Universidade de Aveiro
2022

Jorge
Faustino

# Processamento automático de texto de narrativas clínicas

# Automatic text processing of clinical narratives

Jorge
Faustino

# Processamento automático de texto de narrativas clínicas

# Automatic text processing of clinical narratives

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor José Luís Guimarães Oliveira, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

**o júri / the jury**

presidente / president

**Doutor Augusto Marques Ferreira da Silva**
Professor Associado da Universidade de Aveiro

vogais / examiners committee

**Doutor Rui Pedro Sanches de Castro Lopes**
Professor Coordenador do Instituto Politécnico de Bragança

**Doutor José Luis Guimarães Oliveira**
Professor Catedrático da Universidade de Aveiro

**agradecimentos / acknowledgements**

Em primeiro lugar à minha família, em especial aos meus pais, por todo o apoio que me deram durante estes anos.

Ao meu orientador, Professor Doutor José Luís Oliveira, e coorientador, João Almeida, pelo acompanhamento e ajuda durante o desenvolvimento desta dissertação.

Aos meus amigos pelo apoio e por me animarem em momentos difíceis.

Por último, mas não menos importante, à Raquel, pela paciência que teve durante esta fase e por toda a ajuda que me deu.

**Palavras Chave**

Mineração de Texto, Processamento de Linguagem Natural, Reconhecimento de Conceitos, Recuperação de Informação.

**Resumo**

A informatização dos sistemas médicos e a subsequente tendência por parte de profissionais de saúde a substituir registos em formato de papel por registos electrónicos de saúde, permitiu que os serviços de saúde se tornassem mais seguros e eficientes. Além disso, estes registos electrónicos apresentam também o benefício de poderem ser utilizados como fonte de dados para estudos observacionais. No entanto, estima-se que 70-80% de todos os dados clínicos se encontrem na forma de texto livre não-estruturado e os dados que estão estruturados não seguem todos os mesmos padrões, dificultando o seu potencial uso nos estudos observacionais.

Esta dissertação pretende solucionar essas duas adversidades através do uso de processamento de linguagem natural para a tarefa de extrair conceitos de texto livre e, de seguida, usar um modelo comum de dados para os harmonizar. O sistema desenvolvido utiliza um anotador, especificamente o cTAKES, para extrair conceitos de texto livre. Os conceitos extraídos são, então, normalizados através de técnicas de pré-processamento de texto, Word Embeddings, MetaMap e um sistema de procura no Metathesaurus do UMLS. Por fim, os conceitos normalizados são convertidos para o modelo comum de dados da OMOP e guardados numa base de dados.

Para testar o sistema desenvolvido usou-se o conjunto de dados i2b2 de 2010. As diferentes partes do sistema foram testadas e avaliadas individualmente sendo que na extração dos conceitos obteve-se uma precisão, recall e F-score de 77.12%, 70.29% e 73.55%, respectivamente. A normalização foi avaliada através do desafio N2C2 2019-track 3 onde se obteve uma exatidão de 77.5%. Na conversão para o modelo comum de dados OMOP observou-se que durante a conversão perderam-se 7.92% dos conceitos. Concluiu-se que, embora o sistema desenvolvido ainda tenha margem para melhorias, este demonstrou-se como um método viável de processamento automático do texto de narrativas clínicas.

**Keywords** Concept Recognition, Information Retrieval, Natural Language Processing, Text Mining.

**Abstract**

The informatization of medical systems and the subsequent move towards the usage of Electronic Health Records (EHR) over the paper format by medical professionals allowed for safer and more efficient healthcare. Additionally, EHR can also be used as a data source for observational studies around the world. However, it is estimated that 70-80% of all clinical data is in the form of unstructured free text and regarding the data that is structured, not all of it follows the same standards, making it difficult to use on the mentioned observational studies.

This dissertation aims to tackle those two adversities using natural language processing for the task of extracting concepts from free text and, afterwards, use a common data model to harmonize the data. The developed system employs an annotator, namely cTAKES, to extract the concepts from free text. The extracted concepts are then normalized using text preprocessing, word embeddings, MetaMap and UMLS Metathesaurus lookup. Finally, the normalized concepts are converted to the OMOP Common Data Model and stored in a database.

In order to test the developed system, the i2b2 2010 data set was used. The different components of the system were tested and evaluated separately, with the concept extraction component achieving a precision, recall and F-score of 77.12%, 70.29% and 73.55%, respectively. The normalization component was evaluated by completing the N2C2 2019 challenge track 3, where it achieved a 77.5% accuracy. Finally, during the OMOP CDM conversion component, it was observed that 7.92% of the concepts were lost during the process. In conclusion, even though the developed system still has margin for improvements, it proves to be a viable method of automatically processing clinical narratives.

# Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| **API** | Application Programming Interface |
| **CDM** | Common Data Model |
| **CUI** | Concept Unique Identifier |
| **EHR** | Electronic Health Record |
| **EMR** | Electronic Medical Record |
| **ETL** | Extract Transform Load |
| **FN** | False Negative |
| **FP** | False Positive |
| **ICD** | International Classification of Diseases |
| **NEN** | Named Entity Normalization |
| **NLP** | Natural Language Processing |
| **NER** | Named Entity Recognition |
| **OHDSI** | Observational Health Data Sciences and Informatics |
| **OMOP** | Observational Medical Outcomes Partnership |
| **OMOP-CDM** | Observational Medical Outcomes Partnership Common Data Model |
| **POS** | Part of Speech |
| **TN** | True Negative |
| **TP** | True Positive |
| **UMLS** | Unified Medical Language System |

# Chapter 1

# Introduction

## 1.1 Context and motivation

Over the years efforts were made to improve the healthcare system through informatization, with healthcare providers moving towards the usage of Electronic Health Records (EHR) replacing the old paper based format. An EHR is a digital version of a patient's chart [56]. Its digital format inherently makes it easier to keep up-to-date across multiple systems, allowing for a faster lookup of a patient's history as well as speeding up appointments and office visits. This leads to a safer and more reliable prescribing with less medical errors due to information being outdated or due to illegible handwriting [28]. For instance, in Louisiana, Jane Herwehe et al. [61] implemented a system using the Louisiana Public health Information Exchange (LaPHIE) as a source of patient data to alert medical providers when a patient with HIV/AIDS had not received care in over twelve months. Overall, hospitals with electronic medical record systems that have automated notes and records and clinical decision support had fewer complications and lower mortality rates from medical errors, as well as lower costs [35][80]. Additionally the storing of the records is also safer, as paper is prone to deterioration and damages while digital data can be backed up periodically [28].

The other benefit of EHRs, and the one that will be the main focus of this dissertation, comes from its secondary use, its use as a data source for observational studies. EHRs are an increasingly important source of real-world healthcare data for observational research [43], as they include large and diverse populations that represent real-world patterns of disease and treatment. This diversity of the data sources also helps mitigate a common problem when using data from a single medical unit or hospital, in that the data leads to heavily biased results. Analyzing existing data also tends to be less expensive, less time consuming and overall more convenient than creating a new curated dataset. As pointed out in the work of Victor M.Castro et al. [46], researchers have used these data sources to test targeted associations between drugs and possible adverse effects of those drugs [96][45] and to compare the effectiveness of established therapies [47] Overall, having access to this much patient data worldwide creates the potential to vastly improve medical discoveries and advancements.

Regarding the format of the data that can be present in an EHR, one could split it into two categories, structured and unstructured. Structured data typically encodes lab values, encounters and medication lists. Alternatively unstructured data is typical natural language free text, commonly found in the form of medical notes or clinical narratives [68]. Free text makes it harder for automated systems to operate normally as computers don't inherently

understand the data present in natural language texts. One may question why don't medical professionals simply use the regular data fields and insert the data in a structured way, but clinical narratives play an important role in the healthcare system. For example, it was shown in the work of Preethi Raghavan et al. [85] that clinical narratives are essential to solving 59% of the chronic lymphocytic leukemia (CLL) trial criteria and 77% of the prostate cancer trial criteria, specifically, for resolving eligibility criteria with temporal constraints. Writing the narrative helps clinicians describe the current clinical practice in a way that is more convenient to share and discuss with other professionals. Additionally the narrative can help them reflect on their practice or better analyze a peculiar clinical case [29]. This combination of factors creates a dilemma, as it's both important to keep free text, as well as making it more structured and easier for automated systems to extract information from it, demonstrating the importance of information retrieval and data uniformization methods.

## 1.2 Objectives

The focus of this dissertation is the development of tools for processing of the high volume of unstructured data stored in clinical notes using Text mining and Natural Language Processing (NLP) techniques to extract relevant medical concepts from them, with the purpose of then normalizing the extracted data into a common format (data model). The goal of this dissertation can thus be divided in the following four main steps:

- Explore the current state of the art NLP and text mining software as well as text processing techniques;;
- Define a dataset, which must already have existing normalized data available in order to validate the results of the system being developed;
- Develop text processing tools as well as employ existing NLP ones to extract the data from the clinical notes;
- Normalize the data extracted from the dataset into the common data model.

## 1.3 Outline

This dissertation is organized into five more chapters, which are described below.

**Chapter 2 - State of the art**, presents current existing technologies, tools and methods related to the main topics of this project. It begins by presenting some of the existing methods of extracting information from unstructured data in the form of text, as well as the metrics used to grade those extraction methods, then it proceeds to a comparison between multiple state of the art annotating software, and the chapter concludes by discussing common data models.

**Chapter 3 - N2C2 challenge**, describes the N2C2 2019 challenge track-3 and the system developed to complete it. It starts by explaining what the goals of the challenge are, as well as what dataset and evaluation methods will be used. A brief explanation regarding the motivation to complete this challenge and how it fits within this dissertation's theme is also provided. Proceeding those introductory sections of the chapter, the steps taken to complete the challenge, through the development of a Named Entity Normalization (NEN) system, are described.

