than any listed catalog entry are also visible to users.

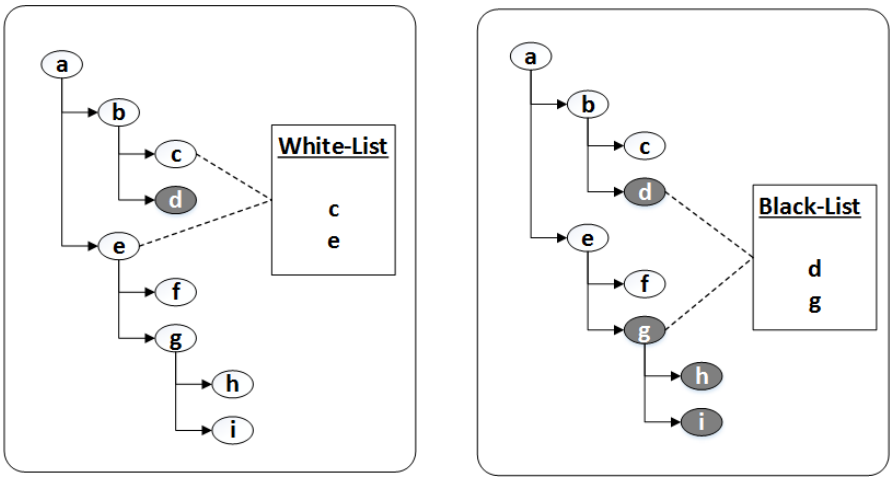


Figure 3.7: **Permission management concept of the catalog.** Examples for black and white listing for the hierarchical structure.

Figure 3.7 shows an example of the permission management. The left picture is an example for white listing: entry “c” and “e” are included in the list. That means, that “c” and “e” are visible as well as all sub entries. Hierarchically higher entries are visible per default, which only leaves out “d” as not visible entry. The black list on the right side includes “d” and “g”. These two entries and their sub entries are not visible.

In addition to catalog entries, the data itself can be administrated as well: Patient cases can be organized with the same rights management as catalog entries. A white-list or a black-list can be used to grand or deny the access of patient cases to user groups.

3.1.12 Interfaces

The PaDaWaN system offers several interfaces for data exchange and user interactions.

GUI. The graphical user interface is used to create and execute queries for any user [52]. Results are presented in various ways and can be exported in some data formats. This is explained in detail in Section 3.2.

REST. Representation State Transfer (REST) is a modern programming paradigm and architecture for web services that is mainly used for a machine to machine (M2M) communication and aims to ensure interoperability [37].

The REST interface is the main interface of all services that communicate with the PaDaWaN server outside of a secure environment, like the web GUI, which runs in the client browser.

The following resources can be accessed: catalog attributes, users, groups, facts and cases. Data can be read, modified, inserted and deleted via the HTTP request methods *GET*, *POST*, *PUT* and *DELETE*.

Data Import. Data can be imported in different ways. The first way is via the REST interface. Second, there are importers that can read *PMDs* (parametrierbare medizinische Dokumentation, engl. *parameterizable medical documentation*) and CSV (Comma-separated values) files. CSV importer can be configured and they accept two data formats: (1) entity attribute value (EAV) oriented documents and (2) patient attribute tables.

Moreover, data can also be imported from other databases.

Data Export. Data can also be exported in multiple ways. The REST interface offers powerful options to export data selectively or extensively, by accessing single facts or the entire list with all facts of patients (see Section 3.1.12.) Furthermore, exporter can write data to files in the two formats: CSV files or patient attribute tables. The entire catalog table can be transformed to an Excel file.

Moreover, the result tables of the query process can be exported as CSV, Excel or JSON.

3.2 Implementation: PaDaWaN User Interface

The PaDaWaN CDW has a web application as graphical user interface. It is written in the JavaScript framework qooxdoo[6]. If offers a compact surface for queries to the data warehouse. It's *easy-to-use* layout allows users, e.g. physicians, the creation and execution of queries to gather information. The main component is the query surface where users can model their questions into queries. The result can be viewed in this surface as well. An attribute analyzer shows characteristics like key values and diagrams (e.g. histograms, time bars etc.) of catalog entries, representing medical concepts. Configuration parameters can be defined in an administration interface.

3.2.1 Query Surface

The query surface is divided into three parts: A catalog viewer lists all attributes that are contained in the CDW. These entries, which correspond to medical concepts, can be put together to a query. That can be modeled in a query view. After the execution of a query, the results are presented in a special view. Figure 3.8 shows the query surface of the PaDaWaN .

[6] `https://www.qooxdoo.org/`, accessed: October 2018

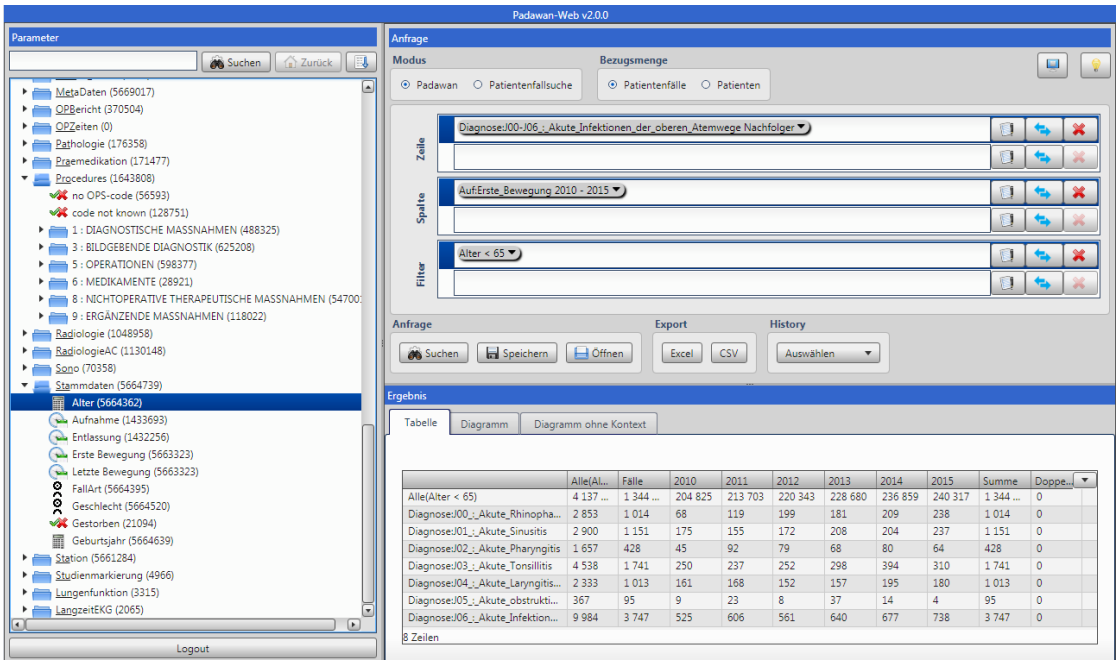


Figure 3.8: **Main query surface of the PaDaWaN.** On the left is the catalog view. Top right is the query view. Results are presented below.

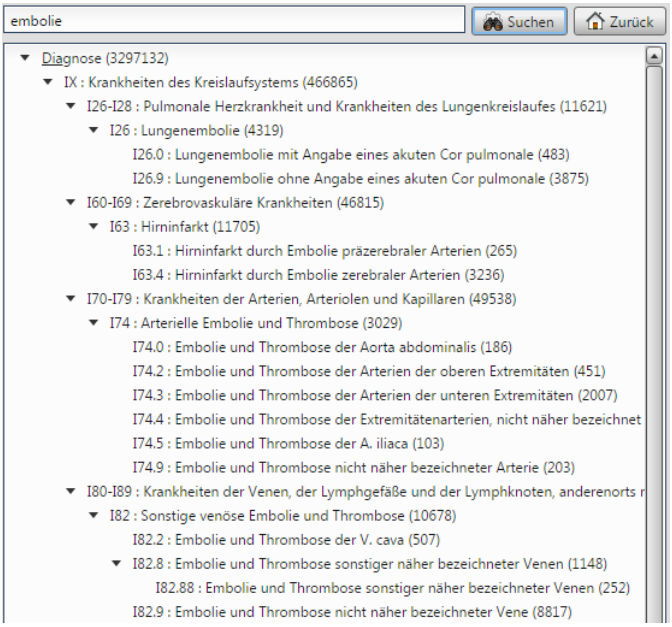


Figure 3.9: **Catalog view of the query surface.** The catalog view shows diagnoses structured by the ICD10 catalog, filtered by “Embolie”.

3.2.1.1 Catalog View

The catalog view contains all medical concepts, which are included in the CDW. A concept is represented with an attribute. These attributes are grouped in various domains such as: Diagnoses, core data, departments, operation codes and laboratory findings. The attributes within a domain are hierarchically structured. Examples for attributes may be: patient age, discharge letter, haemoglobin, stay at the department of internal medicine, etc. Figure 3.9 shows a sub-section of the diagnoses catalog. Diagnoses are structured using the ICD10 catalog. A catalog attribute is represented with its name and a number corresponding to its occurrences in patient cases the data warehouse. The catalog of the Würzburg University Hospital consists of 160.000 attributes[7]. It can be filtered and queried by keywords to ensure a quick access. Figure 3.9 shows an example with the query term "Embolie".

3.2.1.2 Query Definition View

The PaDaWaN CDW has two query modes with two different surfaces: The *statistic mode* and the individual *case query mode*.

Statistic Query. The statistic mode returns aggregated results in a table and works like explained above. One query contains a lot of sub-queries. More precisely, each cell of the result table is a separate query. Therefore, one query provides the counts of many patient groups. The statistic mode is very suitable for time series analyses or a search for subgroups. A lot of information can be gained with a single query.

The query definition view with the statistic query mode is visible in Figure 3.8 on the top right. A query definition view of the *statistic mode* describes a result table: Every row and every column are defined in a query. The query definition contains a section for rows and for columns. All desired columns/rows are defined in their corresponding part. Every row or column is represented with a line. Filters can be defined in addition constraining the result set. Examples of filters are "age > 80" or "sex = male".

Medical concepts / attributes are selected in the catalog view and transferred to the query view into the desired part (row, column filter). There, constraints (filters) can be created based on the attributes by defining operators and values. For example, the attribute "haemoglobin" can be assigned with operator "$\geq$" and the value "16". Query attributes can be combined wit logical operators (`AND` and `OR`) and they can be grouped with braces.

Patient Case Query. The second function of the query tool is the individual *case query mode*. Figure 3.10 shows an example. In contrast to the *statistic mode*, which only returns aggregated results, the *case query mode* shows the individual values of patients.

[7]Status: 2018

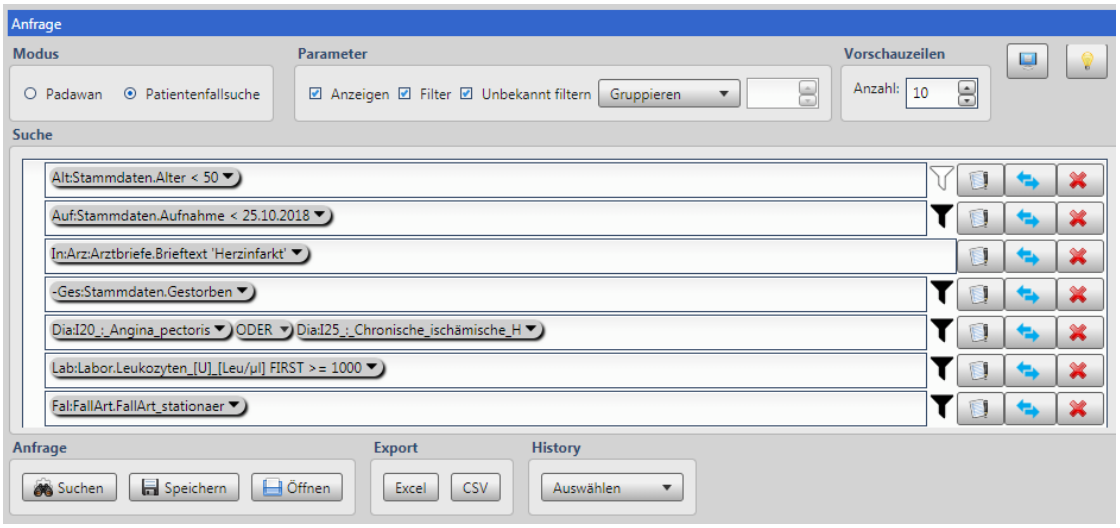
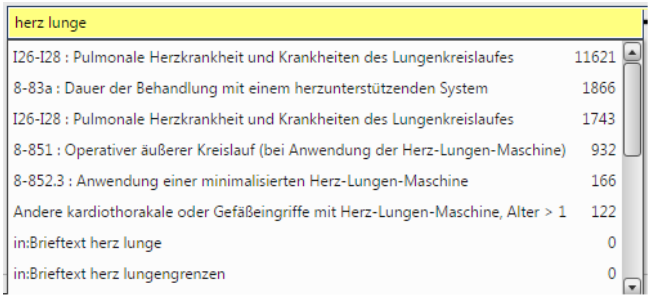
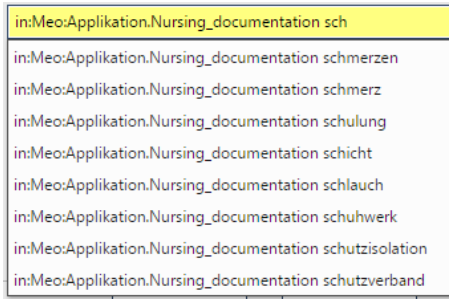


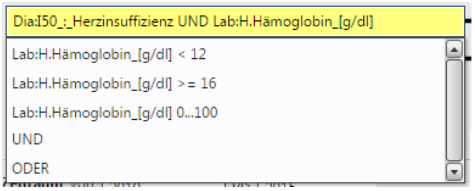
Figure 3.10: **Query creation definition view of the patient case mode.** Lines with a black filter symbol at the end of the line serve as filters, white symbols do not filter unknown values. Attributes without filter symbol do not constraint the result set, but their values will be included in the result table. "Alter", "Aufnahme" have numerical or temporal constraints. "Arztbrief" must contain the term "Herzinfarkt". "Gestorben" is negated and must not occur. The diagnoses "I20" and "I25" are *OR*ed: A patient case must contain at least one of them.



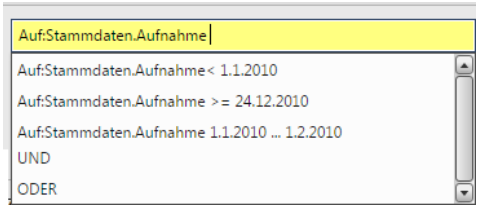
(a) Boosted attribute suggestion



(b) Attr. specific content suggestion



(c) Query syntax suggestion



(d) Query syntax suggestion

Figure 3.11: **Suggestion and auto completion features of the PaDaWaN.** The attribute suggestion return entries matching the keywords, the ranking is boosted by the occurrence count in the CDW. The query syntax structures are proposed as well as attribute specific content query tokens.

All attributes, defined in the query, refer to a column in the result table. Each row represents a patient case and displays the values of the corresponding attributes. This mode is used for a detailed analysis of an issue or as patient recruitment list for a medical trial. Since this mode shows (pseudonymized) data of individual patients, the *patient case query mode* is only available to people with a data protection clearance.

Support features. The process of creating a query is strongly supported by several usability features like suggestions and auto completion tools. A boosted attribute suggestion feature ranks attributes by relevance based on token match and occurrences count. Figure 3.11a shows the suggestion of attributes for the input "herz lunge" (engl. *heart lung*). A query syntax suggestion recommends syntax-keywords like operators. The arguments (boundaries or search terms of text queries) for operators are suggested as well: Attribute specific values are recommended like critical limits or thresholds for laboratory values. Figure 3.11c shows the suggestion of numeric operators and the limits for abnormal values for "haemoglobin" as query example. Figure 3.11d is similar, but shows the treatment for attributes with date values. Each textual attributes has its own suggestion component recommending words of the attribute context. Figure 3.11b gives an example of that feature for the attribute "nursing documentation": The auto-completion recommends words that start with "sch" and are contained in the texts of the attribute "nursing documentation".

3.2.1.3 Result View

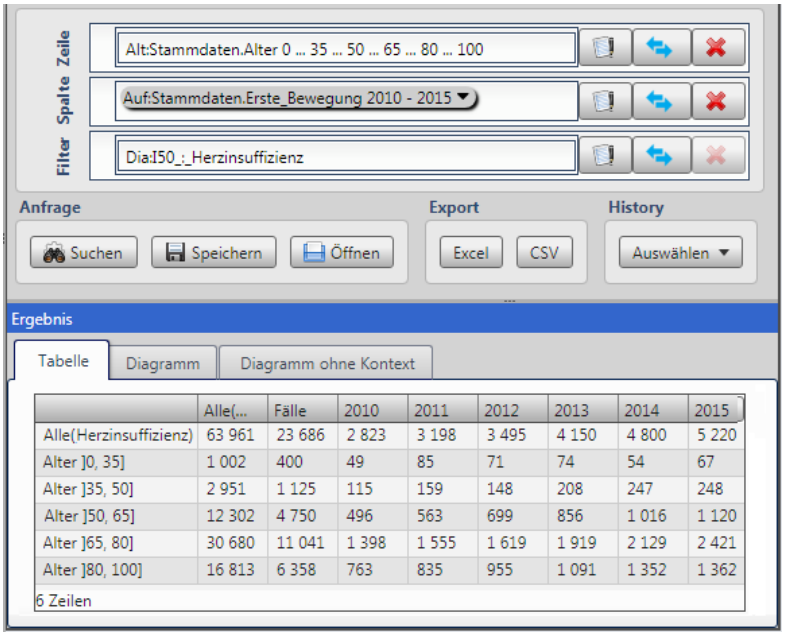
The result representation in the PaDaWaN contains several components: The results of the *statistic mode* queries are presented in a table or as a diagram representation in a bar chart. Figure 3.12a shows a query and the corresponding result table, Figure 3.12b visualizes the table as a bar chart.

The result of a case query contains two components: First, the number of hits is presented. That is the number of patient cases in the CDW that match the query. Second, the personalized values of the patients are listed in tabular style as well. Figure 3.12c is an example of the result of a patient case query. The first column queries the diagnosis "I50 Herzinsuffizienz" (engl. *heart failure*). The result rows (each representing a patient case) list the exact diagnosis like "I50.11 Linksherzinsuffizienz: Ohne Beschwerden" (engl. *Left ventricular failure: Without complaints*). The web application shows a preview of the result set, since the entire result set can comprise thousands of patient cases. These values can be exported in Excel or CSV format for further processing.

3.2.2 Attribute Analyzer

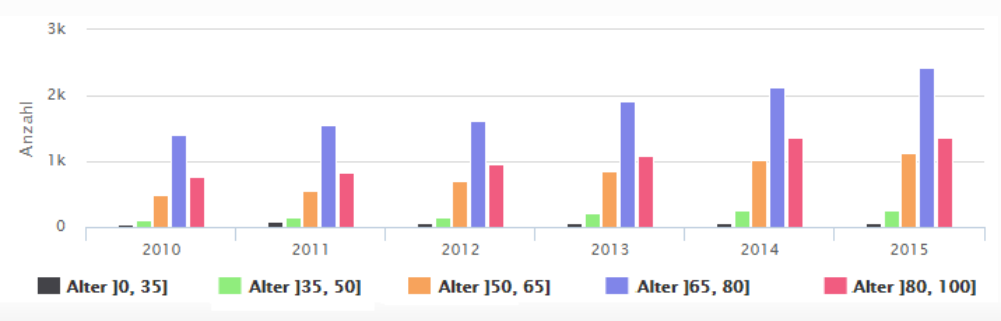
The catalog in a CDW can contain a lot of different attributes, currently[8] about 160 thousand entries. But sometimes, the context of an attribute or differences between

[8]Status: October 2018



	Alle(...	Fälle	2010	2011	2012	2013	2014	2015
Alle(Herzinsuffizienz)	63 961	23 686	2 823	3 198	3 495	4 150	4 800	5 220
Alter]0, 35]	1 002	400	49	85	71	74	54	67
Alter]35, 50]	2 951	1 125	115	159	148	208	247	248
Alter]50, 65]	12 302	4 750	496	563	699	856	1 016	1 120
Alter]65, 80]	30 680	11 041	1 398	1 555	1 619	1 919	2 129	2 421
Alter]80, 100]	16 813	6 358	763	835	955	1 091	1 352	1 362

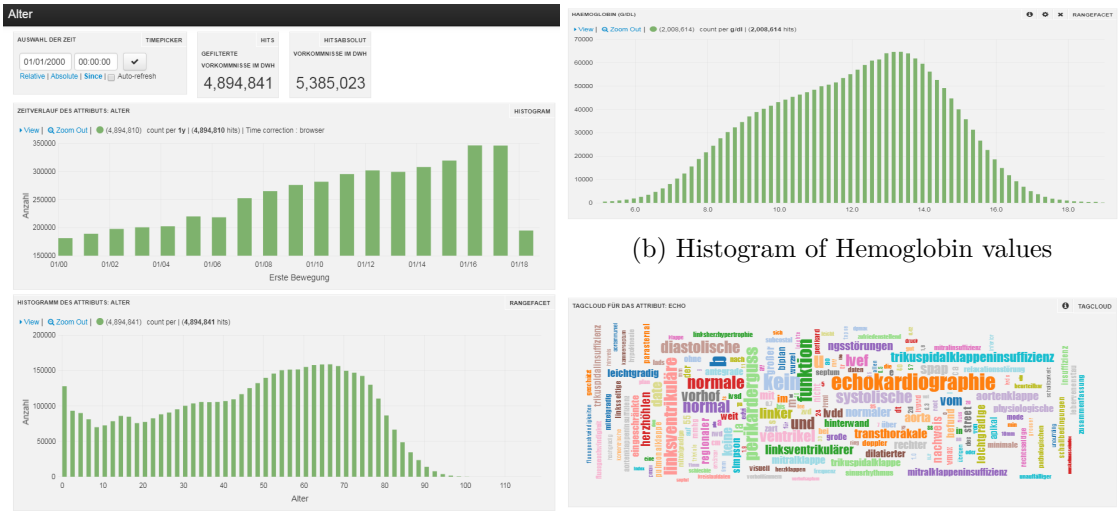
(a) Query view with corresponding result table.



(b) Result of query (a) represented as bar chart.

(c) Result table of an individual case query. (Corresponding query is not shown.)

Figure 3.12: **Result representations in the PaDaWaN GUI.**



(a) Analyzer page for the concept “age”.

(b) Histogram of Hemoglobin values

(c) Tag cloud for echocardiography reports.

Figure 3.13: **Attribute analyzer of the PaDaWaN web GUI.** Useful information can be gained with the attribute analyzer for every attribute in the catalog.

attributes are not clear for everyone. Attributes, representing medical concepts, can be very similar in different clinical departments. Semantically equal attributes are mapped to one attribute during the ETL process. But if they are similar but not equal, they remain as two different attributes in the catalog. Now it is very important to have a good description of these entries to be able to distinguish them and to choose the relevant one. There are several other issues that require good meta information of attributes: Newly introduced attributes need temporal information (introduction date, occurrence, etc.). At what time are the first facts/occurrences documented? Is it used often? How many occurrences exist per day/week/month? Numerical attributes (e.g. laboratory values) need unit information. What is the numerical range? What is the distribution of the values?

To address these issues we implemented an *Attribute Analyzer*. Each attribute in the catalog can be selected. Besides relative and absolute counts of occurrences in the CDW a time-line is given, showing these occurrences over time. It is possible to zoom in and out or to restrict the data to a special time window. Depending on the data type of the attribute, different diagrams are shown. Figure 3.13a presents the front page of the attribute analyzer for the concept “age”. Figure 3.13b shows a histogram of hemoglobin values. Figure 3.13c illustrates a tag cloud for echocardiography reports.

3.2.3 Admin Interface

The PaDaWaN web GUI has an admin interface with several functions:

- Permissions management: This covers the management (creation, modification and deletion) of users and groups and the assignment of users into a group.
- Catalog functions: The catalog can be imported and exported. This allows a comfortable modification.
- Indexing functions: Indexing processes can be started in the admin GUI such as the re-indexation of the catalog or patient data.

4 Methods for Ad Hoc Information Extraction[1]

Common use cases for Clinical Data Warehouses (CDW) are to query frequencies of patients with certain inclusion and exclusion criteria, e.g. for assessing whether there are enough patients available for a clinical trial. If a major part of the required data is not available as structured data but only included in textual reports, such assessments are quite time-consuming by manually checking many text documents. Assistants, like study nurses, have to look at all relevant documents, such as discharge letters or findings reports, and decide whether the patient fulfills the given criteria. This has to be done patient by patient. Recently, intelligent systems have been developed that can support or even solve this problem. One popular method is to preprocess the textual data within the *extract, transform, load (ETL)* process transferring data from Electronic Health Records (EHR) into the CDW with information extraction (IE) methods. Various approaches to extract structured information from unstructured texts exist (e.g. for German texts [127, 218, 227]), but they require computational expensive preprocessing in the integration step and cannot be applied dynamically at query time. Furthermore, a lot of time has to be spend on engineering and building ontologies. For more information please see Chapter 2 Section 2.4.3.

Another approach is to retrieve the information dynamically at runtime. However, most CDWs do not support textual queries very well [55] (see Section 2.3.1.3).

Definition 4.0.1. Ad hoc IE means the technical concept of extracting the existence of any concept (e.g. chronic kidney disease) or any number (e.g. the LVEF value) from a source in real-time, thus allowing the application of the usual query operations (e.g. counting the number of patient cases with "LVEF" < 45) on the extracted concepts.

The Boolean ad hoc IE queries the existence (yes/no) of a medical concept, which is a named entity that may have a feature/property. Examples of Boolean concepts are single findings or assessments (e.g. moderate mitral insufficiency, severe aortic stenosis), drugs (e.g. Aspirin, beta blocker) or diagnoses (e.g. appendicitis, myocardial infarction). A numerical concept is defined as a named entity with a numeric value. For example, this could be the value of a laboratory finding (e.g. cholesterol, glucose, LEVF) or a derived value/index (e.g. BMI, age). Numeric IE extracts the value as a number of a numerical

[1] This chapter is based on previously published work [55]: G. Dietrich, J. Krebs, G. Fette, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Ad hoc information extraction for clinical data warehouses. *Methods of information in medicine*, 57(01):e22–e29, 2018

concept. A numerical condition can be defined optionally, like "LVEF" < 45, matching all mentions of LVEF with a value lower than 45. In some medical reports on diagnostic findings, the exact value of a concept is not given, but there is a formulation indicating an interval or an inequality of a value (e.g. "LVEF lower than 45"). These statements can be queried in conjunction with numeric ad hoc IE exploiting both qualitative and quantitative information from textual reports e.g. for checking inclusion or exclusion criteria of studies. In addition to counting queries, which only asses the presence of a concept or the validity of constraints (e.g. BMI > 25), the actual values can also be returned for further processing.

This chapter is organized as follows: After a short presentation of the objectives in Section 4.1, the data structures used for ad hoc IE are outlined in Section 4.2. Afterwards, the algorithms is described in Section 4.3. Section 4.3.1 begins with the system design showing how the individual modules work together. First, a medical text has to be fragmented. The used methods are explained in Section 4.3.2. Then, Sections 4.3.3 and 4.3.4 discuss in detail how to identify the context of information in a text and how to exclude misleading information. The text query features that help to extract information ad hoc are described in Section 4.3.5.

4.1 Objectives

The goal is to develop a pipeline for ad hoc IE being able to reliably count Boolean and constrained numerical values in clinical textual documents. Many findings in clinical texts are negated, relate to other persons, or have a temporal context. That situation leads to three subtasks for ad hoc IE:

1. Determining the context of information: recognizing and excluding negations, temporal context and their scopes as well as information of other persons in text documents.

2. Extracting Boolean concepts (e.g. moderate mitral insufficiency).

3. Extracting numeric values of concepts fulfilling constraints (e.g. LVEF < 45) with context sensitive search queries.

In addition to counting the occurrences, the actual values should also be extracted and processed further. In particular, they should be made available for application specific tasks. For example, if a task requires the extraction of drugs, an extension of the framework to extract the daily dose taken by the patient should be possible as well.

This ad hoc IE shall not be considered as a replacement for conventional IE, but rather as a supplement allowing quick shallow data aggregation to potentially answer any question in a first approximation without the complex pre-defined specifications required for standard IE.

4.2 Data Structures for Ad Hoc Information Extraction

Ad hoc IE allows a quick extraction of information with rich query and extraction features. To achieve that, the data has to be transformed and stored into a special data structure. For example, medical text documents, such as discharge letters, must be split up and the obtained parts must be stored in a well-organized manner.

4.2.1 Text Data Structures

Medical texts have to be analyzed and preprocessed. This includes the fragmentation of texts into sections and a further, much finer splitting of sections into smaller segments.

4.2.1.1 Sections of Discharge Letters

In the medical domain two types of documents exist: special reports e.g. for heart echo or X-ray etc. and documents summarizing other reports e.g. discharge letters. A discharge letter (also described as *doctor's letter*) usually consists of different sections from various medical domains. Mostly, sections like diagnoses, medical history, therapy and medication are included. For a detailed description see Section 2.1.3.1. For many applications, the information of just a particular medical domain (represented as a section in the discharge letter) is relevant. For example, only the medication section is relevant for an analysis on drugs and their agent groups, since all drugs are mentioned there. For that reason the discharge letter is split up in its different sections. This has several advantages: First, sections can be treated more individually: Sections differ from each other with respect to structure and lexical nature. The language processing can be adapted to their characteristics. As a second benefit, the processing times are shortened by focusing on the relevant sections, a lot of non-relevant text does not have to be considered. This point is particularly important for requests during runtime. The third advantage is that information can be searched more precisely, as sections can be selected or omitted as data source. For example: The section *family history* describes the diseases of the family members, not of the patient. If an application or task is only interested in the diseases of the patient, the section *family history* should be excluded as data source, to avoid confusion.

A section is defined by Krebs [125] as a coherent, semantic unit of a discharge letter. The content of a section consists of data from only one medical domain. A section is typically initiated by a headline, but may also include subheadings.

The following text shows the sections of the example a discharge letter in German from Section 2.1.3.1 [1]. All sections are marked with a frame.

Diagnosen

- Akuter Hinterwandmyokardinfarkt mit ST-Hebung (sogenannter “STEMI”)
- Koronare 2-Gefäßerkrankung

Weitere Diagnosen

- Diabetes Mellitus

Anamnese

Am Aufnahmetag bekam Herr Sowieso plötzlich starke Luftnot und Thoraxschmerzen bei leichter Belastung. Bereits seit ca. 6 Monaten habe er immer wieder leichte thorakale Schmerzen, ...

Labor

Natrium: 144 [135 - 145] mmol/l, Kalium: 4.7 [3.5 - 5] mmol/l, Creatinin: 1.1 [0.5 - 0.9] mg/dl, GOT (ASAT): 7.5 [0 - 15] U/L, GPT (ALAT): 15.0 [0 - 17] U/L, GGT: 20.0 [0 - 18] U/L, Thromboplastinzeit n. Quick: 112 [70 - 130] %, ...

Transthorakales Echocardiogramm:

Linker Ventrikel normal weit und normal kontraktil, keine regionalen Wandbewegungsstörungen, auch keine erkennbare HW-Narbe, Auswurffraktion normal, EF planimetrisch 64 %. grenzwertige LV-Hypertrophie, ...

Herzkatheteruntersuchung

Beurteilung: Es zeigt sich als Ursache des akuten Hinterwandinfarktes ein proximaler Verschluss der dominanten RCA. In gleicher Sitzung erfolgt eine PCI. Nach Vorführen des Führungsdrahtes Thrombussaugextraktion, ...

Epikrise:

Der Patient wurde wegen instabiler Angina pectoris eingewiesen. Im EKG zeigten sich diskrete Hebungen in der Ableitung II und III und ST-Senkungen V1 bis V3. In der Herzkatheteruntersuchung zeigte sich ein Verschluss des RCA, ...

Therapieempfehlung / Medikation:

Simva (Simvastatin) 20 0-0-1
Plavix (Clopidogrel) 75 1-0-0 für weitere 12 Monate ...

The sections have a different sentence structure depending on the medical domain. A narrative style with long and complex sentences is used in sections such as *anamnesis*, *physical Examination* and *therapy*. A rather telegraphic style with short sentences is mainly common for findings reports. Pure enumerations may also occur, e.g. in the section *laboratory*, but often a mixture of enumerations, key words, and complete sentences is used. The diagnosis section is an example for that hybrid style.

4.2.1.2 Segments Within Sections

A section presents all information of one medical domain. For example, the text of the section *physical examination* contains all relevant findings. This text can be split into finer grained *segments*, which are defined as semantically closed clauses. A segment consists of a few words that describe a concept or it is a whole sentence that describes an issue. The segments for a section are created with a section specific splitter. The following examples describe segments of various sections.

Anamnesis

Am Aufnahmetag bekam Herr Sowieso plötzlich starke Luftnot und Thoraxschmerzen bei leichter Belastung.
Bereits seit ca. 6 Monaten habe er immer wieder leichte thorakale Schmerzen, die vom Hausarzt als vertebragen klassifiziert worden seien.
Vom Notarzt hatte er Fentanyl und Nitro Spray bekommen.
In der Familie keine Herzerkrankungen, kein Schlaganfall; fraglich Diabetes.

Examination report (echocardiogram)

Linker Ventrikel normal weit und normal kontraktil,
keine regionalen Wandbewegungsstörungen, auch keine erkennbare HW-Narbe,
Auswurffraktion normal, EF planimetrisch 64 %. grenzwertige LV-Hypertrophie,
Septum edd 10 mm, diastolische Relaxationsstörung Grad II (DT 284 ms,
IVRT 57 ms, LA-Vol.-Index 35,1 ml/m^2). Vorhöfe leicht dilatiert (LA 23 cm^2,
RA 17 cm^2) und rechter Ventrikel normal weit, gute RV-Funktion,
TAPSE 24 mm. Aortenklappe leicht sklerosiert, keine Stenose,
leichtgradige Insuffizienz (PHT nicht messbar). Leichtgradige Mitralinsuffizienz
V. contracta 4 mm. Trikuspidalis unauffällig.
Pulmonalis mit leichtgradiger Insuffizienz. Normfrequenter Sinusrhythmus,

Medication

CSimva (Simvastatin) 20 0-0-1

Plavix (Clopidogrel) 75 1-0-0 für weitere 12 Monate, CoDiovan 160/25 mg 1-0-0, Beloc zok mite (Metoprolol) 1-0-0, Delix (Ramipril) 5 1-0-0

Selbstverständlich können auch vergleichbare eventuell kostengünstigere Medikamente mit gleichen Inhaltsstoffen verabreicht werden.

4.2.2 Text Index

A full-text index is used in information retrieval for storing and retrieving documents. A search engine index parses and stores documents in a sophisticated way, that optimizes the speed and performance of finding relevant documents. Broadly speaking, it reduces the search time from a full scan of all documents to a tree search. Before a document is stored in an index, it is passed through an NLP pipeline which preprocesses the texts. Standard processing steps are tokenizing, stemming, and stop words removal. Thereafter, the tokens are stored in an index structure. For each token, the list of documents in which it occurs is stored allowing fast response times of queries. Modern index libraries have sophisticated indexing mechanisms and efficient search algorithms. The most popular project is Apache Lucene[2] and is used by the search engine frameworks Apache Solr[3] and Elasticsearch[4].

Furthermore, the index is used as a storage engine and is a central component of ad hoc IE. We choose Apache Solr. The following sections provide descriptions of features and concepts that exist in Apache Solr. These functions are available in a similar way in other index libraries for search engines like Elasicsearch.

4.2.2.1 Basic Concepts

Documents. The basic unit of information in Solr is a *document* containing a set of data of one topic [75]. In a clinical data warehouse, a document represents a case of a patient and comprises all data of one patient for one hospital stay or visit. For example, if a patient has a broken leg, the case includes all medical treatments, such as admission, examinations, surgery, stay on the ward, and discharge. The patient does not have to be an inpatient at the clinic, outpatients are treated the same way. All information about a case is stored in one document.

[2] `https://lucene.apache.org/core/`, accessed August 2018
[3] `http://lucene.apache.org/solr/`, accessed August 2018
[4] `https://www.elastic.co/`, accessed August 2018

A document can be nested and can contain child documents.[5] However, the documents cannot be nested arbitrarily as Solr (unlike Elasticsearch) only allows one child hierarchy level.

Fields. A document consists of *fields* that represent specific information [75] and could be the name or the age of a patient. Fields can accept different data types such as texts, numbers, dates, Boolean values or floating point numbers. A field can contain a single value, or a list of values.

4.2.2.2 Document Structure

A lot of information can be stored in a Solr document, e.g. all information of a patient case. For ad hoc IE, text data is stored in a special structure. In addition to the original version of the text, all confirmed findings of each text and all confirmed and currently valid findings of the patient are stored in individual fields. For this purpose, all negated, historical and patient-related information is removed. By storing this information in separate fields, the semantic different versions of the texts can be requested independently of each other. Figure 4.1 gives a schematic representation of the structure of text fields within Solr document.

4.2.2.3 Text Representation

As mentioned in Section 4.2.1.1 and 4.2.1.2, a text can be divided into sections and further split into smaller segments. Each section is stored in a separate field that has the data type *text* and is multi-valued. That means one field contains a list of texts each corresponds to a segment. In other words: a field contains all segments of a section. Queries with several words (e.g "tiredness and headache") that should appear in one section can be located in different segments. That is no problem because queries search and match tokens in all texts of a field per default. However, this is not desired for special requests. For example, a phrase query or a context-sensitive query (e.g. "cardiac decompensation") requires that all query-words occur in one segment. This can be ensured by a configuration parameter that handles cross segment queries.

[5] `https://lucene.apache.org/solr/guide/7_4/uploading-data-with-index-handlers.html#nested-child-documents`, accessed August Aug 2018

original text

Diagnosen:
- NSTEMI und kardiale Dekompensation
- Paroxysmales Vorhofflimmern
- Ausschluss von Niereninsuffizienz

Zustand nach Schrittmacher-Implantation 5/09
CVRF: arterielle Hypertonie (Vater mit 47 Jahren an Myokardinfarkt verstorben)

affirmed findings

Diagnosen:
- NSTEMI und kardiale Dekompensation
- Paroxysmales Vorhofflimmern

Zustand nach Schrittmacher-Implantation 5/09
CVRF: arterielle Hypertonie (Vater mit 47 Jahren an Myokardinfarkt verstorben)

recent affirmed patients findings

Diagnosen:
- NSTEMI und kardiale Dekompensation
- Paroxysmales Vorhofflimmern

CVRF: arterielle Hypertonie

Figure 4.1: **Structure of a text field within a Solr document.** In addition to the original version of the text, all confirmed findings of each text and all confirmed and currently valid findings of the patient are stored in different fields. For this purpose, all negated, historical, and patient-related information is removed. (Schematic representation)

4.3 Algorithms for Ad Hoc Information Extraction

This section introduces the algorithms for ad hoc Information Extraction. This novel technique allows the extraction of concepts and their values from plain texts on the fly, as defined above.

An ad hoc IE system consists of several modules, processes, and techniques. Section 4.3.1 begins with the system design showing how the individual modules work together. First, medical text have to be fragmented, Section 4.3.2 shows how this works. Sections 4.3.3 and 4.3.4 discuss in detail how to identify negations and the context of information in a text and how to exclude information not required, which otherwise might be a potential error source. The text query features that help extract information ad hoc are described in Section 4.3.5.

4.3.1 System Design

During data integration, the texts are preprocessed in an analysis pipeline. First, the sections are identified and separated from each other. Each individual section is segmented into semantic sentence blocks (segments). The Negex and Contex algorithms are applied to these clauses to identify negated phrases and the context of information in the text, respectively all negated parts with their scopes and all information, which are not recent and do not belong the patient are removed. The remaining text with only affirmed and currently valid findings of the patient is passed into the pipeline as is the original text but with a different label. After that, lexical analysis is applied to the texts using standard NLP steps such as tokenizing, stemming and stop-word removal. Finally the texts are stored in an index structure so that they can be queried separately afterwards, like all other information. The system design outlined in Figure 4.2.

Ad hoc IE requests can be made during runtime. In the GUI a user defines a query that request concepts that are occur in text documents. The back-end server receives this query, compiles it and sends to an index server that retrieves the matching documents. The response of the server is returned to the back-end server that creates a result corresponding to the input query of the user. The user interfaces presents the final result to the user. Figure 4.3 gives an overview of the process.

Lexical Analysis The standard NLP steps work very well for most medical domains. In a few areas the pipeline is slightly modified to accommodate the specifics of a domain or to provide additional functionality. For example, we added a sentence splitter in the medication section which separates the individual medication instructions from each other. Furthermore, the stemmer was deactivated because the word endings of the medications should not be touched. Lemmatization also addresses this problem, but could not be applied since not all lemmas are known. However, deactivating the stemmer has the same effect and since this section only contains mentions of drugs, it is the better solution. Finally, a custom tokenizer ensures that the quantity, strength, and dosage

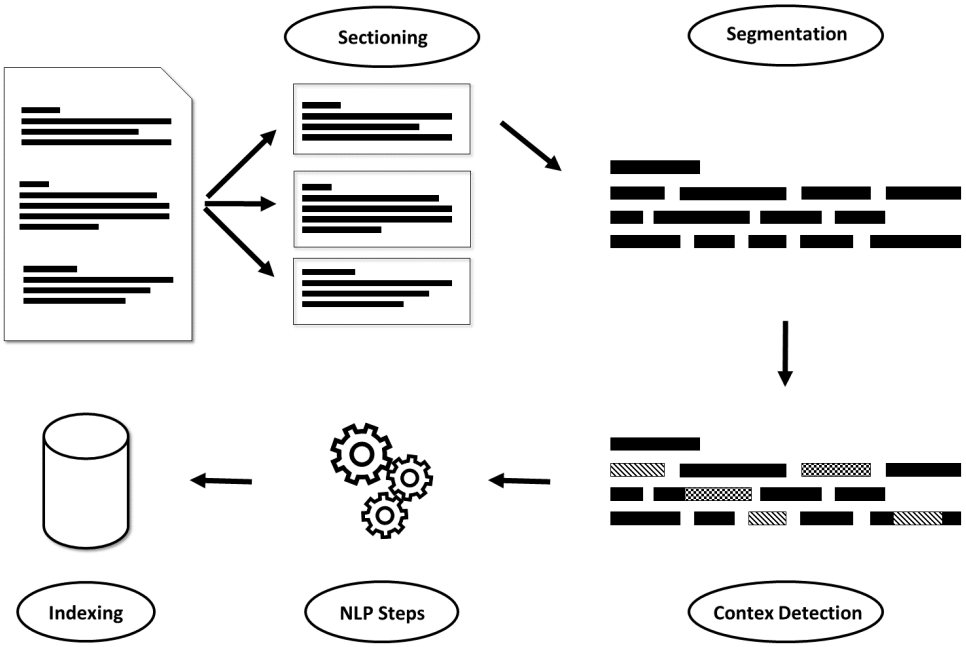


Figure 4.2: **System design of the preprocessing pipeline for medical texts.** A discharge letter as input document is split up by its sections, which are further segmented in smaller, semantic sentence blocks. These clauses are processed with the Negex and Contex algorithms and stored in separate fields in the index after a lexical analysis.