**Chapter 4 - System implementation**, presents the implemented system and provides a discussion regarding its components. It starts by presenting the Named Entity Recognition (NER) component developed, including the stages of its development and why each decision was made in the process of improving its performance. Afterwards it presents the Extract Transform Load (ETL) component, providing a brief explanation of its goal in the system and expected outcomes, followed by the steps taken to make it possible.

**Chapter 5 - Results**, presents the results obtained from the tests done to the system components implemented in chapter 4, providing a discussion regarding the methods used to test and validate them, as well as the performance impact of each step taken to improve the system.

**Chapter 6 - Conclusion**, briefly summarizes the accomplished work. Additionally some ideas for future work are proposed to make the system easier to use from the end user perspective, as well as pointers to improve the effectiveness of the implemented system.

# Chapter 2

# State of the art

This chapter aims to describe the current most relevant technologies, tools and methods related to the main topics of this project. It begins by presenting some of the existing methods of extracting information from unstructured data in the form of text, as well as the metrics used to grade it. Proceeds to describe some of the main annotators, as well as some of the related work done using them, and finalizes with a discussion about common data models (CDM).

## 2.1 Dealing with unstructured data

As already mentioned in the introduction chapter, clinical narratives, which are a form of unstructured free text, are an essential part of healthcare, especially in the more complex cases, where the very act of putting thoughts in writing helps the healthcare professionals in solving such cases. But they pose an added adversity to the automation of electronic medical records systems that rely on data being properly structured. An estimated 70-80% of all clinical data are available in free text documents [55], and although even text documents may contain a few structured fields, such as title and date, the vast majority of its data is in unstructured text form, and in order to use, query and analyze the data it needs to be properly structured with each piece of information being tagged and categorized. With those considerations, in order to derive high quality data from unstructured text one must resort to methods of text mining and to a pivotal subset of text mining - Natural Language Processing (NLP) [94].

Text mining is the process of exploring and analyzing large quantities of unstructured text data [52] aided by software that can identify lexical or linguistic usage patterns in the data with the purpose of extracting high-quality information from that same unstructured text data. This encompasses tasks from information retrieval, concept extraction, to text classification [60]. When using text mining in combination with the previously discussed electronic health records (EHR), it allows for compelling developments like an hospital system capable of estimating future bed demand using only textual information from early medical records from the emergency department [73]; or a surgical site surveillance system that, through pattern-matching based text mining of electronic health and administrative records for patients who underwent surgical procedures recently, predicts whether the patient developed a surgical site infection [37]; or predicting patient readmission by identifying specific factors recorded in primary care [107]. Text mining is for the most part a probabilistic process, it targets

patterns in an attempt to extract probably useful and probably correct information [102].

NLP is a component of text mining, it allows software to process, analyze and derive information from the human natural language. While text mining focuses on the text itself, NLP focuses on the underlying metadata. As described by Anne Kao et al. [66] it can be put roughly as figuring out who did what to whom, when, where, how and why. NLP makes use of linguistic concepts such as Part of Speech (POS) (e.g. noun, verb, adjective) and grammatical structure. It has to deal with ambiguities and figure out what previously mentioned noun is a pronoun referring to [66]. By better understanding the underlying meaning of the text rather than just relying on specific text patterns and triggers, it allows for a deeper, more nuanced analysis of health records. For instance, this enabled the development of a system that is able to compute the chances of hospital fall risk. The system analyzes registered nurses' electronic narrative notes and discover if there is meaningful fall risk, as it was found that these can contain information about clinical as well as environmental and organizational factors that could affect fall risk but are not explicitly recorded by the provider as fall risk factors [40]. Additionally in another recent study NLP was used to process hospital discharge notes to improve the prediction of suicide and accidental death after discharge from hospitals [64]. Ultimately, and especially in more advanced and complex systems, one could say there is some overlap between what's considered pure text mining and what's considered NLP.

## 2.2 Text processing

To ease the process of deriving data from text, certain methods of text processing are typically applied before the actual data retrieval process. The most commonly applied text processing methods are normalization, tokenization and chunking. While the purpose of normalization is transforming the text to improve the information extraction, the other two methods are more centric on the actual extraction of, selecting, tagging and grouping up information [59][65]. It's also during the tagging of information that part-of-speech tagging is performed, where each word is identified as being a noun, pronoun, verb, etc [77]. These methods aren't typically done separated but rather combined and applied sequentially.

### 2.2.1 Normalization

The goal of normalization is to turn every word to their most standard form allowing processing to proceed uniformly. This is done through processes such as lowercasing, stemming, lemmatization, punctuation removal or converting numbers to their word equivalents [1]. Typically, the first step of normalization is lowercasing, which is a process where all text is converted to lowercase, for example "Canada" becomes "canada", like demonstrated on Table 2.1. This makes it so that all variations of casing of a word will be converted to a single uniform one, making the text processor more efficient.

Table 2.1: Demonstration of the lowercasing process used in text preprocessing

| **Original word** | **Lowercase word** |
|---|---|
| Canada | canada |
| PORTUGAL | portugal |
| TomCat | tomcat |

As the goal is to extract valuable information, words that carry no value are removed. In text mining these words are called stop words [86], which are words that add no meaningful data and serve only to aid speech, such as "the", "a", "as", "in". Addiotionally, symbols (e.g. "#", "@") as well as punctuation marks (e.g. quotation marks, question mark, comma, hyphen) are also removed and replaced with a single space. These methods of text uniformization can be seen in Table 2.2.

Table 2.2: Examples of text preprocessing methods of removal of stop words and removal of punctuation marks

| **Original phrase** | **Processed phrase** |
|---|---|
| The "work" is done | work is done |
| John's car? | john car |
| Hello, got the e-book? | hello got e book |

In order to make the text more uniform, word contractions are also expanded [21], meaning that words such as "don't" are converted to "do not", which can be observed in Table 2.3. This step has to be carefully planned with the previously mentioned symbol / punctuation removal, because if the apostrophe is removed from a contraction, the algorithm might not recognize the contraction, this can be solved by either not removing the apostrophe until this step is done, not removing apostrophes in those specific cases, or making the contraction algorithm recognize the contraction without the apostrophe, it's up to the developer to decide on a solution.

Table 2.3: Showcasing the text mining process of expansion of contractions

| **Contraction** | **Expanded contraction** |
|---|---|
| Don't | Do not |
| It's | It is |
| She'll | She will |

While most other steps in normalization are applied sequentially and used together, the next step involves two methods where only one of them is applied, depending on the preference of the developper, these methods are stemming and lemmatization. Stemming consists of reducing inflection in words (e.g. troubled, troubles) to their root form (e.g. trouble). It uses a crude heuristic process that chops off the ends of words, so in some cases the words "trouble", "troubled" could be stemmed to "troubl" instead of "trouble". A stemmer algorithm processes each word without taking context into account, it cannot differentiate between words which have different meanings depending on part of speech. The advantage of stemmers is that they're easier to implement and run faster, and the reduced accuracy may not matter for some applications [78]. Alternatively, while lemmatization has the same goal as stemming (i.e. reducing inflection in words and using their root form), it tries to do it the proper way. Rather than just chopping letters off, it transforms words to their actual root, even if visually they're not similar, for example, the word "better" would map to "good". A comparison between the two methods can be seen in Table 2.4.

Table 2.4: Examples of the comparisons between applying stemming and lemmatization algorithms to the same words.

| **Word** | **Stemming** | **Lemmatization** |
|---|---|---|
| better | better | good |
| computers | comput | computer |
| are | are | be |
| is | is | be |
| wanted | want | want |
| information | inform | information |
| informative | inform | informative |

### 2.2.2 Tokenization and chunking

Tokenization is essentially splitting a phrase, sentence, paragraph or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. One can think of tokens as parts like a word is a token in a sentence, and a sentence is a token in a paragraph [14][20]. It's typically in this step that the previously mentioned Part of Speech (POS) tagging process is applied. This process can be seen in Figure 2.1 where the sentence "We saw the yellow dog" is tokenized and each token is adequately identified with the corresponding POS tag. The tags present in the figure are personal pronoun (PRP), verb (VBD - past tense), determiner (DT), adjective (JJ) and noun (NN - singular) respectively.

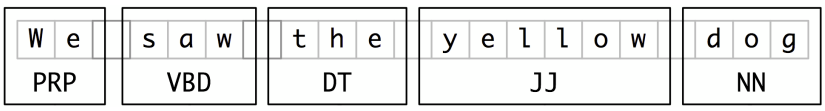


Figure 2.1: Example of the tokenization with part of speech tagging [12].

In order to improve the tokenization process, chunking [30] may be utilized. Chunking is a process where multi-token sequences are combined and labeled, called chunks. It works on top of POS tagging, it uses POS-tags as input and provides chunks as output. One of the main goals of chunking is to group into what are known as "noun phrases", typically referred to as NP. These are phrases of one or more words that contain a noun, maybe some descriptive words, maybe a verb, and maybe something like an adverb. The idea is to group nouns with the words that are in relation to them. Using the previously shown POS tagging Figure 2.1 as baseline, it's shown in Figure 2.2 the result of employing the chunking process to it. The smaller boxes show the word-level tokenization and part-of-speech tagging, while the large boxes show higher-level chunking. Each of these larger boxes is called a chunk. Like tokenization, which omits whitespace, chunking usually selects a subset of the tokens. Also like tokenization, the pieces produced by a chunker do not overlap in the source text.