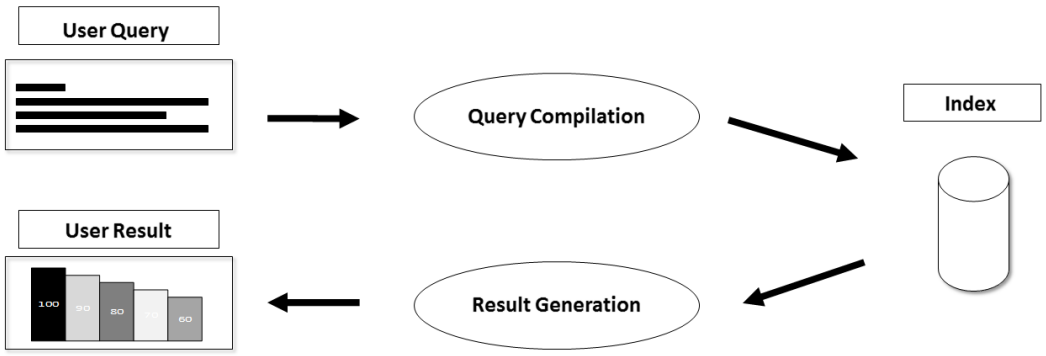


Figure 4.3: **System design of the ad hoc IE at runtime.** The input is a user query focusing on concepts that are hidden in the text. The ad hoc IE query is compiled and sent to an index server. The response of the server is used to create a result corresponding to the input query of the user.

Table 4.1: **Lexical analysis of the medication section in the discharge letter.** A text is split in segments consisting of drug names and medication instruction. A special tokenizer preserves the dosage information.

text	sentences	tokens
Delix 10mg 1-0-0, Belok zok 1/2-0-0, Mono-Mack 20 1-1-0	Delix 10mg 1-0-0 Belok zok 1/2-0-0 Mono-Mack 20 1-1-0	Delix, 10, mg, 1, 0, 0 Belok, zok, 1/2, 0, 0 Mono, Mack, 20, 1, 1, 0

information of the medication instructions are correctly decomposed. Table 4.1 shows an example of the lexical analysis.

4.3.2 Text Fragmentation

The first step in the text preprocessing pipeline is the fragmentation of discharge letters into smaller pieces. This is necessary (1) as prerequisite to identify negations and (2) in order to get small, semantically closed text segments describing a concept that will be used later for the IE. Reports of findings do not have to be split up because they contain information of a single medical domain, which is equivalent to a section of the discharge letter.

First, the text must be fragmented into larger sections before determining smaller segments. As mentioned in sections 4.2.1.1 and 4.2.1.2, the sentence structure of the text of a section can be different depending on the domain. That is why the segments are created in a different way for each section. This is achieved by e.g. the use of different separators.

4.3.2.1 Sectioning

We developed an algorithm to sectionize discharge letters [125]. It splits up the text into semantic sections representing the medical domains. The desired result is described in Section 4.2.1.1. The algorithms to create sections works that way:

Input: The text of a discharge letter.

Output: The text divided in semantic sections representing the medical domains.

---

1. Generate candidates for headlines.
2. Classify the candidates in headings and non-headings and determine their category.
3. Delete consecutive headings of the same category
4. Mark sections from heading to heading.

This is a schematic representation of the algorithm. The actual procedure consists of ten steps. For further information please see at Krebs [125].

4.3.2.2 Segmentation

The segmentation splits a text into finer grained segments or semantically closed clauses which consist of a few words that describe a concept. A segment may also be a whole sentence that describes an issue. The desired result is described in Section 4.3.2.2. The input for the segmentation is a text like a section of a discharge letter or a findings report. The basic procedure for segmenting a text is shown below [125]:

Input: The text of a section or a findings report.

Output: The text divided into segments representing semantically (sub) closed clauses.

---

1. Mark all numbers and abbreviations within a section.
2. Define the splitter set. Determine if the section contains verbs:
   a) Yes: Use the default splitter set.
   b) No: Use the domain specific splitter set. If none is defined, use the default one.
3. Mark all splitter characters in the text and then delete the ones, which are covered by number or abbreviation marks.
4. Mark segments from separator to separator.
5. Check again for every segment if it contains verbs. If not, use the procedure again within this segment. If so, the segmentation process for the current segment is done.

A splitter set is a list of delimiters such as punctuation characters like “, . ; : ! ?” or a line break character. The comma sign and the semicolon may be included in the splitter set, since in many clinical abbreviated texts they serve as a regular period, e.g. “Zungenmotilität normal, keine Zungenfibrillationen, Zungenkraft normal” (engl. *Tongue motility normal, no tongue fibrillation, tongue force normal*). Some exceptions were made to capture *Hearst-Patterns*, i.e. enumerations like “Keine Stauungszeichen, Infiltrate oder Ergüsse.” (engl. *No cramps, infiltrates or effusions.*).

Section 4.2.1.2 shows the result of the segmentation process for several sections.

The quality of the segmentation process depends on the quality of the splitter set which in turn depends strongly on the medical domain. In this work, new delimiter were found for various domains like the diagnoses or medication section.

Table 4.2: **Example for the trigger labeling.** Triggers are assigned according to the longest match sequence. In the first sentence, the prefix negation trigger label is chosen because it has a longer match sequence than the postfix negating trigger in sentence two. The trigger tokens are highlighted.

Token	Text with trigger
PREN	**Keine Anhaltspunkte für** pulmonale Metastasierung.
POST	Für eine pulmonale Metastasierung ergaben sich **keine Anhaltspunkte**.

4.3.3 Negation Detection

A prerequisite for useful counts of search results in clinical texts is the reliable detection of negations. We developed an extended version of the NegEx algorithm [29] to identify negations. Further adaptions were made because the input of the root algorithm has two arguments: a sentence and a concept token. The output is whether the concept is negated or not. In our system, the negation detection is part of the preprocessing. But the concepts that should be classified (affirmed/negated) are not present at preprocessing time, because it is given later within a query by users at runtime. Therefore, we determine the negations and their scope within a sentence. User query medical concepts at runtime and all concepts, that are located in a negated scope, are considered as negated. Like ConText [91], we extended the NegEx algorithm by changing the negation scope from fixed length of six tokens to a variable length to the next trigger token. Moreover, we extended the trigger token set.

4.3.3.1 Trigger Set

We took the trigger list from [42], which is based on the translation to German by [30]. We edited and extended this list to a size of 557 triggers. While Cotik et al. [42] label the trigger tokens in a given sentence according to a fixed precedence list, starting with pre-negating trigger tokens (PREN), followed by post-negated trigger tokens (POST), prepositions (PREP) and pseudonegating trigger tokens (PSEU), we always choose the label with the longest match sequence. In the first example in Table 4.2, the pre-negating trigger "keine Anhaltspunkte für" (engl. *no evidence for*) has a longer match sequence than 'keine Anhaltspunkte" (engl. *no evidence*). As a result, the words "keine Anhaltspunkte für" in the text are labeled with the pre-negating trigger (see Table 4.2).

4.3.3.2 Algorithm Description

The algorithm for the negation detection works as shown below:

Input: A sentence and a trigger list.

Output: Negated scopes in the given sentence.

---

1. All trigger tokens in a given sentence are annotated with their corresponding label.
2. The algorithm iterates over the trigger tokens of the sentence:
   a) At every post-negating trigger token a negation scope is added from the last trigger token (or the begin of the sentence) to the current one.
   b) At every pre-negating trigger token a negation scope is added from the current trigger token to the next trigger token (or the end of the sentence).

4.3.3.3 Sentence Splitting

Since NegEx requires a single sentence as input, but we have entire texts, we have to apply sentence splitting to these texts. We use the segmentation algorithm, explained in Section 4.3.2.2 to divide the text into semantic units.

The original RegEx algorithm expects whole sentences as input. Meanwhile, we use our segments, which are usually smaller phrases. These are more suitable for the telegraphic writing style, which is often used. For example, in a findings report, a comma also represents a delimiter and the sentence should be split into two segments at this point.

Figure 4.4 shows an example of the output of the negation detection algorithm. The negation triggers are bold and the negation scope is underlined. First, the text is split up and then the algorithm is applied.

4.3.4 Context Detection

The context of information in a discharge letter is an important topic. Many pieces of information are negated [28] (e.g. “no fever”, “dizziness is denied”) or they relate to other persons (e.g. within the context of family history). Some information like medications within in the discharge letter have a temporal context and may not be valid anymore (e.g. medication might have been stopped at hospital entry or during hospitalization, like Ramipril in Figure 4.6).

Kein Pleuraerguss, **kein** konfluierendes Infiltrat, **keine** Stauungszeichen. **Keine** malignitätssuspekten Rundherde. Herz links betont vergrößert. Aorta elongiert und sklerosiert. Ramipril abgesetzt.

(a) Echocardiogram reports

Harnröhre zeigt **keinen Hinweis** für eine Striktur, Prostata ist **nicht** obstruktiv, nebenbefundlich enger Blasenhals, Blasenschleimhaut trabekuliert, jedoch **kein Hinweis** für einen exophytischen Blasentumor. Ostien bds. orthotop, schlitzförmig mit klarem Urinjet.

(b) Urethrocystoskopie

Figure 4.4: **Example for negated findings in medical reports.** The negation triggers are bold and the negation scope is underlined.

Table 4.3: **Context dimensions and their values for information in medical report.**

Dimensions	Values
Negation context	affirmed, negated, possible
Temporal context	recent, historical, hypothetical
Experiencer	patient, other patients

4.3.4.1 Context of Information

Depending on the application or evaluation, different types of information are relevant or must be excluded. In most cases, physicians are interested in confirmed and current findings of a patient.

Another decision to consider when choosing the data is whether to include only recent findings or historical information (e.g. "myocardial infarction 25 years ago") as well. For some use cases, errors may occur if the historical findings are not excluded.

A similar issue is the *experiencer context* which describes the referenced person of an information. In other words, to whom does an information relate? In general, a piece of information is related to the patient, but it can also relate to other peoples like relatives. They can be mentioned in a background report or in special sections like family history. The context of this information can easily be overlooked. Table 4.3 lists the context dimensions of information in medical reports and their values.

The context of the information is expressed differently depending on the section. Negated findings appear in examination reports or diagnostic findings reports and exclude findings or diseases.

Information with a temporal dimension may occur, for example, in the diagnosis section. Frequently, important events from the patient's history are listed here like severe illnesses,

Diagnosen:
Aktuell:
- NSTEMI und kardiale Dekompensation
- Coronarangiografie noch ausstehend bei aktuell steigenden Nierenwerten und unklarer mikrozytärer Anämie
- V.a. Raumforderung, bei klinischer Relevanz Endosographie im Verluaf empfohlen.
- V.a. Schrittmacherdysfunktion (s. Epikrise)

Vordiagnosen:
Koronare Herzkrankheit
Z. n. Myokardinfarkt 1991 und 2001
Z. n. Schrittmacher-Implantation 5/09
Intermittierende AV-Block II. Grades
Paroxysmales Vorhofflimmern

CVRF:
arterielle Hypertonie (positive Familienanamnese: Vater mit 47 und Bruder mit 38 Jahren an Myokardinfarkt verstorben)

Figure 4.5: **Example of a diagnosis section of a discharge letter.**

operations or other medical interventions. Historical information can occur in every section, even the medication section: For example, some medications within in the discharge letter are not valid any longer and have to be disregarded for some evaluations, e.g. medication might have been stopped at hospital admission or during hospitalization, like *Ramipril* in Figure 4.6.

Information that is not patient related is often recorded in the anamnesis section, e.g. hereditary diseases of ancestors. For special applications, however, even this information may be important.

Figure 4.5 shows an example of a diagnosis section of a discharge letter and contains examples for the described contexts of information. The statements "V.a. Raumforderung [...]" and "V.a. Schrittmacherdysfunktion" begin with the German abbreviation "V.a", which stands for "Verdacht auf" (engl. "suspected"). Thus, these statements are not confirmed, but only possible. The two lines "Z. n. Myokardinfarkt 1991 und 2001" and "Z. n. Schrittmacher-Implantation 5/09" have a temporal context and are historical. On the one hand, they start with the abbreviation "Z.n." ("Zustand nach". Engl.: "condition after" / "status post"), on the other hand, they contain a historical date ("1991", "2001" and "05/2009"). In the sentence "Vater mit 47 und Bruder mit 38 Jahren an Myokardinfarkt verstorben" (engl. "Father died with 47 and brother with 38 years of myocardial infarction" is the experiencer of the incident "myocardial infarction") the person "father" and "brother", not the patient himself.

Negations can already be detected with the procedure described in Section 4.3.3. This approach does not yet recognize whether a fact is current or historical, nor if it refers to the patient or to another person, such as a relative.

Medikation bei Entlassung:	Medication at discharge:
Beloc-Zok 1/2 – 0 – 1,	Beloc-Zok 1/2 – 0 – 1,
Pantoprazol 20mg 1/2 – 0 – 1/2,	Pantoprazol 20mg 1/2 – 0 – 1/2,
Delix 5 plus 1-0-0, ASS 100 0-1-0, Plavix 0-1-0, Zocor 0-0-1.	Delix 5 plus 1-0-0, ASS 100 0-1-0, Plavix 0-1-0, Zocor 0-0-1.
Ramipril abgesetzt.	Ramipril stopped.
(a) German	(b) English

Figure 4.6: **Example of a medication section of a hospital discharge letter.**

Table 4.4: **Trigger types used in the context algorithm.** The table shows the trigger token types used in the context algorithms with their corresponding dimension and value. In addition, the position of the trigger within its scope is given.

Description	Dimension	Value	Position at scope
prefix negation trigger	negation context	negated	begin
postfix negation trigger	negation context	negated	end
prefix possible trigger	negation context	possible	begin
postfix possible trigger	negation context	possible	end
pseudo negation token			
conjunction			
prefix historical trigger	temporal context	historical	begin
postfix historical trigger	temporal context	historical	end
infix historical trigger	temporal context	historical	in between
infix experience trigger	experiencer	other person	in between

Hence, we extended the NegEx-version to a ConText [91] algorithm implementation that handles not only negations but also the context of an information.

4.3.4.2 Trigger Set

The following listing shows examples for the trigger tokens in German. The complete trigger set is online available.[6]

prefix negation trigger Ausschluss, kein Hinweis, keine Anzeichen, ohne ,negativ in Bezug auf, entwickelte sich nie, keine Klagen über

post negation trigger kann ausgeschlossen werden, nicht präsent, nicht vorhanden, nicht nachweisen, verneint

[6]The complete trigger set is available at: `go.uniwue.de/padawan`

prefix possible trigger unsicher, unwahrscheinlich, vermutlich, scheinbar, kann es sich um Verdacht auf, V.a.

postfix possible trigger unwahrscheinlich, nicht wahrscheinlich, nicht unbdingt, nicht zwangsläufig, weiß nicht, nicht sicher

pseudo negation token keine abnormale, keine verdächtige Veränderung, keinen signifikanten, ohne Schwierigkeit kein Wechsel, keine Zunahme, kein Anstieg

conjunction aber, als Grund der, als Quelle von, als Ursache von, als eine Ätiologie für, außer, obwohl, dennoch, jedoch, unabhängig davon

prefix historical trigger Zustand nach, Z.n.,

postfix historical trigger pausiert, Pause, abgesetzt, abgelehnt

infix historical trigger vor Jahren, vor Monaten, Frühling, Sommer, 19[\d]{2}, 20[\d]{2}, [\d]{2}/20[\d]{2}, Verlaufskontrolle, Kontrolluntersuchung

infix experiencer trigger Vater, Mutter, Oma, Opa, Großvater, Tante, Onke, Angehörige

4.3.4.3 Algorithm Description

We developed a version of the Contex algorithm based on original approach [91].

Input: A sentence and the list of trigger tokens.

Output: The scopes in the sentence, which are negated, hypothetical, historical, or experienced by someone other than the patient.

---

1. All trigger tokens in a given sentence are annotated with their corresponding label.
2. The algorithm iterates over the trigger tokens of the sentence. It determines the different scopes like a sweep line algorithm and keeps all active scopes in memory and performs an action on every trigger type:
   a) No action is performed on pseudo negation tokens.
   b) All types of triggers, which are at the beginning of a scope, are stored in an "active scope list". These are e.g. pre-negating and pre-historical trigger, but also infix-historical triggers.
   c) For each trigger type that is at the end of a scope, scope annotations are created:
      i. At each conjunction or end of sentence, scope annotations are created for all active scopes, from their start to the conjunction. The active scope list is cleared.
      ii. For the remaining trigger tokens, which themselves cause a scope and are at the end of it, scope annotations are created from the last conjunction or sentence beginning to the current token.

Identification of blocks with context. The context algorithm identifies the context of information within sentences. In medical documents it is possible that entire paragraphs or sections to have a specific context. For example, in Figure 4.5 , there is a section titled *Vordiagnosen* (engl. *preliminary diagnoses*) that describes historical diagnoses of the patient. These blocks and their context are identified with the simple following procedure:

Input: The text of a section and a list of trigger tokens.

Output: Scopes in the text, which are negated, hypothetical, historical, or experienced by someone other than the patient.

---

1. Identification of paragraphs and blocks
2. Keyword matching with the headline of the paragraph, block or section
   a) If there is match, mark the entire paragraph, block or section with the appropriate context.
   b) If there is no match, no action is performed.

4.3.5 Text Query Features

During run time, the index can be requested through an interface. That's where the ad hoc IE takes place. We developed query and output features to extract information and to make them available for further processing [51].

There are well-known functions like Boolean retrieval, wildcards and phrase queries, and more advanced features like a context specific query, a regular expression query with filter options and output definition.

4.3.5.1 Basic Query Features

Boolean retrieval: Boolean retrieval filters document that contain the given query tokens (bag of words).

Logical Operators: They can be combined via logical operators *and*, *or*, *not*. A query for heart failure could look like:

$$\text{(cardiac decompensation) OR (heart failure)} \tag{1}$$

- Matching snippets for query (1):
  - The patient has a heart failure.
  - The cardiac examination revealed a decompensation.
- Non matching snippets for query (1):
  - The patient showed a decompensation.
  - There is a cardiac failure.

Wildcard Query: A wildcard character is used to substitute any characters in a word, e.g. in the German compound words like (2). (mitral insufficiency)

Mitral*insuffizienz (2)

- Matching snippets for query (2):
  - Minimale Mitralinsuffizienz.
  - Leichtgradige Mitralklappeninsuffizienz.
- Non matching snippets for query (2):
  - Mitrale Insuffizienz
  - insuffiziente Mitralklappe

Phrase Query: A phrase query matches texts containing a particular sequence of words. The entire sequence must match like:

“diabetes mellitus” (3)

- Matching snippets for query (3):
  - . . . mit nachfolgender Verdopplung des Blutzucker-spiegels (bekannter Diabetes mellitus).
  - Weitere kardiovaskuläre Risikofaktoren: Diabetes mellitus Typ 2, arterielle Hypertonie
- Non matching snippets for query (3):
  - Vordiagnosen: Diabetes.(Genauer: mellitus, Typ 2)
  - Typ-2-Diabetes

Table 4.5: **Example for a context-sensitive query.** The query (4) matches any text containing the two terms "dilatiert Vorhof" in one sentence with not more than seven words (default value) between them. The order of words does not matter. In contrast to the query (5): Here, the order of words matters and the gap between the query tokens must not be more than three words.

Query	Matching Text	
[dilatiert Vorhof]	Der linke Vorhof ist deutlich dilatiert	(4)
[3+ Suffiziente Mitralklappe]	Suffiziente Aorten- und Mitralklappe	(5)

Table 4.6: **Example of the regular expression feature.** It queries (6), constraints (7) and extracts (8) a numeric concept (Puls = pulse, ZAHL = NUMBER). "$1" is a reference to the extracted concept (the first expression in round parentheses or its equivalent predefined class, i.e. "ZAHL").

Syntax	Alternative Syntax	
/Puls ZAHL/	/Puls [0–9]+/	(6)
/Puls ZAHL/[ZAHL > 150]	/Puls ([0 − 9]+)/[$1 > 150]	(7)
/Puls ZAHL/[ZAHL > 150] ZAHL	/Puls ([0–9]+)/[$1 > 150]$1	(8)

4.3.5.2 Context-Sensitive Query

In contrast to a Boolean query, where the terms can be located anywhere in the text, in a context-sensitive query the user has control over the proximity and order of the terms in the query. The given terms must occur in the same sentence (see Table 4.5). The query (4) would match any text that contains these two terms in one sentence with less than eight words (default value) between them. The order of words does not matter. In contrast to the query (5), here the order of words is important and the gap between the query tokens must not be more than three words.

4.3.5.3 Advanced Regular Expression Query

The regular expression (regex) query feature is a further function to filter texts. Experienced user can write a regular expression using the standard regular expression syntax with predefined character classes, quantifiers, alternatives, and grouping. An introduction in regular expressions and their definitions is given on this info page[7] The regular expression is defined between slashes (see Table 4.6). For users with no computer science background we added predefined classes for convenience, like *ZAHL* (engl. *number*), which are compiled to a regular expression automatically. Table 4.6 shows an example of the regex query feature for a numeric concept.

Line (6) queries the existence of the concept in the text. Line (7) adds a numeric condition, which is defined in brackets. That query would match all texts with the token

[7] `https://www.regular-expressions.info/reference.html`, accessed November 2018

Puls followed by a number that is bigger than 150. This number would be returned within the result. Line (8) extracts the numeric value of the concept for further computation. That is defined in the query syntax by writing the desired group (ZAHL or "$1") at the end of the query.

If a query is run with that extraction mode, the engine returns a list with all type safe extracted numbers for the queried concept. Further features of the query syntax are given in the example (9) and (10).

```
/Blutdruck ZAHL \/ZAHL/[$2 >150]                    (9)
/([0-9]+)\.([0-9]+)\.([0-9]+)/$3-$2-$1              (10)
```

If the query contains more than one number like 'Blutdruck 150/90' (blood pressure), the numbers can be referenced using the $-notation (see examples 8 and 9). The escape character is the backslash. Not only can the predefined class *NUMBER* be referenced and extracted, but also self-created regex groups can be used. The groups are defined in parentheses in accordance with the regex syntax (used in lines 6-10). As an additional use case example, the runtime IE mechanism can be used to transform notations like a German date (i.e. "dd.MM.yyyy") into the English equivalent ("yyyy-MM-dd") for further computation (see 10). The regex query, containing a numeric condition, is compiled into the regular expression. (11) is the compiled result of query (7). This regular expression is passed to the index server as a normal constraint query and can be evaluated efficiently. So no post-processing of the results is necessary.

$$\text{Puls}(15[1-9])|(1[6-9][0-9]\{1,\})|([2-9][0-9]\{2,\})|([1-9][0-9]\{3,\}) \quad (11)$$

Table 4.7 shows further examples for numeric ad hoc IE. Query 1 retrieves all documents that contain the word "Puls" followed by a number. The result is a text snippet that contains the query tokens. Query 2 matches the same texts and extracts the numbers from them. Query 3 has an additional constraint that restricts the matching documents: The number must be bigger than 120. Query 4 is an example how synonyms can be defined. The term "Herzfrequenz" (engl. *heart rate*) is defined as an alternative to "Puls". The query does not contain a constraint, but a result output definition. "$0" means *group 0* and refers to the entire query. This definition corresponds to the syntax of regular expressions. Group 1 is "(Puls|Herzfrequenz)" and group 2 is the number.

4.3.5.4 Spelling Error Tolerant Query

Since medical reports are often typed manually, some words are misspelled. For such typos we added a spelling error tolerant query feature using the Damerau-Levenshtein distance that is a string metric for measuring the edit distance between two sequences. In our system it is used to assess how much two names differ. The distance measures includes a transposition operation (transposition of two adjacent characters) in addition

Table 4.7: **Example queries for nurmeric IE.** Query 1 retrieves all documents that contain the word "Puls" followed by a number. Query 2 matches the same texts and extracts the numbers of them. Query 3 has an additional constraint: The number must be bigger than 120. Query 4 shows the definition of a synonym: "Herzfrequenz" (engl. *heart rate*) is an alternative to "Puls".

Query / Text	**Match**	**Result**
Query 1: /Puls ZAHL/		
Blutdruck 107/52 mmHg, Puls 116/min,	✓	Puls 116/min
RR 152/98 mmHg, Puls 70/min, regelmäßig.	✓	Puls 70/min
Des Weiteren läge der Puls oft niedrig.		
Pat. in gutem AZ und EZ, Puls regelmäßig.		
RR 134/76 mmHg, Puls 88 /min, regelmäßig	✓	Puls 88 /min
O2-Sättigung 97 %, Puls 145/min.	✓	Puls 145/min
Query 2: /Puls ZAHL/ZAHL		
Blutdruck 107/52 mmHg, Puls 116/min,	✓	116
RR 152/98 mmHg, Puls 70/min, regelmäßig.	✓	70
Des Weiteren läge der Puls oft niedrig.		
Pat. in gutem AZ und EZ, Puls regelmäßig.		
RR 134/76 mmHg, Puls 88 /min, regelmäßig	✓	88
O2-Sättigung 97 %, Puls 145/min.	✓	145
Query 3: /Puls ZAHL/[ZAHL > 120] ZAHL		
Blutdruck 107/52 mmHg, Puls 116/min		
RR 152/98 mmHg, Puls 70/min, regelmäßig.		
RR 134/76 mmHg, Puls 123 /min, regelmäßig	✓	123
O2-Sättigung 97 %, Puls 145/min.	✓	145
Query 4: /(Puls|Herzfrequenz) ZAHL/$0		
Blutdruck 107/52 mmHg, Puls 116/min	✓	Puls 116
RR 152/98 mmHg, Puls 70/min, regelmäßig.	✓	Puls 70
RR 134/76 mmHg, Puls 88 /min, regelmäßig	✓	Puls 88
O2-Sättigung 97 %, Herzfrequenz 145/min.	✓	Herzfrequenz 145

Table 4.8: **Damerau–Levenshtein distance for typos in drug names.** Examples of misspelled drug names and their Damerau–Levenshtein distance

Product name	Misspells	Distance	Operation
Ibuhexal	Ibohexal	1	substitution
Cordarex	Kordarex	1	substitution
Warfarin	Wafarin	1	insertion
Euphylong	Euphyllong	1	deletion
Repaglinid	Repagilnid	1	transposition
Ramipril	Rampiril	1	transposition
Repaglinid	Repagilid	2	transposition, insertion

Table 4.9: **Promximity searches in the medication domain.** Example for promximity searches to query the daily dose of a medication instruction

Query	Expanded Query	Matching	Not matching
Delix 5 mg	"Delix 5 1 0 0" OR "Delix 5 1/2 1/2 0"	Delix 5mg 1-0-0 Delix 5mg 1/2-0-1/2 Delix 5-mg 0 1 0	Delix 5mg 1-0-1 Delix 5mg 0-0-1/2 Delix 5 mg 0-1-1/2

to three edit operations, i.e. insertion, deletion and substitution [14]. Table 4.8 shows selected examples of misspellings of drugs and their Damerau–Levenshtein distance to the actual name.

4.3.5.5 Proximity Search

For a special application we implemented a further query feature to extract the daily dosage of the medication taken by the patient. The extraction requires two pieces of information: the strength and the cumulative daily amount of the drug. The strength is given in digits using a standard unit (usually milligrams or micrograms) with the drug name. The dosing interval is usually coded by a number-hyphen notation like 1/2-0-1/2. The numbers represent the units that must be taken in the morning, at noon, and in the evening. A possible fourth digit refers to the number before going to bed. The daily dose is obtained by adding these three or four numbers and then multiplying by the strength. We added a feature that makes it easier to query the daily dose. The proximity query searches the given tokens next to each other. The order of these tokens is irrelevant. Proximity queries do not match across sentence boundaries. Since each medication instruction is provided in a segmented fashion as a single sentence during the import, proximity queries do not match dosage information of other medications. Table 4.9 shows an example of how a daily dose can be extracted. The corresponding request is displayed as well as matching and not matching text snippets. With this technique facilitates queries that assess different drug strengths and daily dosages.

5 Implementation of Ad Hoc Information Extraction

The methods and procedures, described in Chapter 4, are implemented in the PaDaWaN CDW, outlined in Section 3, in order to gain the ability to perform ad hoc Information Extraction. Text analysis functions, such as text sectioning, negation detection and the context detection of information, have to be created and integrated as well as query features. In addition, a powerful and convenient storage engine must be found, which facilitates that kind of requests.

The negation and context detection can be implemented in any CDW. The query features depend on the used storage engine in the CDW. The implementation of ad hoc IE query features in a DB-based data-warehouses are possible, but a DB offers limited functionality, which results in a limited number of query features. A search index based system offers more functionality and better fits the needs of ad hoc IE.

We implemented the ad hoc IE methods, described in Chapter 4 in PaDaWaN CDW, outlined in Chapter 3. The integration is described in Section 5.1. Apache Solr is used as index server. Its function and its configuration is characterized in Section 5.2. The implemented parts are described more in detail: The text processing in Section 5.3, the query parsing in Section 5.4 and result creation features in Section. 5.5.

5.1 CDW Integration

The ad hoc IE is implemented at two parts in the PaDaWaN system: Some *preprocessing* procedures are added in the ETL-process, more precisely in the indexing step in the PaDaWaN-CDW.

The second part of the ad hoc IE is implemented in the runtime system. This includes two tasks: The query features are integrated in the query creation process, the result features are implemented in the result creation process.

5.1.1 Integration in the ETL-Process

The sectioning (see Section 4.3.2) of discharge letters takes place at the end of the data import process into the DB. A discharge letter is divided into sections, each referencing a medical domain. These sections are stored as regular facts in the DB.

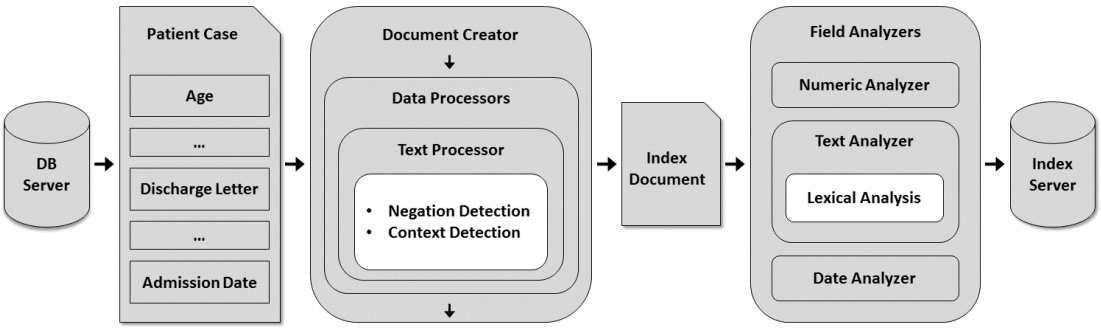


Figure 5.1: **Ad hoc IE integration in the indexing process in the PaDaWaN CDW.** Processes in white boxes are integrated in the existing system (grey color). The negation and context detection is performed in the text processor. Lexical analysis is done by the text field analyzer with NLP steps.

The text fragmentation, the negation detection and context detection are part of the indexing process (see Section 3.1.7). These algorithms are integrated into the *document creator*, which receives all information of a patient as a list and merges them in a single index document, which is stored in the index server: The document creator has three major sub-tasks: (1) the structure building, (2) the information propagation and (3) the data transformation by processors based on their data type. Ad hoc IE procedures are integrated into the text processor pipeline: The passed texts are segmented and divided into smaller pieces with the algorithms presented in Section 4.3.2.2. The output of this procedure are semantically closed (sub-)clauses. They are the input for the negation detection and context detection. These algorithms (described in Section 4.3.3 and 4.3.4) identify and remove all negated parts with their scopes. The remaining text with only affirmed findings and the original text are stored in fields as shown in Section 4.2.2.2.

A lexical analysis of this text is done by the field analyzers, especially by the text analyzer. It defines a pipeline for NLP steps that are applied as outlined in Section 4.3.1. Figure 5.1 presents the architecture.

5.1.2 Integration in the Query Process

The ad hoc IE query features (see Section 4.3.5) are integrated in the query process of the PaDaWaN CDW.

Query Creation Process. The features are integrated in the data model of the PaDaWaN and extend the Medical XML Query Language Example (MXQL) (see Section 3.1.10). The Web-GUI (see Section 3.2) offers input-options for the ad hoc IE. They are also integrated in the input syntax and can be used like other query features. The Web-GUI creates a query in MXQL-format and sends it to the REST-Server, which compiles it to a Solr Query (see Section 3.1.9). This is done by the *query parser*. Section 5.4 gives

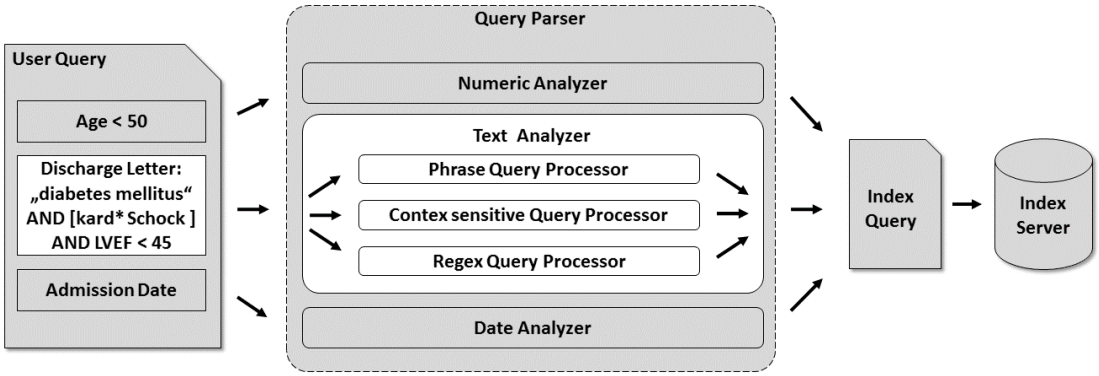


Figure 5.2: **Ad hoc IE integration in the query process in the PaDaWaN CDW.** The ad hoc IE query features are compiled to Solr query parts by the text analyzer. The query parser assembles these parts to a complete Solr query.

a closer look at the parsing process. Every attribute in the query is transformed to Solr query syntax with a special analyzer, depending on the data type of the attribute. Every query feature of the ad hoc IE is implemented in the text analyzer. The output of such a feature handler is a Solr query part. For example, a context-sensitive query (see Section 4.3.5.2) is transformed to surround query statements in Solr syntax, which is later processed by the Solr surround query parser[1]. Figure 5.2 shows the integration of the ad hoc IE features in the query process.

Result Creation. The PaDaWaN has two query modes: the *statistic query* mode and the *patient case query* mode. The statistic query mode (see Section 3.1.8) shows aggregated results and only returns the size of queried patient case groups. Hence, no implementations are necessary for the creation of the result in this query mode.

The patient case query mode (see Section 3.1.8) shows individual values of patients. The result output generator for the ad hoc IE query features is integrated here. In the PaDaWaN -System, a user query is processed and send to the Solr server, as explained in the previous section. The index server returns a response, which is received by the REST-Server. The *result creator* builds a user result based on this response. Each attribute in the user query is processed with a data type specific handler. The ad hoc IE implementations are integrated in the text result creator. The implementations comprise new features for the result creation. The input for this procedures are the queried texts of the patient and the output are e.g. highlighted text snippets as result for the user. They are described in detail in Section 5.5. Figure 5.3 pictures the ad hoc IE integration in the process of the result creation for patient case queries.

[1] `https://lucene.apache.org/solr/guide/7_3/other-parsers.html`, accessed: November 2018

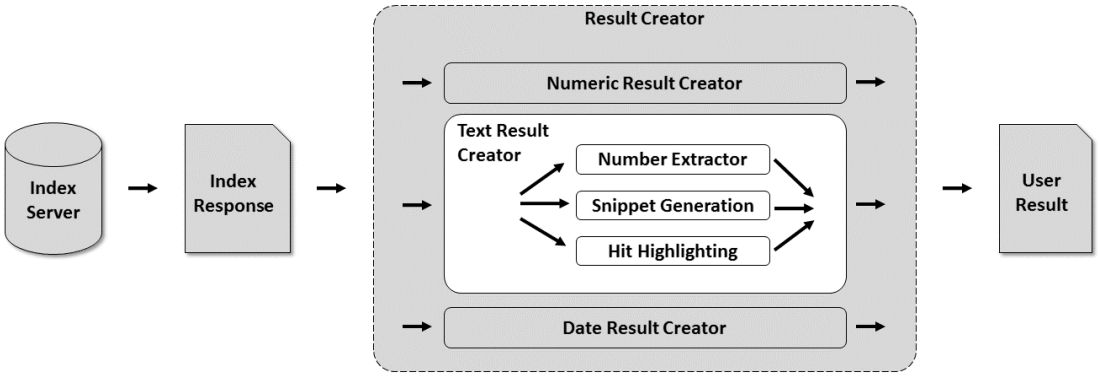


Figure 5.3: **Ad hoc IE integration in the result creation process in the PaDaWaN CDW.** The response of a Solr server is received by the REST-Server and a user result is generated. The text result creator produces the user result using the integrated ad hoc IE specific features, like the highlighting engine.

5.2 Apache Solr

An index server is a core element for ad hoc information extraction. It has several benefits for this use case:

- Index structures providing fast response times for queries
- Built-in NLP functions for text data
- Rich text query features
- Document oriented storage
- Scalability for big applications

We choose Apache Solr[2] as index server, which is one of the most popular enterprise search platforms and offers a lot more features: Some example are: (1) Standards open interfaces supporting XML, JSON and HTTP, which facilitates the integration. (2) Extensible plugin architecture, that allow the extension of index and query features. (3) Optimization for high volume traffic. Solr can handle a large scale of query and index requests. There is a long list of companies that use Apache Solr[3] including Apple (`apple.com`), Instagram (`instagram.com`), Netflix (`netflix.com`), Reddit (`reddit.com`) and others.

[2] `http://lucene.apache.org/solr/`, accessed: November 2018
[3] `https://lucidworks.com/2012/01/21/who-uses-lucenesolr/`, accessed: November 2018

Solr is build on top of the index library Apache Lucene[4]. Lucence is directly used (without the Solr framework) by many companies (e.g. Twitter[5] (`twitter.com`)) and web applications as well[6].

Documents, Fields, and Schema Design. All values of attributes, each representing a medical concept, are stored in *fields* in the index. All information (attributes with values) of a patient case are saved in a single document. There exist two document schemas in the PaDaWaN system. An extensive description of the document schemas and fields, used in the PaDaWaN CDW, is in Section 4.2.2.2.

The fields are mapped to Solr specific field types. The mapping is defined in a schema configuration file [74]. These field types have *analyzers* for *request handlers* or *search components*. An important request handler is the index procedure, a major search component is the query procedure. An analyzer in the Solr system is a pipeline containing a *tokenizer* and a list of *filters*. Text values of a field are processed with the corresponding analyzer according to its field type. Texts in queries are treated the same way as texts in the index process. Queries or sub-queries are mapped to fields and address the values in this field. For example: "valvular aortic stenosis" may be a sub-query and could be searched in the field of echocardiogram reports.

An example of a sub-query is the retrieval of a "valvular aortic stenosis" in an echocardiogram report. The tokens of a query are processed by their predefined analyzer. The query token, which can be seen as the value of a query, are processed in a similar style like the indexed texts, which are stored as the values of fields. This is necessary in order to ensure the match of query tokens to the same or similar tokens in the stored texts.

NLP pipeline. The PaDaWaN system contains several analyzers for text fields. They define a pipeline with some NLP operations. Listing 5 shows the definition in XML for a standard text field. The text is tokenized, lowercased, stop words filtered, normalized and stemmed.

Load Balancing. PaDaWaN uses the cloud mode of Apache Solr. It provides a highly available, fault tolerant environment for distributing the indexed content and query requests across multiple servers with automatic load balancing.[7] Solr can be started with one node or a fixed size of nodes and various others nodes can be added during runtime. Small PaDaWaN system can be run on one server with a single node or with multiple nodes on the same server. If the applications grows, the amount of index server can be increased easily.

[4] `http://lucene.apache.org/`, accessed: November 2018
[5] `https://blog.twitter.com/engineering/en_us/a/2011/twitter-search-is-now-3x-faster.html`, accessed: November 2018
[6] `https://wiki.apache.org/lucene-java/PoweredBy`, accessed: November 2018
[7] `https://lucene.apache.org/solr/guide/7_5/getting-started-with-solrcloud.html`

```
<fieldType name="text_de" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true"
    ↪  words="lang/stopwords_de.txt" format="snowball" />
    <filter class="solr.GermanNormalizationFilterFactory"/>
    <filter class="solr.GermanLightStemFilterFactory"/>
  </analyzer>
</fieldType>
```

Listing 5: **The index-analyzer pipeline of the default text field in Solr.** The text is tokenized first, before the tokens are lowercased. The next filter removes stop words, which are defined in a list. A normalization filter transforms special characters like umlauts (e.g. "äöü"). At the end, all tokens are stemmed with a less aggressive stemmer for the German language.

Parallelization. A Solr *collection* contains all documents of one system. That means in the PaDaWaN context, that all patient cases, which are represented by documents, are grouped together in one collection. This collection can be organized in multiple pieces, called *shards* that can be hosted on multiple nodes. The amount of shards of a collection, determines the amount of parallelization that is possible for an individual search request [74]. The default configuration of PaDaWaN is two nodes with one shard per node.

Cross Segment Search. Texts, like discharge letters, are transformed before they are stored in the index as depicted in Section 4.3.1. Discharge letters are split into sections in a first step (see Section 4.3.2.1). This is not necessary for findings reports, because their content only addresses one medical topic. The following procedure is the negation detection and the context detection, which is applied to all texts (see Sections 4.3.3 and 4.3.4). The input for this procedures is a segmented text. The segmentation algorithms splits the text in sub clauses (see Section 4.3.2.2). To obtain a text with affirmed findings only, the negated parts within the segments are removed. The original text and the modified text without negations are both stored in the index. While original text is stored as plain text, the text with the affirmed findings is stored as a list of segments.

Figure 4.1 shows an example for the steps described above. Figure 5.4a represents the input text, here: a transthoracic echocardiogram. Figure 5.4b is the corresponding subtext after the negation detection. It consists of list of segments with only affirmed findings. Both texts are stored in Solr fields. Figure 5.4c shows a part of an input document for Solr. (JSON was chosen as syntax language in this example. Others languages, that structure a document, like XML, are accepted by Solr as well.)

A normal token text query with multiple words, searches these words in an entire text. These words can appear at any position in the text. For other requests, a semantic relationship of the query words may be desired. A segment provides such a semantic unit,

Linker Ventrikel normal weit und normal kontraktil, keine regionalen Wandbewegungsstörungen, auch keine erkennbare HW-Narbe, Auswurffraktion normal, EF planimetrisch 64 %. grenzwertige LV-Hypertrophie, Septum edd 10 mm, diastolische Relaxationsstörung Grad II

(a) Normal text (transthoracic echocardiogram).