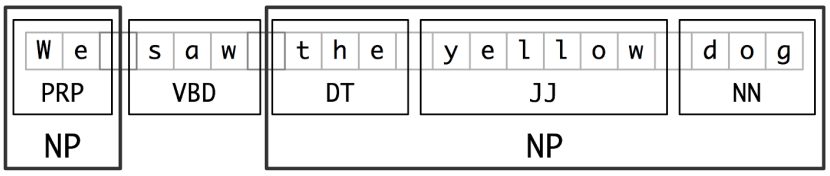


Figure 2.2: Segmentation and labeling at both the token and chunk levels.

### 2.2.3 Named Entity Recognition (NER) and Normalization (NEN)

These methods of text processing, in the context of this dissertation, ultimately culminate in Named Entity Recognition (NER) and Named Entity Normalization (NEN). Named Entities are defined as proper names and quantities of interest. This includes person and location names, as well as dates, times, percentages and monetary amounts [50][81], or, in the context of this dissertation, they can be thought of as medical concept names, dosages, etc.

NER refers to the task of locating those same entities in free text and subsequently classifying them into predefined categories such as person names, locations, medical codes or quantities. This is typically done through the usage of annotators, which like the name suggests, are software used to annotate those named entities in free text. These annotators will be discussed in more detail further in this chapter. NER plays an important role in applications of information extraction and can be used as a source of information for NLP applications [79].

Named Entity Normalization, like the name suggests, is a normalization task where it's assigned suitable identifiers to recognized entities, and it's typically done after the NER process. NEN is a challenging task, especially in the medical field, as many terms have multiple synonyms and variations and medical professionals often refer to them using abbreviations [70]. Several NER and normalization studies have been conducted in the past years to improve and resolve these ambiguities, for instance, in the work of Hyejin Cho et al. [51], where word embeddings created with the data from National Center for Biotechnology Information (NCBI) were employed.

## 2.3 Evaluation methodology of NER systems

This section will go over the methodology typically employed when evaluating the performance of a NER system. Before discussing some of NER tools and some of the work done using each one of them, which will be presented in the next section, one must understand the terminology used when grading their performance, what metrics are used, and why those metrics are used. Evaluation metrics make use of four base definitions:

- True Positive (TP): correct prediction that an instance is positive;
- False Positive (FP): incorrect prediction that an instance is positive;
- True Negative (TN): correct prediction that an instance is negative;
- False Negative (FN): incorrect prediction that an instance is negative;

These metrics form the confusion matrix, observable in table 2.5, effectively measuring the totality of outcomes [69][91].

These metrics form the confusion matrix, observable in table 2.5, effectively measuring the totality of outcomes [69][91].

Table 2.5: Confusion matrix of evaluation metrics for a NER system.

| | | Predicted | |
|---|---|---|---|
| | | Positive | Negative |
| Actual | Positive | TP | FN |
| | Negative | FP | TN |

Through the presented metrics that form the confusion matrix it's possible to derive the standard metrics: precision, recall, accuracy and F-score, which all take values between zero and one. Precision (Equation 2.1) measures the ability of a system to present only the relevant entities, it's the proportion of relevant entities among the retrieved entities. In contrast, recall (Equation 2.2) measures the ability of a system to present all relevant entities, it's the proportion of relevant entities that were retrieved out of all relevant entities in the collection [84].

$$Precision = \frac{relevant\ items\ retrieved}{total\ items\ retrieved} = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{relevant\ items\ retrieved}{relevant\ items\ in\ collection} = \frac{TP}{TP + FN} \tag{2.2}$$

Accuracy (Equation 2.3) represents the proportion of correct classifications out of all classifications. Although a high accuracy value is generally good, the number can be very misleading as it does not take into account how the data is distributed. For instance, in screening for a relatively rare condition, one can achieve a high accuracy ignoring all evidence and classifying all cases as negative. If only 5% of patients have the condition in question, a physician who always blindly states that the condition is absent will be right 95% of the time so, for this reason accuracy may be considered as being of limited usefulness as an index of diagnostic performance [76].

$$Accuracy = \frac{correct\ classifications}{all\ classifications} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

For the reasons previously presented, the metric that is typically used to evaluate NER systems is the F-score (Equation 2.4), also commonly referred to as F1-score or F-measure, which is the harmonic mean of precision and recall. The harmonic mean gives a better measure of incorrectly classified cases as it penalizes extreme values, thus being more resistant to being misleading in cases of highly imbalanced data sets.

$$Fscore = 2\ \frac{Precision\ .\ Recall}{Precision + Recall} \tag{2.4}$$

It should be noted that no metric is infallible, although F-score has good qualities, the metric has received some criticism, namely from David Hand and Peter Christen [58] for giving equal importance to precision and recall, with the argument that different types of mis-classifications incur different costs. It has also been pointed out by David Powers [84] that, due to the nature of its formula, it ignores True Negatives.

## 2.4 Available software

This section will go over some of the more relevant available software for annotating medical concepts. It will go over the following software: cTAKES [8], MetaMap [18], MedEx [106], MedTagger [17], BeCAS [5] and Neji [19]. Some of which are already equipped with a pipeline that include the tools for text pre-processing before deriving the information from the text. A brief description of each annotator will be presented, as well as some of the work that other teams have accomplished using said software. The choice of these annotators is based on a multitude of factors including popularity in the scientific community, the amount of research papers in which they are mentioned and performance that other researchers achieved while using them.

### 2.4.1 cTAKES

cTAKES, originally developed by a team of physicians, computer scientists and software engineers at the Mayo Clinic, is an open source, NLP system. It's an acronym for clinical Text Analysis and Knowledge Extraction System and, like the name suggests, was designed aiming at processing clinical free texts originating from EHRs, identifying relevant entities, such as diseases, treatments and drugs, and mapping them to their corresponding Unified Medical Language System (UMLS) [41] Metathesaurus concepts. It was built upon the Unstructured Information Management Architecture (UIMA) [2][54] framework using components trained specifically for clinical text [92]. One of its most appealing features is its modular, fully customizable pipeline, and although by default it includes a version of the UMLS Metathesaurus as its dictionary, it allows the usage of custom dictionaries provided by the user.

Kersloot MG et al. [67] included cTAKES in their developed web-based Medical Language Processing system. Their system uses cTAKES for the pre-processing and processing of free-text clinical narratives, since it uses the UMLS as its dictionary, providing a generic way of concept matching and detection of syntactic relationships.

Another team used cTAKES to build a speech transcriber for Electronic Medical Records use [101]. The team developed a speech transcriber for a web-based Electronic Medical Record (EMR) with the goal of dealing with a major challenge faced by clinicians who use EMR, which is the lowered perceived quality of patient-doctor communication and interaction as a result of doctors being distracted with typical EMR forms use during consultations. In their developed system cTAKES is later used for clinical annotation on the transcribed speech.

### 2.4.2 MetaMap

MetaMap is a biomedical text annotation tool developed by Dr. Alan Aronson [36] at the National Library of Medicine (NLM). It's a rule-based tool that uses NLP and computational linguistic techniques to identify possible biomedical concept mentions in text and map them to the UMLS Metathesaurus. It evaluates the annotations based on centrality, variation, coverage and cohesiveness, scoring them based on their probability of being a match. It provides some degree of configurability, allowing users to set option flags that control distinct modules, as well as the output format, including XML, JSON and more human-readable text formats.

Its capabilities as a tool for extracting relevant information from EHRs have been demonstrated in the work of Jinying Chen et al. [48] - FOCUS (Finding impOrtant medicalConcepts

most Useful to patientS), a system designed to help patients better understand their own EHR notes. Many health organizations allow patients to access their own EHR notes through online patient portals, however, EHR notes are typically long and contain abundant medical jargon that can be difficult for patients to understand. FOCUS first identifies candidate terms from each EHR note using MetaMap and then ranks the terms using a support vector machine-based learn-to-rank algorithm.

MetaMap's concept recognition capabilities can be used to ease the creation of subgroups of populations, this can be seen in the work of Ruth Reátegui and Sylvie Ratté [87]. In their work they analyzed discharge summaries of overweight and diabetic patients from the i2b2 (Informatics for Integrating Biology to the Bedside) Obesity dataset and used MetaMap in conjunction with UMLS to identify both the Concept Unique Identifiers (CUI) and the preferred name that correspond to two semantic types: (1) Disease or Syndrome, and (2) Mental or Behavioral Dysfunction. Three subgroups were identified after the diseases were extracted and aggregated, which correspond to patients with sleep apnea, patients with heart diseases, and patients with communicable diseases.

### 2.4.3 MedEx

MedEx was originally developed by Dr Hua Xu et al. [106] at the Vanderbilt University Medical Center. The system was designed specifically for the task of extracting drug related details from clinical notes, which means it extracts drug names as well as signature information about drug administration, strength, route, and frequency. The names of the drugs are derived from the RxNorm and UMLS [53]. The system consists of two main components: a semantic tagger that labels words or phrases with a semantic category and a parser that uses a context-free grammar to parse textual sentences into structured forms based on pre-defined semantic patterns[105].

In a recent work MedEx was used on free-text clinical notes, as part of an automated Electronic Health Record (EHR) data extraction system. It extracts information on eye medications and compares it to information from medication orders to aid measuring visual acuity (VA) and intraocular pressure (IOP) outcomes of cataract and glaucoma surgeries, which are two types of surgeries undertaken to improve VA and lower IOP [100].

In a different work [103], MedEx's capabilities in extracting information from medical summaries were used to detect candidate noncancer drugs that can potentially be used for the treatment of cancer. In order to achieve this, they used the synthetic derivative (SD) as a data source, which contains comprehensive clinical data for more than 2.3 million patients, and used MedEx to extract medication information from both structured (e.g. electronic physician orders) and unstructured (i.e. clinical notes) data.

### 2.4.4 Becas

BeCAS is an online based biomedical concept recognition system that also presents a visual delineation of the annotated concepts in the text [82]. It was built upon a modular biomedical concept recognition system, integrating modules for PubMed article fetching, tokenization, lemmatization, POS tagging, chunking, concept identification, abbreviation resolution and interactive visual concept highlighting. Motivated by the lack of offers of no-installation, no-maintenance and online modular solutions for concept annotation that can be easily integrated in any text-processing pipeline, BeCAS provides its features through three interfaces: an

HTTP REST API, a widget embeddable in web pages and an interactive web application.

Facihul Azam et al. [38] in an effort to provide a global overview of prostate cancer research in genetics, used the BeCAS API for information retrieval and named entity recognition demonstrating how to integrate text mining with network analysis investigating research contributions of countries and collaborations within and between countries. Alternatively, using BeCAS in conjunction with the already discussed MetaMap as terminology-driven baselines, Noha Alnazzawi et al. [34] developed PhenoNorm which integrates a number of different similarity measures to allow automatic linking of phenotype concept mentions to known concepts in the UMLS to help uncover new disease-phenotype associations.

### 2.4.5 Neji

Neji [44] is a modular, open source framework specialized in biomedical concept recognition, integrating modules for biomedical NLP, such as sentence splitting, tokenization, lemmatization, part-of-speech tagging and chunking. It uses a hybrid method of concept recognition, combining dictionary matching and machine learning, and it supports overlapped concept names and disambiguation techniques due to a concept tree implementation. It's particularly appealing for developers and researchers as a result of the ease it provides in implementing new modules or using pre-defined pipelines and its support for the most popular input and output formats, namely Pubmed XML, leXML, CoNLL and A1.

João Rafael Almeida and Sérgio Matos [33] proposed extracting family history information, from clinical notes in EHRs using NLP, and using this knowledge to help in diagnosis and prognosis of patients. They used a Neji annotation server with a disease dictionary compiled from the UMLS Metathesaurus, combined with Stanford CoreNLP tools to identify disease mentions in the clinical notes.

Another example of Neji's application can be seen on the work of Sérgio Matos et al. [75], as they used Neji to pre-annotate documents, before importing them to Egas [11], a web-based platform for text-mining assisted literature curation, to help on the task of identifying mentions of human genomic variants in the biomedical literature, and associating these mentions to corresponding genes, phenotypes and clinical attributes.

### 2.4.6 MedTagger

MedTagger [71] is an open-source NLP pipeline based on the Unstructured Information Management Architecture (UIMA) framework for indexing based on dictionaries, information extraction, and machine learning–based NER from clinical text. It's composed of three main components: MedTagger for indexing based on dictionaries, MedTaggerIE for information extraction based on patterns, and MedTaggerML for machine learning-based NER.

Using MedTagger to break down sentences into words and identify concepts related to peripheral artery disease (PAD), Naveed Afzala et al. [31] aimed to identify critical limb ischemia (CLI) from clinical notes. They developed a NLP algorithm, extended from a previously validated NLP algorithm for PAD identification, and after MedTagger identified the relevant concepts, their algorithm mapped specific categories to those concepts that were later used for patient classification.

Additionally Sijia Liu et al. [72] proposed a rule-based information extraction system to extract lab test results from clinical notes, mainly aimed at lab test results for referral patients or lab tests that can be done in a non-clinical setting, in both cases the results can

be captured in unstructured clinical notes. The sentence detector, tokenizer, part-of-speech tagger and chunker are from MedTagger. The sentence boundaries obtained from the sentence detector are used for the separation of semantic concepts.

### 2.4.7 Comparing and combining annotators

Over the years several teams benchmarked and compared some of these NLP systems against each other. While these annotators have the same general goal of extracting relevant information from free text they have vastly different implementation aspects, scalability potential and even resource requirements of the computer system hosting and running it. As a means to determine which annotator, or combination of annotators, is more adequate to further achieve the objective of this dissertation, an analysis was conducted on previous studies about the subject of benchmarking annotators, as well as the possibility of combining them in a single system.

#### Benchmarks

Alejandro Rodríguez-González et al. [90] compared MetaMap and cTAKES in their research. They developed a software that extracts diagnosis-related content from Web pages, then applied a named-entity recognition approach based on MetaMap and cTAKES to extract all relevant terms. The evaluation was performed by doing a manual analysis of the results and parameters were computed in order to calculate precision, recall, specificity and F1 score values. They found that overall the systems have similar results, however there were some cases where one of the systems would have a slightly higher precision for specific drugs while still having similar F1 scores.

Ruth Reátegui and Sylvie Ratté [88] compared the performance of MetaMap and cTAKES in the task of entity extraction in clinical notes from the i2b2 (Informatics for Integrating Biology to the Bedside) Obesity Challenge data. The results were evaluated with manually annotated medical entities and it was found that MetaMap had an average of 0.88 in recall, 0.89 in precision, and 0.88 in F-score and cTAKES had an average of recall, precision and F-score of 0.91, 0.89, and 0.89, respectively.

Eugene Tseytlin et al. [97] benchmarked 5 state-of-the-art semantic annotators, NOBLE Coder (which they developed), cTAKES, MetaMap, Footnote 3 ConceptMapper and MGrep. The benchmarking was done on two publicly available human-annotated corpora, ShARe, consisting of annotated clinical notes, and CRAFT, consisting of annotated biomedical literature. All the tools performed better on the clinical notes corpus (ShARe) On the ShARe corpus, NOBLE Coder, cTAKES, MGrep and MetaMap were of comparable performance, while ConceptMapper lagged behind. On the CRAFT corpus, NOBLE Coder, cTAKES, MetaMap and ConceptMapper had very similar results, whereas MGrep performed significantly worse. In terms of speed, ConceptMapper was the fastest one, followed by cTAKES and NOBLE Coder with similar results, and MGrep with slightly slower results. MetaMap was by far the slowest (30 times slower than the best performing tool).

#### Combining systems

Alternatively to comparing systems to pick one to use, it's possible that combining NLP systems could yield better results. Yunqing Xia et al. [104] used a combination of MetaMap

and cTAKEs for disorder recognition for the ShARe/CLEF eHealth 2013 task 1. They implemented two baseline systems, one system is built using MetaMap and the other using cTAKES. Afterwards they developed a system with the two previous systems combined that performed better than either single system.

#### Applying NLP in social media and other non formal settings

One other method to understand the capabilities of these annotators may be to look at some of its applications in tasks other than extracting data from clinical notes. Health forums enable patients to learn and communicate on health issues online. These online social platforms have millions of users and mining such large scale user generated content (UGC) could help better understand users and patients on many health related topics.

Hongkui Tu et al. [98] randomly selected 100 posts, split the posts into sentences, and later the sentences into words, and labeled all words with MetaMap. As a control method the posts were also manually annotated. However the precision obtained on the word labels was only 43.75%. They noticed that only a very small percentage of UMLS concepts are discussed in medical forums, as most semantic types are not of interest (e.g., Reptile) or too domain-specific (e.g., Cell Function). Their results show that more general semantic types such as "Body Part, Organ, or Organ Component" (e.g. ears, hair) obtained high precisions (greater than 85%). But very specific semantic types such as "Immunologic Factor" obtained very low precisions (below 10%). Due to the nature of this data, another semantic type that had high precision combined with a big amount of occurrences was "Mental Process" (e.g. think, hope), words that are widely used in online discussions but not for communicating medical related issues. This combination of factors led them to the conclusion that directly applying MetaMap on social media data on healthcare leads to low quality word labels. Like most NLP systems, MetaMap is designed for processing medical data written by professionals rather than UGC in online forums.

## 2.5 Common data models CDM

### 2.5.1 Context

As mentioned previously, the data present in EHRs is an important contributor to healthcare related observational studies around the world. In order to take full advantage of the data in observational studies, the data often needs to be compared and contrasted, however observational databases don't all follow a standard when storing data, healthcare data can vary greatly from one organization to the next, and even vary within the same organization. For instance, EHRs are aimed at supporting clinical practice at the point of care, while administrative claims data are built for the insurance reimbursement processes. Each has been collected for a different purpose and may be stored in different formats using different database systems and information models. Each organization follows their local rules, for instance, in one database the patient information could be stored in a table named "patient", in another it could be "person", in one of them there could be a field called "name", while in another it's "patient_name". As a result one would need a specific query for each database just to retrieve the information [25].

Despite the growing use of standard terminologies in healthcare [9], the same concept may be represented in a variety of ways from one setting to the next, even if the database format

(i.e. table names, field names) is the same, different naming conventions regarding medical concepts could be used. As illustrated in Figure 2.3, four real observational databases, all containing an inpatient admission (i.e. person who has been admitted to a hospital for bed occupancy purposes) for a patient with a diagnosis of "acute subendocardial infarction", and yet they all have different table names, column names, table structures, ICD code writing conventions (with and without decimal points) and different name conventions (ICD9 and ICD10).

**Truven MarketScan Commercial Claims and Encounters (CCAE)**
**INPATIENT_SERVICES**

| enrolid | admdate | pdx | dx1 | dx2 | dx3 |
|---|---|---|---|---|---|
| 1570337021 | 5/31/2000 | 41071 | 41071 | 4241 | V5881 |

**Optum Extended SES**
**MEDICAL_CLAIMS**

| patid | fst_dt | diag1 | diag2 | diag3 | diag4 |
|---|---|---|---|---|---|
| 259000476532 | 5/30/2000 | 41071 | 27800 | 4019 | 2724 |

**Premier**
**PATICD_DIAG**

| pat_key | period | icd_code | icd_pri_sec |
|---|---|---|---|
| -17197140 | 36526 | 410.71 | P |
| -17197140 | 1/1/2000 | 414.01 | S |
| -17197140 | 36526 | 427.31 | S |
| -17197140 | 1/1/2000 | 496 | S |

**Japan Medical Data Center**
**DIAGNOSIS**

| member_id | admission_date | icd10_level4_code |
|---|---|---|
| M0041437 | 41582 | I214 |
| M0041437 | 4/11/2013 | A539 |
| M0041437 | 41582 | B182 |
| M0041437 | 4/11/2013 | E14- |

Figure 2.3: Example of an inpatient admission diagnosed with acute subendocardial infarction from four real observational databases [27].