Linker Ventrikel normal weit und normal kontraktil,
Auswurffraktion normal, EF planimetrisch 64 %. grenzwertige LV-Hypertrophie,
Septum edd 10 mm, diastolische Relaxationsstörung Grad II

(b) Segmented text with removed negated findings.

```
{
  "normal_text" : "Linker Ventrikel normal weit und normal kontraktil, keine regionalen
  ↪  Wandbewegungsstörungen, auch keine erkennbare HW-Narbe, Auswurffraktion normal, EF
  ↪  planimetrisch 64 %. grenzwertige LV-Hypertrophie, Septum edd 10 mm, diastolische
  ↪  Relaxationsstörung Grad II",

  "affirmed_findings" : [
        "Linker Ventrikel normal weit und normal kontraktil:",
        "Auswurffraktion normal",
        "EF planimetrisch 64 %",
        "grenzwertige LV-Hypertrophie",
        "eptum edd 10 mm",
        "diastolische Relaxationsstörung Grad II"]
}
```

(c) Resulting solr document with both texts, represented in JSON.

Figure 5.4: **Representation of a segmented text in a Solr document.** A normal text (a) (here: a transthoracic echocardiogram) is segmented and the negation detection is applied. Both texts, the original and the text with removed negations are stored in the Solr index: As normal text and as a list of affirmed findings.

```
[7= (komp | reduzierte | eingeschränkte | verminderte)* (link| sys)* ventr* funkti* ] OR
↪  [7= (komp | reduzierte | eingeschränkte | verminderte)* lv funkti* ] OR [7= ventrik*
↪  (komp| reduzierte | eingeschränkte | verminderte)* funkti* ] OR [5= sys* dysfunkti*]
```

Figure 5.5: **Syntax example of an ad hoc IE query.** The extracted concept is a "reduced left ventricular function" of the heart in German texts. Four context sensitive queries are combined with `OR`s. Some of them have a inner logical structure defined with round brackets.

per definition. Therefore, some requests only search within a segment. This is ensured by a parameter which prevents cross-segment search. It is called `positionIncrementGap` and specifies a distance between the segments. It it set to an high value, so that surround queries cannot match across diverse segments. Listing 5 shows an example of a definition of this parameter. It is specified for the element `fieldtype` and is set 100. This means that the position index of tokens is incremented by 100 after each segment. This position index is used to calculate the distance between tokens by the Solr surround-query-parser, which is used by the context sensitive query feature of the ad hoc IE.

5.3 Text Processing Implementation

The ad hoc IE system contains some text processing steps, which are executed at preprocessing time, such as the sectioning, segmenting, negation detection and context detection. All these algorithms are implemented in Java using the Apache UIMA[8] (Unstructured Information Management Architecture) framework. It is the state of the art framework for text processing. Annotations on text parts can be created in the CAS (Common Analysis System). Their possible types and features are defined in a type system.

5.4 Query Parsing

Multiple ad hoc IE features can be combined to search a concept in a text. For example, various synonyms of a concept can be queried at once. Each alternative formulation can be defined in a sub-query. These query parts can use different query features like wildcard queries, phrase queries, context context sensitive queries, etc. The sub queries can be combined to a normal query. For more details of the various query features, see Sections 4.3.5.

Figure 5.5 shows an example of several context sensitive queries that are combined with `OR`s.

[8] `https://uima.apache.org/`, accessed November 2018

These statements are parsed with the text analyzer in the query parser. Figure 5.2 pictures this process. The parsing of the query language is done with ANTLR[9] (ANother Tool for Language Recognition). Lexical and grammar rules are defined in ANTLR-syntax that describe the query language. The rules are compiled with ANTL to a parser program. This program is used to parse and process the user queries.

5.5 Result Creation Features

The patient case query searches and presents individual values of patients in a result view. For example, if a laboratory value is queried, the value of the measurement is returned. This can be done straightforward for numeric, date or Boolean queries. Text queries have to be treated in a special way. An entire text of an attribute (e.g. a discharge letter) can be quite long and the result tables can be confusing. As a solution for this problem, only text snippets are returned, that contain the queried words or tokens. These tokens are highlighted to increase the readability and to allow a quick recognition of the relevant information. A custom snipped creation and highlight engine was developed for this use case. Figure 5.6 shows an example of an ad hoc IE query with the corresponding result view.

[9] `http://www.antlr.org/`, accessed November 2018

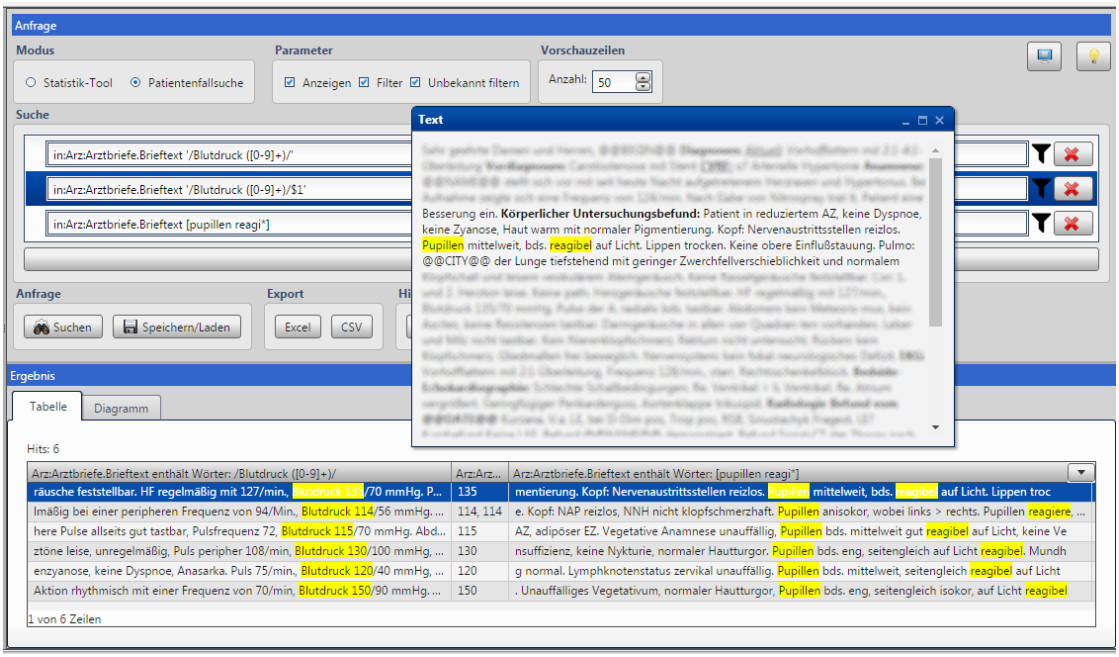


Figure 5.6: **Ad hoc IE queries in the PaDaWaN with result presentation.** The query defines three lines: The first two lines query "Blutdruck" (engl. *blood pressure*) and its numeric value. The first line has no explicit output definition, so the matched snippet is returned. The second line extracts the number of the concept. That can be seen in the second column of the result view on the bottom of the picture. The third line is a context sensitive query that contains the two terms "Pupillen reagi*" (engl. *pupils reacti**). These two terms must occur next to each other in the text. The wildcard "*" matches any character sequence. The generated hit snippets are listed in the third column of the result view. It can be seen that the two tokens have a close distance and refer to each other. The order of terms is not specified here. In addition to the snippet, the complete document can be shown as well. A double-click on the desired snippet opens the entire text of the underling attribute. In this example, it is a discharge letter. The highlighting engine emphasize the matched query terms, too.

6 Experiments & Evaluations[1]

The methods and implementations of ad hoc information extraction (IE) presented in Chapter 4 and 5, are evaluated in this section. We conduct several comprehensive tests to ensure the quality and the efficiency of the system. Therefore we carry out experiments for the whole system as well as for the sub-components.

All tests are performed with real patient data in order to achieve real world findings instead of results of artificial laboratory experiments. For the evaluations, texts are randomly selected out of the PaDaWaN CDW, installed at the UKW. To protect privacy, the texts are de-identified and in addition must not leave the clinical network.

This chapter starts with an overview of the most important evaluation results in Section 6.1. The following parts present the several experiments in detail. Section 6.2 describes the evaluation of the negation detection. The functionality of ad hoc IE is tested in Section 6.3 and 6.4. The experiments are carried out for Boolean and numeric concepts. Section 6.3 examines the accuracy and Section 6.4 evaluates the speed and efficiency.

The discussion and the comparison of the results are conducted in the subsequent chapter (see Chapter 7).

6.1 Overview

Table 6.1 summarizes the important results: Negated medical concepts are found with an F1 score between 0.96 and 0.99. The result varies on the text type (findings report or discharge letter). If a medical concept is negated is determined by whether the concept is covered in a negation scope. These scopes are found with an F1 score of 0.986 and their length is determined to 91% exactly. The accuracy of ad hoc IE is between 0.987 and 1 for Boolean concepts and between 0.991 and 1 for numeric concepts. The runtimes for assessing the existence of Boolean concepts with ad hoc IE is less than half a second. Numeric concepts are extracted with and without constraints (e.g. LEVEF < 45) between 15 seconds and one minute. Runtimes are tested with one million discharge letters.

[1] This chapter is based on previously published work:
[55], G. Dietrich, J. Krebs, G. Fette, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Ad hoc information extraction for clinical data warehouses. *Methods of information in medicine*, 57(01):e22–e29, 2018

Table 6.1: **Overview of the evaluation results.** The main results of the various experiments are listed. Runtimes are measured on one million texts. The exact experimental setup is explained in the corresponding section.

Evaluation	Result
Finding negated medical concept	F1: 0.96 - 0.99
Finding negated scopes	F1: 0.986
Exact computed length of negated scope	91%
Accuracy of Boolean ad hoc IE	F1: 0.987 - 1
Accuracy of numeric ad hoc IE	F1: 0.991 - 1
Runtime for Boolean baseline	14 sec - 33 min
Runtime for Boolean ad hoc IE	0.02 sec - 0.41 sec
Runtime for numeric baseline	15 sec - 35 min
Runtime for numeric ad hoc IE	8 sec - 61 sec

6.2 Negation Detection

A major component for ad hoc IE in clinical texts is a well performing negation detection. This section evaluates the algorithm described in Section 4.3.3.

6.2.1 Experimental Setup

Document Selection. For the evaluation of the negation detection experiments, we take two different domains. The first one is chest X-ray reports which have a rather telegraphic style text structure with short sentences, mostly containing noun phrases. In addition to chest X-ray reports, we use a second domain with a more complex sentence structure and a larger vocabulary, i.e. discharge letters.

Gold Standard Creation. We create a manually annotated gold standard of 100 reports: First, the texts are automatically pre-annotated to save time, using an information-extraction terminology created by physicians [126]. Afterwards these texts are manually corrected in the ATHEN[2] environment to annotate the data and to achieve the gold standard.

We also create a gold standard for 50 discharge letters. The procedure is similar to the chest X-ray gold standard, but because we have no terminology for entire discharge letters, we try to identify findings and medical concepts in the texts. Therefore we take the German list of Alpha-ID[3], a list with more than 80,000 diagnoses, and the German version of MeSH (Medical Subject Headings)[4], a list with more than 60,000 medical

[2] `http://www.is.informatik.uni-wuerzburg.de/research_tools_download/athen/`, accessed: June 2017

[3] `https://www.dimdi.de/static/de/klassi/alpha-id/`, accessed: June 2017

[4] `https://www.dimdi.de/static/de/klassi/mesh_umls/mesh/`, accessed: Jone 2017

Table 6.2: **Performance of the negation detection.** Medical concepts are evaluated in the two domains.

	Chest X-ray	Discharge letter
Documents	100	50
Negations	619	397
True positives	608	366
False positives	1	1
False negatives	11	31
Precision	0.998	0.997
Recall	0.982	0.922
F1	0.990	0.958

concepts. Additionally, we use a part-of-speech tagger [200] to label all named entities and nouns with no lemma, i.e. nouns which are unknown to the tagger. These are technical terms of a specific domain, in this case: mostly medical concepts. Next, we use our negation detection algorithm to label negation trigger and their span. This pre-annotated data is again corrected manually to obtain a gold standard.

Procedure. The negation detection algorithm described in Section 4.3.3, detects and labels all negated scopes. Every concept covered by a negation scope is considered as negated. The retrieval of negated concepts is a common evaluation task in the literature, which is why our results can be directly compared and discussed.

Evaluation measure. This standard measure of retrieval tasks used in the literature is the precision, recall and the combined F1 score that we take as well for our tests.

6.2.2 Evaluation Results

The evaluation shows good results for detecting negated scopes and negated entities.

Negated Entities: The F1-score for the negation detection is 0.99 points in the telegraphic-style chest X-ray reports and 0.96 in the more complex discharge letters. Table 6.2 shows the detailed results of the evaluation of the negation detection of medical concepts. (TP = true positives, FP = false positives, FN = false negatives).

While the precision with just one false positive in each domain is high, the recall contains several false negatives, which refer to different error sources. In 67% of all errors the negation triggers are not contained in the trigger set. This is the main problem for discharge letters especially (77% of its errors). Due to the natural flow of speech the variety of the negation trigger is much greater than in the chest X-ray reports. This explains the difference in the overall performance (F1-scores) between the two domains:

Table 6.3: **Error analysis of misclassified concepts in the negation detection.** Errors are grouped into clusters.

	Chest X-ray reports	Discharge letter	Combined
Sentence splitting	3 (25%)	2 (6%)	5 (12%)
Wrong documentation	1 (8%)	0 (0%)	1 (2%)
Complex sentence structure	3 (25%)	5 (16%)	8 (19%)
Missing negation triggers	5 (42%)	24 (77%)	29 (67%)

Table 6.4: **Performance of retrieval of negated scopes.** The detection of scopes are measured in discharge letters. TP: true positives, FP: false positives, FN: false negatives

TP	FP	FN	Precision	Recall	F1
348	4	6	0.989	0.983	0.986

chest X-ray 0.99, discharge letter 0.96. Table 6.3 summarizes a categorization of the error sources.

The ten most common missing negation triggers are: "kein Anhalt für" (engl. *no clue for*), "nicht erforderlich" (engl. *not mandatory*), "nicht angegeben" (engl. *not specified*), "keine Notwendigkeit" (engl. *no need*), "nicht anzuraten" (engl. *not recommended*), "nicht bekannt" (engl. *not known*), "nicht mehr nachweisbar" (engl. *no longer detectable*), "nicht tastbar" (engl. *not palpable*), "traten nicht mehr auf" (engl. *did not occur anymore*), "keine Indikation" (engl. *no indication*).

Scope of Negations: In the previous paragraph, the detection of medical concepts within a negation scope is evaluated. This paragraph focuses on the negation scopes themselves. The retrieval and the computation of the length of the scope is tested.

The negated scopes are detected with a F1-score of 0.97. The exact length is determined in 91% of all cases. Table 6.4 shows the detailed results of the retrieval of the negated scopes: 348 are found correctly, six are not found and four scopes are found by mistake. Table 6.5 presents the results of the determination of their length in the discharge letter domain.

Table 6.5: **Performance of scopes length computation of the negated scopes.** The correct computation of scope length are measured in discharge letters.

Scope Length	Exact	Too Narrow	Too Wide
Number	318	2	28
Percent	0.91	0.01	0.08

Table 6.6: **Error analysis of wrongly determined negation scope in discharge letters.** Errors are summarized in the most common error classes.

	Number	Percent
Sentence splitting	8	0.28
Documentation fault	3	0.10
Complex sentence structure	5	0.17
Wrong labeling of filler words	6	0.21
Other errors	7	0.24

Table 6.7: **Queries for accuracy tests for Boolean ad IE.** The concepts are extracted with context sensitive queries, defined by "[ ]". Wildcards "*" at the end of a word match different formulations.

Concept	Query
Mild mitral insufficiency	`[Leicht* Mitral*insuffizienz]`
High mitral insufficiency	`[Hoch* Mitral*insuffizienz]`
Mild aortic stenosis	`[Leicht* Aortensten*]`

Many errors by processing chest X-ray report are made by splitting the sentence at wrong positions. The reasons are: Some abbreviations are unknown and the corresponding period is misinterpreted. Furthermore enumerations are not recognized and also some filler words like "and", "too" or commas are mistakenly included at the end of negations scope. However, these are no severe problems and they can be fixed quickly in the algorithm by changing the rule labeling to exclude such filler words from the scopes.

Table 6.6 summarizes a categorization of the error sources.

6.3 Accuracy of Ad Hoc Information Extraction

The accuracy of Boolean and numeric ad hoc IE is tested in this section.

6.3.1 Experimental Setup

The ad hoc IE is evaluated in two domains: echocardiogram reports and discharge letters. We randomly pick 1000 texts from each domain. The ad hoc IE queries are run in the PaDaWaN-system and the results are manually evaluated. The inputs for the Boolean ad hoc IE are medical concepts and their synonyms written in query syntax. All queries are presented in Table 6.8.

Similar, regular expressions describing the medical concept and the value to be extracted are the inputs for the numeric ad hoc information extraction. All queries are defined in Table 6.8.

Table 6.8: **Queries for accuracy tests for numeric ad hoc IE.** Numerical concepts are extracted with regular expression queries.

Concept	Query
Left ventricular ejection fraction	`/LVEF [A-Za-z.=<() ]{0,23}ZAHL/`
Body-mass-index	`/BMI ZAHL/`

Cholesterol	`/Cholesterin ; CHOL ; ZAHL/`
	`/Cholesterin: ([a-zA-Z ]+)? ZAHL/`
	`/(Chol|CHOL|chol)([.;])? ZAHL mg/`

Glucose	`/Glucose: ZAHL/`
	`/Glucose([a-zA-Z ]+)? ZAHL mg/`
	`/Glucose ; GLUC ; ZAHL)/`
	`/(Gluc|GLUC|gluc)[:; ]? ZAHL mg/`

Table 6.9: **Performance of Boolean ad hoc information extraction.** The context sensitive query feature is used for the medical concepts: (1) mild mitral insufficiency, (2) high mitral insufficiency and (3) mild aortic stenosis. Concepts are extracted from echocardiography reports. The corresponding queries are stated in Table 6.7.

		FP	FN	TP	Recall	Precision	F1
1	**Leichtgradige Mitralinsuffizienz**	0	7	304	0.977	1	0.987
2	**Hochgradige Mitralinsuffizienz**	0	0	14	1	1	1
3	**Leichtgradige Aortenstenose**	0	3	160	0.982	1	0.991

The PaDaWaN-system uses an Apache Solr 7.0 server and it is run with one instance, two nodes and two shards. Caching is disabled during the tests to measure the execution times.

6.3.2 Evaluation Results

Boolean Ad Hoc Information Extraction The evaluation of the Boolean ad hoc IE in the heart echo documents shows excellent results with a F1-score between 0.98 and 1 (see Table 6.9). Three concepts with modifiers are queried in 1000 chest echocardiography reports. Every query contained synonyms and wildcards to match the concepts in the texts. The context sensitive query feature is used to ensure that the queried tokens relate to each other. All errors refer to an incorrect sentence splitting in the preprocessing because the some findings are divided into different segments. The average processing time is 72 ms to query the hit count and 2.8 s to export all extracted information.

Table 6.10: **Performance of numeric ad hoc information extraction.** The regex query feature is used for the medical concepts: (1) Cholesterol, (2) Glucose, (3) BMI and (4) LVEF. Concepts are extracted from echocardiography reports (echo) and discharge letters (letter). The corresponding queries are stated in Table 6.8.

		Dataset	FP	FN	TP	Recall	Precision	F1
1	**Cholesterin**	letter	0	2	158	0.988	1	0.994
2	**Glucose**	letter	0	6	336	0.982	1	0.991
3	**BMI**	letter	0	0	44	1	1	1
4	**LVEF**	ehco	6	0	452	1	0.987	0.993

Numeric Ad Hoc Information Extraction: Table 6.10 shows the results of numeric ad hoc information using the regex query feature with examples from two datasets. Some regex-queries (cholesterol, glucose, age) contained synonyms of the concepts and all queries accepted multiple notations. All F1-scores are above 0.99.

The error analysis revealed that the eight false negatives of the first four concepts in Table 6.10 are caused by an incorrect sentence splitting in the preprocessing, while the six false positives result from incorrect recognition of intervals instead of single numbers.

The average time is 1.1 s to query the number of hits and 1.1 s to export all extracted information.

Beside classic findings of examination, an attempt is made to extract the concept "age". However, extracting the age of a person is not necessary in practice as it is accessible as structured data. The extraction of this concept is a difficult task, since the age can be stated in different formulations, e.g. "35 Jahre alter Patient" (engl. *35 years old patient / patient, aged 32*), "Alter des Patienten: 35", (engl. *Age of the patient: 35*). The most common unit of age is "years", however, many mentions of age lack of any unit. But, some other concepts use the unit "years" as well, for example events in the past or in the future. This leads to the risk that these numbers are mistakenly extracted, resulting in a high number of FPs.

We achieved a high recall, but a low precision (see Table 6.11, error analysis see Table 6.12). 97% of the errors refer to the wrong context for extracting the concept "age" in the discharge letter, as it can be seen in Table 6.12. The errors are subdivided in four parts: age in the patient history ("First occurrence at the age of 30 years."), age in family history ("The grandmother died with 87 years."), relative years in the patient history "5 years ago"), relative years to future events ("next examination in 2 years").

Table 6.11: **Performance of extracting the numeric concept.** The regex query feature is used to perform ad hoc IE for the numeric concept "age". Concepts are extracted from echocardiography reports (echo) and discharge letters (letter).

Concept	Dataset	FP	FN	TP	Recall	Precision	F1
Age	letter	136	4	49	0.93	0.27	0.41

Table 6.12: **Error analysis of ad hoc information extraction of the concept age.** The concepts are searched in the entire discharge letter.

Error group	Number of errors	Percentage
Age in patient history	18	0.13
Age in family history	15	0.11
Relative years in patient history	81	0.60
Relative years to future events	18	0.13
Unexpected syntax	4	0.03

6.4 Efficiency of Ad Hoc Information Extraction

Information extraction includes several issues: For many application is the first task the terminology/ontology creation or learning. If a new concept is introduced and shall be extracted, several tasks have to be accomplished: The extraction knowledge must be engineered in the system. This is done by defining extraction rules in a rule based system. In a learning system, data must be labeled and the model has to be trained. After that, in both systems the extraction step itself must be performed.

The goal of our work is that this procedure is much faster than this conventional approach. So fast, that user requests are answered ad hoc with a short processing time. Ad hoc IE is defined as "extracting the existence of any concept (e.g. chronic kidney disease) or any number (e.g. the LVEF value) from a source in real-time" (Definition 4.0.1). The efficiency and the speed of ad hoc IE is measured in this section with the aim to check if it meets these requirements. The Boolean and the numeric ad hoc IE are measured with various queries in two different domains and with a different amount of texts. Furthermore, a baseline is defined and evaluated. The speed of the ad hoc IE is improved compared to the accuracy tests in the previous section by implementing the export functions more efficient.

6.4.1 Overview

Table 6.13 gives a brief overview of the most important results. The ad hoc IE queries extract the information drastically faster than the corresponding regular expressions. SQL functions only support a limited amount of query types. The `CONTAINS` function returns results in a few seconds. However ad hoc IE outperforms all other alternatives.

Table 6.13: **Overview of results of efficiency of ad hoc information extraction.** Concepts are extracted of one million discharge letters. If a technique does not provide a functionality the corresponding cell is marked with "×". Concepts of query-types: *Tokens*: NYHA, *Tokens with wildcard*: pulmora edema, *Context sensitive with wildcard*: Normal sized left ventricle, *Context sensitive with wildcard*: Mild mitral insufficiency, *Numeric IE*: BMI (< 25). For detail of experimental setting see Section 6.4.2.

Query Type	Ad hoc IE	Regex	SQL Like	SQL Contains
Boolean ad hoc IE				
Tokens	0.03 s	14.7 s	151.9 s	1.2 s
Tokens with wildcard	0.03 s	36.9 s	×	×
Context sensitive	0.11 s	31.5 min	×	3.4 s
Context sensitive with wildcard	0.06 s	> 6 h	×	×
Numeric ad hoc IE				
Value	2.0 s	14.0 s	×	×
Constrained Value	2.3 s	14.7 s	×	×

6.4.2 Experimental Setting

This subsection describes the experimental setup of ad hoc IE and baseline evaluations. For each test, we describe the document selection and the extracted concepts with their query tokens.

6.4.2.1 Ad Hoc Information Extraction

General Setup. The evaluation is conducted at the Würzburg University Hospital (UKW) with real patient data. Patient information are pseudonymized and the texts are anonymized. Patients with special texts data (discharge letters or echocardiogram reports) are selected and indexed with the PaDaWaN pipeline, including segmentation and negation detection (see Section 5.1.1). Ad hoc IE queries are defined and executed. The duration of the query, starting with request and ending with the returned result, is measured. Each query is executed in two modes: (1) The query extracts the information and returns the number of matches. (2) The extracted values are returned in a CSV file. Extracted information of the numeric ad hoc IE are numbers and text snippets with highlighted match tokens for the Boolean ad hoc IE. The following example shows a result snippet of a query extracting myocardial infarctions. The words "Verschluss" and "RCA" match the context sensitive query "[RCA Verschlus]" and are highlighted with guillemet "«»"

> Infarkt bei hochgradiger Stenose sowie »Verschluss« von Rcx und »RCA«.

Table 6.14: **Queries for runtime tests for Boolean ad IE in echocardiogram reports.** The concepts are extracted with context sensitive queries, defined by "[ ]". Wildcards "*" at the end of a word match different formulations.

Concept	Query
Normal Sized Left Ventricle	`[normal* gro* link* Ventrik*]`
Normal Sized Right Ventricle	`[normal* gro* recht* Ventrik*]`
Mild aortic stenosis	`[Leicht* Aortensten*]`
Moderate aortic stenosis	`[Mittel* Aortensten*]`
High aortic stenosis	`[Hoch* Aortensten*]`
Mild mitral insufficiency	`[Leicht* Mitral*insuffizienz]`
Moderate mitral insufficiency	`[Mittel* Mitral*insuffizienz]`
High mitral insufficiency	`[Hoch* Mitral*insuffizienz]`

Document Selection. Texts from two domains are selected for this evaluation: Echocardiogram reports, which have a telegraphic style with short sentences, mostly containing noun phrases, and discharge letters, that are long and comprehensive documents with different sections and different sentence structures. They contain finding reports, descriptions, explanations and other content.

The PaDaWaN CDW in the UKW contains just over 100,000 echocardiogram reports and a bit more than 1,000,000 discharge letters. We determine two sample sizes for two evaluation runs: We selected 100,000 texts of each domain for the first iteration. The second run is conducted with 1,000,000 discharge letters.

The average length of echocardiogram reports is 1,833 characters, whereas discharge letters are almost twice as long with 3,137 characters on average.

Hardware Setting. The evaluation is carried out on a server with an Intel Xeon E5 3.3 GHz processor with 32 GB RAM.

Solr Configuration. A Solr server in the version 7.2 is used with one node and eight shards. We turned of all caching, including the filter cache, query result cache, document cache and the block join cache. Additionally the `queryResultMaxDocsCached` parameter is set to 0.

Queries. Boolean and numeric concepts are extracted from texts of each domain. Frequently and less common occurring concepts are chosen.

Table 6.14 shows the concepts for Boolean ad IE. The concepts are extracted from echocardiogram reports with context sensitive queries. Different word endings, formulation and synonyms are covered by wildcards "*". For example the query "Mitral*insuffizienz" matches the words "Mitralinsuffizienz" and "Mitralklappeninsuffizienz".

Table 6.15 lists the concepts for the Boolean ad IE in discharge letters. The list contains two diagnosis appendicitis and myocardial infarction. Furthermore, findings and characteristics of heart failure are used. They are taken from the case study "Prevalence of heart failure in hospital inpatients" (see Section 8.2). The miscellaneous characteristics of heart failure are queried separately and they are combined in one query named "any heart failure characteristic".

Table 6.15: **Queries for runtime tests for Boolean ad IE in discharge letters.** Characteristics and sub-findings of heart failure are selected, in addition to the diagnosis appendicitis and myocardial infarction. HF: Heart failure, NYHA: New York Heart Association, LV/RV: left/right ventricular

Concept	**Query**
Appendicitis	`appendix OR appendizitis`
Myocardial Infarction	`I21* OR Hebungs-Infarkt OR Hebungsinfarkt* OR Hebungsmyokardinfarkt* OR Herzinfarkt* OR Hinterwandinfarkt* OR Myocardinfarkt* OR Myokardinfarkt* OR Nstemi OR Stemi OR Vorderwandinfarkt* OR Hauptstammstenose* OR Hauptstammverschluss* OR [Verschl* RIVA] OR [Verschl* RCA]`
Heart Failure Characteristics	
Cardiac Decompensation	`(card* OR kard* OR kardiopulmo* OR cardiopulmo* OR hydrop* OR herz* OR link) AND dek*`
Diastolic Dysfunction	`((komp* OR eingeschr* OR vermind*) AND (ventr* OR dias*) AND funkti*) OR (dias* AND (dysfunkti* OR relax*stör*))`
Dilated Cardiomyopathy	`(dilat* AND (kardiomy* OR cardiomy*)) OR dcm`
Heart Failure	`herzschw* OR herzinsuff*`
Left Atrial Enlargement	`(link*vorho* OR link*atri* OR la) AND (verg* OR dilat* OR hypertr)`
Left HF	`(kard* OR linksherz*) AND insuff*`
Left Ventricular Hypertrophy	`(lv hypertr*) OR (link*ventr* hypertr*)`
NYHA	`nyha`
Pulmonary Edema	`lung*ödem* OR lung*stau* OR stau*lung*`
Reduced LV Function	`((komp* OR reduzierte* OR eingeschränkte* OR verminderte*) AND (link* OR sys*) AND ventr* funkti*) OR ((komp* OR reduzierte* OR eingeschränkte* OR verminderte*) AND lv funkti*) OR (entrik* AND (komp* OR reduzierte* OR eingeschränkte* OR verminderte*) AND funkti*) OR (sys* dysfunkti*)`

Reduced RV Function	`((komp* OR reduzierte* OR eingeschränkte* OR verminderte*) AND (rechts* OR dias*) AND ventr* funkti*) OR ((komp* OR reduzierte* OR eingeschränkte* OR verminderte*) AND rv funkti*) OR (dias* dysfunkti*)`
Right HF	`(rechtsherz* OR diast*) AND insuff*`
Systolic-Failure	`((card* OR kard* OR cardiopulmo* OR kardiopulmo* OR hydrop* OR herz*) AND pumpvers* OR vorwärtsversag*) OR ((kard* OR card*) AND schock*)`

Table 6.16 lists the numeric concepts and the corresponding query text. Two queries are created for every concept: The first one extracts the existence of the numeric concept in the text. The second query has a numeric filter and selects abnormal values only. The used thresholds are defined in the last column. For example, the body mass index (BMI) is defined as abnormal if its value is bigger than 25. The first query returns all mentions of the concept BMI (name and value) in the text and second one returns all mentions with a value bigger than 25.

6.4.2.2 Baselines

The first baseline for the ad hoc IE are SQL-queries that request a SQL-server. There are several relational database systems on the market. We selected one as a reference system for comparisons. In order to obtain a high significance, the tests are carried out on real patient data. Since the PaDaWaN is implemented and in productive use at the Würzburg University Hospital (UKW), we chose their DB system, that is supported by their service center (SMI): Microsoft (MS) SQL Server[5]. It is the second most popular DB-system after MySQL[6] according to an considerable developer survey [216]. MS SQL offers some possibilities to query texts. In addition to the *equals*-operator, which requires that the entire text must be equal to the requested text, the *like* operator allows wildcards (%) queries like "%appendicitis%". The wildcard can be matched with arbitrary fragments of a character string.[7] Other placeholder represent a single random character. However, quantifiers cannot be defined.

MS SQL server offers the option to install a full-text search component. This features is available in advanced editions, not in the basic *Express* edition.[8] It contains two *predicates* to query words in texts. The *CONTAINS*-function searches a single word

[5] `https://www.microsoft.com/en-us/sql-server/`, accessed: October 2018
[6] `https://www.mysql.com/`, accessed: October 2018
[7] `https://docs.microsoft.com/en-us/sql/t-sql/language-elements/like-transact-sql?view=sql-server-2017`, accessed: October 2018
[8] `https://docs.microsoft.com/en-us/sql/t-sql/language-elements/like-transact-sql?view=sql-server-2017`, accessed: October 2018

or a phrase in the text. An exact match is required. The *FREETEXT*-function is a fuzzy search. It accepts several words or a small text as input. The queried string is divided into tokens, which are stemmed afterwards. For each token, a list of expansions or replacements is identified, taken of a thesaurus. They are added to the query tokens. The query returns a result of texts that may match the query or may be in semantic relation to it. This fuzzy semantic search returns texts that may be relevant, however it is no exact search with hard filters and constraints.

The CONTAINS-predicate can search in addition to a single word, multiple words with an *NEAR*-operator. The tokens must occur anywhere in the text, per default. A maximum distance of the tokens to each other can be specified optionally, as well as the order of the terms. The queried value can be words, phrases or prefix-terms. A prefix-term is matched with any term in the text that start with the specified sequence. For example the query "ware*" would match the word "warehouse" [39].

However, no MS SQL query function allows the usage of regex statements or wildcard operators within a word with quantifiers. So the simple query "Mitral*insuffizienz" that matches the words "Mitralinsuffizienz" and "Mitralklappeninsuffizienz", cannot be answered by any MS SQL query function.

Hence, a SQL server cannot be used as a baseline for all query types that are provided with ad hoc IE. Because it cannot answer some queries types and does not provide the required functionality.

SQL Baseline. We carry out a baseline test with SQL-queries. As explained above, not all query types can be tested, but some basic queries can be performed with SQL. The concept "high aortic stenosis" is mapped to SQL queries. However they are not semantically identical and return other results as explained above. Table 6.17 shows the SQL statements.

Regex Baseline. We use a search with regular expressions as a second baseline, which is functionally equivalent with our search mechanisms. We created regular expressions to search the concepts in the texts.

Table 6.18 lists the concepts and the corresponding regular expressions. Context sensitive queries are translated to a special regular expression, called *near*-regex. For example, the concept "high aortic stenosis" with the query "`[Hoch* Mitral*insuffizienz]`" is mapped to a regex that defines, that the two tokens must occur next to each other. A maximum of six words may be placed between them. Because that expression is quite time consuming, alternative formulations are evaluated as well. The *sequence*-regex specifies that the words must appear one after the other only separated by a blank space. The third version just requires that both words appear anywhere in the text. The *sequence*-regex and *anywhere*-regex can be evaluated faster than the *near*-regex, but they are less equivalent to a context sensitive query. The *near*-regex is similar to context sensitive query, but it matches across sentence boundaries and does not ensure the same

Table 6.16: **Queries for runtime tests for numeric ad hoc IE.** Numerical concepts are extracted with the according queries. The existence of the concept is counted and extracted in a first query. The abnormal values are returned in a second independent query. *LVDD*: left ventricular diastolic dysfunction, *LVEF* left ventricular ejection fraction, *LVMI*: left ventricular mass index, *BMI*: body-mass-index

Concept	Query	abnormal
Aortic Root	`/Ao-root(= )? ZAHL/`	`> 40`
LVDD	`/LVDD(= )? ZAHL/`	`> 54`
LVEF	`/LVEF [A-Za-z.=<() ]{0,23}ZAHL/`	`< 45`
LVMI	`/LVMI(= )? ZAHL/`	`< 45`
BMI	`/BMI ZAHL/`	`> 25`

Cholesterol	`/Cholesterin ; CHOL ; ZAHL/`
	`/Cholesterin: ([a-zA-Z ]+)? ZAHL/`
	`/(Chol|CHOL|chol)([.;])? ZAHL mg/`
		`> 220`

Glucose	`/Glucose: ZAHL/`
	`/Glucose([a-zA-Z ]+)? ZAHL mg/`
	`/Glucose ; GLUC ; ZAHL)/`
	`/(Gluc|GLUC|gluc)[:; ]? ZAHL mg/`
		`> 106`

Table 6.17: **SQL-queries of the baseline tests for ad hoc IE runtime evaluation.** Some queries are translated to SQL `LIKE` and SQL `CONTAINS` statements. *Myo. Infac.*: Myocardial Infarction, *NYHA*: New York Heart Association, *Mitral Insuf.*: High Mitral Insufficiency, *Left Ventricle:* Normal Sized Left Ventricle

Concept	Query text
Like	
Myo. Infac.	`LIKE ('%Myokardinfarkt%' OR '%Nstemi%' OR '%Stemi%' OR '%Hebungsinfarkt%' OR '%Vorderwandinfarkt%' ...)`
NYHA	`LIKE ('%NYHA%')`
Contains	
Mitral Insuf.	`CONTAINS(i.value,'NEAR ("Hoch*", Mitralinsuffizienz)')`
Left Ventricle	`CONTAINS(i.value,'NEAR ("normal*", "gro*", "link*","Ventrik*")')`

Table 6.18: **Regular expressions of the baseline tests for ad hoc IE runtime evaluation.** Queries of Boolean and numeric concepts are converted to regular expressions. Context sensitive queries are transformed to three different versions. Used predefined character classes: word characters (`\w`), word boundaries (`\b`) and letter.

Concept	Regular Expression
High mitral insufficiency	
Near regex	`(\b(Hoch\w*)\b(\p{L}+\b){0,6}(Mitral\w*insuffizienz\w*)\b)` `|(\b(Mitral\w*insuffizienz\w*)\b+(\p{L}+\b){0,6}(Hoch\w*)\b)`
Sequence	`Hoch\w*\s+Mitral\w*insuffizienz\w*` `Mitral\w*insuffizienz\w*\s+Hoch\w*`
Anywhere	`Hoch\w*` and `Mitral\w*insuffizienz\w*`
Myocardial infarction	`(I21\w*)|(Hebungs-Infarkt)|(Hebungsinfarkt\w*)` `|(Hebungsmyokardinfarkt\w*)|(Herzinfarkt\w*)` `|(Hinterwandinfarkt\w*)|(Myocardinfarkt\w*)` `|(Myokardinfarkt\w*)|(Nstemi)|(Stemi)` `|(Vorderwandinfarkt\w*)|(Hauptstammstenose\w*)` `|(Hauptstammverschluss\w*)` `|Riva\s+Verschl\w*|Verschl\w*\s+Riva` `|RCA\s+Verschl\w*|Verschl\w*\s+RCA`
Appendicitis	`(Appendix\w*)|(Appendizitis\w*)`
BMI	`BMI\s+([0-9]+(,[0-9]+)?)`
BMI abnormal	`BMI\s+(2[6-9])|([3-9][0-9]{1,10})|([1-9][0-9]{2,10})`

Table 6.19: **Match examples of the baseline regular expressions.** The concept to be extracted is "high mitral insufficiency". The actual occurrence and the machetes of the three regular expression versions are marked with an "✓".

Text	Actual Match	Query-Match Near	Sequence	Anywhere
Hochgradige Mitralinsuffizienz	✓	✓	✓	✓
Hochgradige und chronishce Mitralinsuffizienz	✓	✓		✓
Hochgradige Aortenstenose, leichtgradige Mitralinsuffizienz				✓

context. The used predefined character classes in the regular expression represent word characters (`\w`), word boundaries (`\b`) and letters (`\p{L}`).[9] Table 6.19 gives an example how the various regular expressions match example text pieces.

The tests are carried out with discharge letters and echocardiogram reports in this procedure: The regular expression are loaded and compiled to patterns. Then the patterns are searched in four different ways:

1. The texts are sequentially processed: Every text is loaded and the pattern is searched in the text.
2. The texts are processed in parallel. The loading time and the process time is measured.
3. All texts are pre-loaded first and then sequentially processed. The loading time is not included in the measurement.
4. All texts are pre-loaded first and then processed in parallel. Again, only the processing time is counted.

Comparability of results: The regular expressions and the SQL queries do not take negated findings into account. They do not consider historical of information of other persons. This circumstance is neglected for this runtime test and not taken into account.

Ad hoc IE and the baselines (SQL and regular expression) do not extract the identical information. However, the deviations are not relevant for these runtime tests and therefore the results are comparable.

6.4.3 Evaluation Results

The results of the baseline tests are presented first, followed by the ad hoc IE results.

6.4.3.1 Baseline Results

Table 6.20 shows the result of the regular expression baseline. Several findings can be deduced: The *near*-regex are very inefficient. The *sequence*-regex is up to ten times faster. If the terms can be located anywhere in the texts, the runtime is shortened again. However, they are not semantically equivalent to the *near*-query. The runtime increases if the regular expression contains more words. NYHA is up to eight times faster than myocardial infarction. The execution time also increases with the number of processed texts in a linear manner.

Loading the texts is a very time consuming task. If the texts are processed sequentially, loading takes four times as long as searching the regular expressions. In parallel, it is factor eleven, for one million texts even 25.

[9] `https://www.regular-expressions.info/unicode.html`, accessed: November 2018

If the regular expressions are searched in parallel, the computing time is massively reduced. The process is 7.4 (texts are loaded from disk) times respectively 8.0 (texts are loaded from disk) times faster, which is almost the number of cores in the CPU (8). Loading the text from RAM instead of disk has a big impact as well: The process is between 3.2 and 4.4 times faster, if the texts are taken from RAM. Table 6.21 summarizes the impact of parallel processing.

Table 6.22 presents the execution times of the SQL queries. The `CONTAINS` functions uses a full text index and is dramatically faster than the `LIKE` functions that scans all texts. The runtime of the `LIKE` function strongly increases with the number of texts to be searched, whereas it only goes up slightly with the `CONTAINS` function.

6.4.3.2 Boolean Ad Hoc Information Extraction

Table 6.23 lists the results for the Boolean ad hoc IE in 100,000 echocardiogram reports. Each concept is queried separately and in addition, some concepts are combined to a query. The count queries, assessing the existence of a concept, are all answered in less than half a second. The queries containing multiple concept are processed in a few milliseconds, too. This applies for the flat data model as well as for the hierarchical model.

The export times in the hierarchical data model are bigger in general. However, the export of about 1000 extracted information is faster than one second in the flat model and about one second on average for the hierarchical model. The export times increase with the number of extracted information. All facts (up to 76,000) are exported in less than 30 seconds.

Table 6.24 presents the result of the Boolean ad hoc IE query in discharge letters. All queries, counting the existence of a concepts, are processed within milliseconds. This applies for single and multiple concepts in both data structures (flat and hierarchical). The runtimes for exporting all extracted information in a CSV table are longer than the queries that only count their existence. The export times hardly differ in the two data schemes, but the flat structure is a bit faster. The export duration increases with the number of exported information. While about 1000 pieces of information are extracted and exported in about a second, it takes over a minute for several tens of thousands. Loosely speaking, the export time is about one second for one thousand facts, if one concept is queried. The duration is also extended with the number concepts that are contained in a query. The query “any heart failure characteristic” contains 13 concepts as previously mentioned. Each concepts is extracted and exported in a separate column.