The purpose of a CDM is to standardize both the format and content of observational data, thus allowing common software applications and analytics tools to be easily applied across datasets from multiple healthcare organizations [22].

### 2.5.2 Observational Medical Outcomes Partnership Common Data Model (OMOP-CDM)

The OMOP-CDM can accommodate both administrative claims and EHRs [25], as it defines a set of data structures to ease the integration of disparate observational databases, with minimal information loss, to a standardized vocabulary and allows the gathering of information in the same way across different institutions. The OMOP-CDM is patient-centric, having tables for data commonly needed in clinical trials and observational studies such as drug use, procedures performed, etc. and only events that actually occurred are considered relevant,

as such situations like canceled appointments are not stored. It's optimized for identifying patient populations based on what healthcare interventions patients had and their outcomes enabling the characterization of these populations for multiple parameters like demographic information, disease history, healthcare delivery, cost, treatments and sequence of treatments [6]. This allows for an easier prediction of these occurrences in individual patients and helps estimating the effect of such interventions on a population.

The overall approach is to create an open network of observational data holders and require that the data holders translate their data into the OMOP-CDM, having every element in the participant database mapped to the approved CDM vocabulary and placed in the common data schema. It provides data organized in a way optimal for analysis while keeping the patient's personal information private, all data that might be used to identify them (e.g. names, precise birthdays) are limited. It's designed to record healthcare data from observational databases from different sources from all over the world while allowing each institution to keep and continue using their preferred vocabulary and format locally. This is achieved by creating a direct mapping between the original data and standardized vocabularies containing all necessary and appropriate corresponding standard healthcare concepts and, even though all codes are mapped to the standardized vocabularies, the model also stores the original source code to ensure no information is lost. It's database platform independent, meaning it can be used in any relational database (e.g. Oracle, SQL server), not requiring a specific technology, and it's optimized for data processing of data sources that vary in size, including databases with up to hundreds of millions of patients [7]. These features allow for a multi-center global analysis to be performed and enable the ability to pursue cross-institutional collaborations accelerating research due to easy access to de-identified data that is mapped onto these standard vocabularies.

The OMOP-CDM is maintained by Observational Health Data Sciences and Informatics (OHDSI), which is an international collaborative with the goal of creating and applying open source data analytic solutions to a large network of health databases in order to improve healthcare. The OHDSI team comprises academics, industry scientists, healthcare providers, and regulators whose mission is to improve medical decision making through the creation of reliable scientific evidence about disease history, healthcare delivery, and the effects of medical interventions by applying large-scale analyses to observational health databases [63]. The consortium also oversees the maintenance and the development of free analysis tools, such as ATLAS [4] and HADES [13], which are all available as open source.

## 2.6 Summary

This chapter aimed to present the current knowledge about the topic of text mining and NLP in the context of information retrieval of EHRs. There's a multitude of advantages in the usage of EHRs, such as more efficient billing, validating physician's prescriptions to prevent medical errors, checking on patients for risks of self harm and the one that will be the main focus of this dissertation, their use in observational studies.

Most of the previously mentioned advantages require data to be properly structured and an estimated 70-80% of all clinical data currently available are in free text documents, which poses an added adversity to the automation and effective functioning of these systems. Preventing medical professionals from writing clinical narratives is also not a viable option as it has been shown that these are vital to the solving of more complex cases.

One possible solution is to extract the data from unstructured free text using text mining and NLP, more specifically using annotators, which are programs specialized in the task of extracting concepts from text using text mining and NLP. In this chapter a select few annotators were presented, namely cTAKES, MetaMap, MedEx, Becas, Neji and MedTagger. These annotators serve the same general purpose of extracting relevant concepts from free text but are all very different, for instance, how they're executed (e.g. online based through a web browser or locally), how they're operated (e.g. command line or graphical interface), or even how customizable they are. It's up to the users to select the annotator that is more suitable to help them achieve their goals.

The data present in EHRs is an important contributor to healthcare related observational studies around the world. In order to take full advantage of the data in observational studies the data often needs to be compared and contrasted but observational databases don't all follow a standard when storing data. Healthcare data can vary greatly between organizations, with each following their own local rules. For instance, in one database the patient information could be stored in a table named "patient", while in another it could be "person" or there could be a field called "name", while in another the field is called "patient_name". As a result, one would need a specific query for each database just to retrieve the information. The other issue being that concepts can be represented in a variety of ways based on different conventions.

This disparity regarding how the data is stored emphasizes the need for a CDM for these observational databases. The purpose of a CDM is to standardize both the format and content of observational data. In this dissertation we will be focusing on one specific CDM, which is the OMOP-CDM. The approach from the creators of this CDM was to create an open network of observational data holders where each data holder is required to translate their data into the OMOP-CDM by creating a direct mapping between the original data and standardized vocabularies. This way every element in the participant databases is mapped to the approved CDM vocabulary and placed in the common data schema while allowing each institution to keep and continue using their preferred vocabulary and format locally. These features allow for a multi-center global analysis to be performed and enable the ability to pursue cross-institutional collaborations accelerating research due to easy access to de-identified data that is mapped onto these standard vocabularies.

# Chapter 3

# N2C2 Challenge

## 3.1 Introduction

As a means to incrementally build the final system to achieve the goals of this dissertation, it was decided to complete this assignment as an intermediary task. This particular task was selected as it provides clear objectives that align with part of the goals of the system being developed. It provides a data set, as well as the expected results from the processing of the data set. Another benefit is the fact that it was part of a challenge that's been solved already by multiple teams, from which the median accuracy of all participating teams was 77.33%. This provides additional perspective and guidance to understand what methods are more effective as well as setting an expected goal.

### 3.1.1 Description

The aim of the N2C2 2019 challenge track-3 [74] was to normalize medical entities to standard medical vocabularies. The challenge focuses on Named Entity Normalization (NEN) rather than Named Entity Recognition (NER). When discussing NER systems regarding clinical notes, it generally consists of identifying mentions of relevant clinical terms, while NEN involves linking named entities to concepts in standardized medical terminologies, thereby allowing for better generalization across contexts. In this challenge the relevant clinical mentions are already identified, and the task is to associate the mentioned text with its corresponding Concept Unique Identifier (CUI). The metric taken into consideration throughout this challenge to evaluate the performance of the developed system is the accuracy on the test dataset, as discussed previously the F1-score tends to be preferred, but this was the metric that the task organizers' decided to use and evaluate with.

### 3.1.2 Dataset

The task utilizes part of the i2b2 2010 data set [99], all records have been fully de-identified and manually annotated for concept information. The dataset provided contains a total of 100 annotated discharge summaries and is split evenly in training and test subsets. In both subsets it's provided the position on the clinical text files of every annotated entity, for the training subset it's also provided the CUI for the entities.

## 3.2 Implementation

This section will present and discuss the methods employed over the course of this challenge in order to achieve the final result.

### 3.2.1 Text pre-processing

In order to compare the text of the annotations and the results from looking up terms all text had to be normalized, so text pre-processing rules were applied to each annotation, as well as the results. The first step to normalize the text was making it all lower case, afterwards, contractions were expanded (i.e. "couldn't" is converted to "could not") and punctuation marks (i.e. commas, periods, quotation marks, question marks, etc) as well as symbols such as "#" were removed. HTML entities, which are pieces of text that begin with an ampersand ( & ) and end with a semicolon ( ; ), were converted to their meaning. These entities are used to display reserved characters, for example "<", which is represented as "<" (short for "less than"), which would otherwise be interpreted as HTML code [15].

Common word replacements were applied, this was done mostly for abbreviations, for example "b/l" was replaced with "bilateral". The source of the replacements was based on manual checking of the training dataset, common medical abbreviations and some chemical elements, for instance i.e. "o2" replaced with "oxygen". Additionally some words that are commonly used such as "last" and "former" were replaced with their gold standard equivalent "previous".

In the course of the pipeline both stemming and lemming algorithms are used but not simultaneously and are not applied in every case. What this means is that both texts being compared will be compared with all the normalization methods discussed in this section with the exception of lemmatization or stemming, and only then will they be lemmatized and compared, and then the original texts will be stemmed and compared. This comparison is only used in the Exact Matching method, that will be discussed later, the other methods only used lemming as it proved to have better results. Stop words, which are words that carry no meaningful data (e.g. "the", "a", "as"), were also removed. It should be noted that the order in which each method is applied is important as to avoid conflicts, some systems would render the following ones useless if applied before. For instance if the apostrophe punctuation mark was removed in an earlier step, the contraction expander wouldn't detect them.

Table 3.1: Methods of text pre-processing applied on examples of text snippets from the clinical notes.

| **Method** | **Example input** | **Example output** |
|---|---|---|
| Lower casing | The Patient | the patient |
| Remove symbols | prescription for Tylenol #3 | prescription for Tylenol 3 |
| | hospital day # 3 | hospital day 3 |
| Replace HMTL entities | 20 's rang | 20's rang |
| | SBP < 100 or HR < 55 | SBP <100 or HR <55 |
| Contractions expanded | isn't | is not |
| Word replacements | b/l | bilateral |
| | o2 | oxygen |
| | vit | vitamin |
| | former | previous |
| | last | previous |
| | u/s | ultrasound |
| Remove stop words | A leg | leg |
| | The patient | patient |
| | The patient was again seen | patient |

### 3.2.2 Word embedding

As a way to improve the comparison between the text of the annotations and the gold standard, even with the use of text normalization to make them as similar as possible, one would rather compare the actual meaning of the text than the text itself. Word embeddings are a type of representation, typically in the form of a real-valued vector, where words with similar meaning have similar representation, and as such words with similar meaning would be closer in the vector space. Word embedding is a machine learning technique, thus the meaning which will result in the value of the word vectors has to be learned. To achieve this the publicly available BioWordVec model [49] was employed. This model was created using the fastText library [42] and was generated from over 30 million documents from PubMed articles and clinical notes from the MIMIC-III database.

In order to compare the degree of similarity between vectors cosine similarity is used. Cosine similarity measures the similarity between two vectors of an inner product space. By measuring by the cosine of the angle between two vectors it's determined whether two vectors are pointing in roughly the same direction [57], similarly to the example shown in Figure 3.1. The closer in meaning the word vectors are, the smaller the angle will be and the similarity will approach the value of one.

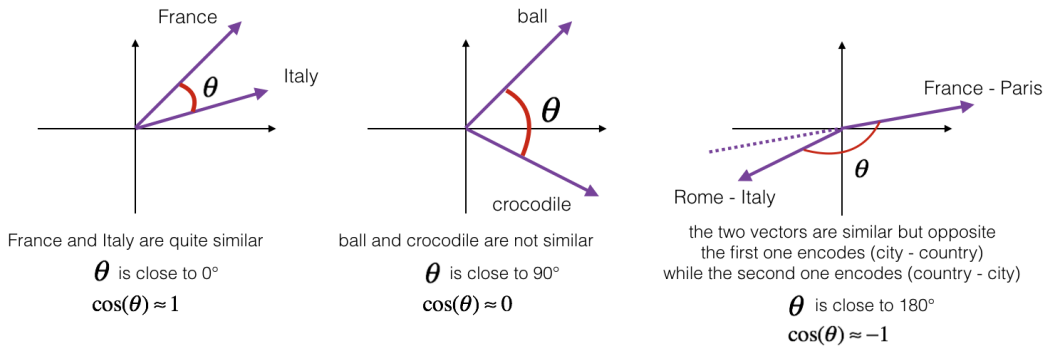


Figure 3.1: Example showcasing the graphical representation of the cosine similarity between different words [39].

### 3.2.3 Sieves

In earlier tests it was noted that the algorithms used didn't reach high enough accuracies individually. Due to this reason, and based on the work of João Rafael Almeida et al. [93] done for this same task, a sieve based approach was decided upon. Rather than trying to recognize every concept with a single method, in a sieve based system, each method is applied sequentially. Every method used is seen as a sieve or a layer, so annotations that don't get identified in a sieve, whether with a certain degree of certainty or they weren't identified at all, are passed to the next sieve. The deciding factor for the order of the sieves was based on the precision of each algorithm, with algorithms that are more strict and have higher precision being used in the first sieves, and the ones with lower precision on later ones. In simpler terms, initially it's important to be certain during identifications, as incorrect identifications would also render the next sieves useless as the annotation would already be marked as identified, and as the annotation progresses through the sieves it starts to allow identifications with lower degrees of certainty.

### 3.2.4 cTAKES

As an initial test, a state of the art Natural Language Processing (NLP) software, specifically cTAKES, was used keeping its default settings and default dictionary. In order to test its performance the previously discussed text normalization wasn't applied to the annotations beforehand and instead relied solely on its own capabilities. The default dictionary was the UMLS AB 2016 which uses Snomed and RxNorm as a source of vocabularies. This resulted in an accuracy of 22%. The dictionary was updated to its most up to date version, at the time of this project was the 2019 AB, and a custom install of the dictionary was made to include extra vocabulary sources, specifically NCI, ICD10 and Metathesaurus. These additional vocabulary sources were picked based on their usage in the training dataset and, with these changes, it achieved an accuracy of 40%.

### 3.2.5 Exact Matching

The first method developed was an exact matching algorithm that used the training dataset as a source of data. In information retrieval an exact matching method consists of using validated text - meaning pairs from the already validated training annotations as a

dictionary in order to assign a meaning (in this case the meaning is the associated CUI) to the text in the annotations being evaluated. As the name implies, it matches when the text is exactly the same, but even in such cases there can be dictionary collisions, meaning that in the creation of the dictionary there can be multiple CUIs associated with the same snippet of text. This is partially due to the nature of free text since clinical professionals don't always type the full medical terms, instead they might type the term in a more generic way and not fully specify it by typing the complete scientific name, as it would be said in a conversation. The other cause is the use of abbreviations, while some abbreviations are well established, others rely on context. Two different terms that start with the same letters could be abbreviated the same way.

In an attempt to mitigate some of the dictionary collisions several tests were done, as demonstrated in Table 3.2, using a combination of lemmatization, stemming and lower casing. The tests consisted of associating the resulting text of separate normalization methods to the CUI of the original annotation. For example in the table row where it's presented "Low+Lem", it means that both the results of lower casing and of the lemmatization are associated with the CUI of the original text. This method achieved an accuracy of 58% and a precision of 95%, which means that while it will only provide a match on roughly half the annotations, although when it does provide a match it's almost certain to be a correct prediction.

Table 3.2: Results of the combinations of normalization methods in the exact matching algorithm. In this table "Low" refers to the Lower casing algorithm, "Stem" refers to the stemming algorithm, "Lem" refers to the lemmatization algorithm, and "Base" refers to the usage of the original text with no normalization.

| **Normalization method** | **Accuracy** | **Precision** |
|---|---|---|
| Low+Stem+Lem | 57,91% | 95,52% |
| Stem+Lem | 57,04% | 95,23% |
| Low+Lem | 57,01% | 96,95% |
| Base+Lem | 57,18% | 96,70% |
| Low+Base+Lem | 57,26% | 96,68% |
| Low+Base+Stem+Lem | 57,99% | 95,60% |
| Low+Base+Stem+Lem | 56,95% | 96,15% |

### 3.2.6 UMLS lookup

The next approach involved looking up the text annotation on the database of UMLS Thesaurus concepts. The database had concepts from the following sources: Metathesaurus, Snomed and RxNorm and was built with the following, publicly available tool py-umls [83], which stands for "UMLS for Python". The results were low, as presented in Table 3.3, having a 20% accuracy and 32% precision, and although selecting only results that only had one possible match resulted in an increase of precision to 47%, the accuracy dropped to 3%. In either case, the results were too low to be considered. This happens mainly because, while the annotations are normalized to be as close to the gold standard as possible, the database being looked up isn't normalized.

Table 3.3: Results achieved from using the UMLS lookup method.

| **Picking method** | **Accuracy** | **Precision** |
|---|---|---|
| First result was picked | 20,7% | 32,2% |
| Only cases with 1 result | 3,2% | 47,1% |

In an attempt to improve this method the previously discussed word embedding was applied. The text was still looked up in the local database, but every result was then embedded and the resulting embeddings were compared with the embedding of the annotation being looked up using cosine similarity. The similarity was used for more than just selecting the most likely match in cases where the UMLS lookup provided multiple matches, it also served as a threshold for the results, meaning that results that didn't have an high enough similarity were discarded. After testing different values for the similarity threshold it was shown that, with a 0.99 similarity threshold it achieved an accuracy of 25% and precision of 78.7%. Using a lower threshold it achieved a very slightly higher accuracy of 26.4% however it also resulted in a substantially lower precision of 61.6%. Alternatively, on the opposite side of the spectrum with the highest possible similarity threshold it was possible to achieve an 81.2% precision yet the accuracy dropped drastically, as such 0.99 was picked as the threshold with most well rounded results, and for these reasons it was used as the second sieve.

Table 3.4: Results from using word embeddings and cosine similarity on the results of the UMLS Lookup method as a deciding factor.

| **Sim threshold** | **Accuracy** | **Precision** |
|---|---|---|
| 0,85 | 26,4% | 61,6% |
| 0,86 | 26,0% | 63,2% |
| 0,87 | 25,9% | 64,8% |
| 0,88 | 25,8% | 66,1% |
| 0,89 | 25,7% | 68,5% |
| 0,9 | 25,7% | 70,4% |
| 0,91 | 25,6% | 72,6% |
| 0,92 | 25,4% | 73,2% |
| 0,93 | 25,3% | 73,8% |
| 0,94 | 25,2% | 74,1% |
| 0,95 | 25,2% | 75,0% |
| 0,96 | 25,2% | 75,8% |
| 0,97 | 25,2% | 78,1% |
| 0,98 | 25,2% | 78,6% |
| 0,99 | 25,2% | 78,7% |
| 1 | 19,9% | 81,2% |

### 3.2.7 MetaMap

Following the first test that involved using a NLP software, namely cTAKES, as is with no pre-processing of the text, the next logical step was to use it combined with the text normalization and the previously discussed word embedding. For the third sieve, it was decided to use MetaMap instead of cTAKES, as it was shown by the work of Alejandro

Rodríguez-González et al. [90] that they have very similar performance. For this specific case, there was already a python module [89] to help the integration and usage of MetaMap through python, the programming language being used in the rest of the pipeline. MetaMap processed the annotations after they were normalized with the text pre-processing discussed earlier and it resulted in an accuracy of 48% and 48% precision. MetaMap has a built-in scoring system for its predictions called MetaMap Indexing (MMI), and, in an attempt to improve the results, it was used as a deciding factor for whether a result is kept or ignored. Multiple values for this threshold were tested, as can be seen in table 3.5, and it was decided that a value of 5.1 yielded the best results, with 46% accuracy and 62.6% precision.