6.4.3.3 Numeric Ad Hoc Information Extraction

Table 6.25 presents the runtime results of the numeric ad hoc IE. The existence of numerical concepts is ascertained in about one second in 100,000 texts by count-queries. That numeric count-query works faster in echocardiogram reports. An obvious reason

Table 6.20: **Result of the regular expressions baseline for the ad hoc IE runtime evaluation.** Numerical and Boolean concepts are searched with regular expressions in echocardiogram reports (echo) and discharge letters (letter). Context sensitive queries ("High mitral insufficiency" and "Cardiac-Decompensation") are translated and searched with three different regular expressions (regex). Runtimes are specified in seconds.

		100,000 text documents				**1,000,000 text documents**			
Texts Storage		Disk	Disk	RAM	RAM	Disk	Disk	RAM	RAM
Processing		Sequential	Parallel	Sequential	Parallel	Sequential	Parallel	Sequential	Parallel
Numerical Concepts									
Aortic root	echo	122.8	17.7	1.5	1.0				
Aortic root abnormal	echo	124.6	13.0	4.1	1.2				
Body mass index	letter	121.9	16.7	6.2	1.2	1215.6	1643.4	50.0	14.0
Body mass index abnormal	letter	118.4	15.6	5.3	1.4	1307.1	2109.5	50.1	14.7
Boolean Concepts									
High mitral insufficiency									
Near-regex	echo	314.7	22.1	140.7	12.6				
Sequence-regex	echo	132.6	18.3	10.3	1.7				
Anywhere-regex	echo	123.3	13.1	3.7	1.2				
Mild mitral insufficiency									
Near-regex	letter	377.7	27.7	121.6	18.6	> 6 h	> 6 h	> 6 h	> 6 h
Sequence-regex	letter	149.1	12.1	16.4	3.4	1558.4	176.3	196.8	28.0
Anywhere-regex	letter	661.9	10.5	6.2	1.2	1312.0	171.7	78.1	10.9
Normal sized left ventricle									
Near-regex	letter	6.2 h	5.0 m	28.3 m	4.7 m	6.3 h	46.4 m	4.9 h	31.5 m
Sequence-regex	letter	157.2	13.9	21.9	4.6	1699.7	188.6	290.7	35.8
Anywhere-regex	letter	129.2	11.0	6.0	1.8	1323.3	159.7	86.6	14.2
Appendicitis	letter	140.2	17.0	21.5	3.0	1504.7	1953.5	220.9	32.9
Myocardial infarction	letter	266.3	21.4	111.5	11.6	3073.9	1879.8	1156.7	129.7
NYHA	letter	119.5	13.4	5.0	1.4	1360.8	2053.0	65.1	14.7
Pulmora edema	letter	156.7	13.2	21.4	4.6	1634.4	198.0	272.4	36.9

Table 6.21: **Runtime improvement for regular expressions baseline tests.** The runtime reduction is calculated for loading the text from RAM instead of loading them from disk in the left part of the table. In the right part, the speed-up is shown for processing the text parallel instead of sequential. Processing the texts in parallel is 8 times faster on average than sequential, if texts are taken from RAM. *Echo*: Echocardiogram reports, *Letter*: discharge letter

		RAM Storage			**Parallel Processing**		
		Processing			Text Storage		
Domain	Texts	Sequential	Parallel	AVG	Disk	RAM	AVG
Echo	100,000	5.1	4.7	4.9	9.7	9.0	9.4
Letter	100,000	2.4	1.4	1.9	10.0	6.1	8.0
Letter	1,000,000	2.0	6.9	4.5	2.5	8.7	5.6
Average		3.2	4.4		7.4	8.0	

Table 6.22: **Runtime result of the SQL baseline.** SQL baseline statements are executed with the two SQL functions `LIKE` and `CONTAINS` on echocardiogram reports (echo) and discharge letters (letter). Runtimes are specified in seconds.

Concept	**Domain**	**Number of Texts**	**LIKE** [min]	**CONTAINS** [sec]
High mitral insufficiency	echo	100,000	-	0.463
Myocardial infarction	letter	100,000	2.48	1.046
Normal sized left ventricle	letter	100,000	-	2.285
NYHA	letter	100,000	12 s	0.556
Myocardial infarction	letter	1,000,000	35.7	1.648
Normal sized left ventricle	letter	1,000,000	-	3.402
NYHA	letter	1,000,000	2.52	1.207

Table 6.23: **Runtime results of Boolean ad hoc IE in echocardiogram reports.** Queries that count or export the occurrences of concepts in texts are measured. Concepts are requested separately and in combination. The flat and the hierarchical data structure are evaluated. Runtimes are specified in seconds. Number of texts: 100,000

Concept	Hits	Flat		Hierarchical	
		Count-Time	Export-Time	Count-Time	Export-Time
Single Concept					
Mild aortic stenosis	1,119	0.17	0.89	0.03	1.50
Moderate aortic stenosis	1,049	0.16	0.45	0.03	1.33
High aortic stenosis	1,160	0.33	1.79	0.28	4.45
Mild mitral insufficiency	25,555	0.11	6.47	0.03	10.25
Moderate mitral insufficiency	4,560	0.05	1.94	0.03	2.56
High mitral insufficiency	1,039	0.13	0.52	0.02	0.97
Normal sized left ventricle	65,425	0.27	18.66	0.13	25.12
Normal sized right ventricle	47,736	0.09	12.70	0.11	19.42
Multiple Concept					
Any aortic stenosis	3,165	0.08	1.56	0.06	2.75
Any mitral insufficiency	29,430	0.08	7.49	0.11	13.67
High mitral insufficiency & high aortic stenosis	30	0.05	0.08	0.05	0.16
Normal sized left and right ventricle	36,986	0.13	10.09	0.17	15.96
Normal sized left or right ventricle	76,175	0.25	19.13	0.17	27.88

Table 6.24: **Runtime results for Boolean ad hoc IE in discharge letters.** Queries that count or export the occurrences of concepts in texts are measured in a flat and a hierarchical data structure. Concepts are requested separately and in combination. Runtimes are specified in seconds. *HF*: Heart failure, *LV/RV*: left/right ventricular

	100,000 text documents					**1,000,000 text documents**				
		Flat		**Hierarchical**			**Flat**		**Hierarchical**	
	Hits	Count-Time	Export-Time	Count-Time	Export-Time	Hits	Count-Time	Export-Time	Count-Time	Export-Time
Single Concept										
Mild mitral insufficiency	1,537	0.05	1.60	0.03	2.02	13,511	0.04	12.56	0.06	9.08
Normal sized left ventricle	65,425	0.25	19.08	0.13	21.37	65,426	0.14	20.93	0.11	22.62
Appendicitis	573	0.06	0.44	0.03	0.86	5,396	0.16	4.54	0.41	5.73
Myocardial infarction	4,664	0.17	4.59	0.05	5.23	45,519	0.27	46.89	0.25	70.65
Cardiac Decompensation	3,575	0.17	2.79	0.05	3.18	29,965	0.22	24.54	0.25	59.11
Diastolic-Dysfunction	4,044	0.11	4.17	0.03	4.10	35,076	0.16	34.27	0.19	75.05
Dilated Cardiomyopathy	823	0.06	0.76	0.02	1.03	7,162	0.05	6.76	0.05	6.35
Heart failure	2,505	0.03	1.98	0.02	1.45	23,884	0.03	14.62	0.05	28.80
Left Atrial Enlargement	2,785	0.05	2.36	0.03	2.50	25,349	0.08	16.41	0.06	35.91
Left Heart Failure	1,175	0.03	1.06	0.03	1.17	8,985	0.03	7.07	0.03	10.62
Left Ventricular Hypertrophy	795	0.05	0.75	0.03	0.81	7,949	0.05	6.19	0.05	7.88
New York Heart Association	1,829	0.03	1.64	0.02	1.29	15,891	0.03	10.12	0.03	14.68
Pulmonary Edema	28	0.05	0.13	0.02	0.19	292	0.05	0.42	0.03	0.42
Reduced LV-Function	2,641	0.05	5.52	0.05	5.16	23,707	0.08	45.94	0.09	55.82
Reduced RV Function	1,935	0.06	3.40	0.03	3.56	16,529	0.06	28.52	0.08	28.50
Right Heart Failure	730	0.03	0.84	0.02	0.97	5,468	0.03	5.16	0.03	5.79
Systolic Failure	423	0.05	0.59	0.02	0.76	3,494	0.05	2.95	0.06	2.90
Multiple Concepts										
Myocac. infar. & Card. Decom.	527	0.06	1.22	0.05	1.67	4,496	0.13	8.05	0.13	12.71
Any HF Characteristic	10,344	0.28	66.96	0.20	59.00	93,159	0.31	897.59	0.31	722.02

is that the length of discharge letters as they have more content. The flat and the hierarchical data structures show now big differences. The query times for multiple concepts are between two and three seconds.

The runtimes for identifying and counting concepts increases with the number of analyzed texts. It takes between two and five seconds in 1,000,000 texts for a single concept and about eight seconds for multiple concepts. The elapsed time is independent of the number of hits. It is also not influenced if the value of the concept is constrained or not. It solely depends on the amount of analyzed text and the number of queried concepts.

The export of the extracted information takes between a few seconds and two minutes, depending on the document structure: The hierarchical data structure is much slower than the flat one. The speed also depends on the number of extracted information: 1,000 pieces of information are exported in a CSV-file in about 0.5 seconds using the flat data structure and about one second for the hierarchical structure.

Table 6.25: **Runtime results for numeric ad hoc IE.** Queries assess the existence of numerical concepts with any value or abnormal values only in echocardiogram reports (echo) and discharge letters (letter). Queries that count or export the mentions of concepts are measured in a flat and a hierarchical data structure. Runtimes are specified in seconds. *BMI*: body-mass-index, *LVDD*: left ventricular diastolic dysfunction, *LVEF*: left ventricular ejection fraction, *LVMI*: left ventricular mass index, *HF*: heart failure

		100,000 documents					**1,000,000 documents**				
			Flat		**Hierarchical**			**Flat**		**Hierarchical**	
Concept	Text	Hits	Count Time	Export Time	Count Time	Export Time	Hits	Count Time	Export Time	Count Time	Export Time
Single Concept											
BMI	letter	3,029	0.64	1.54	0.52	2.92	25,943	2.03	13.81	2.04	18,49
BMI abnormal	letter	1,910	0.59	1.01	0.44	1.98	16,916	1.92	8.18	2.34	12.46
Cholesterol	letter	4,925	2.29	4.32	1.37	9.75	55,027	4.13	20.50	3.90	52.93
Cholesterol abnormal	letter	475	2.23	1.97	1.31	3.07	9,064	4,39	6.52	4.40	15.18
Glucose	letter	17,263	1.75	8.71	1.28	16.97	166,053	3.90	61.43	4.23	114.33
Glucose abnormal	letter	7,385	1.53	4.54	1.31	12.17	70,149	4.20	32.48	4.38	52.96
Aortic root	echo	72,509	0.73	12.70	0.34	23.24					
Aortic root abnormal	echo	3,846	0.34	1.37	0.28	2.82					
LVDD	echo	80,844	0.36	13.45	0.37	24.12					
LVDD abnormal	echo	15,680	0.34	3.42	0.30	6.54					
LVEF	echo	67,413	1.26	11.37	1.19	28.94					
LVEF abnormal	echo	7,726	1.48	3.34	1.36	13.34					
LVMI	echo	31,372	0.19	6.65	0.27	12.00					
LVMI abnormal	echo	4,585	0.20	1.26	0.25	3.23					
Multiple Concepts											
Ao-r. & LVEF: abnorm.	echo	559	1.67	2.81	1.67	5.34					
Any HF Char. or abn. LVEF	both	17,683	1.72	68.64	1.58	66.57					
Gluc. & Chol: abnorm.	letter	130	2.75	3.65	2.08	7.02	1,529	7.99	8.66	8.11	17.13

7 Discussion[1]

This chapter summarizes the results and discusses the proposed approach of ad hoc IE. It starts with a recap of the benefits in Section 7.1. The evaluation results are compared to related worked reported in literature in Section 7.2. Section 7.3 discusses the approaches of conventional IE and ad hoc IE in general and characterizes their similarities and differences. Section 7.4 presents the novel features of ad hoc IE compared to the existing CDWs systems. Section 7.5 shows the limitations and Section 7.6 concludes this chapter with the gained insights and possible improvements.

7.1 Benefits of Ad Hoc Information Extraction

This section starts with a brief recap of the current situation summarizing the state of the art: Information extraction algorithms work well to extract structured data from unstructured data, such as texts in natural language. However, the number of clinical domains with an existing IE system is severely limited. Building new IE system is a very complex and laborious work requiring difficult resources such as ontologies. The features of current CDWs to query texts are poor. This gap is addressed with the introduced novel technique *ad hoc information extraction* that comes with several benefits and advantages.

Extraction of concepts ad hoc. Ad hoc IE supports the extraction of information from text ad hoc. The existence of Boolean and numeric concepts can be assessed (yes/no). Furthermore numeric concepts can be constrained ("LVEF < 45") as well. Ad hoc means that the extraction takes place at runtime and returns the result in milliseconds for most cases.

Autonomous working of clinicians. The main advantage is that clinicians can use the ad hoc IE independently and make evaluations on their own obtaining good results immediately. This enables fast and autonomous working, since no engineer is required.

[1] This chapter contains minor section of previously published work:
[55], G. Dietrich, J. Krebs, G. Fette, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Ad hoc information extraction for clinical data warehouses. *Methods of information in medicine*, 57(01):e22–e29, 2018
[56], G. Dietrich, J. Krebs, L. Liman, G. Fette, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Replicating medication trend studies using ad hoc information extraction in a clinical data warehouse. *BMC medical informatics and decision making*, 19(1):15, 2019

Interactive and Flexibley. The presented approach does not only counts the occurrences of the extracted information, it furthermore is able to display extracted values in a clear and fast readable way with tabular structure. Paired with the fast response time, this method enables interactive working. The description of the concepts to be extracted can interactively be refined to increase the extraction result.

Terminologies are not required. Unlike conventional information extraction, ad hoc IE does not require a terminology. Users of a data warehouse are usually clinicians and physicians who possess the required knowledge (technical terms used in the texts). The inclusion of these experts makes the process very adaptive for different domains, especially since these experts are familiar with respective domain and the text corpus with its characteristics. This makes ad hoc IE particularly attractive for the many areas where no terminology exists.

Extendable for further applications The introduced approach is not a closed system, but it is open to extensions, such as the presented extraction of the daily intaken drugs (see Section 8.1).

Easy to use. The easy-to-use interface of PaDaWaN CDW offers a Goolge-like interactive style. PaDaWaN supports an implementation of ad hoc IE with various features that facilitates a comfortable, interactive usage.

7.2 Comparison of Evaluation Results

This section compares the results of the evaluations of the negation detection and the ad hoc IE system with similar results in the literature.

7.2.1 Negation Detection

The negation detection performs very well in both domains: F1-score of 0.99 in Chest X-ray and 0.91 in discharge letters. Hence, our approach is slightly better as Cotik et al, who achieves a F1-score of 0.96 on clinical notes and 0.91 on discharge letters (see Table 7.1) [42].

The computation of the length of a negation scope performs well, too. In 91% of all detected negations scopes, the length of the scope are determined correctly. That score is lower, but keep in mind, that the F1-score of negated concepts is quite good. So some of these miscalculated scopes do not contain relevant information or clinical concepts. In fact, the determination is a difficult task, Gros and Stede could only compute in 54% the exact scope in medical texts (see Table 7.2) [84].

Table 7.1: **Comparison of scope retrieval results to related work.** F1-scores of the negation detection are compared to other approaches using German clinical texts.

Data set	Cotik et al.	Our approach
Discharge letter	0.91	0.96
Clinical notes	0.96	–
Chest X-ray	–	0.99

Table 7.2: **Comparison of results of the scope length determination to related work.** F1-scores are compared other approaches using German clinical texts.

	Gros and Stede	Our approach
Data set	cardiology report	discharge letters
Exact	0.54	0.91
Too narrow	0.34	0.01
Too wide	0.12	0.08

7.2.2 Ad Hoc Information Extraction

The accuracy of Boolean and numeric ad hoc IE and the speed of ad hoc IE is discussed.

Boolean Ad Hoc Information Extraction. The extraction for Boolean values using the context sensitive query works well. The text is split up into sentences and logical parts and negations are removed at preprocessing time. Afterwards queries can be run against the index. The tokens of a query must match the tokens in one sentence. Wildcards in the query tokens match many variants of word spellings. That simple mechanism is a very powerful tool. Even the span-limitation-feature between the words is often not necessary. The F1-scores between 0.99 and 1.0 confirm that approach.

Numeric Ad Hoc Information Extraction. The regular expression queries for the numeric ad hoc information extraction provides very good results as well. But they show the limitation of that approach, too. The extraction of the desired values works fine, but the context must be clear. If the concept always refers to the patient, the regex query is a powerful feature as well, which extracts values with a F1-score bigger than 0.99.

Speed of Ad Hoc Information Extraction. Ad hoc IE outperforms the presented baselines in terms of speed by far. Ad hoc IE extracts Boolean concepts in milliseconds and numeric concepts in a few seconds (< 3 sec). The SQL `LIKE` baseline for Boolean concepts is slower by several orders of magnitude (152 sec) and the SQL `CONTAINS` baseline returns results in a few seconds but on average more than 30 times slower than ad hoc IE. In addition, SQL queries support a very limited amount of ad hoc IE features.

Statements can be created with regular expressions that approximate the functionality of ad hoc IE, but do not reach it. The elapsed time for extracting Boolean concepts ranges from 15 seconds to more than 6 hours. Ad hoc IE accomplishes this task in milliseconds. The difference of the systems for extracting values of numeric concepts is not as drastic as for extracting Boolean concepts. Ad hoc IE process takes 2 seconds and is seven times faster than regular expressions, which finishes in 14 seconds.

7.3 Conventional Versus Ad Hoc Information Extraction

This section compares the approaches of conventional IE and ad hoc IE in general.

7.3.1 Conventional Information Extraction

IE turns unstructured information embedded in texts into structured data [110]. More precisely, it is the automatic extraction of concepts, entities and events, as well as their relations and associated attributes [243]. It consists of subtasks, i.e. entity recognition, relation extraction, event extraction (including time and date), and template filling [110]. In a conventional IE application, information are computed by many expensive processing steps [197]. Therefore, each text is annotated several times, e.g. with part of speech tagging, syntactic or dependency parsing or word list labeling. The output of the tagging process is the input for the next step. Two major approaches exist to extract information: Rule-based systems apply rules on these annotations to extract information. Machine learning approaches use a trained model for the extraction step.

7.3.2 Ad Hoc Information Extraction

In ad hoc IE, a segmentation step separates non-related concepts. On these segments, a one-step annotation can be made effectively. This step is quite fast, due to the index, and in contrast to the conventional IE, there are not ”many expensive processing steps” [197]. Thus, ad hoc IE is suitable for domains that can be handled with a one-step annotation. A survey revealed that 65% of clinical information extraction systems are rule-based and often use a regular expression as a search pattern [243]. Hence, they are interesting for ad hoc IE and could possibly be implemented with it. Ad hoc IE shifts the time of extraction from the data-integration phase to runtime, enabling a flexible IE at runtime for all users.

Ad hoc IE does not address all sub-tasks of a conventional IE application. However, the tasks important to the medical domain are supported: Named entity recognition is ensured by the query functions, relation extraction for medical concepts is accomplished by segmentation and for patient identification by context detection.

Table 7.3: Comparison between ad hoc information extraction and conventional IE.

	Ad hoc IE	**Conventional IE**
Scope	specific concept	entire domain
Development effort	low	high
Engineer required	no	yes
Promptness	fast	slow
Adaptability by user	yes	no
Accuracy	lower	higher
Evaluation results	no	yes

7.3.3 Comparison

Ad hoc IE has several advantages in comparison to conventional IE, but also some shortcomings. The main advantage is that clinicians can use the ad hoc IE independently and make evaluations on their own obtaining good results immediately. This enables fast and autonomous working, since no engineer is required. A big difference is the development effort, which is very low for ad hoc IE (just entering the concept with its variants) and high for conventional IE requiring the definition of a terminology and learning or engineering the extraction patterns. Other advantages are the low development effort, the promptness of the results and the adaptability by the user to his or her particular questions.

The main disadvantage is that the accuracy of the results is usually lower and there are no evaluation results available resulting in a lower confidence. Table 7.3 summarizes the comparison. It would be attractive to integrate concepts from the ad hoc IE into the permanent part of the CDW by enriching its catalog of concepts.

Another disadvantage of conventional IE in the medical domain in German language is that is suffers from the availability of fewer resources than IE in English [218].

7.4 Query Features of Other CDWs

Text query features are poorly supported in CDWs [55]. Most of them, like the well-known i2b2, store their data in SQL-DBs and just support the *like*-operator[2] a SQL full text index.

Texts can be queried via the `LIKE`-operator, which is used to perform wildcard queries. However, this is limited in many ways: Error tolerant queries, which deal with misspellings, are barely supported. Medical concepts (e.g. drug names) that consist of several words are difficult or cumbersome to find with SQL methods. Especially, if these words are not next to each other and, e.g., separated by a brand name. SQL functions only support a limited amount of query types used for ad hoc IE. Table 7.4 is a summary of Table 6.13 and illustrates the restricted capability of SQL for ad hoc IE.

[2] `http://community.i2b2.org/wiki/display/DevForum/Text+search+in+i2b2`

Table 7.4: **Support of SQL for ad hoc IE features.** Summary of Table 6.13. Concepts are extracted of one million discharge letters. If a technique does not provide a functionality the corresponding cell is marked with "×". MS SQL is uses as SQL reference. For detail of experimental setting see Section 6.4.2 Computation times are given in seconds.

		SQL	
Query Type	**Ad hoc IE**	**Like**	**Contains**
Boolean ad hoc IE			
Tokens	0.03	151.911	1.207
Tokens with wildcard	0.03	×	×
Context sensitive	0.11	×	3.402
Context sensitive with wildcard	0.06	×	×
Numeric ad hoc IE			
Value	2.0	×	×
Constrained Value	2.3	×	×

Other CDWs index their textual data with index libraries as Apache Solr (e.g. tranSMART [92] or Roogle [44]) or with SQL full text index (e.g. STRIDE[143]). Dr. Warehouse is the most similar CDW compared to our approach and to PaDaWaN . It uses a search engine to store text data as well and performs a negation and context detection. Texts are split up in sentences, which are classified whether they are negated, historical or are describe family history [80]. Our approach splits texts in sentences and further in smaller segments. These fine grained segments are semantically closed clauses. The negation and context detection is performed within this segments, achieving a more specific classification. Dr. Warehouse makes no statements about the accuracy of their system. Therefore, the approaches can only be compared in their structure.

However, no CDW system except Dr. Warehouse has query features that exceed a token search. Queries are matched with documents, if query tokens appear at any position in the document. The semantic relationship of tokens (e.g. "high mitral insufficiency") is not ensured. Our approach facilitates context sensitive queries by searing in semantically closed clauses and with control of the span between the tokens and their order.

No CDW supports any feature to query or to extract numeric concepts from texts. Our approach (integrated in the PaDaWaN CDW) facilitates the extraction of numeric concepts and offers valuable features: (1) the existence can be assessed, (2) their values can be constrained (e.g. "BMI" < 25) and (3) their values are extracted, displayed and exported for further analysis.

7.5 Limitations

Ad hoc IE does not address all fields of information extraction, which comprises the extraction of concepts, entities and events, as well as their relations and associated

attributes [243]. Ad hoc IE is not suited for all types of relation extraction. For example, the coreference resolution is not supported in the current version. The described one-step-annotation-function is inappropriate for complex and multi-layered issues.

However, ad hoc IE supports the tasks important to the medical domain: Named entity recognition is ensured by the query functions, relation extraction is accomplished by segmentation for medical concepts and relation extraction is ensured by context detection for the patient identification.

As mentioned above, ad hoc IE will usually have a lower accuracy of the results than conventional IE. Moreover no evaluation results are available in everyday use, resulting in a lower confidence.

However, ad hoc IE does not claim to replace conventional IE, it rather should be considered a supplement for quick analysis to get a good and detailed overview for further investigations.

7.6 Gained Insights and Possible Improvements

Ad hoc IE proved to be a useful and reliable tool in the all three case studies (see Chapter 8). Ad hoc IE requires as user input a concept name to be extracted and alternative formulations (includes synonyms and homonyms) that may occur in the text. The completeness of these terms has a direct influence on the quality of extraction. For the medication trend replication study, we used the ATC dictionary that contained all necessary drug names. For the other two studies, we had to determine some of the terms ourselves. The gathering of terms can be quite time-consuming, especially when this is done by people who are less familiar with the domain than physicians. That issue could be addressed by the inclusion of conventional dictionaries or lists, such as MeSH or Alpha-ID. They could be integrated in the user interface to support and simplify the query process. An approach could aim to identify concepts in the entered user input and try to map these concepts to entries in dictionaries (e.g. Alpha-ID). Synonyms and related terms could be suggested and selected by the user. A more progressive approach could automatically add synonyms and homonyms to the request as query expansion. These terms should ensure a high precision and could increase the recall in order to raise the quality of the entire extraction.

8 Case Studies

Three real-world case studies using ad hoc information extraction are conducted at the Würzburg University Hospital. Several medical trend studies are replicated using ad hoc IE and the medical findings are matched with the original publications in Section 8.1. Section 8.2 presents a case study assessing the prevalence of heart failure in hospital inpatients via ad hoc IE from discharge letters and ascertains an underestimation in contrast to the encoded ICD-10 diagnoses. The consistency of diagnoses (ICD-10 encoded and described in discharge letter) are checked with ad hoc IE in Section 8.3 A method is developed to extract any diagnosis from discharge letters by generating synonyms.

8.1 Replication of Medication Trend Studies

This case study replicates medical trend studies that show the changes of medication over the years. The medication usage data is gained with ad hoc IE using the PaDaWaN CDW installed at the Würzburg University Hospital.

This section contains a previously published article [56]:
G. Dietrich, J. Krebs, L. Liman, G. Fette, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Replicating medication trend studies using ad hoc information extraction in a clinical data warehouse. *BMC medical informatics and decision making*, 19(1):15, 2019[1]

8.1.1 Summary

Background Medication trend studies show the changes of medication over the years and may be replicated using a clinical Data Warehouse (CDW). Even nowadays, a lot of the patient information, like medication data, in the EHR is stored in the format of free text. As the conventional approach of information extraction (IE) demands a high developmental effort, we used ad hoc IE instead. This technique queries information and extracts it on the fly from texts contained in the CDW.

Methods We present a generalizable approach of ad hoc IE for pharmacotherapy (medications and their daily dosage) presented in hospital discharge letters. We added import and query features to the CDW system, like error tolerant queries to deal with

[1]The article is published with the Creative Commons Attribution 4.0 International License (`https://creativecommons.org/licenses/by/4.0/`), minor changes are made.

misspellings and proximity search for the extraction of the daily dosage. During the data integration process in the CDW, negated, historical and non-patient context data are filtered. For the replication studies, we used a drug list grouped by ATC (Anatomical Therapeutic Chemical Classification System) codes as input for queries to the CDW.

Results We achieve an F1 score of 0.983 (precision 0.997, recall 0.970) for extracting medication from discharge letters and an F1 score of 0.974 (precision 0.977, recall 0.972) for extracting the dosage. We replicated three published medical trend studies for hypertension, atrial fibrillation and chronic kidney disease. Overall, 93% of the main findings could be replicated, 68% of sub-findings, and 75% of all findings. One study could be completely replicated with all main and sub-findings.

Conclusion A novel approach for ad hoc IE is presented. It is very suitable for basic medical texts like discharge letters and finding reports. Ad hoc IE is by definition more limited than conventional IE and does not claim to replace it, but it substantially exceeds the search capabilities of many CDWs and it is convenient to conduct replication studies fast and with high quality.

8.1.2 Background

Reliable information on the use of medication in a hospital and its changes over time is of great importance for many acute and chronic diseases – from a hospital, patient and payor perspective. This is reflected by many studies reporting medication trends: e.g. attention deficit hyperactivity disorder (ADHD) [253], atrial fibrillation (AF) (US [65], Denmark [79, 217]), chronic kidney disease (CKD) [122, 245], rheumatoid disease [114] or hypertension (HT) [23] (England [64], France [82], Germany [198], Sweden [241], US [85, 206]).

However, medical research (like many other disciplines) is affected by the so called replication crisis, addressed in an article in 2012 reporting that only 11% of the pre-clinical cancer studies could be replicated [15]. The Nature Journal conducted a survey of 1,500 scientists in 2016, in which 70% of them stated that they had failed to reproduce another scientist's experiment [13].

The ability to reproduce findings reported in a clinical study is a cornerstone of scientific progress. Replication of medication trend studies can be performed using a CDW, which is an important, albeit little exploited and published use case.

CDWs can deal with structured data very well. Unfortunately, a lot of the patient information in the electronic health record (EHR) is still stored in free text. E.g. Jensen et al. retrieved on average 146 unstructured text documents for each patient from EHR of their hospital for their study [109]. Medication, too, is usually documented as free text within the discharge letter. As a solution, advanced CDW systems offer a query language that can extract data from free text (e.g. in [55]).

The conventional approach is to perform information extraction (IE) in the ETL[2] process. A well-known system for IE of medication is MedEx [249]. Beside other rule based-systems like [215], hybrid systems exist using machine learning techniques [211]. A good overview on IE from free text is given by Wang et al. [243].

Rule based systems require a high volume of handcrafted rules and learning systems need a large amount of manually labeled training data. Either way, a lot of expert work is necessary. Besides high developmental efforts, another disadvantage of conventional IE is its slow promptness and non-adaptability by users [55].

A novel way to retrieve information from plain text is ad hoc IE. Ad hoc IE is described as extracting the existence of any concepts (e.g. chronic kidney disease) or any numbers, like the left ventricular ejection fraction (LVEF) value, from textual sources in real-time. The Boolean ad hoc IE queries the existence (yes/no) of a medical concept. A medical concept is a named entity that may have a feature/property or a numeric value. Examples of Boolean concepts are single findings or assessments (e.g. moderate mitral insufficiency, severe aortic stenosis), drugs (e.g. Aspirin, beta blocker) or diagnoses (e.g. appendicitis, myocardial infarction). Numeric IE extracts the value as number of a numerical concept. That could be for example the value of a laboratory finding (e.g. cholesterol, glucose, LEVF) or a derived values/indexes (e.g. BMI, age). A numerical condition can be defined optionally, like LVEF < 45, matching all mentions of LVEF with a value lower than 45. In some finding reports, the exact value of a concept is not given but there is a formulation indicating an interval or an inequality of a value (e.g. “LVEF lower than 45"). These statements can be queried in conjunction with numeric ad hoc IE exploiting both qualitative and quantitative information from textual reports e.g. for checking inclusion or exclusion criteria of studies. In addition to count queries, which only asses the presence of a concept or the validity of constraints (e.g. BMI > 25), the actual values can also be returned for further processing.

This technique showed good results and requires little developmental effort, since the text is indexed efficiently and can be queried with powerful features [55].

8.1.3 Objectives

This work introduces ad hoc IE for medication and their daily dosage from hospital discharge letters. We present and evaluate query features for a CDW. As an example of use, we show medication trend estimations. Therefore we replicate existing studies from the literature in a large CDW of the University Hospital of Würzburg using ad hoc IE. The results will be compared with the corresponding published data describing similarities and differences.

[2]Extract, Transform, Load

Table 8.1: Example for the processing of the drug names.

Product name	Processed name	Alternative name
Bayer Aspirin forte 100mg	Aspirin	
Levothyroxin-Natrium	Levothyroxin Natrium	Levothyroxin Na
Paracetamol-Ratiopharm 500mg	Paracetamol	
ACC akut 200mg Hustenlöser	ACC	

8.1.4 Methods

The developmental steps included extensions and features for the data integration process and the development of new data query tools. For study replication, the drug names had to be acquired and transformed.

8.1.4.1 Query Token Generation

The Anatomical Therapeutic Chemical (ATC) Classification System is an international classification of active ingredients of drugs.[3] In the literature, ATC codes are used to encode drugs and active agents groups. In order to get all brand, drug and agent group names of an ATC-group like *C07 Beta Blocking Agents*, we use the ABDA-DB[4], which contains all names in English and German. Since medical reports rarely contain the full name of a drug, we processed the names from the ABDA-DB in various ways: a) names were simplified by omitting the names of the manufacturers and the strength of the drug; b) other unnecessary words were removed; that includes modifiers concerning the effect like *forte* and the administration form like *oral*; c) abbreviations and alternative spellings were considered. Table 8.1 shows examples of the processing of drug names. The resulting tokens were used for the queries. Hyphens do not need to be treated because they are removed by the tokenizing procedure.

8.1.4.2 Evaluation

We performed tests to evaluate our development and conducted case studies aiming to replicate findings reported in selected medication trends studies.

Medication extraction. Since medication studies only consider the use of drugs, the replication requires just Boolean IE. Therefore we carried out a comprehensive test. We further evaluated the requests for the daily dosage using ad hoc IE. To protect privacy, these texts were de-identified and in addition they must not leave the clinical network.

[3] `https://www.whocc.no/atc_ddd_index/`
[4] `http://abdata.de/datenangebot/abda-datenbank/`

Extraction of drugs. For the evaluation of the medication extraction 600 documents were randomly selected from the disease domains hypertension, atrial fibrillation and chronic kidney disease. From each domain, 100 medication sections from 2005 and 100 sections from 2015 were sampled, resulting in a total of 600 documents. A manually annotated gold standard was created for these documents. All medications, brands, drug and substance names were annotated using the Apache UIMA CAS type system. In order to save time, the text was first automatically pre-announced using the medication tokens gained in Section 8.1.4.1. Then, the texts were manually corrected to obtain the gold standard. The ATHEN environment[5] was used to perform this work [128]. Afterwards the original texts were imported into the PaDaWaN-CDW with the data integration pipeline. Then queries were made with all drug names and the hits detected were annotated. At the end, all hits found by the system were compared to the gold standard.

Daily dosage. The extraction of the daily medication dosage was evaluated with several drugs: Antihypertensive drugs: Esidrix® (Thiazide-Diuretika, ATC: C03A), Concor® (β-blocker, C07A), Delix® (ACE inhibitor C09A) and novel oral anticoagulants (NOAC) used for atrial fibrillation: Eliquis®, Pradaxa®, Xarelto®. For each drug, 100 medication sections containing this drug from 2015 were selected. For the antihypertensive drugs another 100 units were selected for the year 2005. This was not possible for the NOACs, since they did not exist at that time. Queries were made in the PaDaWaN system and evaluated manually. For the evaluation, all dose strengths were extracted. The proximity query feature was used to extract the dose.

Study replication To evaluate the quality of the study replication, we chose five studies from the literature covering three domains (hypertension, atrial fibrillation, chronic kidney disease) and compared the major and sub-findings with the results of the University Hospital of Würzburg in total, respectively restricted to its Department of Internal Medicine I (Med1) using the ad hoc query feature with of the CDW. The drugs were extracted from the medication section of the discharge letter. That contains in almost every case the medication at discharge representing the recommended / prescribed medication. Additionally the medication at admission is described in 18% (Med1: 13%) of all cases. At discharge from hospital, patients receive 8% (Med1: 19%) more medication than at admission, while nearly all medications from admission were continued at discharge. (Tested for the main drug agent groups for hypertension.) We used the whole medication section with all medication descriptions as data source to identify weather a drug is taken or not.

This was conducted with the PaDaWaN-CDW including about 1 million patients with 5 million patient cases and more than 600 million pieces of single information. We applied the same in- and exclusion criteria as in the respective publications. However, we did not compute age-adjusted values. Not every single evaluation in the publications was

[5] `http://www.is.informatik.uni-wuerzburg.de/research_tools_download/athen/`

reproduced; we rather focused on the main statements and central result tables of the studies or took the most interesting parts of the publications to show the power of our approach.

Hypertension. We chose [85] as first drug trend study, because it is a highly cited study addressing a large population. The analyzed data was acquired during the National Health and Nutrition Examination Survey (NHANES) [165]. We further aimed to replicate the results of Shah and Stafford [206] concerning the findings on systolic blood pressure. These authors used data from the National Disease an Therapeutic Index (NDTI), a nationally representative physician survey. We extracted this information from the discharge letter via numeric ad hoc IE [55].

Atrial Fibrillation. In the replication of the study for atrial fibrillation [79] the ad hoc IE from unstructured texts was combined with structured data from the CDW and differentiated according to these. Subgroups such as comorbidity and age groups were investigated by Gadsbøll et al. [217]. The data sources of these studies were the Danish National Patient Registry, the (Danish) National Prescription Registry and the (Danish) Civil Registration System, containing various information on all prescriptions dispensed in Danish pharmacies since 1995.

Chronic Kidney Disease. We also selected a study to examine temporal trends and treatment patterns by patients with CKD and type 2 diabetes mellitus (T2DM) [245]. In this work, medication groups are evaluated. In a more detailed analysis, CKD was broken down into different severity levels (stages), and the medicative effect of the medication groups was considered [245]. This study also used the data from NHANES.

Tables 8.2 and 8.3 map all drug and diagnostic group designations used in respective publications to ATC and ICD10 codes, respectively. These codes were used for the replication of these studies. Table 8.4 summarizes the replicated studies and shows their inclusion and exclusion criteria.

Table 8.3: **Mapping of drug group designations in the literature to ATC codes.** The drug group designations used in the literature are mapped to ATC codes used for the replication.

Designation in paper	ATC-Codesystem
Insulin	A10A: Insulins and analogues
Oral antidiabetes medication	A10B: Blood glucose lowering drugs, excluding insulins
Biguanides	A10BA: Biguanides
Sulfonylureas	A10BB: Sulfonylureas
Antidiabetes combinations	A10BD: Combinations of oral blood glucose lowering drugs

α-Glucosidase inhibitors	A10BF: Alpha glucosidase inhibitors
Thiazolidinediones	A10BG: Thiazolidinediones
DPP-4 inhibitors	A10BH: Dipeptidyl peptidase 4 (DPP-4) inhibitors
Meglitinides	A10BX: Other blood glucose lowering drugs, excluding insulins
Vitamin K antagonists (VKA)	B01AA: Vitamin K antagonists
Warfarin	B01AA03: Warfarin
ADP receptor antagonists	B01AC04: Clopidogrel, B01AC05: Ticlopidine, B01AC22: Prasugrel, B01AC24: Ticagrelor
Oral anticoagulations (OAC)	VKA & NOAC
Non-vitamin K antagonist oral anticoagulants (NOAC)	Dabigatran, Rivaroxaban, and Apixaban
Rivaroxaban	B01AF01: Rivaroxaban
Apixaban	B01AF02: Apixaban
Dabigatran	B01AE07: Dabigatran etexilate
Aspirin	B01AC06 ASS
Dipyridamole	B01AC07: Dipyridamole
Digoxin	C01AA05: Digoxin
Diuretics	C03: Diuretics
Thiazide diuretics	C03A: Low-ceiling diuretics, thiazides
Hydrochlorothiazide	C03AA03: Hydrochlorothiazide
Loop diuretics	C03C: High-ceiling diuretics
Furosemide	C03CA01: Furosemide
Hydrochlorothiazide; triamterene	C03EA01: Hydrochlorothiazide and potassium-sparing agents
β-blockers	C07: Beta blocking agents
Metoprolol	C07AB02: Metoprolol
Atenolol	C07AB03: Atenolol
Carvedilol	C07AG02: Carvedilol
Calcium channel blockers	C08: Calcium channel blockers
Amlodipine	C08CA01: Amlodipine
Nifedipine	C08CA05: Nifedipine
Verapamil	C08DA01: Verapamil
Diltiazem	C08DB01: Diltiazem
RAAS	C09: Agents acting on the renin-angiotensin system
Renin-angiotensin system inhibitors:	C09A: ACE inhibitors, plain
Lisinopril	C09AA03: Lisinopril

Lisinopril; hydrochlorothiazide	C09BA03: Lisinopril and diuretics
Angiotensin receptor blockers	C09C: Angiotensin II antagonists, plain
Losartan	C09CA01: Losartan
Valsartan	C09CA03: Valsartan
Olmesartan	C09CA08: Olmesartan medoxomil
Non-steroidal antiinflammatory drugs:	M01A: Anti-inflammatory and antirheumatic products, non-steroids

8.1.5 Results of Ad Hoc Information Extraction Evaluation

8.1.5.1 Extraction of Drugs

Table 8.5 shows the performance of the ad hoc extraction of medications with an overall F1-score of 0.983 (precision 0.997 and recall 0.970).

Most errors were caused by abbreviations. The misspelling based errors could be significantly reduced by the error tolerant query feature. Table 8.6 shows the error analysis of the ad hoc extraction of medications. The most common occurrences of the error groups are shown below.

Abbreviation Fraxi (20), Tiotropium (6), Mg Verla (4), Dreisavit (3), Dabigatran (2), Insuman (2), Isosorbid (2)

Not in DB Eunerpan (9), Polybion (4), Aclidinium (2), Calcetat (2), Natriumperchlorat (2), Cranoc (2), Calcetat (2)

Alternative notation Glycopyrronium (2), Dikalium Clorazepat (2), Humaninsulin (1), Diuretikum (1), Ca Carbonat (1)

Misspelling Ferrosanol (4), Eins alpha (2), Amphomoronal (2), Beclometasondipropionat (2), Klazid (2), Rehnagel (2), Cardular (2), Calciumdiacetat (2)

Search to fuzzy diabetes ≈ diabetex (4), diagnostik ≈ diagnostika (1), antihypertensiven ≈ antihypertensives (1)

Incorrect extracted medication thrombozyten (1), cholesterin (1), albumin (1), kalium (1), natrium (1)

8.1.5.2 Extraction of Daily Drug Dose

An analysis on the data set for the daily dose, that contains 900 mentions of selected drugs, revealed that 5% of the mentioned drugs were discontinued or reduced. 90% had an indicated strength, 92% an instruction and 89% a strength and an instruction. See Table 8.7.

The most common daily taken dose was one unit (57%) followed by two units (31%), see Table 8.8.

Table 8.2: **Mapping of diagnostic group designations in the literature to ICD10 codes.** The diagnostic group designations used in the literature are mapped to ICD10 codes used for the replication.

Designation in paper	ICD-10-Code	Abbr.
Abnormal liver function	K77: Liver disorders in diseases classified elsewhere	
Alcohol abuse	F10: Alcohol related disorders	
Atrial Fibrillation	I48: Atrial fibrillation and flutter	AF
Bleeding	R58: Hemorrhage, not elsewhere classified	
Chronic Kidney Disease	N18: Chronic kidney disease	CKD
Deep vein thrombosis	I82: Other venous embolism and thrombosis	
Diabetes mellitus Typ 2	E11: Type 2 diabetes mellitus	T2DM
Heart failure	I50: Heart failure	
Hypertension	I10: Essential (primary) hypertension	HT
Ischemic heart disease	I20-25: Ischemic heart diseases	
Myocardial infarction	I21: Acute myocardial infarction	
Peripheral artery disease	I73.9: Peripheral vascular disease, unspecified	
Pregnant	O00-099: Pregnancy, childbirth and the puerperium	
Pulmonary embolism	I26: Pulmonary embolism	
Stroke	I63: Cerebral infarction	
Valvular disease	I05-I09: Chronic rheumatic heart diseases I34-I37: Nonrheumatic mitral/aortic/tricuspid/pulmonary valve disorders Q22-Q23: Congenital malformations of pulmonary and tricuspid valves / aortic and mitral valves	

Table 8.4: Overiew of replicated studies and their inclusion and exclusion criteria.

Study topic	Paper	Filters
Hypertension: Trends	[85]	Hypertension, age $\geq$ 18, not pregnant
Hypertension: Systolic BP	[206]	Hypertension, 1.1.2014- 1.1.2015
Atrial Fibrillation: Trend & Age Groups	[79]	Atrial Fibrillation, 2005 - 2018, age [30, 100], no valvular disease, no pulmonary embolism, no deep vein thrombosis
Atrial Fibrillation: Characteristics & Brands	[217]	Atrial Fibrillation, 22.8.2011 -1.1.2016, age [30, 100], no valvular disease, no pulmonary embolism, no deep vein thrombosis
CKD & T2DM	[245]	CKD,T2DB, Age $\geq$ 18, 2012-2017

Table 8.5: **Performance of the ad hoc extraction of medications.** Documents were selected form the years 2005 and 2015 and for the diagnoses essential hypertension (I10), atrial fibrillation and flutter (I48) and chronic kidney disease (N18).

Dataset	Documents	Medications	TP	FP	FN	Precision	Recall	F1
Overall	600	5701	5529	15	172	0.997	0.970	0.983
2005	300	23000	2176	13	124	0.994	0.946	0.969
2015	300	3041	3353	2	48	0.999	0.986	0.993
I10	200	1817	1768	3	49	0.998	0.973	0.986
I48	200	1795	1741	1	54	0.999	0.970	0.984
N18	200	2089	2020	11	69	0.995	0.967	0.981

Table 8.6: **Error analysis of the ad hoc extraction of medications.** Errors are summarized into the most common error groups.

	Medications		Occurrences	
	Number	Percent	Number	Percent
Abbreviation	40	33%	76	41%
Not in DB	22	18%	39	21%
Alternative notation	9	7%	10	5%
Misspelling	38	31%	47	25%
Search to fuzzy	3	2%	6	3%
Incorrect extracted medication	9	7%	9	5%

Table 8.7: **Presence of strength and instruction application of medication.** Data is taken form the evaluation set.

	Number	Percent
Intake (not discontinued)	852	95%
With strength	814	90%
With instruction	829	92%
With strength and instruction	800	89%

Table 8.8: **Summed daily dose of the medication units in the evaluation set.** All units of drug that are token over a day are summed up.

Daily units	Number	Percent
0.25	1	0.1%
0.5	85	10.0%
1	489	57.4%
1.5	7	0.8%
2	264	31.0%
3	5	0.6%
4	1	0.1%

The overall F1-score for the extraction of the daily medication dose was 0.974. The precision was the same or slightly higher than the recall in all tests. The extraction results were slightly better on the antihypertensive drug set (F1: 0.982) than on the NOACs drug set (F1: 0.958). The documents from 2015 also showed slightly better results than those of 2005 (F1: 0.977 vs 0.968). The complete results can be found in Table 8.9.

Most errors were caused by an unusual notation. See Table 8.10 and listing below. Other error sources were supplements, which contained numbers, incorrect splitting of the tokenizer, double mentions in same document, segmentation faults, and a too wide gap between the drug name and the instructions.

Notation Esidrix 1x1, Pradaxa 150-0-150 mg

Supplement Pradaxa 110 mg 1-0-1 (bitte 1 Tag vor stationären Aufnahmetermin pausieren);

Tokenizer Euthyrox®

Double mention Medikation bei Entlassung: Esidrix 12,5 mg 1-0-0; Medikamente bei Entlassung: Esidrix 25 pausiert

Gap Concor 5 mg (bei Bedarf) 1 – 0 – 0 – 1

8.1.6 Result of Study Replication

The presented results for the University Hospital of Würzburg (UKW) and the Department of Internal Medicine I (Med1) were computed via ad hoc IE (see Section 8.1.4.2). Since the ad hoc IE had an F1 score of 0.974, there may be small deviations from the exact values.

Table 8.9: **Performance of the ad hoc extraction of the daily medications dose.** Texts were selected form the years 2005 and 2015 of patients with the diagnosis atrial fibrillation and flutter. New oral anticoagulant (NOAC) and antihypertensive drugs are extracted.

Dataset	**Documents**	**TP**	**FP**	**FN**	**Precision**	**Recall**	**F1**
Overall	900	875	21	25	0.977	0.972	0.974
Xarelto	100	100	0	0	1.0	1.0	1.0
Eliquis	100	95	3	5	0.960	0.950	0.955
Pradaxa	100	92	6	8	0.939	0.920	0.929
NOACs	300	287	12	13	0.960	0.957	0.958
Esidrix	200	197	2	3	0.990	0.985	0.987
Concor	200	196	4	4	0.980	0.980	0.980
Delix	200	195	3	5	0.985	0.975	0.980
Antihypertensive drug	600	581	9	12	0.985	0.980	0.982
2015	600	586	13	14	0.978	0.977	0.977
2005	300	289	8	11	0.973	0.963	0.968

Table 8.10: **Error analysis of the ad hoc extraction of the daily medications dose.** Errors were summarized in the most common error groups.

Error	**Number**	**Percent**
Notation	23	50%
Supplement	6	13%
Tokenizer	6	13%
Doublet	5	11%
Segmentation	4	9%
GAP	2	4%

Table 8.11: **Replication of the medication group trend study for hypertension.** Drug agent groups compared to the reference paper [85] with all patients and Med1 clinic patients from University Hospital of Würzburg (UKW) during 2000-2010.

		2000 -2001	2003 -2004	2005 -2006	2007 -2008	2009 -2010	Overall
n	Paper	1669	1750	1564	2169	2168	9320
	UKW	4720	12267	17823	20187	23646	78643
	Med1	3485	5938	6690	7596	9189	32898
Diuretics	Paper	30%	32%	34%	35%	36%	34%
	UKW	48%	46%	45%	46%	48%	46%
	Med1	48%	56%	61%	60%	59%	58%
Thiazide-Diuretics	Paper	22%	24%	26%	27%	28%	26%
	UKW	14%	21%	20%	18%	18%	18%
	Med1	13%	24%	24%	20%	17%	20%
β-blockers	Paper	20%	25%	30%	28%	32%	27%
	UKW	58%	52%	50%	52%	56%	53%
	Med1	62%	69%	73%	72%	71%	70%
CC-Blocker	Paper	19%	21%	22%	19%	21%	20%
	UKW	27%	24%	24%	25%	28%	26%
	Med1	27%	30%	33%	34%	36%	33%
ACE inhibitors	Paper	26%	30%	29%	29%	33%	30%
	UKW	49%	46%	42%	44%	46%	45%
	Med1	51%	57%	56%	57%	55%	56%
ARB	Paper	11%	15%	15%	20%	22%	17%
	UKW	10%	11%	13%	14%	16%	14%
	Med1	11%	14%	16%	19%	20%	17%

8.1.6.1 Hypertension

Study: Trends in antihypertensive medication use and blood pressure control among United States adults with hypertension
Table 8.11 shows the results of the replication of the medication trend study to hypertension for the years 2000 to 2010. The findings of the referenced paper and their reproducibility by our results are listed in Table 8.12. The computation time to query the data for Table 8.11 from the CDW was 2 minutes (min) 26 seconds (sec).

Current trends of hypertension treatment in the United States. Table 8.13 shows the group systolic blood pressure of hypertensive patients. The findings of the referenced paper and their reproducibility by our results are listed in Table 8.15. The computation time to query the data for Table 8.13 and 8.14 from the CDW was aggregated 49 min 55 sec.

Table 8.12: **Comparison of findings to the antihypertensive medication study.** Study: *Trends in antihypertensive medication use and blood pressure control among United States adults with hypertension clinical perspective*

	Finding	Rep.
	Main findings	
1	Any antihypertensive drug increased	(yes)
	Other findings	
2	diuretics remained the most commonly used antihypertensive drug class	no
3	more than one third of hypertensive adults reported taking diuretics	yes
4	Use of thiazide diuretics accounted for three fourths of all diuretic use.	no
5	The prevalence of thiazide diuretic use increased slightly	yes
6	The overall prevalence of use of β-blockers increased	yes
7	Approximately 20% use CCBs in each survey period	yes
8	the use of CCBs remained relatively constant	yes
9	ACE inhibitors were the second most commonly used antihypertensive drug class	no
10	The use of ACE inhibitors increased significantly overall.	no
11	The use of ARB increased significantly	yes

8.1.6.2 Chronic Kidney Disease

Study: Understanding CKD among patients with T2DM: prevalence, temporal trends, and treatment patterns – NHANES 2007-2012

Figure 8.1 is an additional evaluation showing all severity levels of CKD over time. The computation time to query the data from the CDW was 14 sec.

Figure 8.2 shows the hypertension medication agent groups by degrees of severity of CKD for all patients with hypertension and CKD for the years 2013-2016. The computation time to query the data from the CDW for Figure 8.2 was 1 min 3 sec.

Table 8.16 compares the findings of Wu et al. [245] to our findings for the UKW and the Med1 concerning medication and agent groups for patients with CKD and T2DM. It shows the medication for diabetes as well as the hypertension. The findings of the referenced paper and their reproducibility by our results are listed in Table 8.18. The computation time to query the data from the CDW was 3 min 16 sec for Table 8.16 and 5 min 9 sec for Table 8.17.

Table 8.13: **Systolic blood pressure (SBP) in mm Hg of hypertensive patients.** Results are compared to [206].

	< 130	$[130-139]$	$[140-149]$	$[150-159]$	≥ 160
Paper	32%	26%	19%	9%	15%
UKW	23%	12%	11%	10%	45%
Med1	25%	13%	11%	9%	42%

Table 8.14: **Use of drug agent groups and systolic blood pressure groups.** Evaluation is made on hypertensive patients. Results are compared to [206]. SBP: systolic blood pressure, measured in mm Hg

SBP		Thiazide	β-Blocker	CCB	ACEI	ARB
< 130	Paper	25,1%	20,4%	20,0%	31,1%	21,1%
	UKW	14,3%	61,7%	27,3%	38,6%	21,4%
	Med1	15,5%	67,0%	30,8%	38,0%	23,1%
[130-139]	Paper	27,8%	17,2%	23,1%	29,7%	22,3%
	UKW	14,9%	54,7%	35,4%	42,9%	24,2%
	Med1	13,3%	61,9%	40,7%	44,2%	27,4%
[140-149]	Paper	24,7%	17,8%	23,7%	27,7%	22,5%
	UKW	17,2%	52,4%	33,1%	44,1%	24,8%
	Med1	17,0%	67,0%	41,5%	45,7%	34,0%
[150-159]	Paper	25,4%	17,9%	24,9%	25,6%	23,0%
	UKW	22,9%	52,7%	38,9%	48,9%	23,7%
	Med1	22,9%	61,4%	48,2%	54,2%	21,7%
≥ 160	Paper	26,0%	20,6%	26,0%	25,4%	20,5%
	UKW	22,9%	51,4%	37,0%	52,1%	23,4%
	Med1	16,5%	57,4%	41,2%	51,6%	23,9%

Table 8.15: **Comparison of findings to the hypertension treatment study.** Study: *Current trends of hypertension treatment in the United States*

	Finding	Rep.
	Main finding	
1	BP control widely varied among this medication-treated group of patients.	yes
	Other findings	
2	ACEI use was significantly more likely in patients with SBP < 130 compared with those with BP ≥ 160.	no
3	The use of CCBs was less likely among those with SBP < 130, but more likely among those with SBP ≥ 160	yes

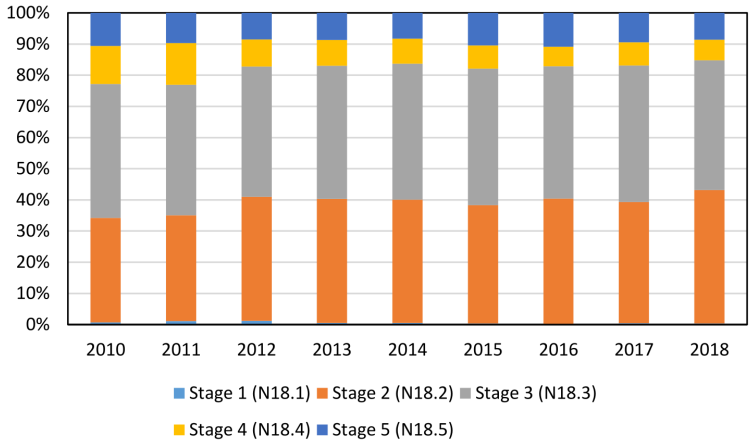


Figure 8.1: **Temporal trend of CKD stages in the UKW.** The severity degrees of CKD-patients are shown over time.

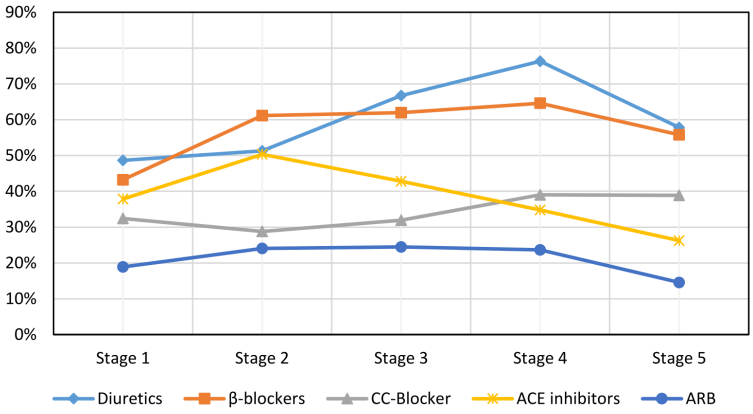


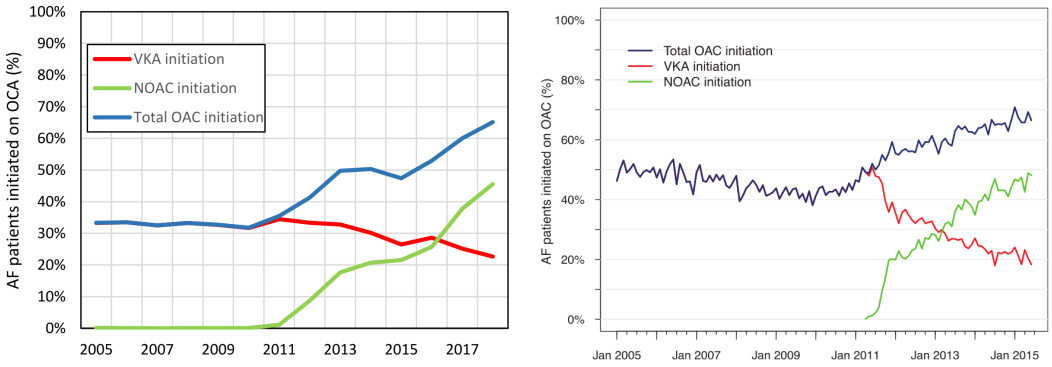
Figure 8.2: **Medication agent groups by degrees of severity of CKD in the UKW.** Patients have CKD and hypertension.

Table 8.16: **Medication and agent groups for CKD with T2DM.** The table subdivided according to CKD stage (S1-S5). Values are compared to [245]. Values with * were omitted due to small sample sizes.

	Overall	**No CKD**	**S 1**	**S 2**	**S 3**	**S 4**	**S 5**
n							
Paper	1380	1122	144	159	258	32	16
UKW	35636	20314	34	4725	7659	1671	1603
Med1	13461	6452	*	2264	3319	735	766
DM medication							
Paper	83%	81%	84%	89%	84%	94%	77%
UKW	60%	59%	59%	69%	62%	55%	44%
Med1	71%	69%	*	79%	72%	69%	61%
Insulin							
Paper	19%	15%	16%	28%	24%	38%	63%
UKW	26%	24%	24%	23%	30%	38%	35%
Med1	38%	39%	*	28%	39%	52%	51%
Oral antidiabetes medication							
Paper	75%	75%	81%	77%	72%	69%	44%
UKW	46%	47%	41%	59%	46%	28%	13%
Med1	51%	50%	*	69%	52%	31%	16%
Biguanides							
Paper	56%	62%	68%	55%	36%	4%	3%
UKW	32%	34%	26%	48%	27%	7%	1%
Med1	34%	33%	*	57%	32%	6%	0%
Sulfonylureas							
Paper	35%	31%	44%	42%	42%	56%	15%
UKW	8%	7%	9%	10%	10%	7%	2%
Med1	7%	6%	*	11%	9%	7%	2%
DPP-4 inhibitors							
Paper	7%	7%	4%	8%	8%	23%	7%
UKW	12%	11%	24%	14%	17%	13%	7%
Med1	17%	15%	*	19%	20%	17%	10%

Table 8.17: **Medication and agent groups for CKD with T2DM.** The table subdivided according to CKD stage (S1-S5). Values are compared to [245]. Values with * were omitted due to small sample sizes.

	Overall	**No N18**	**S 1**	**S 2**	**S 3**	**S 4**	**S 5**
n							
Paper	1380	1122	144	159	258	32	16
UKW	10314	15315	34	4723	7656	1671	1601
Med1	6452	7009	*	2266	3319	734	765
Hypertension medication							
Paper	76%	69%	63%	90%	92%	100%	97%
UKW	77%	68%	71%	89%	90%	89%	79%
Med1	85%	75%	*	96%	96%	96%	90%
Diuretics							
Paper	36%	30%	22%	42%	58%	76%	34%
UKW	53%	39%	56%	60%	76%	82%	64%
Med1	63%	47%	*	65%	84%	90%	76%
Thiazide diuretics							
Paper	24%	23%	18%	24%	30%	33%	0%
UKW	14%	13%	24%	22%	15%	10%	2%
Med1	12%	10%	*	23%	14%	7%	1%
Loop diuretics							
Paper	14%	7%	3%	21%	31%	54%	34%
UKW	40%	26%	41%	40%	64%	78%	63%
Med1	51%	36%	*	43%	74%	88%	74%
Potassium-sparing diuretics							
Paper	6%	6%	1%	4%	7%	8%	9%
UKW	11%	8%	6%	14%	20%	14%	6%
Med1	16%	11%	*	18%	27%	16%	9%
β-blockers							
Paper	31%	24%	15%	45%	46%	76%	82%
UKW	52%	43%	38%	62%	66%	68%	58%
Med1	64%	52%	*	74%	77%	78%	71%
CC-Blocker							
Paper	20%	15%	13%	37%	25%	33%	57%
UKW	29%	24%	29%	33%	35%	43%	37%
Med1	34%	28%	*	36%	39%	50%	45%
ACE inhibitors							
Paper	40%	38%	43%	51%	42%	28%	41%



(a) UKW (b) Paper

Figure 8.3: **Temporal trend of VKA and OACs.** Results are compared to [217].

UKW	38%	35%	41%	50%	44%	34%	27%
Med1	43%	38%	*	56%	48%	37%	32%
ARB							
Paper	22%	19%	11%	25%	32%	35%	16%
UKW	19%	16%	18%	24%	26%	25%	15%
Med1	24%	19%	*	30%	32%	32%	18%
RAAS							
UKW	58%	52%	59%	74%	69%	59%	42%
Med1	68%	58%	*	86%	80%	68%	50%

8.1.6.3 Atrial Fibrillation

The studies on atrial fibrillation (AF) investigate the characteristics and the temporal trend of the use of oral anticoagulants (OAC).

Study: Increased use of oral anticoagulants in patients with atrial fibrillation: temporal trends from 2005 to 2015 in Denmark
Gadsbøll et al. investigate the increased use of oral anticoagulants in patients with atrial fibrillation [79]. Figure 8.3 shows the temporal trend of VKA and OACs compared to [217]. The findings of the referenced paper and their reproducibility by our results are listed in Table 8.19. The computation time to query the data from the CDW for Figure 8.3 was 25 sec.

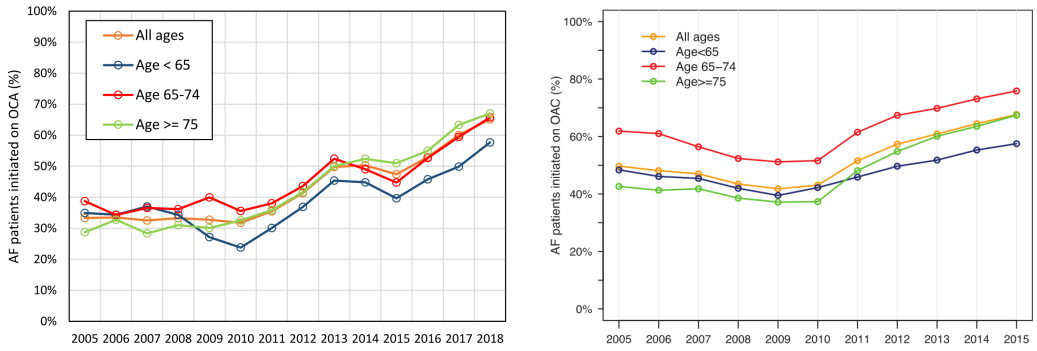
Figure 8.4 shows the temporal trend for AF patient age groups using OACs like in [217]. The computation time to query the data from the CDW for Figure 8.4 was 55 sec.

Table 8.18: **Comparison of findings to the CKD and T2DM study.**
Study: *Understanding CKD among patients with T2DM: prevalence, temporal trends, and treatment patterns—NHANES 2007–2012*

	Finding	**Rep.**
	Main findings: The use of antidiabetic and antihypertensive medications generally followed treatment guideline recommendations:	
1	The use of metformin was significantly limited with increasing CKD severity	yes
2	The use of insulin increased sharply in severe CKD stages	yes
3	Antihypertensive medications were used extensively	yes
4	The level of RAAS inhibitor (including ACE inhibitors and ARBs) use was consistent, even in patients without CKD and with mild-to-moderate CKD	yes
5	Use of thiazide diuretics was more prevalent than other diuretic agents with mild-to-moderate CKD	yes
6	Thiazide diuretics were replaced by loop diuretics among those with moderate CKD to kidney failure	yes
	Other findings	
	Antidiabetes medications:	
7	Overall, 83.1% of individuals with T2DM received antidiabetic medications	no
8	The use of insulin, biguanide (metformin), and sulfonylurea (SU) was significantly different between patients without CKD, those with mild-to-moderate CKD, and those with moderate CKD to kidney failure	yes
9	The use of dipeptidyl peptidase-4 (DPP-4) inhibitors was similar	yes
10	The use of sulfonylurea (SU)s increased in later CKD stages (3b and 4)	no
11	Sulfonylurea SU use dropped in CKD stage 5	yes
	Antihypertensive medications:	
12	Overall, 75.7% of individuals with T2DM received antihypertensive medications	yes
13	Use was extensive in those with CKD stage 2 or higher	yes
14	Fewer than two-thirds were taking some form of RAAS inhibitor	(yes)
15	There was a difference in the use of ACE inhibitors and ARBs between patients without CKD, those with mild-to-moderate CKD, and those with moderate CKD to kidney failure	yes
16	The use of β-blockers, diuretics, and CCBs was statistically different	yes
17	ARBs appeared to be more commonly used in stages 3a–4	yes
18	The use of β-blocker and CCBs trended upward with increasing CKD severity	(yes)
19	Diuretic use also increased from stage 1 through stage 4, but sharply fell in stage 5	yes
20	Dhiazide diuretics were more commonly used by individuals without CKD or with mild-to-moderate CKD compared with other diuretic subclasses	yes
21	In later CKD stages, the dominance of thiazide diuretics was replaced with loop diuretics	yes
22	β-Blocker use increased with stages 4 and 5 CKD	no

Table 8.19: **Comparison of findings to the OAC study (2005-20015).**
Study: *Increased use of oral anticoagulants in patients with atrial fibrillation: temporal trends from 2005 to 2015 in Denmark*

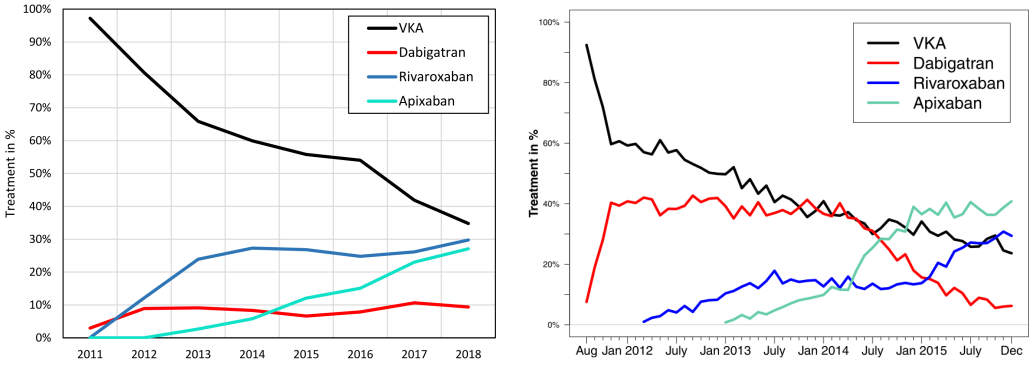
	Finding	Rep.
	Main findings	
1	since 2010, more incident AF patients were initiated on OAC treatment	yes
2	NOACs have replaced VKA as the OAC of choice in AF	yes
	Other results	
3	OAC initiation rates among the incident AF patients decreased from January 2005 to December 2009	yes
4	From 2010, more patients were initiated on OAC therapy	yes
5	From 2011, more prevalent AF patients were treated with an OAC	yes
6	From 2011, a decreasing proportion of the newly diagnosed AF patients was initiated on VKA	yes
7	This decrease in VKA initiation was followed by a rapid increase in NOAC initiation	yes



(a) UKW

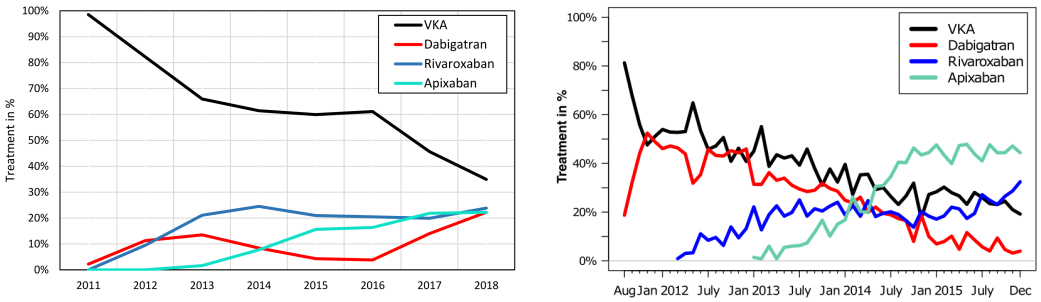
(b) Paper

Figure 8.4: **Temporal trend of OAC clustered by age groups.** Results are compared to [217].



(a) UKW (b) Paper

Figure 8.5: **Temporal trend of VKA and OAC usage of all AF patients.** Results are compared to [217].



(a) UKW (b) Paper

Figure 8.6: **Temporal trend of VKA and NOACs of AF patients aged $\geq$ 85.** Results are compared to [217].

Study: Non-vitamin K antagonist oral anticoagulation usage according to age among patients with atrial fibrillation: Temporal trends 2011–2015 in Denmark

Staerk et al. made a detailed research for the years 2011 and 2015, since NOAC became relevant [217]. Figure 8.5 and 8.6 is a detailed analyses of the temporal trend OACs listing its representatives: Dabigatran, Rivaroxaban, Apixaban. The findings of the referenced paper and their reproducibility by our results are listed in Table 8.20. The computation time to query the data from the CDW was 36 sec for Figure 8.5 and 29 sec for Figure 8.6.

Table 8.21 shows the distribution among sex and age groups. Table 8.22 analyses the comorbidities and table 8.23 lists the concomitant medication. The values in the referenced paper refer to the time period between 22.8.2011 and 1.1.2016. We computed the values for the same period (named UKW_11) and for the period 1.1.2016 - 1.1.2018 (named UKW_16). The computation time to query the data from the CDW was 1 min 10 sec for Table 8.21, 1 min 40 sec for Table 8.22 and 2 min 10 sec for Table 8.23.

Table 8.20: **Comparison of findings to the OAC study (2011-2015).**
Study: *Non-vitamin K antagonist oral anticoagulation usage according to age among patients with atrial fibrillation: Temporal trends 2011–2015 in Denmark*

	Finding	Rep.
	Main findings	
1	The absolute number of patients initiating OAC has increased among patients aged < 65, 65 to 74, and ≥ 85 years	yes
2	The utilization of VKAs has decreased since the introduction of NOACs	yes
3	From 2014 [to 2015] the utilization of dabigatran has decreased, especially among patients aged ≥ 85 years	yes
4	Apixaban has increased significantly and was the most used NOAC drug among patients aged ≥ 85 years	(yes)
	Other results	
5	For patients aged 75 to 84 years, number of patients initiating OAC treatment stayed approximately the same	no
6	The utilization of dabigatran increased within a couple of months since its introduction to the market	yes
7	A fairly constant level of dabigatran utilization was seen from December 2011 of approximately 40%	no
8	Rivaroxaban has steadily increased usage and at study end 29%	yes

Table 8.24 summarizes the results of the study replication. Main findings were replicated and confirmed by us to 93%, sub-findings to 68% and overall to 75%.

Daily medication dose extraction. As an additional evaluation, we extracted the daily dose of patients with AF using ad hoc IE. All three OACs agent groups with their drugs where analyzed: Xarelto (Rivaroxaban) (see Table 8.26a), Eliquis (Apixaban) (see Table 8.26b) and Pradaxa (Dabigatran) (see Table 8.26c).

The average daily dose was 19,31 mg of Xarelto, 7,4 mg of Eliquis and 232,3 mg of Pradaxa.

8.1.7 Discussion

First, the results of the replication studies are discussed, and second, the ad hoc IE tests and the system itself are compared to other approaches.

8.1.7.1 Study Replication

Major result & comparison. One study (AF Trend from 2005 to 2015 [79]) could be completely replicated, i.e., all main findings and sub-findings were confirmed by

Table 8.21: **Characteristics of patients with atrial fibrillation using VKAs or OAC.** Results are compared to [217].

		VKA	**Dabigatran**	**Rivaroxaban**	**Apixaban**
N (%)	Paper	42%	29%	13%	16%
	UKW_11	66%	8%	22%	6%
	UKW_16	48%	9%	26%	19%
Males (%)	Paper	57%	55%	50%	50%
	UKW_11	59%	62%	61%	63%
	UKW_16	61%	66%	62%	58%
Age < 65	Paper	22%	24%	17%	15%
	UKW_11	12%	21%	25%	17%
	UKW_16	10%	9%	21%	15%
Age 65 to 74	Paper	33%	35%	33%	31%
	UKW_11	28%	29%	28%	22%
	UKW_16	25%	25%	29%	25%
Age 75 to 84	Paper	31%	28%	29%	31%
	UKW_11	45%	35%	34%	40%
	UKW_16	46%	49%	36%	42%
Age ≥ 85	Paper	13%	13%	21%	22%
	UKW_11	15%	15%	13%	21%
	UKW_16	19%	17%	14%	18%

Table 8.22: **Comorbidities of patients with atrial fibrillation using VKAs or OAC.** (Continuation of Table 8.21)

		VKA	Dabigatran	Rivaroxaban	Apixaban
Stroke	Paper	15%	15%	18%	21%
	UKW_11	2%	13%	5%	13%
	UKW_16	3%	26%	3%	2%
Myocardial infarction	Paper	11%	7%	6%	7%
	UKW_11	3%	1%	2%	1%
	UKW_16	2%	2%	4%	1%
Ischemic heart disease	Paper	26%	20%	20%	21%
	UKW_11	32%	26%	23%	31%
	UKW_16	29%	29%	31%	30%
Heart failure	Paper	19%	14%	15%	16%
	UKW_11	31%	25%	26%	34%
	UKW_16	35%	26%	31%	38%
Diabetes mellitus	Paper	14%	11%	12%	13%
	UKW_11	32%	22%	22%	28%
	UKW_16	32%	24%	23%	29%
Hypertension	Paper	47%	44%	44%	43%
	UKW_11	69%	68%	63%	67%
	UKW_16	67%	71%	61%	64%
Chronic kidney disease	Paper	8%	2%	4%	5%
	UKW_11	58%	54%	49%	51%
	UKW_16	49%	43%	46%	49%

Table 8.23: **Concomitant medication of patients with AF using VKAs or OAC.** (Continuation of Table 8.21)

		VKA	Dabigatran	Rivaroxaban	Apixaban
ADP receptor antagonists	Paper	10%	8%	10%	11%
	UKW_11	4%	8%	3%	4%
	UKW_16	5%	10%	11%	3%
ASS	Paper	43%	38%	38%	36%
	UKW_11	11%	15%	13%	11%
	UKW_16	9%	15%	11%	8%
Non-steroidal antiinflammatory drugs	Paper	15%	15%	14%	14%
	UKW_11	6%	5%	5%	3%
	UKW_16	8%	9%	8%	5%
Loop diuretics	Paper	22%	15%	18%	19%
	UKW_11	59%	42%	42%	52%
	UKW_16	60%	40%	41%	54%
Beta-blockers	Paper	45%	38%	39%	37%
	UKW_11	77%	76%	77%	78%
	UKW_16	77%	72%	75%	76%
Calcium channel blockers	Paper	29%	26%	27%	26%
	UKW_11	32%	29%	30%	30%
	UKW_16	32%	33%	29%	28%
Renin-angiotensin system inhibitors	Paper	43%	42%	41%	43%
	UKW_11	46%	40%	38%	42%
	UKW_16	39%	42%	35%	38%

Table 8.24: **Summary of the of the study replication results.** Results include main, sub and overall findings. The table shows the amount of findings, which were replicated and confirmed by us.

Paper topic	Ref	Main finding	Sub finding	Overall
HT: Trends	[85]	50%	50%	50%
HT: SBP	[206]	100%	50%	67%
CKD & T2DM	[245]	75%	75%	82%
AF Trend 2005-2015	[79]	100%	100%	100%
AF: Characteristics & Brands	[217]	88%	50%	69%
Overall		**93%**	**68%**	**75%**

Table 8.25: **Extraction of the daily medication dose for patients with atrial fibrillation.**

(a) Xarelto. Average dose: 19,3 mg

d. u.	10mg	15mg	20mg	50mg
1	0,9%	26,6%	67,4%	0,5%
1,5	0,0%	0,0%	0,0%	0,0%
2	1,4%	1,4%	1,4%	0,0%
3	0,0%	0,0%	0,5%	0,0%
sum	2,3%	28,0%	69,3%	0,5%

(b) Eliquis. Average dose: 7,4 mg.

d. u.	2,5mg	5mg
1	3,7%	3,2%
1,5	0,0%	0,0%
2	43,2%	49,5%
3	0,0%	0,5%
sum	46,8%	53,2%

(c) Pradaxa. Average dose: 232,3 mg.

daily units	10mg	75mg	110mg	150mg
1	0,0%	1,1%	5,6%	3,3%
1,5	0,0%	0,0%	0,0%	0,0%
2	1,1%	3,9%	51,1%	33,3%
3	0,0%	0,0%	0,6%	0,0%
sum	1,1%	5,0%	57,2%	36,7%

us. Overall, 93% of the main findings, 68% of other detailed findings and 75% of all findings could be replicated. Table 8.24 lists the results of the individual replications. As mentioned in section 1, many researchers have tried to reproduce other researchers work, but 70% failed. 24% researchers reporting a successful replication of experiments were able to publish their work. In case of unsuccessful reproduction this proportion was only 13% [13]. Of course, when conducting replication experiments, some deviations have to be expected. Concerning the sources of variation, not only the exact reproduction of the study design is important, but also the population under study and time trends observed regarding diagnosis and therapy matter. E.g., Gu et al. reported that the control of blood pressure (BP) levels "varied greatly between recent publications" [85]. Staerk et al. mentioned that the most frequently used NOAC agent in their study was different to a previous study owing to changes in prescription patterns over time [217] .

Study details. The distribution among the groups of active substances for hypertension in the UKW was slightly different compared to the paper [85]. In Med1, patients got substantially more drugs, probably indicating treatment preferences of a certain clinic.

In the CKD study, 75% of all findings agreed with our results, but there were also some deviations. Some observations differed only in stage 5 of CKD. This could be explained with different sizes of population of the subgroups with level 1, 4 and 5. These were caused by the basic population (population-based sample vs. hospital patients). The trends in the studies of atrial fibrillation could be replicated by us, however with a

surprisingly small temporal shift. The comorbidities and the concomitant medication differed slightly, but many agreed.

Data acquisition & study population. The studies differed regarding the data acquisition approach: The hypertension [85] and CKD [245] studies were based on NHANES, the AF studies [79, 217] on the Danish National Prescription Registry and the hypertensive study with SBP used a physician survey. The medication in NHANES was "self-reported data (via a patient survey questionnaire)" [245]. We took the medication information from the discharge letter written by physician, which should be reflected in higher accuracy. NHANES is a representative sample of the U.S., i.e. both healthy and sick people, whereas a CDW collects information on hospitalized or ambulatory patients. There are even differences within a hospital. The medication use was found higher in almost all cases at the Med1 compared to the entire clinic. This is comprehensible, because hypertension, atrial fibrillation and chronic kidney diseases are usually treated there. The studies also differed regarding the number of analyzed cases. The AF studies used a nation-wide data source, i.e. three to four times more patients than which were present in the local CDW. For the hypertension study, we analyzed eight times more cases, in the CKD even 25 times more cases.

Analysis duration. While our queries took only a few minutes, it probably took a few weeks or months to conduct the studies for the referenced papers.

8.1.7.2 Ad Hoc Information Extraction

Ad hoc IE possesses features of a conventional IE and query functions of CDWs. Therefore, the evaluation results and the system itself are compared with other approaches.

Comparison of evaluation results According to [243] MedEx is the most widespread used tool for extracting medication information from clinical texts. In their original paper they achieved an F1-score of 93,2% for extracting drug names, a score of 94,6% for the strength and 96,0% for the frequency [249]. Two years later they published a case study around the medication *warfarin* and pushed the F1 score to 95% (recall 99,7%, precision 90,8%) for extracting the daily dosage [248]. In another study, they tried to calculate the daily dosage for the drug *tacrolimus* with an extended MedEx version and reported precisions of 90-100% and recalls of 81-100%. For discharge summaries they achieved F1 measures of 96% for strength and 88% for daily dosage [247].

Some papers mention, that they had to deal with more complex medication instructions like dosing in 2-hour intervals [210, 247–249]. This may complicate the calculation of the dosage and explain the inferior results compared to ours (F1 97,4%, precision 97,7%, recall 97,2%).

The results of the extraction of the drug names alone were only partially comparable with ours. First, no lists of medications were used in the literature, and second, these

are all conventional IEs. We applied ad hoc IE, which extracts the information on the fly during runtime.

8.1.7.3 Limitations

Limitations for conducting medication trend studies in a CDW relate to complex inclusion and exclusion criteria that cannot appropriately be mapped, like complex temporal constraints. Some techniques frequently used in clinical analyses are more difficult to apply like adjustment for important confounders, e.g. sex and age. This is not a technical limitation, but it would require a laborious recalculation.

The feasibility of replication studies depends as well on the data embedded in the CDW. Only integrated concepts or texts can be queried. The populations of studies are always different, so the population of a specific hospital department does not correspond to the overall population.

8.1.8 Conclusion

With the presented approach of the ad hoc IE for medications, which provides equally good results for this task as the conventional approach, it is possible to quickly carry out analyses like the study replications shown here. We combined ad hoc IE with additional filters based on structured and unstructured data: We stratified the data by year and severity of the respective condition, and analyzed subgroups like age, comorbidities and concomitant medication. Furthermore, we used ad hoc IE to transform unstructured data from the discharge letters to structured data (e.g. systolic blood pressure groups) and extracted the daily dosage per drug on the fly.

To calculate daily medication dosages, each strength unit combination must still be queried individually. It is intended to calculate this automatically, e.g. with the use of function queries.

8.2 Prevalence of Heart Failure in Hospital Inpatients

This sections presents a case study of ad hoc information extraction (IE) at the University Hospital of Würzburg that investigates the prevalence of heart failure in inpatients. The study evaluates the existence of heart failure by extracting medical concepts from discharge letters. The characteristics are mapped by physicians to medical findings and their descriptions that appear in the letters. These are extracted with ad hoc IE using the PaDaWaN CDW (see Table 8.27). The analysis and interpretation of medical data and results are performed by physicians as well.

The case study shows the application of ad hoc IE and its advantages in a real-world scenario. It is demonstrated which evaluations are possible with ad hoc IE and how a domain can benefit from the extracted data: This study focuses on the medical interpretation of the results. This section contains a previously published article [113]: M. Kaspar, G. Fette, G. Güder, L. Seidlmayer, M. Ertl, G. Dietrich, H. Greger, F. Puppe, and S. Störk. Underestimated prevalence of heart failure in hospital inpatients: a comparison of icd codes and discharge letter information. *Clinical Research in Cardiology*, pages 1–10, 2018[6]

8.2.1 Summary

Background. Heart failure is the predominant cause of hospitalization and amongst the leading causes of death in Germany. However, accurate estimates of prevalence and incidence are lacking. Reported figures originating from different information sources are compromised by factors like economic reasons or documentation quality.

Methods. We implemented a clinical data warehouse that integrates various information sources (structured parameters, plain text, data extracted by natural language processing) and enables reliable approximations to the real number of heart failure patients. Performance of ICD-based diagnosis in detecting heart failure was compared across the years 2000–2015 with (a) advanced definitions based on algorithms that integrate various sources of the hospital information system, and (b) a physician-based reference standard.

Results Applying these methods for detecting heart failure in inpatients revealed that relying on ICD codes resulted in a marked underestimation of the true prevalence of heart failure, ranging from 44% in the validation dataset to 55% (single year) and 31% (all years) in the overall analysis. Percentages changed over the years, indicating secular changes in coding practice and efficiency. Performance was markedly improved using search and permutation algorithms from the initial expert-specified query (F1 score of 81%) to the computer-optimized query (F1 score of 86%) or, alternatively, optimizing precision or sensitivity depending on the search objective.

[6]The article is published with the Creative Commons Attribution 4.0 International License (`https://creativecommons.org/licenses/by/4.0/`), minor changes are made.

Conclusions Estimating prevalence of heart failure using ICD codes as the sole data source yielded unreliable results. Diagnostic accuracy was markedly improved using dedicated search algorithms. Our approach may be transferred to other hospital information systems.