Table 3.5: Analysis of the effect of Metamap's MMI score threshold on the accuracy and precision of the results.

| **MMI** | **Accuracy** | **Precision** |
|---|---|---|
| 3 | 48,12% | 48,1% |
| 3,2 | 48,12% | 48,1% |
| 3,4 | 48,12% | 48,1% |
| 3,6 | 48,12% | 49,0% |
| 3,8 | 47,07% | 61,8% |
| 4 | 46,25% | 62,7% |
| 4,2 | 45,98% | 62,6% |
| 4,4 | 45,98% | 62,6% |
| 4,6 | 45,98% | 62,6% |
| 4,8 | 45,98% | 62,6% |
| 5 | 45,98% | 62,6% |
| 5,1 | 45,98% | 62,6% |
| 5,18 | 45,98% | 62,6% |
| 5.18 | 0,04% | 20,00% |

After the first cull of results based on the MMI, a second pass is done using word embeddings and cosine similarity. This pass serves two purposes, deciding on results with equal MMI and removing results that aren't as reliable. As seen on table 3.6 multiple values were tested, and the one that was decided on was 0.84, less strict than the one used on the exact matching algorithm, resulting in a final accuracy of 42.8% and a precision of 73%.

Table 3.6: Analysis of the cosine similarity threshold on the precision and accuracy of the results.

| **Sim threshold** | **Precision** | **Accuracy** |
|---|---|---|
| 0 | 62,8% | 62,8% |
| 0,8 | 71,5% | 45,5% |
| 0,81 | 72,2% | 45,1% |
| 0,82 | 72,2% | 44,2% |
| 0,83 | 72,4% | 43,7% |
| 0,84 | 73,0% | 42,8% |
| 0,85 | 73,2% | 42,3% |
| 0,86 | 73,5% | 41,2% |
| 0,87 | 74,0% | 40,6% |
| 0,88 | 74,3% | 40,1% |
| 0,89 | 74,2% | 39,6% |
| 0,9 | 74,0% | 39,2% |
| 0,91 | 74,0% | 38,9% |
| 0,92 | 74,3% | 38,7% |
| 0,93 | 74,2% | 38,4% |
| 0,94 | 74,3% | 38,3% |
| 0,95 | 74,1% | 37,9% |
| 0,96 | 74,1% | 37,9% |
| 0,97 | 74,2% | 37,8% |
| 0,98 | 74,3% | 37,7% |
| 0,99 | 74,3% | 37,7% |
| 1 | 75,4% | 30,0% |

### 3.2.8 Final sieve and architecture

And lastly in the final sieve all annotations present in the training dataset were embedded and the cosine similarity was calculated. For this case, since it was the final sieve, there was no minimum similarity threshold. The pipeline, observable in Figure 3.2, using all sieves, managed to achieve a 77.5% accuracy through the task organizers' evaluation tool.

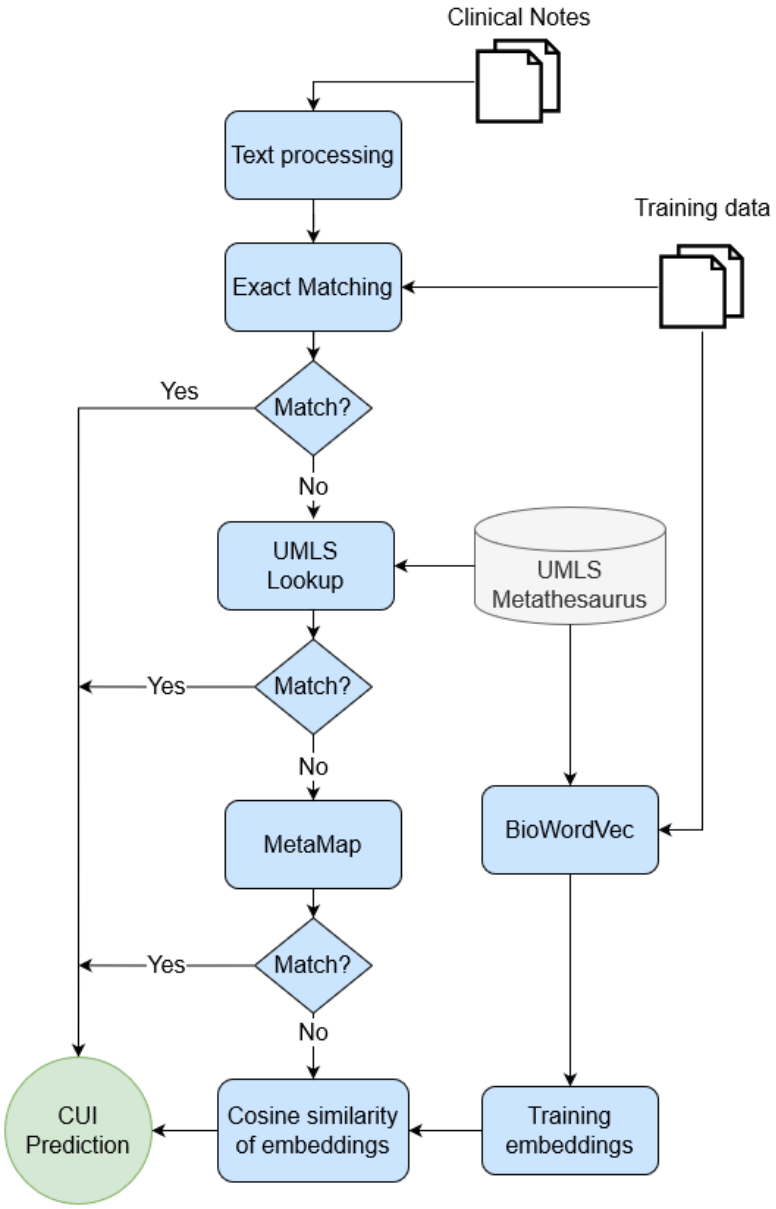


Figure 3.2: Final system architecture employed for the N2C2 challenge.

## 3.3 Summary

In this chapter, a NEN system was developed in the process of completing the N2C2 2019 challenge track 3. The challenge provided clear guidelines, a dataset and metrics. The task consisted of classifying snippets of text extracted from discharge summaries with the appropriate UMLS CUI.

The motivation behind the decision to complete this challenge was that it allowed the gradual development of the final system. Since the challenge was already completed by other

teams in the past, it was possible to analyze what methods performed better as well as compare the results of the system being developed with the results of other teams, which gave a better perspective of how it performed. The employed system used a sieves based approach, which means rather than attempting to classify every concept using a single method each method is applied sequentially. So annotations that don't get identified on a sieve are passed on to the next.

The first sieve applied an exact matching algorithm that uses the training data in an attempt to match the testing data.The second sieve consisted of looking up the text of the annotation in the UMLS dictionary to find a matching term. The third sieve employed the annotator MetaMap to attempt to classify the annotations with the appropriate CUI. The fourth and final sieve used BioWordVec to create word embeddings of the testing annotations and then used cosine similarity to compare them to the word embeddings of both the training data and terms from the UMLS dictionary. The word embeddings method was also used as a deciding factor in the two previous sieves (UMLS lookup and MetaMap) in cases where they returned multiple possible Concept Unique Identifier (CUI) for a single annotation. The system was able to achieve a 77.5% accuracy with the described methods, which is similar to the median of the results of all participating teams at 77.33%.

# Chapter 4

# System implementation

This chapter describes the implemented system for the automatic processing of clinical notes. The purpose of the system is to process discharge notes, extract relevant medical concepts in the text, classify the concepts with the appropriate CUI, normalize the concept into the OMOP-CDM format and finally store the normalized concepts in a database according to the OMOP schema.

The pipeline, observable in Figure 4.1, is essentially divided in three parts, the first part being the Named Entity Recognition (NER) system, the second part being the Named Entity Normalization (NEN) system, which was previously developed and tested for the N2C2 challenge in the previous chapter, and finally the third part being the Extract Transform Load (ETL) system that performs the normalization of the annotations into the OMOP-CDM, and subsequently stores the relevant data in a database.

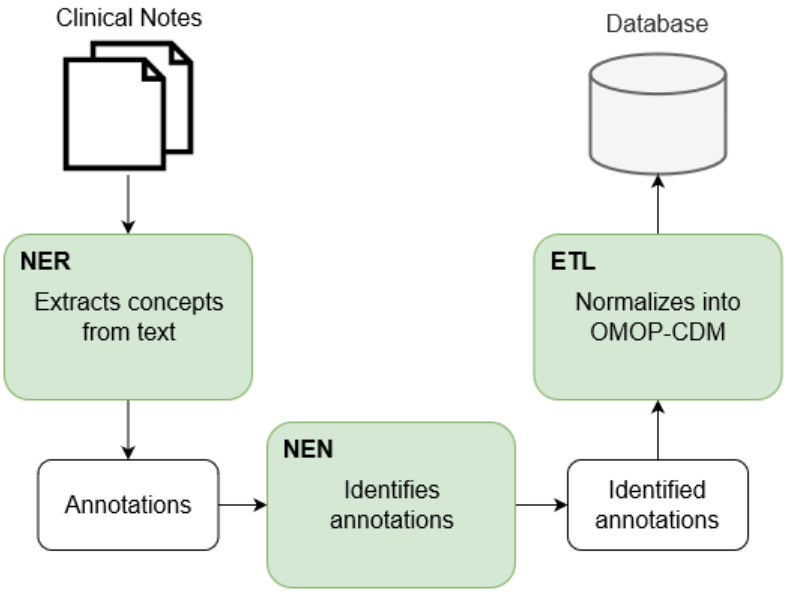


Figure 4.1: Simplified overview of the clinical notes processing system.