8.2.2 Background

Heart failure has become the leading diagnosis at hospital discharge and the most important driver of in-hospital mortality in Germany [2]. These estimates are based on counts provided by hospitals utilizing the German modification of the International Classification of Diseases (ICD)-10. Respective ICD-10 codes identifying heart failure within the German ICD-10 catalogue are I11.*, I13.0, I13.2, and I50.*. However, estimating the true number of patients suffering from heart failure based on this catalogue is unreliable for several reasons. The heart failure syndrome, especially in its early stages, may go unrecognized or may not be encoded as an explicit diagnosis; further, various financial incentives provided by the German health care system drive the likelihood of a specific ICD code entering a patient's list of discharge diagnoses. These incentives favor the encoding of diagnoses associated with the most favorable reimbursement profile and may therefore considerably affect the "true number" of patients burdened by heart failure. After hospital discharge, however, only the most important diagnoses (e.g., the top three) are reported to and collected by higher level organizations, e.g., health insurances or statutory registries. Thus, the "prevalence" of a certain condition may be augmented or suppressed depending on its re-imbursement profile and subsequent quality of coding. Furthermore, the quality of documentation itself, e.g., staff training [11] and the marked changes imposed on respective workflows (e.g., change from paper-based records to electronic record systems [177]), have a major influence on disease statistics. Despite these shortcomings, the above-mentioned approach of collecting ICD diagnoses remains the prime source for public health decisions [87, 199].

Beside the statutory census of disease statistics, attempts have been made towards a more reliable and comprehensive identification of diagnoses from clinical routine data. Most of them, however, based their algorithm on coded diagnoses [194]. Because of these reasons, a better and earlier recognition of heart failure patients is of utmost importance [220]. Since modern hospitals can provide a wealth of electronic patient-based information, this data may be used to improve or corroborate diagnostic certainty and comprehensiveness.

The objective of the current study was to approximate the "true number" of patients suffering from heart failure at a tertiary care center. Against a physician-based reference standard, we compared the performance of ICD-based diagnosis versus advanced definitions based on algorithms that integrate various sources of the hospital information system. We hypothesized that (a) ICD-based diagnosis may underestimate the true prevalence of heart failure and (b) a catalogue of criteria defining heart failure utilizing various sources of the hospital information system will advance diagnostic accuracy.

8.2.3 Methods

8.2.3.1 The Würzburg Data Warehouse

The clinical data warehouse (DWH) implemented at the Würzburg University Hospital provides a homogeneous and structured access to pseudonymized data of 100The only current exceptions are data from psychiatric and child care facilities for data protection reasons. The technical set-up is based on open-source systems and has been described elsewhere [51, 68, 112]. Data of the DWH can be queried in (a) structured form (e.g., patient demographics, diagnoses as ICD codes, procedures as codes of the German procedure classification "Operationen- und Prozedurenschlüssel" (OPS), and laboratory values); (b) semi-structured form (e.g., echocardiography, cardiac catheterization), and (c) unstructured form (e.g., discharge letters). The most innovative add-on to the DWH is the unique information extraction and ad hoc text search functionality, which allows to create parametrized information from semi- and unstructured reports and to search for any textual item (e.g., search within discharge letters for text combinations including variants and negations or extract numeric parameters from echocardiographic reports) [228].

8.2.3.2 Patient Selection

The Medical Department I of the Würzburg University Hospital specializes in, but is not limited to, emergency medicine, intensive care, cardiology, pulmonology, nephrology, and endocrinology. For the current analysis, we used all cases of patients treated at the Medical Department I between the years 2000 and 2015 for whom a discharge letter was available.

8.2.3.3 Reference Standard for the Definition of Heart Failure

A sample of consecutive patients treated at the Medical Department I was drawn from the DWH within a randomly selected period (January 1 to January 31, 2009), yielding 1042 cases. These patients were manually checked by a cardiologist with long-standing experience in the care of heart failure patients (GG). Information used by the physician included ICD codes, the discharge letter, and the echocardiographic report (if available). The physician assigned a label ("heart failure: yes/no") to each case, which was then used as reference standard for subsequent analyses.

8.2.3.4 Algorithms for Automated Detection of Heart Failure

In order to investigate heart failure detection algorithms, 18 subqueries of relevant heart failure-related concepts were defined within the user interface of our DWH and presented in the rows of Table 8.27. Each subquery considers a specific fact and either is a restriction on a numeric DWH parameter (e.g., subquery Echo-EF ≤ 45 represents a

left ventricular ejection fraction (LVEF) $\leq 45\%$ captured from echocardiographic reports after information extraction), the existence of an ICD diagnoses (e.g., subquery ICD-Any-HF represents the existence of any heart failure related ICD) or text searches within the discharge letter suggesting presence of heart failure (e.g., subquery Text-Left-HF represents the occurrence of a textual synonym for "left ventricular heart failure"). Text searches were specified to account for typing errors, synonyms and negations [6].

The algorithms used to detect patients with heart failure (i.e., MICD, MExpert, APrecision, ASensitivity, AF1) are presented in the right-hand columns of Table 8.27, each by a selection of subqueries that needed to be combined for the full algorithm. Each hit of any of an algorithm's subqueries stands for the presence of heart failure. Two of the algorithms were manually specified: MICD indicates an algorithm that solely utilizes ICD codes and MExpert indicates an algorithm (i.e., subqueries used for this DWH interrogation) pre-specified by cardiologists based on clinical experience. The other three algorithms (APrecision, ASensitivity, and AF1) originated from iterative permutation testing utilizing all defined subqueries. They were optimized to yield the most favorable results regarding the chosen measures (i.e., precision, sensitivity, and F1 score; for definitions see "Data analysis") with regard to the reference standard definition of heart failure described in the previous section. The algorithms were computed utilizing exactly the same data that the physician used to evaluate the reference standard: the discharge letter, the ICD codes, and the echocardiographic report (if available).

Legend for Table 8.27: M_{Expert} indicates the initially defined query by the clinical expert and a computer scientist. M_{ICD} indicates the algorithm using sole ICD codes for comparison. A_{F1}, $A_{Precision}$, and $A_{Sensitivity}$ indicate search algorithms optimized using permutation testing. Queries use Boolean "OR" operators, which means that each single hit justifies presence of heart failure *LV* left ventricular, *RV* right ventricular, *EF* ejection fraction, *HF* heart failure, *NYHA* New York Heart Association, *DCM* dilated cardiomyopathy, *NT-proBNP* N-terminal prohormone of brain natriuretic peptide.

[a] **Sources:** *Echo* echocardiography report
ICD ICD diagnosis, *Text* unstructured text from discharge letter
Lab laboratory value from routine laboratory testing

8.2.3.5 Data Analysis

The data for the current analyses were exclusively taken from the DWH via its graphical user interface and the subqueries defined in Table 8.27. Analysis was done using the software package R [3]. Presence of heart failure was captured based on data of individual hospital visits of individual patients (i.e., one case); each patient was counted once per year. The proportions of true positive, false positive, true negative, and false negative matches were calculated. Further, precision, sensitivity, and the F1 score were computed to provide integrated measures of the accuracy of the match between automated heart failure detection and the reference standard. Precision (also called positive predictive value) describes the share of algorithmically labeled heart failure patients who indeed

Table 8.27: Automated advanced data warehouse interrogation to detect heart failure.

Subquery name[a]	Search terms (including synonyms and word parts in German language)	HF detection algorithms				
		M_{ICD}	M_{Expoert}	$A_{\text{Precision}}$	$A_{\text{Sensitivity}}$	A_{F1}
Echo-EF ≤ 45	lvef ≤ 45		✓	✓		✓
Echo-EF ≤ 50	lvef ≤ 50					
ICD-Any-HF	I13.2, I13.0, I11, I50 (including more specific diagnosis)	✓	✓		✓	✓
ICD-Any-HF	I13.2, I13.0, I11, I50 (including more specific diagnosis)	✓	✓		✓	✓
Text-Heart-Failure	herzschw* OR herzinsuff*		✓	✓	✓	✓
Text-Cardiac-Decompensation	(card| kard| kardiopulmo| cardiopulmo| hydrop| herz| link)* dek*		✓	✓	✓	
Text-Systolic-Failure	di (card| kard| cardiopulmo| kardiopulmo| hydrop| herz)* pumpvers* OR vorwärtsversag* OR (kard| card)* schock*		✓	✓	✓	✓
Text-Dilated-Cardiomyopathy	dilat* (kardiomy| cardiomy)* OR dcm			✓		
Text-NYHA	nyha	✓		✓		
Text-Left-HF	(kard|linksherz)* insuff*	✓		✓		
Text-Right-HF	(rechtsherz| diast)* insuff*	✓				
Text-Reduced-LV-Function	(komp| reduzierte| eingeschränkte| verminderte)* (link| sys)* ventr* funkti* OR (komp| reduzierte| eingeschränkte| verminderte)* lv funkti* OR ventrik* (komp| reduzierte| eingeschränkte |verminderte)* funkti* OR sys* dysfunkti*	✓		✓		
Text-Reduced-RV-Function	(komp| reduzierte| eingeschränkte| verminderte)* (rechts| dias)* ventr* funkti* OR (komp| reduzierte| eingeschränkte| verminderte)* rv funkti* OR dias* dysfunkti*	✓				
Text-Pulmonary-Edema	lung*ödem* OR lung*stau* OR stau*lung*	✓				
Text-Left-Ventricular-Hypertrophy	lv hypertr* OR link*ventr* hypertr*				✓	
Text-Left-Atrial-Enlargement	(link*vorho*| link*atri*| la) (verg| dilat| hypertr)*				✓	
Text-Diastolic-Dysfunction	(komp| eingeschr| vermind)* (ventr| dias)* funkti* OR dias* (dysfunkti*| relax*stör*)				✓	
Lab-NT-proBNP ≥ 1000	nt-probnp (pg/mL) ≥ 1000					
Lab-NT-proBNP ≥ 3000	nt-probnp (pg/mL) ≥ 30					

have heart failure, out of all algorithmically labeled heart failure patients; e.g., a precision of 100% means that all selected patients truly have heart failure. Sensitivity (also called true positive rate or recall) is the share of algorithmically labeled heart failure patients who indeed have heart failure, out of all patients with heart failure; e.g., a sensitivity of 100% means that all patients with heart failure are selected. The F1 score is the harmonic mean of precision and sensitivity and is used as the overall accuracy measure in this analysis; e.g., an F1 score of 100% means that exactly the patients who truly have heart failure are selected and an F1 score of 85% would describe the prevalence of heart failure with an estimated error of 15%. Measures of any permutation of the subqueries were computed in R, utilizing single DWH exports of each subquery, to maximize the F1 score and aiming to yield a precision and sensitivity of at least $> 90\%$ but still have a corresponding sensitivity and precision $> 60\%$, respectively. Frequencies and percentages were used to present aggregated data across periods under study.

8.2.4 Results

From 2000 to 2015, 110,742 individual patients were treated at and received a discharge letter from the Department of Medicine I of the Würzburg University Hospital. Of these patients, 71,625 had at least one inpatient visit. After splitting the 16-year period into four 4-year periods (i.e., 2000–2003, 2004–2007, 2008–2011, and 2012–2015), respective counts for all patients (inpatients) were 25,753 (17,941), 32,301 (19,592), 37,300 (21,743), and 42,119 (25,692).

8.2.4.1 Verification of the Heart Failure Detection Algorithm

Table 8.28 presents the performance characteristics derived from cross-validating the heart failure detection algorithms (defined in Table 8.27) against the reference standard set, i.e., the 1042 manually labeled inpatients, in whom 222 subjects (21%) were identified by the expert to suffer from heart failure.

The algorithm that was solely based on ICD codes (MICD) resulted in a good precision of 94%, but a low sensitivity of 50%, and a F1 score of 65%. The low sensitivity illustrates the low share of patients with heart failure detected by this algorithm. The missing 6% to a precision of 100% were caused by seven out of the 1042 patients who had a heart failure-related ICD diagnosis, but were not labeled to have heart failure. The expert-specified heart failure algorithm (MExpert) improved the detection rate and resulted in a precision of 76%, a sensitivity of 87%, and a F1 score of 81%. Divergent conclusions between the algorithm and the reference standard were found in 89 cases, with 60 patients mistakenly classified to have heart failure and 29 patients mistakenly classified to not have heart failure. Since the manually defined algorithms resulted in low scores, the algorithm MExpert was refined further. Three algorithms were developed and tested, each optimizing certain aspects of diagnostic accuracy: APrecision aimed to increase the reliability of the classification as heart failure patient (reduced false

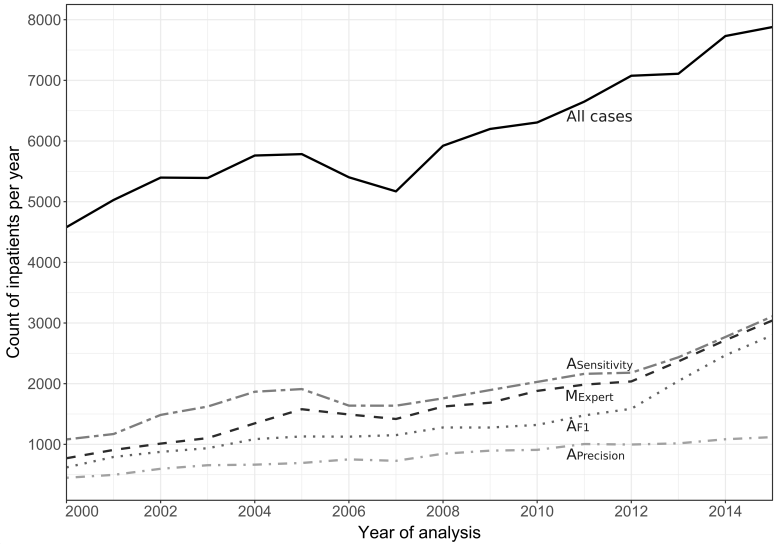


Figure 8.7: **The solid line indicates all patients; each patient is counted once per year.** Intermittent lines represent patients with heart failure identified using different automated heart failure detection algorithms: M_{Expert} originates from the variable set pre-specified by the clinical expert; $A_{\text{Precision}}$ optimizes count of false positives; $A_{\text{Sensitivity}}$ optimizes count of false negatives; A_{F1} optimizes overall accuracy (for details refer to Methods).

positives), ASensitivity aimed to reduce the number of patients not classified as heart failure patient (reduced false negatives), and AF1 aimed for an overall improved accuracy of the classification as heart failure patient (balanced precision and sensitivity). The algorithm with the highest F1 score (i.e. AF1) resulted in a precision of 89%, a sensitivity of 84%, and an F1 score of 86%. The missing 14% to an F1 score of 100% was caused by 59 false matches that originated from "borderline cases" with limited or unclear textual information that opened more room for interpretation and misclassification for both computer and expert. Some errors were the result of missing data in the DWH, e.g., missing LVEF values or terms that indicate negations of heart failure in the discharge letter.

8.2.4.2 Prevalence of Heart Failure

Figure 8.7 illustrates the annual frequencies of all inpatients of the Department of Medicine I with a discharge letter and the subgroup of patients with heart failure identified by the automated algorithms described in Table 8.27. Across the entire period, AF1 identified 18,167 unique patients with heart failure. In the year 2000, the count of patients with heart failure started at n = 620 and showed an average annual gain of 9.3% over the entire period. After the year 2012, the annual gain appeared to accelerate from 7.4% before 2012 to 17.1% thereafter. By contrast, the average annual gain of all inpatients

Table 8.28: **Performance of automated HF detection algorithms versus reference standard.**
Ref reference standard defined by a heart failure specialist inspecting the documents, *HF+* heart failure present, *HF-* heart failure absent, *TP* true positive, *FP* false positive, *FN* false negative, *TN* true negative. Precision: TP/(TP+FP), sensitivity: TP/(TP+FN), F1: 2×(precision×sensitivity)/(precision+sensitivity). For details refer to Section"Methods" 8.2.3

Algorithm		**N**	**Ref. HF+** $n = 222$	**Ref. HF-** $n = 820$	**Precision (%)**	**Sensitivity (%)**	**F1 score (%)**
M_{ICD}	HF+	117	110 (TP)	7 (FP)	94	50	65
(for comparison)	HF-	925	112 (FN)	813 (TN)			
M_{Expoert}	HF+	253	193 (TP)	60 (FP)	76	87	81
(expert specified)	HF-	789	29 (FN)	760 (TN)			
$A_{\text{Precision}}$	HF+	140	134 (TP)	6 (FP)	96	60	74
(precision optimized)	HF-	902	88 (FN)	814 (TN)			
$A_{\text{Sensitivity}}$	HF+	286	204 (TP)	82 (FP)	71	92	80
(sensitivity optimized)	HF-	756	18 (FN)	738 (TN)			
A_{F1}	HF+	209	186 (TP)	23 (FP)	89	84	86
(F1 score optimized)	HF+	833	36 (FN)	797 (TN)			

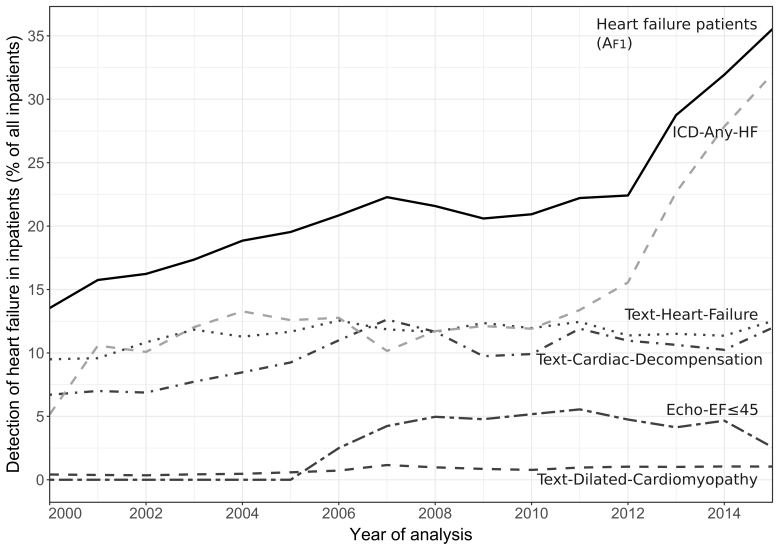


Figure 8.8: **Detection of heart failure in inpatients using different approaches.** The solid line indicates the prevalence detected when applying the automated algorithm A_{F1} (for details refer to Methods). Intermittent lines indicate detection using ICD codes or other information tags that dominantly contributed to the detection of heart failure. Each patient entered analysis only once per year; if patients attended the hospital multiple times, the first case of each patient per year was used. The percentage of all inpatients is depicted.

was 3.4%. The application of APrecision and ASensitivity resulted in 10,786 and 25,084 patients identified with heart failure, respectively.

Several patients were treated multiple times over the years, which resulted in sums of unique patients per year reported in Fig. 1 that were higher than the above-reported sum of unique patients of all years. This included 3115 unique patients with 4583 heart failure-related re-hospitalizations after an initial heart failure-related hospitalization within the entire period. A characterization of inpatients with heart failure identified by the application of AF1 is presented in Table 8.29 for the four 4-year periods from 2000 to 2015, grouped by age and sex.

Each search term of the detection algorithm AF1 contributed with varying impact to the identification of heart failure. In the reference standard, the largest contributions emerged from “Text-Heart-Failure” (59% capture rate), “ICD-Any- HF” (56%), and “Text-Cardiac-Decompensation” (53%). Further important contributors were “Echo-EF $\leq$ 45” (24%) and “Text-Systolic-Failure” (4%). In the case of “ICD-Any- HF”, for example, this means that 44% of all patients with heart failure did not have an ICD code indicative of heart failure. The contribution of the individual search terms to the overall analysis varied substantially over the years, as presented in Figure 8.8.

This illustration presents the search terms of the first heart failure related hospitalization per patient and year. Noteworthy is the relatively small contribution of the term LVEF

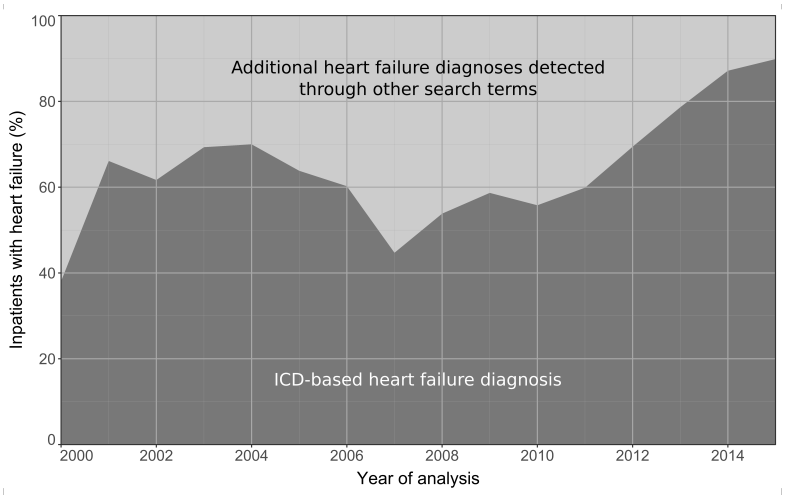


Figure 8.9: **Detection of heart failure via related ICD codes.** Detection of heart failure via related ICD codes (dark gray) and the additional detection through other search terms[7](light gray), in inpatients with heart failure across the entire sampling period (years 2000-2015). The percentage of patients found via selective ICD code search increased in recent years, which might be explained by the foundation of the Comprehensive Heart Failure Center Würzburg in the year 2010, i.e. a facility devoted to the integration of research and care of patients with heart failure.

$\leq$ 45% from echocardiography, although echocardiography was frequently performed: in 58% of patients on cardiologic ward and 29% of patients on other wards of internal medicine.

Figure 8.9 illustrates the contribution of ICD codes to the detection of heart failure in contrast to the additional contribution of other search terms (text/echo terms) over the entire period using the AF1 algorithm. Within the years 2000–2015, the overall share of patients with heart failure identified by ICD codes was 69% of the total sample of patients with heart failure, which means that 31% of patients with heart failure remained undetected throughout the entire period.

8.2.4.3 Comorbidities and Heart Failure

We further analyzed, whether the comorbidity profile differed in subjects in whom presence of heart failure was identified via ICD codes versus subjects in whom heart failure was identified via additional sources of the DWH. Table 8.30 lists the most frequent comorbidities reported in the 18,167 patients with heart failure detected by the AF1 algorithm in the mutually exclusive subgroups "detected by ICD codes" or "detected by other search terms" (specific DWH interrogation other than ICD code). Reported

[7]Executed via application of the automated algorithm A_{F1} (for details refer to Methods).

Table 8.29: **Frequencies of all patients with heart failure identified by the $A_{\mathbf{F1}}$ algorithm.** Algorithm depends on age group, gender and the 4-year periods (each with unique patients per time period).
Data are count (percent). Inpatients with heart failure subdivided by sex and age categories. Some patients were admitted more than once; on average, one patient contributed 1.8 patient cases

	All patients with heart failure $n = 18,167$	Period			
		2000–2003 $n = 3100$	2004–2007 $n = 4244$	2008–2011 $n = 4905$	2012–2015 $n = 8197$
Sex, n (%)					
Male	10,636 (59)	1772 (57)	2466 (58)	2929 (60)	4945 (60)
Female	7531 (41)	1328 (43)	1778 (42)	1976 (40)	3252 (40)
Age category, n (%)					
≤45 years	702 (4) [67% male]	108 (3) [67% male]	173 (4) [70% male]	156 (3) [68% male]	301 (4) [64% male]
46–54 years	1294 (7) [74% male]	167 (5) [72% male]	287 (7) [76% male]	317 (6) [73% male]	589 (7) [75% male]
55–64 years	2766 (15) [73% male]	474 (15) [73% male]	571 (13) [77% male]	637 (13) [74% male]	1268 (15) [70% male]
65–74 years	5169 (28) [67% male]	956 (31) [65% male]	1262 (30) [67% male]	1311 (27) [68% male]	2062 (25) [68% male]
≥ 74 years	9163 (50) [51% male]	1431 (46) [46% male]	2004 (47) [47% male]	2540 (52) [52% male]	4052 (50) [54% male]

comorbidities were identified by their respective ICD code. Patients with ICD-coded comorbidities more frequently also had an ICD code for heart failure. Of note, the subgroup identified without ICD codes appeared to have a slightly lower burden of comorbidity.

8.2.5 Discussion

The current analysis sheds new light on the magnitude of underestimation of heart failure prevalence in hospitalized patients. Identifying patients with heart failure from the hospital information system solely based on ICD-coded discharge diagnoses substantially underestimated the "true number" that could be gleaned after adding specific text searches and echocardiographic parameters to the search profile.

We observed a large degree of heart failure underestimation when using ICD codes only: within a single year it was up to 55% (average 31%) lower than the "true number" of heart failure. The last years of the analysis showed a trend towards better patient identification. The decreasing gap of underestimation became considerably smaller over time and indicates that coding strategies as diagnostic and therapeutic algorithms may indeed affect the "prevalence" of a disease. The detection gap came together with a marked increase in the absolute frequency of encoding ICD diagnoses for heart failure, starting with the year 2012 for inpatients. Furthermore, the percentage of patients with heart failure increased from about 15% in 2000 to about 35% in 2015. Reasons for such high proportions might be that we only included patients from the Medical Department I (hosting wards for intensive care, cardiology, pulmonology, endocrinology, nephrology), where heart failure is a frequent diagnosis, but also identified patients having heart failure as a secondary or tertiary diagnosis. Another explanation for these developments might be that the Comprehensive Heart Failure Center was founded at the Würzburg University Hospital in the year 2010, i.e., a facility devoted to the integration of research and care of patients with heart failure. This spurred numerous structural and research projects involving several hospital departments, led to a higher degree of awareness for the heart failure syndrome, and ultimately might not only have increased the count of patients with heart failure admitted to the hospital, but also improved the coding ratio.

Verifying the diagnosis of heart failure patients based on physician claims or hospital data has been attempted earlier [5, 7, 77, 116, 140, 181, 190, 194, 202] . However, most of these studies focused on confirming or refuting the diagnosis of heart failure with the help of experts in subjects pre-identified via several variants of ICD codes, via study inclusion/exclusion criteria or manual screening. Subsequently, reported identification figures were fairly precise (i.e., yielded high precision). Frolova et al. for example aimed to verify ICD-based diagnosis of acute heart failure amongst patients admitted to the hospital with suspected acute heart failure [77] and found a precision of 93% (sensitivity of 76%) leading to an F1 score of 84%. In contrast, we aimed to identify "true heart failure" amongst all-comers, i.e., without an increased pre-test likelihood for the presence of such diagnosis. As expected, performance of the algorithm relying solely on ICD-based identification (= MICD) was worse in our data set (F1 score 65%). There are few studies

Table 8.30: **Frequency of comorbidities in inpatients with heart failure.** Results detected by ICD codes and additionally detected via data sources provided by the data warehouse.
Comorbidities with ICD codes in descending order by prevalence in the total sample. Numbers are count (%).
HF heart failure, *COPD* chronic obstructive pulmonary disease.
[a] Heart failure detection was done using the automated algorithm A_{F1} (for details refer to Section "Methods" 8.2.3)

	All patients with heart failure[a] n=18,167	Heart failure detected by ICD codes n=13,361	Heart failure detected on top via other sources n=4806
Essential primary hypertension (I10)	11,335 (62.4)	8441 (63.2)	2894 (60.2)
Chronic ischemic heart disease (I20, I25)	9025 (49.7)	6801 (50.9)	2224 (46.3)
Heart valve disorders (I34-I39)	3495 (19.2)	3045 (22.8)	450 (9.4)
COPD (J44)	2874 (15.8)	2392 (17.9)	482 (10.0)
Acute myocardial infarction (I21)	2837 (15.6)	1990 (14.9)	847 (17.6)
Anemia (D60-D64)	1942 (10.7)	1529 (11.4)	413 (8.6)
Cardiomyopathy (I42)	1283 (7.1)	1041 (7.8)	242 (5.0)
Depression (F32, F33)	1031 (5.7)	848 (6.3)	183 (3.8)
Cerebral hemorrhage, infarction, stroke (I61, I63, I64)	387 (2.1)	313 (2.3)	74 (1.5)
Sleep apnea (G47.3)	287 (1.6)	240 (1.8)	47 (1.0)
Kidney failure (N19)	196 (1.1)	156 (1.2)	40 (0.8)

focusing on all-comers for detecting heart failure [182, 183, 202], all reporting lower F1 scores (82, 53–67, 80%, respectively) compared to our analysis (86%).

Applying text extraction methods to detect heart failure has rarely been attempted. Meystre et al. [157], for example focused on the information extraction of a few highly selected parameters (e.g., LVEF value and medication) from texts in contrast to an overall detection of heart failure. They utilized a pre-defined data set of heart failure patients in contrast to all-comers and, subsequently, received high F1 scores of up to 99% for single parameters (e.g., LVEF value). While interesting to demonstrate feasibility, such concepts do not mirror clinical reality. No related work was found utilizing information extraction to detect heart failure in all-comers.

Another major finding of the current study is that readily available information from the hospital information system considerably improves the identification of heart failure patients beyond the traditional identification via ICD codes. The option to enrich the search strategy by clinical variables supporting or denying the presence of heart failure is not new, but a variety of problems may impede its implementation: (1) the information is only selectively documented in clinical routine; (2) the desired information is stored in a non-structured format and appropriate data extraction tools are unavailable or unreliable; (3) the information is stored in a structured format but cannot be accessed for analysis (e.g., because it is stored in dedicated research data bases); (4) the quality of the stored data (structured or non-structured) is unreliable; (5) the information behind variables (meta data) is highly flexible but cannot be connected to the source data; (6) the individual patient and the corresponding cases of a patient (repeat hospitalizations) cannot reliably be discerned. Our approach utilized the hospital's clinical DWH as described earlier [51, 112, 228] and integrated the full spectrum of digital information collected per patient in the hospital information system.

The most elaborated part in providing a DWH is the implementation of the data extract–transform–load (ETL) process to transfer data from the information systems to a unified database, which often—but not always—requires to consider local peculiarities depending on the available information systems. We implemented this process for most of the information stored within our systems, be it structured or unstructured. Our DWH query system utilizes a locally developed add-on [51] to provide text search functionality to DWH systems. This add-on could be instantaneously added as an extension to the often utilized i2b2 DWH system [162] or, with little extra work, to other similar DWH systems. Importantly, these tools were tested and optimized across their repeat utilization for various studies, including data validation against the primary systems after DWH extraction [228].

The combined use of these interfaces and generation of automated detection algorithms markedly improved the identification of patients with heart failure. We found better albeit still unsatisfactory accuracy when employing the algorithm based on "clinical information" alone (i.e., the algorithm MExpert). We therefore tested other, data-optimized algorithms, and observed another major improvement of heart failure detection: the algorithm AF1 optimized precision and sensitivity and yielded the overall best results.

Importantly, our approach allowed to adjust and optimize the detection algorithm for different scenarios or use cases, e.g., to identify potential study participants via the algorithm (and thus enabling a study nurse to fine-tune the results) ASensitivity might yield best results. For the scenario of a post hoc analysis, AF1 or APrecision might be the preferred solutions. Interestingly, the NT-proBNP queries "Lab- NT-proBNP $\geq$ 1000" and "Lab-NT-proBNP $\geq$ 3000" (see Table 8.27) were not selected by the permutation analysis for any algorithm. This may be explained by the collinearity contained in other terms indicative for heart failure; e.g., for the AF1-algorithm: "Echo-EF $\leq$ 45", "ICD-Any-HF", "Text- Heart-Failure", "Text-Cardiac-Decompensation", and "Text-Systolic-Failure". We also considered using Framingham heart failure signs and symptoms [154] for detection of heart failure (see [25, 240]) either alone or in combination with borderline echocardiographic data, but were unsuccessful in demonstrating superior precision and sensitivity.

Our analyses support the notion that comorbidities of heart failure may also affect coding practices for heart failure. When comparing the presence of common comorbidities with the detection of heart failure via ICD-based versus alternative approaches, the differences where highly significant for almost all conditions. Interestingly, a sizeable proportion of patients with heart failure received an ICD code for the respective comorbidity, but not the ICD code for heart failure itself. This might indicate that heart failure was not at the focus of their hospitalization visit and not a dominant contributor from the reimbursement perspective. From a health policy perspective this means that many patients with heart failure as a concomitant condition leave the hospital without being reported to statutory data banks as heart failure patients. This not only adds to the detection gap, but also constitutes a major information gap for care providers after hospital discharge who play a key role in the treatment of heart failure in Germany [221].

8.2.5.1 Limitations

A limitation of this study is that the reference standard was only defined by a single cardiologist with long-standing experience in heart failure instead of multiple experts. The count of true heart failure patients may vary considerably depending on the care setting, the type of catchment area, and numerous other influencing factors. Hence, absolute counts are likely not directly comparable between hospitals. Similarly, the successful implementation of adapted detection algorithms needs to be confirmed before our results may become generalizable to other hospitals, both in Germany as internationally.

8.2.6 Conclusions

Coded discharge diagnoses substantially underestimate the number of heart failure patients compared to the added information available within discharge letters and echocardiographic reports. Therefore, statistics about heart failure solely based on ICD codes might be misleading. The degree of underestimation might vary substantially

across case types (inpatients versus outpatients) and the course of subsequent years. The latter might be influenced by internal factors, e.g., improved coding practices, and/or external factors, e.g., the set up of specialized centers as the Comprehensive Heart Failure Center Würzburg.

8.3 Consistency of Diagnoses

Ad hoc IE can be used to evaluate data quality by exploiting redundancy, e.g. between textual and structured data. In this case study the consistency of diagnoses is assessed by comparing structured ICD-10[8] encoded diagnoses with diagnoses mentioned in the diagnosis section of the discharge letter. These diagnoses are extracted with ad hoc IE from texts using terms of the Alpha-ID list and with synonyms generated by a novel method to obtain synonyms from discharge letter.

8.3.1 Introduction

Data quality is an important topic of all information systems. The reliability of a data warehouse strongly depends on the quality of its data. Syntactic checks can ensure the correct data types, e.g. "number" for "age", which guarantees a basic level of data quality. However the semantic correctness of data cannot be rated with such methods. Consistency analysis can reveal discrepancies within the data. In a case study at the University Hospital of Würzburg (UKW), we assess the data quality of the clinical Data Warehouse by reviewing the consistency of diagnosis. We use two different data sources that should contain the same information and compare them to each other, assuming them to match. Diagnoses are encoded with ICD-10 as structured information. This information is mainly used for billing-relevant processes with *diagnosis related groups* (DRG) numbers. The medically relevant document is the discharge letter, containing diagnoses listed in a separate section. The encoded ICD-10 diagnoses should be contained in the discharge summary. Not every diagnosis described in the discharge letter must be coded, but it is expected for the most severe diagnoses. The amount of matched diagnoses reflects the quality of the data, increasing with their correspondence: The higher the correspondence of the diagnoses, the higher is the quality of the data.

8.3.2 Methods

A prerequisite to match diagnoses is the information, if a diagnosis is present in both data sources. Diagnoses encoded with ICD-10 are structured data whereas diagnoses in the discharge letter must be extracted first. This is done with ad hoc Information Extraction: The input for queries are textual descriptions (synonyms) of the diagnoses to be assessed. In other words: The names of diagnoses and their synonyms are queried in the diagnosis section of the discharge letter, in order to extract and to assess the existence of the diagnoses.

With this method, consistency checks can be made for a huge amount of diagnoses, since ad hoc IE enables the extraction of diagnoses from texts and only their synonyms required as resource. Alpha-ID[9] aims to provide everyday language diagnosis names for ICD-10

[8] International Classification of Diseases

[9] `https://www.dimdi.de/dynamic/de/klassifikationen/icd/alpha-id/`, accessed: February 2019

diagnostic codes. *Alpha-ID* is described in detail in Section 2.2.5.7. As a second approach, we generated synonyms of diagnoses using the resources of the Data-Warehouse.

8.3.2.1 Synonym Generation

The synonyms are generated using the diagnosis section in the discharge letters and the structured diagnoses encoded with ICD-10. All texts are first preprocessed before the synonyms for the associated diagnoses are gathered. Therefore we defined a pipeline with text as input and synonym candidates as output:

1. Filtering current diagnoses: Removing historical, hypothetical and non patient concerning diagnoses
2. Lexical analysis: tokenizing, stemming, stop word removal
3. N-gram building: adjacent words are combined
4. Filtering relevant tokens
5. Synonym generation
   a) Candidate generation
   b) Synonym selection

Filtering current diagnoses in the text. The diagnosis section contains the current findings and their causes. In addition it may contain important medical events in the past or even of diseases of relatives.

Hence, we filter the text with the context algorithm (see Section 4.3.4) and remove all historical, hypothetical and non patient concerning diagnoses. This also includes minor subsections like "Vordiagnosen" (engl. *preliminary diagnoses*) and "CVRF" (Kardiovaskuläre Risikofaktoren, (engl. *cardiovascular risk factors*)).

Lexical analysis: tokenizing, stemming, stop word removal. Afterwards a lexical analysis is conducted to the texts. A tokenizer splits up the text, a stemmer reduces the tokens (words) to basic versions and a stop word remover eliminates unnecessary words. (*Note:* The terms "token" and "word" are used synonymously.)

N-gram building. Some diagnosis are compounded of two or more words, e.g. "diabetes mellitus" or "kardiale Dekompensation" (engl. *cardiac decompensation*). This problem is addressed with n-grams: N-grams are built for tokens next to each other by concatenating these words to one token and adding it to the text. The transformation of the sentence "chronische kardiale Dekompensation" (engl. *chronic cardiac decompensation*) can be seen in line (2) for $n = 2$. The new compounded n-grams "chronische_kardiale" and "kardiale_Dekompensation" are added to the sentence.

Input:	chronische kardiale Dekompensation	(1)
Output:	chronische kardiale chronische_kardiale Dekompensation kardiale_Dekompensation	(2)

Filtering relevant tokens. Some tokens or n-grams clearly are no synonyms and are removed in the next step. Two filter operations are applied: (1) Tokens or n-grams with very few occurrences are eliminated and (2) Tokens or n-grams that do not contain a noun are removed as well. The first procedure removes all non relevant tokens, which are a lot according to Zipf's law [178], the second filter deletes all adjectives, verbs and other tokens and n-grams without a noun, because they cannot be a synonym.

Previously, a dictionary was created that contains all tokens and n-grams of all documents and their number of occurrence. The dictionary also contains the most likely part of speech tag for each token. A part of speech tagger labeled all words in numerous discharge letters and stored the most common tag for every token in the dictionary.

The first step removes all tokens and n-grams in the current pipeline that have a very low number of occurrences in the corpus (all documents). In the second step, the part of speech tag is looked up for each token in the pipeline. Tokens that are no nouns are dropped. N-grams are passed through, if at least one token of the n-gram is a noun. This look-up procedure with a dictionary is chosen because tagging the words in the diagnosis section would cause to many errors due to the telegraphic language style. These filter steps drastically reduces the tokens and the following computation times.

Synonym generation. Two approaches are tested to generate the synonyms.

Counting words. This method counts words (tokens and n-grams) for each diagnosis. The pairwise occurrence of a word and a diagnosis is counted. If a word occurs in combination with a diagnosis in many patient cases, it has a high recall. Words only occurring with one diagnosis and with no other diagnoses have a high precision.

Two settings are evaluated: (1) Reliable synonyms are selected with a high-precision-setup. Words with minimum precision of 0.9 and recall ≥ 0.05 are considered to be synonym. These words are relatively trustworthy, but rare and maybe exotic. (2) Synonym candidates are generated with a high F1-score (> 0.1) (combined measure of precision and recall). This setting selects word that frequently used with a F1 > 0.1 and a precision of > 0.5.

Word embeddings. The second approach uses word embeddings to compute alternative diagnosis names. The ICD-10 diagnoses are added as special tokens to the diagnosis section by inserting them after every 15 tokens. All words, including the ICD-10 tokens, are mapped into a low dimensional vector space with the use of GloVe[10] [172]. GloVe

[10] `https://nlp.stanford.edu/projects/glove/`, accessed January 2019

(Global Vectors) is an algorithm that maps words into a meaningful vector space aiming to group them by their semantic. Words with a high semantic similarity are close to each other. The statistical model uses the co-occurrences of words for the transformation. Each word can represented as a vector. The similarity of each token (T) to the special ICD-10 diagnosis token (D) is computed with the cosine similarity as (sim). The highest similarity is indicated by the value 1, the lowest one by the value 0. Furthermore, the similarity of a token to all other diagnoses is computed as well. The maximum similarity to another diagnosis is selected as ($dist$). The final score of a token (T) for a diagnosis (D) is calculated as: $score(D,T) = (k * sim + (1 - dist))/(k + 1)$.

k is a factor defining the importance of the similarity/proximity of a token to a diagnosis in contrast to the distance to other diagnoses. k is adjusted to 5 after a several experiments. Synonyms are selected with a score > 0.6 for the high precision setting and with a score > 0.55 for the high recall setting.

Baseline. The diagnosis names are used as baseline for the synonym generation evaluation. The IDC-10 terminology provides the names for all diagnoses. These are refined to achieve a higher recall. E.g "I10-I15 Hypertonie [Hochdruckkrankheit]" is transformed to "Hypertonie" and "C90.0 : Plasmozytom [Multiples Myelom] (In kompletter Remission)" is mapped to "Plasmozytom". The resulting words are queried with wildcards like "*Hirn*infarkt*" that matches the words "Hirninfarkt", "Hirnstamminfarkt" and "Kleinhirninfarkt".

The diagnoses are hierarchically ordered in the ICD-10 terminology. Diagnoses with a deeper hierarchical level are finer grained diagnoses (hyponym) and are used as well.

8.3.2.2 Evaluation Setup

Synonyms are generated based on diagnosis sections of 100,000 discharge letter, which are randomly picked from the Würzburg University Hospital. The hospital consists of various special departments, like the medical department or the surgery. Most diseases are commonly treated in one department like all fractions in the surgery department.

An assumption is, that physicians in the specialized department use other descriptions and synonyms for diagnoses than not expert physicians of other departments. For example, the word "infarction" on the medical department I refers to a myocardial infarction and to a cerebral infarction in the medical department II. To test this assumption, synonyms are also generated only using discharge letters the of specialized department, were they are commonly treated.

All synonym candidates that are generated this way, are reviewed manually. Incorrect candidates are removed from the list.

The evaluation is conducted with 20 diagnoses. In order to cover a broad spectrum of diseases, diagnoses of different chapters of the ICD-10 terminology are selected, covering

Table 8.31: Department of the Würzburg University Hospital and their common subject.

Abbreviation	Department name	Subject
Chir1	Surgical Clinic I	general surgery
Chir2	Surgical Clinic II	trauma surgery
Med1	Department of internal medicine I	diseases of circulatory system
Med2	Department of internal medicine II	neoplasms
Neuro	Neurological Clinic	neurological diseases
UKW	Würzburg University Hospital	all subject

the diseases of nervous system, circulatory system, respiratory system, digestive system as well as injuries.

Each diagnosis evaluation task queries the baseline tokens, the generated high precision tokens and the generated high F1 tokens. The number of hits (diagnoses extracted from text) are compared with the corresponding encoded ICD-10 diagnoses. Precision, recall and F1 values are computed based on these results. A precision value of 1 means that for each diagnosis that is extracted, there is also a ICD-10 diagnoses encoded. A recall value of 1 means that all encoded ICD-10 diagnoses are contained in the text and extracted. That evaluation run is performed with both synonyms lists: (1) Terms that are generated using discharge letters of the entire hospital and (2) terms that just consider texts of the respective specialist department. Table 8.31 shows the specialist departments and their main subject area.

8.3.3 Results

This section presents the generated synonyms, the result of the consistency tests and an error analysis.

8.3.3.1 Synonym Generation

For each diagnosis, synonyms are listed for the entire Würzburg University Hospital (UkW) and the department, where the disease is mainly treated. Synonym candidates produced with the high-precision-method are in bold type, non bold tokens are generated with the-high-F1-procedure. Tokens that are striked out are no synonyms and have been removed manually.

G20 Parkinson disease

- UKW: **Rigide Typ**, **Parkinson Rigid**, Parkinson, **Yahr Stadium**, ~~Äquivalenztyp~~, ~~Wirkfluktuation~~, ~~idiopathisch Morbus~~
- Neuro: **Morbus**, **Yahr Typ**, **Rigid Typ**, **Parkinson**, **Parkinsonsyndrom** ~~Äquivalenztyp~~, ~~Stadium~~, ~~Wirkfluktuation~~,
- GloVe: **Parkinson**, Yahr Stadium

G25 Other extrapyramidal and movement disorders

- UKW: **Stiff Person Syndrom**
- Med1: **Stiff Person Syndrom**
- GloVe: ~~Stiff~~

I21 Acute myocardial infarction

- UKW: **STEMI**, **Hebungsinfarkt**, NSTEMI, Myokardinfarkt, RIVA Verschluss, ~~BMS~~, ~~Vorderwand~~, ~~Hinterwand~~, ~~PCA Implantation~~
- Med1: **STEMI**, **NSTEMI**, **Hebungsinfarkt**, Myokardinfarkt, RIVA Verschluss, Hebungsmyokardinfarkt, ~~BMS~~, ~~Vorderwand~~, ~~Hinterwand~~, ~~Metal Stent~~, ~~PTCA Implantation~~
- GloVe: ~~**PTCA**~~, **STEMI**, **NSTEMI**, ~~RCS~~, ~~RCA~~, ~~BMS~~, ~~KHL~~, ~~RIVA~~, ~~CVRF~~, ~~Stent~~, ~~Stenos~~, ~~Gefäss~~, ~~Verschluss~~, ~~Vorderwand~~, ~~Hinterwand~~, ~~Implantation~~, Hebungsinfarkt, ~~Stentimplantation~~, ~~Koronarangiographi~~

I71 Aortic aneurysm and dissection

- UKW: Aortenaneurysma, Aneurysma Aorta, Bauchaortenaneurysma, ~~Querdurchmess~~
- Chir1: **Bauchaortenaneurysma**, AAA, Endoleak, Aneurysmas, Aortenaneurysma, Aneurysmaausschaltung, ~~Aorta~~, ~~Querdurchm~~, ~~Stentgraft~~, ~~qQerdurchmess~~, ~~Angiographi CT~~
- GloVe: ~~**Querdurchmesser**~~, **Bauchaortenaneurysma**, Aortenaneurysma, Aortenaneurysmas, ~~Bild~~

I74 Arterial embolism and thrombosis

- UKW: **Ischämie links**, Akute Ischämie, Ischamie rechts, komplette Ischämie, inkomplette Ischämie, embolischer Verschluss
- Chir1: **Ischämie komplett links**, Embolie, Ischämie, mbolischer Verschluss
- GloVe: **Ischämie**

J09-J18 Influenza and pneumonia

- UKW: Pneumonie
- Med1: Pneumonie, pneumogene Sepsis, Stauungspneumonie
- GloVe: **Pneumonie**, ~~Typ~~, ~~Zyklus~~, ~~Verlauf~~, ~~Stadium~~, ~~Therapie~~, ~~Vordiagnose~~, ~~Einleitung~~, ~~Nierenversagen~~, ~~Nebendiagnose~~, ~~Vorerkrankungen~~

K35 Acute appendicitis

- UKW: **Appendizitis acuta**, **perityphlitischer Abszess**, **Appendizitis phlegmonosa**, Appendizitis, Appendicitis, perityphlitischer Abszess

- Chir1: **Appendicitis**, **Appendizitis**
- GloVe: Appendizitis, ~~Tubulusnekros~~

K40 Inguinal hernia

- UKW: **EHS**, **Nyhus**, **indirekte Leistenhernie**, Leistenhernie
- Chir1: **EHS**, **Nyhus**, **Leistenhernie**
- GloVe: **Leistenhernie**, EHS

K42 Umbilical hernia

- UKW: **MU PV**, **Nabelhernie**
- Chir1: **EHS**, **Nyhus**, **Leistenhernie**
- GloVe:

K85 Acute pancreatitis

- UKW: biliäre Pankreatitis, ödematöse Pankreatitis, nekrotisierende Pankreatitis, ~~Pankreatitis unklar~~,
- Med2:

  biliäre Pankreatitis, ödematöse Pankreatitis, exsudative Pankreatitis, ~~Pankreatitis unklar~~
- GloVe: ~~Koma~~, Pankreatitis

K92O ther diseases of digestive system

- UKW: Teerstuhl, GI Blutung , gastrointestinale Blutung
- Med1: **Teerstuhl**, GI Blutung , Blutabgang, gastrointestinale Blutung
- GloVe: ~~Blutung~~, Teerstuhl

R11 Nausea and vomiting

- UKW: **Erbrechen**
- Med2: **Erbrechen**
- GloVe:

S00-S09 Injuries to the head

- UKW: **Commotio cerebri**, **Schädelprellung**, ~~Hirn~~, ~~Trauma~~, ~~Platzwund~~, Orbitabodenfraktur,
- Chir1: **Commotio cerebri**, **Schädelprellung**
- GloVe: Jochbeinfraktur, Nasenbeinfraktur, Orbitabodenfraktur

S10-S19 Injuries to the neck

- UKW: **Anderson Typ**, **Anderson Densfraktur**, ~~Anderson~~, Densfraktur, Axis Fraktur, Distorsion HWS, Atlasbogenfraktur
- Chir2: **Distorsion HWS**, Dens Fraktur, Densfraktur, ~~HWS~~, ~~HWK~~, ~~Anderson~~
- GloVe: **Densfraktur**, ~~Anderson~~

S20 Superficial injury of thorax

- UKW: **rechte Thoraxprellung**, Thoraxprellung
- Chir2: **BWS Prellung**, **Rippenprellung**, **Thoraxprellung**
- GloVe: Thoraxprellung

S42 Fracture of shoulder and upper arm

- UKW: **11 AOO**, **12 AO**, 13 AO, **suprakondyläre Humerusfraktur**, Humerusfraktur, Tuberculum, Claviculafraktur, Humeruskopffraktur, Claviculaschaftfraktur,
- Med2: **Humerusfraktur**, **Claviculafraktur links**, Claviculafraktur
- GloVe: ~~Schulter~~, ~~Fraktur~~, ~~Durchblutung~~

S50-S59 Injuries to the elbow and forearm

- UKW: 22 AO, **23 AO**, **rechte Radiusfraktur**, **linke Radiusfraktur AO**, **intraartikular Radiusfraktur**, Radiusfraktur, Unterarmfraktur, Unterarmschaftfraktur, Ulnafranktur ~~radius~~
- Chir1: **Radiusfraktur**, **Unterarmfraktur**, **Unterarmschaftfraktur**
- GloVe: ~~Flexion~~, Tibiaschaftfraktur, Unterarmschaftfraktur,

S52 Fracture of forearm

- UKW: 22 AO, **23 AO**, **rechte Radiusfraktur**, **linke Radiusfraktur AO**, **intraartikuläre Radiusfraktur**, Radiusfraktur, Unterarmfraktur, Unterarmschaftfraktur, Ulnafranktur, ~~Radius~~
- Chir1: **Radiusfraktur**, **Unterarmfraktur**, **Unterarmschaftfraktur**,
- GloVe: ~~Flexion~~, Tibiaschaftfraktur, Unterarmschaftfraktur

S72 Fracture of femur

- UKW: **31 AO links**, **31 AO rechts**, **AO Femurfraktur links**, **AO Femurfraktur rechts**, **pertrochantäre Femurfraktur** 31 AO, 32 AO, Femurfraktur, Femurschaftfraktur, ~~Grad~~
- Chir2: **Femurschaftfraktur**, **Femurschaftspiralfraktur**, **Oberschenkelschaftfraktur**, 31 AO, 32 AO, Femurfraktur, ~~Grad~~
- GloVe: ~~AO~~, ~~Reposition~~, Femurfraktur, Femurschaftfraktur, Schenkelhalsfraktur

S82 Fracture of lower leg, including ankle

- UKW: **41 AO**, 42 AO, 43 AO, **44 AO**, OSG Fraktur, Fibulafraktur, **OSG Luxationsfraktur**, Luxationsfraktur, Unterschenkelfraktur, Unterschenkelschaftfraktur, ~~Web~~, ~~Pilon~~, ~~Fibula~~,
- Chir2: **Fibulafraktur**, **Tibiaschaftfraktur**, **distale Tibiafraktur**, **Tibiaschaftspiralfraktur**, **distale Unterschenkelfraktur**, 41 AO, 42 AO, 43 AO, 44 AO, OSG Fraktur, bimalleolare OSG, trimalleolare OSG, OSG Luxationsfraktur, Unterschenkelfraktur, ~~Web~~, ~~Pilon~~, ~~Fibula~~,
- GloVe: ~~AO~~, ~~Pilon~~, ~~Fraktur~~, ~~Reposition~~

8.3.3.2 Consistency Tests

Table 8.32 presents the result of the diagnoses consistency tests with synonyms for the Würzburg University Hospital. The average precision of the baseline is 0.76 points. That means that if the diagnosis name in ICD-10 description (e.g. "adenoviral pneumonia") is written in the discharge letter, the diagnoses will be encoded with ICD-10 as structured data in about 3 of 4 cases. In about 1 of 4 cases the name of the diagnosis is written in the discharge letter, but not encoded as ICD-10 diagnosis as structured data. The recall of the baseline is 0.06, which is very low: The ICD-10 descriptions of diagnoses appear in only 6% of all cases in the text. That indicates that the retrieval of diagnoses in the discharge letter is not a trivial problem and the diagnosis are covered in the text with synonyms or hyponyms.

The Alpha-ID synonym list achieves a similar precision (0.74), but a much higher recall (0.39) than the baseline.

The high-precision-synonym-list raises the precision to 0.84 points (increase: 10%) and the recall to 0.22 (increase: 120%) on average, compared to the baseline. This leads to an F1-score of 0.32, which is a gain of 0.22 points (increase: 220%).

The high-F1-synonym-list increases the recall to 0.5 points on average, which is 7.3 times more than the baseline. There are high values like 0.92 points for "K35 Acute appendicitis" and low values like 0.02 for "R11 Nausea and vomiting". However, the precision score drops 6% to 0.72 on average, resulting to a F1 score of 0.55, which is an increase of 0.45 points (450%).

Table 8.33 presents the result of the diagnosis consistency tests with synonyms for the specialized departments. The baseline results are quite similar to the more general results of the entire hospital. The results of the high-precision-synonym-lists and high-F1-synonym-lists differ only slightly compared to UKW results, but they are consistently better.

The high-precision-setup achieves a precision of 0.89 points (3% increase to the baseline), a recall of 0.36 (500% increase) and a F1 score of 0.47 (370% increase) on average.

Table 8.32: **Results of the diagnoses consistency tests of the UKW.** Diagnoses are extracted with he baseline, the high precision and the high F1 list synonyms. P: precision, R: recall, F1: F1-score

	Baseline			**Alpha-ID**			**High Precis.**			**High F1**		
Diagnosis	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
G20 Parkinson disease	-	0.00	0.00	-	0.00	0.00	0.94	0.20	0.33	0.79	0.48	**0.60**
G25 Other movement disorders	0.58	0.07	0.12	0.45	0.25	**0.32**	0.75	0.13	0.22	0.66	0.12	0.21
I21 Acute myocardial infarction	0.67	0.02	0.03	0.84	0.60	0.70	0.90	0.37	0.52	0.83	0.75	**0.79**
I71 Aortic aneurysm and dissection	0.50	0.01	0.01	0.61	0.51	**0.55**	0.50	0.01	0.01	0.71	0.45	**0.55**
I74 Arterial embolism and thrombosis	-	0.00	0.00	0.51	0.22	0.31	0.89	0.10	0.18	0.64	0.54	**0.59**
J09-J18 Influenza and pneumonia	0.44	0.00	0.01	0.63	0.48	**0.55**	0.54	0.01	0.01	0.64	0.44	0.52
K35 Acute appendicitis	0.91	0.22	0.35	0.94	0.65	0.77	0.93	0.60	0.73	0.83	0.92	**0.87**
K40 Inguinal hernia	0.73	0.03	0.05	0.80	0.02	0.03	0.94	0.20	0.33	0.79	0.80	**0.80**
K42 Umbilical hernia	0.60	0.03	0.06	0.58	0.73	**0.65**	0.86	0.16	0.27	0.60	0.69	0.64
K85 Acute pancreatitis	0.90	0.28	0.42	0.90	0.27	0.41	0.90	0.28	0.42	0.72	0.56	**0.63**
K92 Other diseases of digestive system	0.79	0.05	0.10	0.72	0.26	0.38	0.86	0.06	0.11	0.67	0.36	**0.47**
R11 Nausea and vomiting	0.52	0.02	0.04	0.43	0.09	**0.15**	0.52	0.02	0.04	0.52	0.02	0.04
S00-S09 Injuries to the head	0.91	0.10	0.18	0.83	0.47	**0.60**	0.95	0.45	0.61	0.94	0.27	0.41
S10-S19 Injuries to the neck	-	0.00	0.00	0.90	0.24	0.37	0.92	0.08	0.15	0.85	0.33	**0.48**
S20 Superficial injury of thorax	1.00	0.04	0.09	0.89	0.28	0.43	0.94	0.09	0.17	0.89	0.30	**0.45**
S42 Fracture of shoulder and upper arm	-	0.00	0.00	0.79	0.36	0.49	0.92	0.24	0.39	0.79	0.57	**0.66**
S50-S59 Injuries to the elbow and forearm	0.92	0.02	0.04	0.87	0.40	**0.55**	0.90	0.23	0.36	0.25	0.53	0.34
S52 Fracture of forearm	1.00	0.02	0.04	0.87	0.52	0.65	0.90	0.38	0.53	0.81	0.64	**0.72**
S72 Fracture of femur	0.82	0.29	0.42	0.79	0.72	0.75	0.87	0.53	0.66	0.80	0.76	**0.78**
S82 Fracture of lower leg including ankle	0.92	0.03	0.05	0.79	0.35	0.48	0.91	0.27	0.42	0.61	0.50	**0.55**
Average	**0.76**	**0.06**	**0.10**	**0.74**	**0.39**	**0.48**	**0.84**	**0.22**	**0.32**	**0.72**	**0.50**	**0.55**

The high-F1-score-setup performs even better with a precision of 0.84 points (2% decrease to the baseline), a recall of 0.52 (766% increase) and an F1 score of 0.6 (500% increase) on average.

While the precision values move between 0.59 ("R11 Nausea and vomiting") and 0.95 ("S50-S59 Injuries to the elbow and forearm") the recall values vary in a bigger range from 0.03 ("S82 Fracture of lower leg including ankle") to 0.92 ("K35 Acute appendicitis").

8.3.3.3 Error Analysis

An error analysis is conducted for two diagnoses. 100 *false negatives* (FNs) and 100 *false positives* (FPs) are randomly selected for "I21: Myocardial Infarction" and all 62 FNs and 67 FPs of "K35: Appendicitis" are audited. The false negatives and false positives of the high-F1-synonym-list in the specialized departments are investigated. The first study exams the false negatives and checks if the encoded ICD-10 diagnoses is described as well in the diagnosis section of the discharge letter. Table 8.34 shows the results of the false negatives analysis. In 24%, the diagnosis is encoded and contained in the text and not extracted by our approach. In 54%, the diagnosis is encoded, but not described in the text. In 14% of all cases the diagnosis is described heuristically, like "Verdacht auf akute Appendicitis" (engl. *Suspected acute appendicitis*) or "Exitus letalis nach kardiagenem Schock, am ehesten durch Myokartinfarkt" (engl. *exitus letalis after cardiogenic shock, most likely due to myocardial infarction*). Heuristic diagnoses are omitted by the context algorithm by design, as in this evaluation only confirmed diagnoses should be found. In 8% of all cases it is unclear to us if the diagnosis is characterized in the text.

These 39 (24%) errors (diagnosis encoded and described, but not extracted) are further investigated and the error causes are grouped into categories. (See Table 8.35.) The most common error source is the occurrence of a new synonym that is not contained in the synonym list. Examples for new descriptions of "I21: Myocardial Infarction" are "95%ige Stenose" (engl. *95% stenosis*), "RCA Verschluss" (engl. *RCA occlusion*), "Vorderwandinfarkt" (engl. *anterior myocardial infarct*), "Hinterwandinfarkt" (engl. *posterior myocardial infarction*), "Nicht-ST-Hebungsinfarkt" (engl. *non-ST-elevation myocardial*), "ST-Hebungs Infarkt" (engl. *Elevation Myocardial*), "Herzvorderwandinfarkt" (engl. *anterior myocardial infarct*), "hochgradige RIVA-Stenose" (engl. *intense RIVA stenosis*) and "ST-Elevationsmyokardinfarkt der Hinterwand" (engl. *posterior ST-elevational myocardial infarction*).

Other errors are typos (8%), incorrect detection of detecting the section in discharge letters (the diagnosis section within was not found) (5%) and incorrect computation of the context of the synonym. Examples for that last error are: "Z.n. Reanimation bei STEMI der Hinterwand" (engl. *condition after reanimation at STEMI of the posterior wall*) "Aktuell: Ausschluss rel. Koronarstenosen bei laborchemisch NSTEMI" (engl. *Current: Exclusion of rel. coronary stenosis in laboratory chemistry NSTEMI*) "Z.n. erfolgreicher Reanimationsbehandlung nach Herz-Kreislauf-Stillstand bei Myokardinfarkt" (engl. *condition after successful resuscitation treatment after cardiovascular arrest for myocardial infarction*)

Table 8.33: **Results of the diagnoses consistency of specialized departments.** Diagnoses are extracted with the baseline, the high-precision and the high-F1-synonym-list. Dep: department, P: precision, R: recall, F1: F1-score

		Baseline			**Alpha-ID**			**High Precis.**			**High F1**		
Diagnosis	**Dep**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
G20 Parkinson disease	neuro	-	0.00	0.00	-	0.00	0.00	0.91	0.62	**0.74**	0.89	0.61	0.73
G25 Other movement disorders	neuro	0.70	0.12	0.21	0.55	0.31	**0.40**	0.80	0.21	0.33	0.80	0.21	0.33
I21 Acute myocardial infarction	med1	0.66	0.02	0.03	0.87	0.59	0.70	0.91	0.74	0.82	0.85	0.81	**0.83**
I71 Aortic aneurysm and dissection	chir1	1.00	0.00	0.01	0.81	0.68	0.74	0.91	0.22	0.35	0.79	0.73	**0.76**
I74 Arterial embolism and thrombosis	chir1	-	0.00	0.00	0.75	0.17	0.28	0.96	0.06	0.12	0.65	0.72	**0.68**
J09-J18 Influenza and pneumonia	med1	0.93	0.00	0.01	0.79	0.58	0.67	0.87	0.00	0.01	0.79	0.60	**0.68**
K35 Acute appendicitis	chir1	0.92	0.22	0.36	0.95	0.65	0.77	0.91	0.92	**0.92**	0.91	0.92	**0.92**
K40 Inguinal hernia	chir1	1.00	0.01	0.02	1	0.01	0.02	0.93	0.80	**0.86**	0.93	0.80	**0.86**
K42 Umbilical hernia	chir1	0.71	0.02	0.04	0.81	0.81	**0.81**	0.91	0.21	0.35	0.79	0.80	0.80
K85 Acute pancreatitis	med2	0.87	0.32	0.47	0.87	0.32	0.47	0.87	0.32	0.47	0.75	0.55	**0.63**
K92 Other diseases of digestive system	med1	0.84	0.07	0.13	0.78	0.28	0.42	0.87	0.14	0.24	0.72	0.45	**0.55**
R11 Nausea and vomiting	med1	0.59	0.05	0.10	0.50	0.21	**0.30**	0.59	0.05	0.10	0.59	0.05	0.09
S00-S09 Injuries to the head	chir1	0.83	0.02	0.04	0.94	0.57	0.71	0.92	0.62	**0.74**	0.92	0.61	**0.74**
S10-S19 Injuries to the neck	chir2	-	0.00	0.00	0.91	0.4	0.56	0.92	0.43	0.59	0.92	0.46	**0.61**
S20 Superficial injury of thorax	chir2	0.94	0.03	0.05	0.92	0.36	0.51	0.92	0.43	**0.58**	0.92	0.43	**0.58**
S42 Fracture of shoulder and upper arm	chir1	-	0.00	0.00	0.9	0.36	0.52	0.91	0.41	0.57	0.91	0.46	**0.61**
S50-S59 Injuries to the elbow and forearm	chir1	1.00	0.00	0.01	0.94	0.37	0.53	0.95	0.40	**0.56**	0.95	0.40	**0.56**
S52 Fracture of forearm	chir1	1.00	0.00	0.01	0.92	0.44	0.6	0.95	0.45	0.61	0.94	0.49	**0.64**
S72 Fracture of femur	chir2	0.91	0.24	0.38	0.9	0.77	**0.83**	0.91	0.24	0.38	0.91	0.24	0.38
S82 Fracture of lower leg including ankle	chir2	0.88	0.03	0.05	0.85	0.38	**0.53**	0.88	0.03	0.05	0.88	0.03	0.05
Average		**0.86**	**0.06**	**0.10**	**0.84**	**0.41**	**0.52**	**0.89**	**0.36**	**0.47**	**0.84**	**0.52**	**0.60**

Table 8.34: **Analysis of false negatives errors.** Analysis is conducted with cases of specialized department and the high-F1-synonym-list. It is evaluated if a diagnosis is described and contained in the text.

	I21: Myocardial Infarction		**K35: Appendicitis**		**Sum**	
	#	%	#	%	#	%
Diagnosis contained	32	32%	7	11%	39	24%
Diagnosis not contained	59	59%	29	47%	88	54%
D. heuristically contained	5	5%	17	27%	22	14%
Indistinguishable	4	4%	9	15%	13	8%
Sum	100		62		162	

Table 8.35: **Categorization of false negative system errors.** Diagnoses that are described in the text, but are not extracted by our approach are examined and their causes are grouped into categories.

	I21: Myocardial Infarction		**K35: Appendicitis**		**Sum**	
	#	%	#	%	#	%
Unknown synonym	25	78%	3	38%	28	70%
Typo	2	6%	1	13%	3	8%
Sectioning error	2	6%	0	0%	2	5%
Context error	3	9%	4	50%	7	18%
Sum	32		8		40	

Table 8.36: **Analysis of false positive errors.** Analysis is conducted with cases of specialized department and the high-F1-synonym-list. It is evaluated if a diagnosis is described and contained in the text.

	I21: Myocardial Infarction		**K35: Appendicitis**		**Sum**	
	#	%	#	%	#	%
Diagnosis contained	20	20%	34	51%	54	32%
Diagnosis not contained	78	78%	22	33%	100	60%
D. heuristically contained	2	2%	0	0%	2	1%
Indistinguishable	0	0%	11	16%	11	7%
Sum	100		67		167	

The second study exams the false positives and checks why our system assess a diagnosis to exist although it is not encoded with ICD-10. Table 8.36 shows the results of the false positive analysis. In 32% of all cases, the diagnosis is not encoded but described and extracted in the text. That is not an error of our system, but states that not every diagnosis that is mentioned in the discharge letter is encoded with ICD-10. In 60%, the diagnosis is encoded and not contained in the text, but erroneously extracted by our approach. In 1% of all cases the diagnosis is described as heuristic, but erroneously extracted by us and in 7% of all cases it is unclear if the diagnosis is characterized in the text.

Table 8.37 summarizes false positive system errors. The most common error source are historic events. In 29% of all cases the date of an historic finding is missing in the processed text. The original discharge letter document contains this date, but it has been removed during the ETL process in the anonymization step, because the dates may allow inferences concerning the patients. Another problem are big historic paragraphs (22%), containing an historic identifier, like a date in a headline, and several sentences or bullet points. Unfortunately, the sentence structure has been destroyed by the transformation of Word file to pain text, or the document creator inserted line breaks in the middle of a sentence instead of an automatic line wrap and these line breaks are interpreted as the end of a paragraph. The following examples shows two lines that are semantically connected to each other, but hard to parse, because they are separated by a line break and no enumeration is clearly visible.

ICD - Neuimplantation am 10.06.2011
cardiale Dekompensation, NSTEMI (3)

Similar errors (19% all of FP errors) are caused by the segmentation algorithms by computing wrong segments, which can be caused by faulty documentation like "Ausschluss. Myokardinfarkt" (engl. *Exclusion. myocardial infarction*).

16% of the errors are caused by incorrectly spelled or missing context trigger tokens, like "Verd auf" as abbreviation for "Verdacht auf" (engl. *suspicion of*).

A patient case sometimes contains several discharge letters. In addition to the current one, sometimes older discharge letter can be attached to it. 6% of the errors are caused due to a mention of the diagnosis in an old discharge letter.

The synonym "Appendizitis" (engl. *appendicitis*) is in 5% of the errors a too general description of the diagnosis. It matched to the words "chronische Appendizitis" (engl. *chronic appendicitis*), although the "K35: Acute appendicitis" was queried.

In 4% of the errors, other sections of the discharge letter are mistakenly classified as diagnosis section and the synonyms are matched in e.g. the medical history section.

Based on this error analysis the results of Table 8.33 are recalculated for the diagnosis "I21: Myocardial Infarction" and "K35: Appendicitis". Diagnoses that are encoded with ICD-10, but not described in the discharge letter are not counted as false negative. FPs are treated accordingly and heuristic diagnoses and indistinguishable cases are excluded

Table 8.37: **Categorization of false positive system errors.** Diagnoses that are not described in the text, but extracted by our approach are examined and their causes are grouped into categories.

	Myocardial Infarction		**Appendicitis**		**Sum**	
	#	%	#	%	#	%
Sectioning error	4	5%	0	0%	4	4%
Segmentation error	7	9%	13	43%	20	19%
Historic event: anonymized date	30	38%	1	3%	31	29%
Historic event: big paragraph	23	29%	1	3%	24	22%
Wrong context, missing triggers	8	10%	9	30%	17	16%
Diagnoses in old discharge letters	6	8%	1	3%	7	6%
Synonym too general	-	0%	5	17%	5	5%
Sum	78		30		108	

Table 8.38: **Retrieval scores based on the error analysis.** The results of the error analysis were used as a basis to recalculate the results of the consistency with synonyms for the specialized departments. See Table 8.33

	Precision	**Recall**	**F1**
I21: Myocardial Infarction	0,88	0,93	0,91
K35: Appendicitis	0,97	0,99	0,98

from the evaluation. The precision and recall values increase, resulting in an F1 score of 0.9 for "I21: Myocardial Infarction" and 0.98 for "K35: Appendicitis". All values are illustrated in Table 8.38.

The actual consistency of the diagnoses is calculated based on the error analysis. It is assumed that diagnoses, encoded with ICD-10 and extracted by our system from the diagnosis section in the discharge letter with synonym lists, are described as valid diagnoses in the discharge letter. The evaluation (see Table 8.39) shows a consistency between 0.88 and 0.97 between encoded and described diagnoses. Encoded ICD-10 diagnoses are described on average in 90% (myocardial infarction: 88%, appendicitis: 97%) of all cases in the diagnosis section of the discharge letter. Diagnoses described in the discharge letter are encoded on average 94% (myocardial infarction: 97%, appendicitis: 96%) of all cases with ICD-10.

8.3.4 Discussion

The feasibility of consistency tests between structured and unstructured data depends on the quality of the extraction of information from unstructured data. Testing the consistency of encoded and described diagnoses is a difficult task, since the recall and the F1 score of the baseline are very low.

Table 8.39: **Consistency of encoded and described diagnoses.** The consistency of the diagnoses either described in the discharge letter or encoded with ICD-10, is calculated based on the error analysis.

	I21: Myocardial Infarction		**K35: Appendicitis**		**Sum**	
	#	%	#	%	#	%
ICD-10 diagnoses						
Described in text	2748	88%	746	96%	3494	90%
Not described in text	373	12%	29	94%	402	10%
Discharge diagnoses						
ICD-10 encoded	2748	97%	746	96%	3494	96%
ICD-10 not encoded	98	3%	34	4%	132	4%

Our approach, extracting the diagnoses with synonym lists, showed good results for severe diseases. But it had a poor performance in common or non-severe diseases, like "nausea and vomiting". That can have two reasons: (a) The diagnosis is mentioned in the text and not extracted by us or (b) the diagnosis is not described in the text. Rather harmless diagnoses may not be listed in the diagnosis section or they could be described in other sections of the discharge letter as such as epicrisis or therapy. For examples "nausea and vomiting" can be considered as a symptoms rather than a diagnosis and would be expected in the anamnesis (medical history) section.

A problem of this analysis is that the results of the consistency checks and the system evaluation are merged. The error analysis addresses this problem by separating errors and assigning them to their respective source (incorrect consistency or system error). The resulting system evaluation showed that our approach worked very well with F1 scores of 0.91 (myocardial infarction) and 0.98 (appendicitis) (See Table 8.33).

The consistency analysis itself revealed that if a diagnosis is described in the diagnosis section of the discharge letter, it will be encoded with ICD-10 in 96% of the cases on average. However, an encoded diagnosis is only mentioned in 90% of all cases on average. Our study only focuses on the diagnosis section, other parts of the discharge letter are not taken into account. Although diagnoses could be described here too, e.g. therapy or epicrisis. This evaluation only includes two diagnoses, but it could be expanded in order to get more generalized statements.

The most difficult part for such consistency checks is the acquisition of synonyms. The most common error source for the extraction of the diagnoses from text are unknown synonyms. More synonyms clearly could improve the results. The presented approach generates synonyms for various diagnoses in a two step procedure. While candidates of the high-precision-setup are almost all synonyms, some candidates of the high-F1-score-setup have to be rejected. In the presented approach, this is done manually, but it could be automatized by using some more resources. For example, a further step could check whether a candidate is just a body part. This test could be performed using an

anatomical terminology and it would eliminate many incorrect candidates. Examples for such erroneous candidates are “fibula”, “aorta” or “posterior wall”.

Another drawback of this approach would be fixed as well: The candidates are currently aggregated first, and afterwards tested on their correctness. For example, the two candidates for myocardial infarction “RCA Komplettverschluss” (engl. *total occlusion of RCA*) and “RCA” are pooled to “RCA”. But “RCA” is no valid synonym and it is dropped. As a result, the valid synonym “RCA Komplettverschluss” (engl. *total occlusion of RCA*) is lost. The above mentioned approach would identify “RCA” as a body part and would remove it; As a result, the valid valid synonym “RCA Komplettverschluss” remains.

Another possibility to improve the synonym generation is to focus on the naming of the diagnosis in the diagnosis section. Diagnoses often are mentioned with more general names like “NSTEMI” for “myocardial infarction” and described more in detail in the text like “95%ige RCA Stenose” (engl. *95% RCA stenosis*). The last part is not relevant for this task and can be omitted. By selecting the relevant terms on the diagnostic section, higher quality synonyms could be created. The selection could be rule-based using formatting information. Decisions concerning the relevance of a phrase can also be considered as a classification task and solved with a machine learning approach taking into account a variety of features such as: if a phrase contains medical terms, if a phrase contains a verb, if a phrase is in parentheses, the line number or text indentation.

9 Conclusion

This chapter summarizes the main contributions of this work and concludes this thesis by pointing to some future research areas.

9.1 Summary

In recent years, EHR have been integrated into CDWs to benefit various medical issues (e.g. clinical trials, knowledge discovery). They consists of structured data (e.g. diagnosis codes, laboratory values, drugs), which can be processed very well by CDWs, and a lot of unstructured data (e.g. free text discharge letters and reports on diagnostic findings) that is hardly supported by CDWs as most of them solely offer keyword search.

The main topic of this thesis is to develop a mechanism to gain information covered in texts stored in existing CDWs, which very rarely support the such features.

Chapter 2 reviews the literature and describes this situation in more detail, including a characterization of CDWs with their most prominent representatives and algorithms in natural language processing relevant to this work.

The architecture and the user interface of the efficient PaDaWaN CDW is outlined in Chapter 3. Built with several state-of-the-art components and procedures, the efficient architecture uses a novel storage mechanism in order to achieve fast query responses: A document centered index server stores all patient information in a query optimized way.

The data structures and algorithms of ad hoc IE are introduced in Chapter 4 and their implementation in the PaDaWaN CDW in Chapter 5. Efficient document structures enable quick access to semantic closed units and their contexts, which are recognized by negation and context algorithms to filter historical or non patient-related information. Desired information can be extracted via mature query features (e.g context sensitive query).

The accuracy and the efficiency of ad hoc information extraction is evaluated with comprehensive experiments in Chapter 6 and discussed in Chapter 7. The detection of negated entities achieves F1-scores between 0.96 and 0.99 and performs better than systems reported in the literature (0.91% - 0.99%). The exact length of the negated scope is determined in 91% correct (literature 54%). Ad hoc IE consistently outperforms the presented baselines in terms of speed and extracts Boolean concepts in milliseconds and numeric concepts in a few seconds ($<$ 3s).

Ad hoc IE is designed to enable persons to extract information from texts autonomously and quickly. This is especially important for many medical areas where no conventional IE system exists. The advantages of ad hoc IE are the promptness of the results and the adaptability by users compared to conventional IE, which also has a long development effort, but usually has a higher accuracy and confidence.

Ad hoc IE clearly exceeds the query features of other CDWs. Most of them only offer lazy token queries via SQL `LIKE` with very limited functionality. The most similar system to our approach is Dr. Warehouse, which provides a negation and context handling at sentence level and a user interface to query texts. Only Dr. Warehouse and our PaDaWaN CDW have query features that exceed a token search. Solely our PaDaWaN CDW supports features to query and to extract numeric concepts from unstructured texts via ad hoc IE.

The benefits of ad hoc IE can be summarized as follows: It extracts Boolean concepts in milliseconds and (constrained) numeric concepts in seconds with a very high accuracy from texts considering negations and the context of information with rich text query features (e.g context sensitive queries, regex query). The presented method empowers clinicians to extract information from texts on their own, by what physicians can work promptly and autonomously. A new feature not available in other CDWs is to query numerical concepts in texts and even to filter them (e.g. BMI > 25). The identified values can be extracted and exported for further analysis. Our approach is interactive as extracted concepts are displayed tabularly and furthermore, it is flexible because users can refine their input. Unlike conventional information extraction, ad hoc IE does not require a terminology. The approach is extendable for further applications and its easy-to-use interface facilitates a comfortable, interactive usage.

Three case studies are conducted at the Würzburg University Hospital in Chapter 8. The first case study replicates medical trend studies that show the changes of medication over the years. We propose a generalizable approach of ad hoc IE for pharmacotherapy (medications and their daily dosage) presented in hospital discharge letters. A drug list grouped by ATC (Anatomical Therapeutic Chemical Classification System) codes is used as input for queries to the CDW. To evaluate the quality of the study replication, we choose five studies from the literature covering three domains (hypertension, atrial fibrillation, chronic kidney disease) and compare the findings with the results of the University Hospital of Würzburg in total and restricted to its Department of Internal Medicine I. We achieve an F1 score of 0.983 (precision 0.997, recall 0.970) for extracting medication from discharge letters and an F1 score of 0.974 (precision 0.977, recall 0.972) for extracting the dosage. We replicate three published medical trend studies for hypertension, atrial fibrillation and chronic kidney disease. Overall, 93% of the main findings are replicated, 68% of sub-findings, and 75% of all findings. One study is completely replicated with all main and sub-findings.

The prevalence of heart failure in hospital inpatients is assessed with an algorithm that integrates information extracted with ad hoc IE from discharge letter and echocardiogram reports (e.g. LVEF < 45) and other sources of the hospital information system. This

study reveals that relying on ICD codes resulted in a marked underestimation of the true prevalence of heart failure, ranging from 44% in the validation data set to 55% (single year) and 31% (all years) in the overall analysis. Percentages decrease over the years, indicating considerable changes in coding practice and efficiency. Performance is markedly improved using search and permutation algorithms from the initial expert-specified query (F1 score of 81%) to the computer-optimized query (F1 score of 86%) or, alternatively, optimizing precision or sensitivity depending on the search objective [113].

The consistency of diagnoses is assessed by comparing structured ICD-10 encoded diagnoses with diagnoses mentioned in the diagnosis section of the discharge letter. These diagnoses are extracted with ad hoc IE from texts using terms of the Alpha-ID list and with synonyms generated by a novel implemented method to obtain synonyms from discharge letter. The proposed systems achieves an accuracy with an F1 score of 0.91 (myocardial infarction) and 0.98 (appendicitis). The consistency analysis itself reveals that if a diagnosis (myocardial infarction or appendicitis) is described in the diagnosis section of the discharge letter, it will be encoded with ICD-10 in 96% on average. An encoded diagnosis is only mentioned in 90% in the diagnosis section in the discharge letter on average. However, diagnoses may be described in other sections of the discharge letter, which are not taken into account for this evaluation.

9.2 Outlook

A possible extension to ad hoc IE in the medical domain is a deeper semantic integration. As mentioned in Section 7.6, the automatic usage of background knowledge, such as synonyms or homonyms of the AlphaID list or the MeSH net could increase the recall and the quality of the information extraction.

Ad hoc IE has been proven as a reliable and beneficial tool to gain information from unstructured clinical texts ad hoc. The approach is also interesting for many other areas. Modern analysis systems and business intelligence systems seek to be flexible by offering tools for customized analyses that exceed a few predefined reports. Ad hoc IE is very well suited for this purpose and offers powerful functions to support various use cases. Still nowadays, a lot of information is stored in textual form, e.g. in social media, all user comments and posts are written in natural language, as well as reviews on products, stores, or movies. These texts are usually characterized by short sentences with a rather telegraphic structure. The presented approach could be applied on these domains with little changes.

Texts with a more complicated sentence structure are more challenging and relations between concepts and co-references must be considered. Examples for that type of texts are novels or newspaper articles. A new research area could try to expand ad hoc IE to this field. However, examples with complex dependencies can be found in medicine as well, for example: a documentation of a syndrome of a patient, e.g. “pain”, including its intensity, localization, cause, induction, and circumstances. These properties are all related to each other and could be queried in a more structured way. A challenging

problem of this approach is the recognition of relationships between concepts. This could be addressed with dependency parsing or grammar applications. However, for this purpose, the data structure used to store the information must be adjusted in order to ensure high performance queries. This approach aims to recognize all grammatical and syntactical relationships and structures of a text and to make them accessible to user queries, in order to empower users to search concepts that have certain characteristics or relationships. This information does not have to be described in close proximity to each other, they could also be mentioned at the beginning and at the end of a text, as long as they have a grammatical connection or refer to the same concept (co-references).

Overall, the topic offers exciting future trends and application areas that require the research of new powerful algorithms in the field of AI. This work provides a foundation on how to gain knowledge from unstructured text documents and makes a valuable contribution to the field of research with medical data.

Bibliography

[1] Discharge letter: Standardfall verschluß rechte kranzarterie. `https://de.wikibooks.org/wiki/Innere_Medizin_kk:_STEMI#Fälle`. Accessed Oct 2018.

[2] Federal statistical office. `http://www.gbe.de`. Accessed Mar 2017.

[3] R development core team (2008) r: a language and environment for statistical computing. r foundation for statistical computing, vienna. isbn 3-900051-07-0. `http://www.R-project.org`. Accessed Apr 2018.

[4] Z. Afzal, E. Pons, N. Kang, M. C. Sturkenboom, M. J. Schuemie, and J. A. Kors. Contextd: an algorithm to identify contextual properties of medical terms in a dutch clinical corpus. *BMC bioinformatics*, 15(1):373, 2014.

[5] S. K. Agarwal, L. Wruck, M. Quibrera, K. Matsushita, L. R. Loehr, P. P. Chang, W. D. Rosamond, J. Wright, G. Heiss, and J. Coresh. Temporal trends in hospitalization for acute decompensated heart failure in the united states, 1998–2011. *American journal of epidemiology*, 183(5):462–470, 2016.

[6] A. V. Aho. Algorithms for finding patterns in strings, handbook of theoretical computer science (vol. a): algorithms and complexity, 1991.

[7] F. Alqaisi, L. K. Williams, E. L. Peterson, and D. E. Lanfear. Comparing methods for identifying patients with heart failure using electronic data sources. *BMC health services research*, 9(1):237, 2009.

[8] N. Anand and M. Kumar. An overview on data quality issues at data staging etl. In *Int. Conf. on Advances in Signal Processing and Communication*. Citeseer, 2013.

[9] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*, 2016.

[10] D. E. Appelt and B. Onyshkevych. The common pattern specification language. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, pages 23–30. Association for Computational Linguistics, 1998.

[11] K. Asakura and E. Ordal. Is your clinical documentation improvement program compliant? hospital finance executives, take note: your organization's clinical documentation improvement program may soon be under a microscope. *Healthcare Financial Management*, 66(10):96–101, 2012.

[12] N. E. M. Association et al. Digital imaging and communications in medicine (dicom). *http.//medical. nema. org/*, 2003.

[13] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533:452–454, May 2016.

[14] G. V. Bard. Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*, pages 117–124. Citeseer, 2007.

[15] C. G. Begley and L. M. Ellis. Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531, 2012.

[16] O. Bodenreider. Issues in mapping loinc laboratory tests to snomed ct. In *AMIA Annual Symposium Proceedings*, volume 2008, page 51. American Medical Informatics Association, 2008.

[17] B. K. Boguraev. Annotation-based finite state processing in a large-scale nlp architecture. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, 260:61, 2004.

[18] B. Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd international conference on computational linguistics*, pages 89–97. Association for Computational Linguistics, 2010.

[19] L. Borin. Something borrowed, something blue: Rule-based combination of pos taggers. In *LREC*, 2000.

[20] G. R. Brämer. International statistical classification of diseases and related health problems. tenth revision. *World health statistics quarterly. Rapport trimestriel de statistiques sanitaires mondiales*, 41(1):32–36, 1988.

[21] S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit. Tiger: Linguistic interpretation of a german corpus. *Research on language and computation*, 2(4):597–620, 2004.

[22] T. Brants. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics, 2000.

[23] S. Bromfield and P. Muntner. High blood pressure: the leading global burden of disease risk factor and the need for worldwide prevention programs. *Current hypertension reports*, 15(3):134–136, 2013.

[24] S. Buchholz and E. Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning*, pages 149–164. Association for Computational Linguistics, 2006.

[25] R. J. Byrd, S. R. Steinhubl, J. Sun, S. Ebadollahi, and W. F. Stewart. Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records. *International journal of medical informatics*, 83(12):983–992, 2014.

[26] V. Canuel, B. Rance, P. Avillach, P. Degoulet, and A. Burgun. Translational research platforms integrating clinical and omics data: a review of publicly available solutions. *Briefings in bioinformatics*, 16(2):280–290, 2014.

[27] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. Crisp-dm 1.0 step-by-step data mining guide. Technical report, The CRISP-DM consortium, Aug. 2000.

[28] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. Evaluation of negation phrases in narrative clinical reports. In *Proceedings of the AMIA Symposium*, page 105. American Medical Informatics Association, 2001.

[29] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310, 2001.

[30] W. W. Chapman, D. Hilert, S. Velupillai, M. Kvist, M. Skeppstedt, B. E. Chapman, M. Conway, M. Tharp, D. L. Mowery, and L. Deleger. Extending the negex lexicon for multiple languages. *Studies in health technology and informatics*, 192:677, 2013.

[31] S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *ACM Sigmod record*, 26(1):65–74, 1997.

[32] C. G. Chute, S. A. Beck, T. B. Fisk, and D. N. Mohr. The enterprise data trust at mayo clinic: a semantically integrated warehouse of biomedical data. *Journal of the American Medical Informatics Association*, 17(2):131–135, 2010.

[33] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[34] M. F. Collen. A vision of health care and informatics in 2008. *Journal of the American Medical Informatics Association*, 6(1):1–5, 1999.

[35] M. F. Collen and M. J. Ball. *The history of medical informatics in the United States*. Springer, 2015.

[36] M. Collins and T. Koo. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70, 2005.

[37] W. W. W. Consortium. Web services architecture. 3.1.3 relationship to the world wide web and rest architectures. `https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest`. Accessed Oct 2018.

[38] R. Cornet and N. de Keizer. Forty years of snomed: a literature review. 8(1):S2, 2008.

[39] M. Corporation. Sql server documentation. `https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation`. Accessed Oct 2018.

[40] R. Costumero, F. López, C. Gonzalo-Martín, M. Millan, and E. Menasalvas. An approach to detect negation on medical documents in spanish. In *International Conference on Brain Informatics and Health*, pages 366–375. Springer, 2014.

[41] R. A. Cote and S. Robboy. Progress in medical information management: Systematized nomenclature of medicine (snomed). *Jama*, 243(8):756–762, 1980.

[42] V. Cotik, R. Roller, F. Xu, H. Uszkoreit, K. BuddeO, and D. SchmidtO. Negation detection in clinical reports written in german. *BioTxtM 2016*, page 115, 2016.

[43] V. Cotik, V. Stricker, J. Vivaldi, and H. Rodriguez. Syntactic methods for negation detection in radiology reports in spanish. *ACL 2016*, page 156, 2016.

[44] M. Cuggia, N. Garcelon, B. Campillo-Gimenez, T. Bernicot, J.-F. Laurent, E. Garin, A. Happe, and R. Duvauferrier. Roogle: an information retrieval engine for clinical data warehouse. In *MIE*, pages 584–588, 2011.

[45] H. Cunningham, H. Cunningham, D. Maynard, D. Maynard, V. Tablan, and V. Tablan. Jape: a java annotation patterns engine, 1999.

[46] H. Dalianis. *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer, 2018.

[47] I. Danciu, J. D. Cowan, M. Basford, X. Wang, A. Saip, S. Osgood, J. Shirey-Rice, J. Kirby, and P. A. Harris. Secondary use of clinical data: the vanderbilt approach. *Journal of biomedical informatics*, 52:28–35, 2014.

[48] D. de Kok. Tüba-d/w: a large dependency treebank for german. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, page 271, 2014.

[49] L. Deléger and C. Grouin. Detecting negation of medical problems in french clinical notes. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 697–702. ACM, 2012.

[50] Y. Denekamp, O. Ogunyemi, A. A. Boxwala, and R. A. Greenes. Using new object oriented expression language (gello) to encode arden syntax's medical logic modules. In *Proceedings of the AMIA Symposium*, page 1006. American Medical Informatics Association, 2002.

[51] G. Dietrich, M. Ertl, G. Fette, M. Kaspar, J. Krebs, D. Mackenrodt, S. Störk, and F. Puppe. Extending the query language of a data warehouse for patient recruitment. In R. Röhrig, A. Timmer, H. Binder, and U. Sax, editors, *German Medical Data Sciences: Visions and Bridges*, volume 243 of *Studies in Health Technology and Informatics*, pages 152–156. IOS Press, 2017.

[52] G. Dietrich, F. Fell, G. Fette, J. Krebs, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Web-padawan: Eine web-basierte benutzeroberfläche für ein klinisches data warehouse. In *HEC 2016, Joint Conference of GMDS, DGEpi, IEA-EEF, EFMI*, page DocAbstr. 421, 2016.

[53] G. Dietrich, G. Fette, M. Ertl, M. Toepfer, M. Kaspar, S. Störk, and F. Puppe. Fallstudie zur validierung eines klinischen data-warehouse mit hintergrundwissen. In *GMDS 2015. 60. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e.V. (GMDS). Krefeld*, page DocAbstr. 167. German Medical Science GMS Publishing House, 2015.

[54] G. Dietrich, G. Fette, and F. Puppe. A comparison of search engine technologies for a clinical data warehouse. In *LWA*, pages 235–242, 2014.

[55] G. Dietrich, J. Krebs, G. Fette, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Ad hoc information extraction for clinical data warehouses. *Methods of information in medicine*, 57(01):e22–e29, 2018.

[56] G. Dietrich, J. Krebs, L. Liman, G. Fette, M. Ertl, M. Kaspar, S. Störk, and F. Puppe. Replicating medication trend studies using ad hoc information extraction in a clinical data warehouse. *BMC medical informatics and decision making*, 19(1):15, 2019.

[57] V. Dinu and P. Nadkarni. Guidelines for the effective use of entity–attribute–value modeling for biomedical databases. *International journal of medical informatics*, 76(11-12):769–779, 2007.

[58] K. Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279, 2006.

[59] W. Dorda, T. Wrba, G. Duftschmid, P. Sachs, W. Gall, C. Rehnelt, G. Boldt, W. Premauer, et al. Archimed: a medical information and retrieval system. *Methods Archive*, 38(1):16–24, 1999.

[60] G. Durrett and D. Klein. Neural crf parsing. *arXiv preprint arXiv:1507.03641*, 2015.

[61] A. Ekbal, S. Mondal, and S. Bandyopadhyay. Pos tagging using hmm and rule-based chunking. *The Proceedings of SPSAL*, 8(1):25–28, 2007.

[62] P. L. Elkin, S. H. Brown, B. A. Bauer, C. S. Husser, W. Carruth, L. R. Bergstrom, and D. L. Wahner-Roedler. A controlled trial of automated classification of negation from clinical notes. *BMC medical informatics and decision making*, 5(1):13, 2005.

[63] M. Ertl. Erfassung von klinischen untersuchungsdaten und transfer in ein data warehouse. Master's thesis, Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt, 2011.

[64] E. Falaschetti, J. Mindell, C. Knott, and N. Poulter. Hypertension management in england: a serial cross-sectional study from 1994 to 2011. *The Lancet*, 383(9932):1912–1919, 2014.

[65] M. C. Fang, R. S. Stafford, J. N. Ruskin, and D. E. Singer. National trends in antiarrhythmic and antithrombotic medication use in atrial fibrillation. *Archives of internal medicine*, 164(1):55–60, 2004.

[66] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[67] H. Feldweg. Implementation and evaluation of a german hmm for pos disambiguation. In *Natural Language Processing Using Very Large Corpora*, pages 1–12. Springer, 1999.

[68] G. Fette, M. Ertl, A. Wörner, P. Kluegl, S. Störk, and F. Puppe. Information extraction from unstructured electronic health records and integration into a data warehouse. In *GI-Jahrestagung*, pages 1237–1251, 2012.

[69] G. Fette, M. Kaspar, G. Dietrich, M. Ertl, J. Krebs, S. Stoerk, and F. Puppe. A customizable importer for the clinical data warehouses padawan and i2b2. *Studies in health technology and informatics*, 243:90–94, 2017.

[70] J. R. Finkel, A. Kleeman, and C. D. Manning. Efficient, feature-based, conditional random field parsing. *Proceedings of ACL-08: HLT*, pages 959–967, 2008.

[71] E. Ford, J. A. Carroll, H. E. Smith, D. Scott, and J. A. Cassell. Extracting information from the text of electronic medical records to improve case detection: a systematic review. *Journal of the American Medical Informatics Association*, 23(5):1007–1015, 2016.

[72] A. W. Forrey, C. J. Mcdonald, G. DeMoor, S. M. Huff, D. Leavelle, D. Leland, T. Fiers, L. Charles, B. Griffin, F. Stalling, et al. Logical observation identifier names and codes (loinc) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. *Clinical Chemistry*, 42(1):81–90, 1996.

[73] K. A. Foth. Eine umfassende constraint-dependenz-grammatik des deutschen. 2006.

[74] A. S. Foundation. Apache solr reference guide. `https://lucene.apache.org/solr/guide/7_3/index.html`. Accessed Oct 2018.

[75] A. S. Foundation. Apache solr reference guide 7.4. Technical report, Apache Software Foundation, 2018.

[76] D. Freitag and A. McCallum. Information extraction with hmm structures learned by stochastic optimization. *AAAI/IAAI*, 2000:584–589, 2000.

[77] N. Frolova, J. A. Bakal, F. A. McAlister, B. H. Rowe, H. Quan, P. Kaul, and J. A. Ezekowitz. Assessing the use of international classification of diseases-10th revision codes from the emergency department for the identification of acute heart failure. *JACC: Heart Failure*, 3(5):386–391, 2015.

[78] M. Gabetta, I. Limongelli, E. Rizzo, A. Riva, D. Segagni, and R. Bellazzi. Bigq: a nosql based framework to handle genomic variants in i2b2. *BMC bioinformatics*, 16(1):415, 2015.

[79] K. Gadsbøll, L. Staerk, E. L. Fosbøl, C. Sindet-Pedersen, A. Gundlund, G. Y. Lip, G. H. Gislason, and J. B. Olesen. Increased use of oral anticoagulants in patients with atrial fibrillation: temporal trends from 2005 to 2015 in denmark. *European heart journal*, 38(12):899–906, 2017.

[80] N. Garcelon, A. Neuraz, V. Benoit, R. Salomon, and A. Burgun. Improving a full-text search engine: the importance of negation detection and family history context to identify cases in a biomedical data warehouse. *Journal of the American Medical Informatics Association*, 24(3):607–613, 2016.

[81] N. Garcelon, A. Neuraz, R. Salomon, H. Faour, V. Benoit, A. Delapalme, A. Munnich, A. Burgun, and B. Rance. A clinician friendly data warehouse oriented toward narrative reports: Dr. warehouse. *Journal of biomedical informatics*, 80:52–63, 2018.

[82] H. Godet-Mardirossian, X. Girerd, M. Vernay, B. Chamontin, K. Castetbon, and C. de Peretti. Patterns of hypertension management in france (enns 2006–2007). *European journal of preventive cardiology*, 19(2):213–220, 2012.

[83] B. Graubner. *OPS Systematisches Verzeichnis 2014: Operationen-und Prozedurenschlüssel-Internationale Klassifikation der Prozeduren in der Medizin Version 2014*. Deutscher Ärzteverlag, 2013.

[84] O. Gros and M. Stede. Determining negation scope in german and english medical diagnoses. *Nonveridicality and Evaluation*, pages 113–126, 2013.

[85] Q. Gu, V. L. Burt, C. F. Dillon, and S. Yoon. Trends in antihypertensive medication use and blood pressure control among united states adults with hypertensionclinical perspective: The national health and nutrition examination survey, 2001 to 2010. *Circulation*, 126(17):2105–2114, 2012.

[86] T. D. Gunter and N. P. Terry. The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions. *Journal of medical Internet research*, 7(1), 2005.

[87] M. J. Hall, S. Levant, and C. J. DeFrances. Hospitalization for congestive heart failure: United states, 2000–2010. *age*, 65(23):29, 2012.

[88] T. Hamon and N. Grabar. Tuning heideltime for identifying time expressions in clinical texts in english and french. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 101–105, 2014.

[89] D. A. Hanauer. Emerse: the electronic medical record search engine. In *AMIA annual symposium proceedings*, volume 2006, page 941. American Medical Informatics Association, 2006.

[90] A. Happe, B. Pouliquen, A. Burgun, M. Cuggia, and P. Le Beux. Automatic concept extraction from spoken medical reports. *International Journal of Medical Informatics*, 70(2-3):255–263, 2003.

[91] H. Harkema, J. N. Dowling, T. Thornblade, and W. W. Chapman. Context: an algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of biomedical informatics*, 42(5):839–851, 2009.

[92] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde. Research electronic data capture (redcap)—a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of biomedical informatics*, 42(2):377–381, 2009.

[93] F. M. Hasan, N. UzZaman, and M. Khan. Comparison of different pos tagging techniques (n-gram, hmm and brill's tagger) for bangla. In *Advances and innovations in systems, computing sciences and software engineering*, pages 121–126. Springer, 2007.

[94] E. Hatcher and O. Gospodnetic. *Lucene in action (in action series)*. Manning Publications, 2004.

[95] G. Héja, G. Surján, and P. Varga. Ontological analysis of snomed ct. 8(1):S8, 2008.

[96] M. Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 278–277. Association for Computational Linguistics, 2000.

[97] P. Heudel, A. Livartowski, P. Arveux, E. Willm, and C. Jamain. The consore project supports the implementation of big data in oncology. *Bulletin du cancer*, 103(11):949–950, 2016.

[98] J. A. Hirsch, T. M. Leslie-Mazwi, G. N. Nicola, R. M. Barr, J. A. Bello, W. D. Donovan, R. Tu, M. D. Alson, and L. Manchikanti. Current procedural terminology; a primer. *Journal of neurointerventional surgery*, 7(4):309–312, 2015.

[99] M. M. Horvath, S. Winfield, S. Evans, S. Slopek, H. Shang, and J. Ferranti. The deduce guided query tool: providing simplified access to clinical data for research and quality improvement. *Journal of biomedical informatics*, 44(2):266–276, 2011.

[100] A. Hotho, A. Nürnberger, and G. Paaß. A brief survey of text mining. In *Ldv Forum*, volume 20, pages 19–62. Citeseer, 2005.

[101] A. Hotho, S. Staab, and G. Stumme. Ontologies improve text document clustering. In *Third IEEE international conference on data mining*, pages 541–544. IEEE, 2003.

[102] G. Hripcsak, P. D. Clayton, T. A. Pryor, P. Haug, O. Wigertz, and J. Van der Lei. The arden syntax for medical logic modules. In *Proceedings. Symposium on Computer Applications in Medical Care*, pages 200–204. American Medical Informatics Association, 1990.

[103] H. Hu, M. Correll, L. Kvecher, M. Osmond, J. Clark, A. Bekhash, G. Schwab, D. Gao, J. Gao, V. Kubatin, et al. Dw4tr: a data warehouse for translational research. *Journal of biomedical informatics*, 44(6):1004–1019, 2011.

[104] Y. Huang and H. J. Lowe. A novel hybrid approach to automated negation detection in clinical radiology reports. *Journal of the American Medical Informatics Association*, 14(3):304–311, 2007.

[105] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[106] N. Indurkhya and F. Damerau. *Handbook of Natural Language Processing, Second Edition*. Chapman & Hall/CRC Machine Learning & Pattern Recognition Series. Taylor & Francis, 2010.

[107] W. H. Inmon. *Building the data warehouse*. John wiley & sons, 2005.

[108] J. P. Isson. *Unstructured Data Analytics: How to Improve Customer Acquisition, Customer Retention, and Fraud Detection and Prevention*. John Wiley & Sons, 2018.

[109] K. Jensen, C. Soguero-Ruiz, K. O. Mikalsen, R.-O. Lindsetmo, I. Kouskoumvekaki, M. Girolami, S. O. Skrovseth, and K. M. Augestad. Analysis of free text in electronic health records for identification of cancer patient trajectories. *Scientific reports*, 7:46226, 2017.

[110] D. Jurafsky and J. H. Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

[111] D. Kalra, T. Beale, and S. Heard. The openehr foundation. *Studies in health technology and informatics*, 115:153–173, 2005.

[112] M. Kaspar, M. Ertl, G. Fette, G. Dietrich, M. Toepfer, C. Angermann, S. Störk, and F. Puppe. Data linkage from clinical to study databases via an r data warehouse user interface. *Methods of information in medicine*, 55(04):381–386, 2016.

[113] M. Kaspar, G. Fette, G. Güder, L. Seidlmayer, M. Ertl, G. Dietrich, H. Greger, F. Puppe, and S. Störk. Underestimated prevalence of heart failure in hospital inpatients: a comparison of icd codes and discharge letter information. *Clinical Research in Cardiology*, pages 1–10, 2018.

[114] H. Katada, N. Yukawa, H. Urushihara, S. Tanaka, T. Mimori, and K. Kawakami. Prescription patterns and trends in anti-rheumatic drug use based on a large-scale claims database in japan. *Clinical rheumatology*, 34(5):949–956, 2015.

[115] A. Khalaf, A. Salah Hashim, and W. Akeel. Clinical data warehouse: A review. 44, 12 2018.

[116] A. U. Khand, M. Shaw, I. Gemmel, and J. G. Cleland. Do discharge codes underestimate hospitalisation due to heart failure? validation study of hospital discharge coding for heart failure. *European journal of heart failure*, 7(5):792–797, 2005.

[117] R. Kimball and M. Ross. *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons, 2011.

[118] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[119] D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, pages 3–10, 2003.

[120] P. Kluegl, M. Toepfer, P.-D. Beck, G. Fette, and F. Puppe. Uima ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, 22(1):1–40, 2016.

[121] P. Klügl. *Context-specific Consistencies in Information Extraction: Rule-based and Probabilistic Approaches*. BoD–Books on Demand, 2015.

[122] M. Komaroff, F. Tedla, E. Helzner, and M. A. Joseph. Antihypertensive medications and change in stages of chronic kidney disease. *International journal of chronic diseases*, 2018, 2018.

[123] K. U. Kortüm, M. Müller, C. Kern, A. Babenko, W. J. Mayer, A. Kampik, T. C. Kreutzer, S. Priglinger, and C. Hirneiss. Using electronic health records to build an ophthalmologic data warehouse and visualize patients' data. *American journal of ophthalmology*, 178:84–93, 2017.

[124] M. D. Krasowski, A. Schriever, G. Mathur, J. L. Blau, S. L. Stauffer, and B. A. Ford. Use of a data warehouse at an academic medical center for clinical pathology quality improvement, education, and research. *Journal of pathology informatics*, 6, 2015.

[125] J. Krebs. Anonymisierung, segmentierung und terminologie-entwicklung zur informationsextraktion aus unstrukturierten, telegrammartigen, medizinischen dokumenten. Master's thesis, Julius Maximilians University of Würzburg, 2016.

[126] J. Krebs, H. Corovic, G. Dietrich, M. Ertl, G. Fette, M. Kaspar, M. Krug, S. Störk, and F. Puppe. Semi-automatic terminology generation for information extraction from german chest x-ray reports. In R. Röhrig, A. Timmer, H. Binder, and U. Sax, editors, *GMDS*, volume 243 of *Studies in Health Technology and Informatics*, pages 80–84. IOS Press, 2017.

[127] H.-U. Krieger, C. Spurk, H. Uszkoreit, F. Xu, Y. Zhang, F. Müller, and T. Tolxdorff. Information extraction from german patient records via hybrid parsing and relation extraction strategies. In *LREC*, pages 2043–2048, 2014.

[128] M. Krug, N. D. T. Tu, L. Weimer, I. Reger, L. Konle, F. Jannidis, and F. Puppe. Annotation and beyond – using athen annotation and text highlighting environment. In *DHd 2018*, 2018.

[129] S. Kübler, R. McDonald, and J. Nivre. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127, 2009.

[130] H. Kučera and W. N. Francis. *Computational analysis of present-day American English*. Dartmouth Publishing Group, 1967.

[131] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[132] C. P. Langlotz. Radlex: a new method for indexing online educational materials, 2006.

[133] C. S. Ledbetter and M. W. Morgan. Toward best practice: leveraging the electronic patient record as a clinical data warehouse. *Journal of Healthcare Information Management*, 15(2):119–132, 2001.

[134] D. Lee, N. de Keizer, F. Lau, and R. Cornet. Literature review of snomed ct use. *Journal of the American Medical Informatics Association*, 21(e1):e11–e19, 2013.

[135] H. Leslie. International developments in open ehr archetypes and templates. *Health Information Management Journal*, 37(1):38–39, 2008.

[136] M. I. Lieberman, T. N. Ricciardi, et al. The use of snomed© ct simplifies querying of a clinical data warehouse. In *AMIA Annual Symposium Proceedings*, volume 2003, page 910. American Medical Informatics Association, 2003.

[137] D. A. Lindberg, B. L. Humphreys, and A. T. McCray. The unified medical language system. *Yearbook of Medical Informatics*, 2(01):41–51, 1993.

[138] C. E. Lipscomb. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265, 2000.

[139] S. Liu, W. Ma, R. Moore, V. Ganesan, and S. Nelson. Rxnorm: prescription for electronic drug information exchange. *IT professional*, 7(5):17–23, 2005.

[140] L. R. Loehr, S. K. Agarwal, C. Baggett, L. M. Wruck, P. P. Chang, S. D. Solomon, E. Shahar, H. Ni, W. D. Rosamond, and G. Heiss. Classification of acute decompensated heart failure: an automated algorithm compared with a physician reviewer panel: the atherosclerosis risk in communities study. *Circulation: Heart Failure*, 6(4):719–726, 2013.

[141] J. W. London and D. Chatterjee. Implications of observation-fact modifiers to i2b2 ontologies. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2011 IEEE International Conference on*, pages 929–930. IEEE, 2011.

[142] H. J. Lowe and G. O. Barnett. Understanding and using the medical subject headings (mesh) vocabulary to perform literature searches. *Jama*, 271(14):1103–1108, 1994.

[143] H. J. Lowe, T. A. Ferris, P. M. Hernandez, and S. C. Weber. Stride–an integrated standards-based translational research informatics platform. In *AMIA Annual Symposium Proceedings*, volume 2009, page 391. American Medical Informatics Association, 2009.

[144] Z. Lu. Pubmed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011, 2011.

[145] Z. Lu, W. Kim, and W. J. Wilbur. Evaluation of query expansion using mesh in pubmed. *Information retrieval*, 12(1):69–80, 2009.

[146] C. Ma, H. Frankel, T. Beale, S. Heard, et al. Ehr query language (eql)-a query language for archetype-based health records. *Medinfo*, 129:397–401, 2007.

[147] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[148] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA, 1999.

[149] H. Marin, E. Massad, M. A. Gutierrez, R. J. Rodrigues, and D. Sigulem. *Global health informatics: how information technology can change our lives in a globalized world*. Academic Press, 2016.

[150] D. Marwede, P. Daumke, K. Marko, D. Lobsien, S. Schulz, and T. Kahn. Radlex-german version: a radiological lexicon for indexing image and report information. *RoFo: Fortschritte auf dem Gebiete der Rontgenstrahlen und der Nuklearmedizin*, 181(1):38–44, 2009.

[151] A. McCallum, D. Freitag, and F. C. Pereira. Maximum entropy markov models for information extraction and segmentation. 17(2000):591–598, 2000.

[152] C. J. McDonald, S. M. Huff, J. G. Suico, G. Hill, D. Leavelle, R. Aller, A. Forrey, K. Mercer, G. DeMoor, J. Hook, et al. Loinc, a universal standard for identifying laboratory observations: a 5-year update. *Clinical chemistry*, 49(4):624–633, 2003.

[153] R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics, 2005.

[154] P. A. McKee, W. P. Castelli, P. M. McNamara, and W. B. Kannel. The natural history of congestive heart failure: the framingham study. *New England Journal of Medicine*, 285(26):1441–1446, 1971.

[155] S. Mehrabi, A. Krishnan, S. Sohn, A. M. Roch, H. Schmidt, J. Kesterson, C. Beesley, P. Dexter, C. M. Schmidt, H. Liu, et al. Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of biomedical informatics*, 54:213–219, 2015.

[156] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70, 2010.

[157] S. M. Meystre, Y. Kim, G. T. Gobbel, M. E. Matheny, A. Redd, B. E. Bray, and J. H. Garvin. Congestive heart failure information extraction framework for automated treatment performance measures assessment. *Journal of the American Medical Informatics Association*, 24(e1):e40–e46, 2016.

[158] S. M. Meystre, G. K. Savova, K. C. Kipper-Schuler, and J. F. Hurdle. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearbook of medical informatics*, 17(01):128–144, 2008.

[159] R. O. Mohammed and S. A. Talab. Clinical data warehouse issues and challenges. *International Journal of u-and e-Service, Science and Technology*, 7(5):251–262, 2014.

[160] R. Morante and W. Daelemans. A metalearning approach to processing the scope of negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 21–29. Association for Computational Linguistics, 2009.

[161] S. N. Murphy, M. E. Mendis, D. A. Berkowitz, I. Kohane, and H. C. Chueh. Integration of clinical and genetic data in the i2b2 architecture. In *AMIA Annual Symposium Proceedings*, volume 2006, page 1040. American Medical Informatics Association, 2006.

[162] S. N. Murphy, G. Weber, M. Mendis, V. Gainer, H. C. Chueh, S. Churchill, and I. Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130, 2010.

[163] I. Neamatullah, M. M. Douglass, H. L. Li-wei, A. Reisner, M. Villarroel, W. J. Long, P. Szolovits, G. B. Moody, R. G. Mark, and G. D. Clifford. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8(1):32, 2008.

[164] G. S. Nelson. Practical implications of sharing data: a primer on data privacy, anonymization, and de-identification. In *SAS Global Forum Proceedings*, 2015.

[165] National center for health statistics. analytic and reporting guidelines: The national health and nutrition examination survey (nhanes). Accessed May 2018.

[166] J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[167] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[168] M. J. O'Connor, S. W. Tu, and M. A. Musen. The chronus ii temporal database mediator. In *Proceedings of the AMIA Symposium*, page 567. American Medical Informatics Association, 2002.

[169] openEHR Foundation. Archetype query language (aql). `https://specifications.openehr.org/releases/QUERY/latest/AQL.html`. Accessed Jan 2019.

[170] W. H. Organization. *International statistical classification of diseases and related health problems*, volume 1. World Health Organization, 2004.

[171] T. B. Pedersen and C. S. Jensen. Research issues in clinical data warehousing. In *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, pages 43–52. IEEE, 1998.

[172] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[173] J. W. Pennington, B. Ruth, M. J. Italia, J. Miller, S. Wrazien, J. G. Loutrel, E. B. Crenshaw, and P. S. White. Harvest: an open platform for developing web-based biomedical data discovery and reporting applications. *Journal of the American Medical Informatics Association*, 21(2):379–383, 2013.

[174] C. Percy, V. v. Holten, C. S. Muir, W. H. Organization, et al. International classification of diseases for oncology. 1990.

[175] K. Pommerening, J. Drepper, K. Helbing, and T. Ganslandt. Guideline for data protection in medical research projects: Tmf's generic solutions 2.0. In ∴ Technologie- und Methodenplattform für die vernetzte medizinische Forschung e.V., 2014.

[176] M. F. Porter. Snowball: A language for stemming algorithms, 2001.

[177] F. Pourasghar, H. Malekafzali, S. Koch, and U. Fors. Factors influencing the quality of medical documentation when a paper-based medical records system is replaced with an electronic medical records system: an iranian case study. *International journal of technology assessment in health care*, 24(4):445–451, 2008.

[178] D. M. Powers. Applications and explanations of zipf's law. In *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*, pages 151–160. Association for Computational Linguistics, 1998.

[179] M. Puppala, T. He, S. Chen, R. Ogunti, X. Yu, F. Li, R. Jackson, and S. T. Wong. Meteor: an enterprise health informatics environment to support evidence-based medicine. *IEEE Transactions on Biomedical Engineering*, 62(12):2776–2786, 2015.

[180] F. Puppe. Medizinische entscheidungsunterstützungssysteme. *Informatik-Spektrum*, 37(3):246–249, 2014.

[181] S. Quach, C. Blais, and H. Quan. Administrative data have high variation in validity for recording heart failure. *Canadian journal of cardiology*, 26(8):e306–e312, 2010.

[182] H. Quan, B. Li, L. Duncan Saunders, G. A. Parsons, C. I. Nilsson, A. Alibhai, W. A. Ghali, and I. investigators. Assessing validity of icd-9-cm and icd-10 administrative data in recording clinical conditions in a unique dually coded database. *Health services research*, 43(4):1424–1441, 2008.

[183] H. Quan, G. A. Parsons, and W. A. Ghali. Validity of information on comorbidity derived from icd-9-ccm administrative data. *Medical care*, pages 675–685, 2002.

[184] A. N. Rafferty and C. D. Manning. Parsing three german treebanks: Lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German*, pages 40–46. Association for Computational Linguistics, 2008.

[185] P. Raghavan, J. L. Chen, E. Fosler-Lussier, and A. M. Lai. How essential are unstructured clinical narratives and information fusion to clinical trial recruitment? *AMIA Summits on Translational Science Proceedings*, 2014:218, 2014.

[186] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing*, 1996.

[187] M. Reich, T. Liefeld, J. Gould, J. Lerner, P. Tamayo, and J. P. Mesirov. Genepattern 2.0. *Nature genetics*, 38(5):500, 2006.

[188] S. Ribaric, A. Ariyaeeinia, and N. Pavesic. De-identification for privacy protection in multimedia content: A survey. *Signal Processing: Image Communication*, 47:131–151, 2016.

[189] E. Roelofs, L. Persoon, S. Nijsten, W. Wiessler, A. Dekker, and P. Lambin. Benefits of a clinical data warehouse with data mining tools to collect data for a radiotherapy trial. *Radiotherapy and Oncology*, 108(1):174–179, 2013.

[190] M. Rosenman, J. He, J. Martin, K. Nutakki, G. Eckert, K. Lane, I. Gradus-Pizlo, and S. L. Hui. Database queries for hospitalizations for acute congestive heart failure: flexible methods and validation based on set theory. *Journal of the American Medical Informatics Association*, 21(2):345–352, 2013.

[191] D. L. Rubin and T. S. Desser. A data warehouse for integrating radiologic and pathologic data. *Journal of the American College of Radiology*, 5(3):210–217, 2008.

[192] P. Ruch, R. H. Baud, A.-M. Rassinoux, P. Bouillon, and G. Robert. Medical document anonymization with a semantic lexicon. In *Proceedings of the AMIA Symposium*, page 729. American Medical Informatics Association, 2000.

[193] S. Sachdeva and S. Bhalla. Implementing high-level query language interfaces for archetype-based electronic health records database. In *International Conference on Management of Data (COMAD)*, pages 235–8. Citeseer, 2009.

[194] J. S. Saczynski, S. E. Andrade, L. R. Harrold, J. Tjia, S. L. Cutrona, K. S. Dodd, R. J. Goldberg, and J. H. Gurwitz. A systematic review of validated methods for identifying heart failure using administrative data. *Pharmacoepidemiology and drug safety*, 21:129–140, 2012.

[195] T. R. Sahama and P. R. Croll. A data warehouse architecture for clinical data warehousing. In *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*, pages 227–232. Australian Computer Society, Inc., 2007.

[196] M. Samwald, K. Fehre, J. De Bruin, and K.-P. Adlassnig. The arden syntax standard for clinical decision support: Experiences and directions. *Journal of biomedical informatics*, 45(4):711–718, 2012.

[197] S. Sarawagi et al. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008.

[198] G. Sarganas, H. Knopf, D. Grams, and H. K. Neuhauser. Trends in antihypertensive medication use and blood pressure control among adults with hypertension in germany. *American journal of hypertension*, 29(1):104–113, 2015.

[199] G. D. Schellenbaum, S. R. Heckbert, N. L. Smith, T. D. Rea, T. Lumley, D. W. Kitzman, V. L. Roger, H. A. Taylor, and B. M. Psaty. Congestive heart failure incidence and prognosis: case identification using central adjudication versus hospital discharge diagnoses. *Annals of epidemiology*, 16(2):115–122, 2006.

[200] H. Schmid. Probabilistic part-ofispeech tagging using decision trees. In *New methods in language processing*, page 154, 2013.

[201] G. Schneider, M. Hess, and P. Merlo. *Hybrid long-distance functional dependency parsing*. PhD thesis, Verlag nicht ermittelbar, 2008.

[202] S. Schultz, D. Rothwell, Z. Chen, and K. Tu. Identifying cases of congestive heart failure from administrative data: a validation study using primary care patient records. *Chronic diseases and injuries in Canada*, 33(3), 2013.

[203] I. Segen's Medical Dictionary. 2011. Farlex. Anonymized data. `https://medical-dictionary.thefreedictionary.com/Anonymized+Data`. Accessed Oct 2018.

[204] R. Sennrich, M. Volk, and G. Schneider. Exploiting synergies between open resources for german dependency parsing, pos-tagging, and morphological analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 601–609, 2013.

[205] A. Seyfang, R. Kosara, and S. Miksch. Asbru's reference manual, asbru version 7.3. 2002.

[206] S. J. Shah and R. S. Stafford. Current trends of hypertension treatment in the united states. *American journal of hypertension*, 30(10):1008–1014, 2017.

[207] S.-Y. Shin, W. S. Kim, and J.-H. Lee. Characteristics desired in clinical data warehouse for biomedical research. *Healthcare informatics research*, 20(2):109–116, 2014.

[208] M. Skeppstedt. Negation detection in swedish clinical text: An adaption of negex to swedish. *Journal of Biomedical Semantics*, 2(3):S3, 2011.

[209] V. N. Slee. The international classification of diseases: ninth revision (icd-9). *Annals of internal medicine*, 88(3):424–426, 1978.

[210] S. Sohn, C. Clark, S. R. Halgrim, S. P. Murphy, S. R. Jonnalagadda, K. B. Wagholikar, S. T. Wu, C. G. Chute, and H. Liu. Analysis of cross-institutional medication description patterns in clinical narratives. *Biomedical informatics insights*, 6:BII–S11634, 2013.

[211] S. Sohn, J.-P. A. Kocher, C. G. Chute, and G. K. Savova. Drug side effect extraction from clinical narratives of psychiatry and psychology patients. *Journal of the American Medical Informatics Association*, 18(Supplement_1):i144–i149, 2011.

[212] S. Sohn, S. Wu, and C. G. Chute. Dependency parser-based negation detection in clinical narratives. *AMIA Summits on Translational Science Proceedings*, 2012:1, 2012.

[213] M. Sordo, A. A. Boxwala, O. Ogunyemi, and R. A. Greenes. Description and status update on gello: a proposed standardized object-oriented expression language for clinical decision support. In *Medinfo*, pages 164–168, 2004.

[214] M. Sordo, O. Ogunyemi, A. A. Boxwala, and R. A. Greenes. Gello: an object-oriented query and expression language for clinical decision support. In *AMIA Annu Symp Proc*, volume 1012, 2003.

[215] I. Spasić, F. Sarafraz, J. A. Keane, and G. Nenadić. Medication information extraction with linguistic pattern matching and semantic rules. *Journal of the American Medical Informatics Association*, 17(5):532–535, 2010.

[216] stackoverflow. Developer survey results. `https://insights.stackoverflow.com/survey/2018/`. Accessed Oct 2018.

[217] L. Staerk, E. L. Fosbøl, K. Gadsbøll, C. Sindet-Pedersen, J. L. Pallisgaard, M. Lamberts, G. Y. Lip, C. Torp-Pedersen, G. H. Gislason, and J. B. Olesen. Non-vitamin k antagonist oral anticoagulation usage according to age among patients with atrial fibrillation: Temporal trends 2011–2015 in denmark. *Scientific reports*, 6:31477, 2016.

[218] J. Starlinger, M. Kittner, O. Blankenstein, and U. Leser. How to improve information extraction from german medical records. *it-Information Technology*, 2016.

[219] M. Q. Stearns, C. Price, K. A. Spackman, and A. Y. Wang. Snomed clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium*, page 662. American Medical Informatics Association, 2001.

[220] S. Störk, R. Handrock, J. Jacob, J. Walker, F. Calado, R. Lahoz, S. Hupfer, and S. Klebs. Epidemiology of heart failure in germany: a retrospective database study. *Clinical Research in Cardiology*, 106(11):913–922, 2017.

[221] S. Störk, R. Handrock, J. Jacob, J. Walker, F. Calado, R. Lahoz, S. Hupfer, and S. Klebs. Treatment of chronic heart failure in germany: a retrospective database study. *Clinical Research in Cardiology*, 106(11):923–932, 2017.

[222] J. Strötgen and M. Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324. Association for Computational Linguistics, 2010.

[223] W. F. Styler IV, S. Bethard, S. Finan, M. Palmer, S. Pradhan, P. C. de Groen, B. Erickson, T. Miller, C. Lin, G. Savova, et al. Temporal annotation in the clinical

domain. *Transactions of the Association for Computational Linguistics*, 2:143, 2014.

[224] W. Sun, A. Rumshisky, and O. Uzuner. Temporal reasoning over clinical text: the state of the art. *Journal of the American Medical Informatics Association*, 20(5):814–819, 2013.

[225] D. R. Sutton and J. Fox. The syntax and semantics of the pro forma guideline modeling language. *Journal of the American Medical Informatics Association*, 10(5):433–443, 2003.

[226] H. C. Tissot. Normalisation of imprecise temporal expressions extracted from text. 2016.

[227] M. Toepfer, H. Corovic, G. Fette, P. Klügl, S. Störk, and F. Puppe. Fine-grained information extraction from german transthoracic echocardiography reports. *BMC medical informatics and decision making*, 15(1):91, 2015.

[228] M. Toepfer, G. Fette, P.-D. Beck, P. Kluegl, and F. Puppe. Integrated tools for query-driven development of light-weight ontologies and information extraction components. In *Proceedings of the workshop on open infrastructures and analysis frameworks for HLT*, pages 83–92, 2014.

[229] T. Tolxdorff, J. Braun, T. M. Deserno, H. Handels, A. Horsch, and H.-P. Meinzer. *Bildverarbeitung für die Medizin 2008: Algorithmen-Systeme-Anwendungen*. Springer-Verlag, 2008.

[230] T. Tolxdorff, J. Braun, H. Handels, A. Horsch, and H.-P. Meinzer. *Bildverarbeitung für die Medizin 2004: Algorithmen-Systeme-Anwendungen*. Springer-Verlag, 2013.

[231] T. Tolxdorff and F. Puppe. Klinisches data warehouse. *Informatik-Spektrum*, 39(3):233–237, 2016.

[232] tranSMART Foundation. The transmartapp documentation. `hthttps://transmart-app.readthedocs.io/en/latest/index.html`. Accessed Jan 2019.

[233] E. Union. General data protection regulation (eu-gdpr). `http://www.privacy-regulation.eu/en/article-4-definitions-GDPR.htm`. Accessed Oct 2018.

[234] M. Unnewehr, B. Schaaf, and H. Friederichs. Arztbrief: Die kommunikation optimieren. *Deutsches Ärzteblatt*, 110(37):A–1672 / B–1478 / C–1454, 2013.

[235] Ö. Uzuner, Y. Luo, and P. Szolovits. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563, 2007.

[236] R. Van de Velde and P. Degoulet. *Clinical information systems: a component-based approach*. Springer Science & Business Media, 2003.

[237] S. Velupillai. Temporal expressions in swedish medical text–a pilot study. *Proceedings of BioNLP 2014*, pages 88–92, 2014.

[238] S. Velupillai, D. Mowery, B. R. South, M. Kvist, and H. Dalianis. Recent advances in clinical natural language processing in support of semantic analysis. *Yearbook of medical informatics*, 10(1):183, 2015.

[239] S. Velupillai, M. Skeppstedt, M. Kvist, D. Mowery, B. E. Chapman, H. Dalianis, and W. W. Chapman. Cue-based assertion classification for swedish clinical text—developing a lexicon for pycontextswe. *Artificial intelligence in medicine*, 61(3):137–144, 2014.

[240] R. Vijayakrishnan, S. R. Steinhubl, K. Ng, J. Sun, R. J. Byrd, Z. Daar, B. A. Williams, S. Ebadollahi, W. F. Stewart, et al. Prevalence of heart failure signs and symptoms in a large primary care population identified through the use of text and data mining of the electronic health record. *Journal of cardiac failure*, 20(7):459–464, 2014.

[241] F. Wallentin, B. Wettermark, and T. Kahan. Drug treatment of hypertension in sweden in relation to sex, age, and comorbidity. *The Journal of Clinical Hypertension*, 20(1):106–114, 2018.

[242] K. C. Wang. Standard lexicons, coding systems and ontologies for interoperability and semantic computation in imaging. *Journal of digital imaging*, 31(3):353–360, 2018.

[243] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, et al. Clinical information extraction applications: A literature review. *Journal of biomedical informatics*, 2017.

[244] C. Weng, X. Wu, Z. Luo, M. R. Boland, D. Theodoratos, and S. B. Johnson. Elixr: an approach to eligibility criteria extraction and representation. *Journal of the American Medical Informatics Association*, 18(Supplement_1):i116–i124, 2011.

[245] B. Wu, K. Bell, A. Stanford, D. M. Kern, O. Tunceli, S. Vupputuri, I. Kalsekar, and V. Willey. Understanding ckd among patients with t2dm: prevalence, temporal trends, and treatment patterns—nhanes 2007–2012. *BMJ Open Diabetes Research and Care*, 4(1):e000154, 2016.

[246] S. Wu, T. Miller, J. Masanz, M. Coarr, S. Halgrim, D. Carrell, and C. Clark. Negation's not solved: generalizability versus optimizability in clinical natural language processing. *PloS one*, 9(11):e112774, 2014.

[247] H. Xu, S. Doan, K. A. Birdwell, J. D. Cowan, A. J. Vincz, D. W. Haas, M. A. Basford, and J. C. Denny. An automated approach to calculating the daily dose of tacrolimus in electronic health records. *Summit on Translational Bioinformatics*, 2010:71, 2010.

[248] H. Xu, M. Jiang, M. Oetjens, E. A. Bowton, A. H. Ramirez, J. M. Jeff, M. A. Basford, J. M. Pulley, J. D. Cowan, X. Wang, et al. Facilitating pharmacogenetic studies using electronic health records and natural-language processing: a case study of warfarin. *Journal of the American Medical Informatics Association*, 18(4):387–391, 2011.

[249] H. Xu, S. P. Stenner, S. Doan, K. B. Johnson, L. R. Waitman, and J. C. Denny. Medex: a medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24, 2010.

[250] S. Yoo, S. Kim, K.-H. Lee, C. W. Jeong, S. W. Youn, K. U. Park, S. Y. Moon, and H. Hwang. Electronically implemented clinical indicators based on a data warehouse in a tertiary hospital: its clinical benefit and effectiveness. *International journal of medical informatics*, 83(7):507–516, 2014.

[251] T. Zeh. Data warehousing als organisationskonzept des datenmanagements. *Informatik Forschung und Entwicklung*, 18(1):32–38, 2003.

[252] J. Zhang, V. Carey, and R. Gentleman. An extensible application for assembling annotation for genomic data. *Bioinformatics*, 19(1):155–156, 2003.

[253] H. Zoega, K. Furu, M. Halldorsson, P. H. Thomsen, A. Sourander, and J. E. Martikainen. Use of adhd drugs in the nordic countries: a population-based comparison study. *Acta Psychiatrica Scandinavica*, 123(5):360–367, 2011.