The employed NEN system, as mentioned previously, is the same one that was developed for the N2C2 challenge, which was already described and evaluated in the previous chapter,

and for that reason it won't be described in as much detail as the other two components of the pipeline over the course of this chapter.

## 4.1 Named Entity Recognition

The employed NER system, in essence, consists of an annotator, specifically cTAKES, and a set of components that directly and indirectly improve that annotator's performance. The system follows a sequential series of steps to achieve that, which can be seen in the overview of the pipeline in Figure 4.2, starting by pre-processing the text of the clinical notes before having the annotator process them. Following the task of identifying concepts in the text, the resulting annotations are then filtered based on a set of rules, as well as possibly merged with other annotations in cases where it's deemed adequate, specifically when there's some overlap between two or more annotations. Those are the methods that can be considered as indirect improvements to the annotator, as their purpose is fundamentally to make the task of annotating easier and also correct the resulting annotations.

The other measure taken, which directly improves the annotator, was creating a custom dictionary of concepts to be used by cTAKES. The remainder of this section will explain in more detail each of the described components.

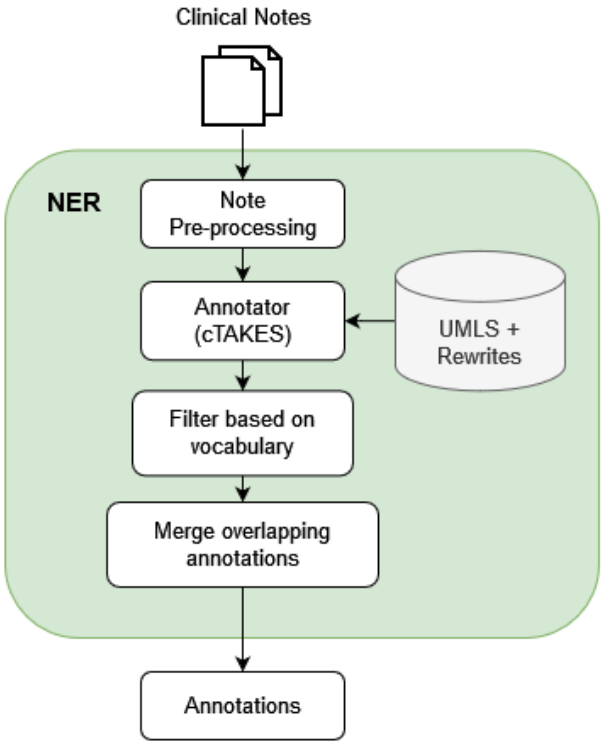


Figure 4.2: Overview of the architecture of the NER module.

### 4.1.1 Text pre-processing

In order to make the task of recognizing relevant concepts in text by the annotator more efficient, a text replacement algorithm was applied to the notes, based on the same text replacements used on the previously discussed work for the N2C2 challenge. HTML entities were replaced with their respective meaning and, as explained earlier, an HTML entity is a piece of text that begins with an ampersand ( & ) and ends with a semicolon ( ; ). They're used to display reserved characters, for example "<", which is represented as "<" (short for "less than"), which would otherwise be interpreted as HTML code. All percentages between 1% and 99% were exchanged with "partial" as this is the standard naming present in the UMLS vocabularies, making it easier for cTAKES to detect them. And lastly, text replacements were applied for common abbreviations (e.g. "o2" for "oxygen, "r hip" for "right hip", etc) as well as some forms of normalizations (e.g. "former" and "last" changed to "previous") and unlike in the previous task, stop words were not removed.

Table 4.1: Methods of text pre-processing used in the NER system.

| **Method** | **Text example** | **Output** |
|---|---|---|
| Replace HMTL entities | 20 apos;s rang | 20's rang |
| | SBP < 100 or HR < 55 | SBP <100 or HR <55 |
| Percentage normalization | revealed a 70% lesion | revealed a partial lesion |
| Text replacement | b/l | bilateral |
| | o2 | oxygen |
| | vit | vitamin |
| | former | previous |
| | last | previous |
| | u/s | ultrasound |
| | r hip | right hip |
| | l arm | left arm |

### 4.1.2 Annotator and dictionary rewrites

Unlike the previous task, cTAKES was chosen over MetaMap as the annotator. This choice was made due to its software package providing a tool to add and customize dictionaries for the software to use, which proved to be a crucial factor during the development of this system as it allowed the use of a customized version of the UMLS Metathesaurus dictionary.

A custom UMLS dictionary was created by applying the Casper tool, created by Kristina M. Hettne et al. [62]. This tool applies rewriting and suppression rules to the UMLS vocabularies, which in their tests resulted in an increase of 3.4% in the number of concepts recognized in the MEDLINE corpus.

The casper rewrite rules, as observable in Table 4.2, comprised of syntactic inversions for terms such as "Failure, Renal" which would be added as "Renal failure", possessive removal by removing the "'s" at the end of words in terms such as "Alzheimer's disease", splitting of short form/long form for terms that have the description followed by an acronym (e.g. in the terms "Selective Serotonin Reuptake Inhibitors (SSRIs)" it would add "Selective Serotonin Reuptake Inhibitors" and "SSRIs" separately), and removal of information about semantic types between parentheses within the term (e.g. "Surgical intervention (finding)"), removal

of text within angular brackets anywhere in a term (e.g. "Chondria <beetle>") which in UMLS are used to specify the meaning for words with multiple meanings, as well as other non-essential parentheticals.

Table 4.2: Examples of the Casper rewrite rules applied to terms of the UMLS dictionary.

| **Rewrite rule** | **Example text** | **Result** |
|---|---|---|
| Syntactic inversions | Failure, Renal | Renal failure |
| Possessive | Alzheimer's disease | Alzheimer disease |
| Short form/long form | Selective Serotonin Reuptake Inhibitors (SSRIs) | Selective Serotonin Reuptake Inhibitors |
| | | SSRIs |
| Semantic types | Surgical intervention (finding) | Surgical intervention |
| Angular brackets | Chondria beetle | Chondria |

Additionally to the rewrite rules, there are also suppression rules, which can be seen in Table 4.3, in these cases the terms are not edited and/or split into multiple terms like in the rewrite rules, instead they're completely removed. Among them, some of the more relevant suppressions were removal of dosages in terms that include a percentage (e.g. "Oxygen 2%"), keeping only the more general version of the term and removal of concept classifications such as "not elsewhere classified", "unclassified", "without mention", which are present at the end of some terms' text in the form of acronyms (e.g. "NEC"). There were also some miscellaneous removals such as terms containing "other" at the beginning, "deprecated", "unknown" or "obsolete".

Table 4.3: Examples of the Casper suppression rules applied to terms of the UMLS dictionary.

| **Suppression rule** | **Example text** |
|---|---|
| Dosages | Oxygen 2% |
| At-sign | ADHESIVE @@ BANDAGE |
| Enzyme classification numbers | EC 2.7.1.112 |
| Any classification | Ventriculoscopy NEC |
| | Abnormality of white blood cells, not elsewhere classified |
| Any underspecification | Hemophilia, NOS |
| Miscellaneous | Robitussin DM (obsolete) |

Using the discussed work of Kristina M. Hettne et al. on the Casper software as a baseline, some additional rewrite rules were implemented, which can be observed in Table 4.4. These changes were mainly based on the text observed in the clinical notes and created made in an attempt to better reflect the more natural way that medical professionals describe and discuss these concepts. It's important to note, that while they're technically referred to as rewrites, they're essentially alternative options, as these rewrites are added to the dictionary but the old rules are also kept.

The first and most straightforward rule was the simplification of the spinal discs, by removing the word "disc" from the terms (e.g. "C4/5 disc" changed to "C4/5"). Specifications of terms through the use of the word "while" or a comma were also simplified (e.g. "Accident while engaged in sports activity" changed to "Accident", "Root amputation, per root" changed to "Root amputation"). Alternatively some specifications are done differently through the colon punctuation mark, for example "Medication administration: inhalation",

in these cases everything before the colon is removed and the term would become just "inhalation". Terms that are presented as multiple alternatives with a slash, for example "droperidol / fentanyl", were added back as separate terms "Droperidol" and "Fentanyl". Terms that contained the words "in" or "of", as well as "in the" or "of the", were processed in order to create all the possible variations used in regular natural speech used by medical professionals, for example the term "muscle cramps in the calf" results in "calf muscle cramps", and "MRI of the spine" results in "MRI of spine" and "Spine MRI".

Finally, a more general clean up was applied where slashed numbers to represent partials were removed (e.g. "Anterior 2/3 of tongue" changed to "Anterior of tongue") as well as other term specifications, done through the usage of dashes, parenthesis and/or numbers, that aren't normally described in normal speech (e.g. "Amylo-(1,4,6)-transglycosylase" changed to "Amylo transglycosylase"). These rules were applied in cycles with the purpose that a term might be eligible to a rule only after being processed and changed by a different one in the previous cycle.

Table 4.4: Examples of the developed rewrite rules applied to terms of the UMLS dictionary.

| **Rewrite rule** | **Example text** | **Result** |
|---|---|---|
| Simplify "discs" | C4/5 disc | C4/5 |
| "while" specifications | Accident while engaged in sports activity | Accident |
| Comma specifications | Root amputation, per root | Root amputation |
| Colon specifications | Medication administration: inhalation | inhalation |
| Slashed terms | droperidol / fentanyl | Droperidol |
| | | Fentanyl |
| Dash definition | GBL - gamma-butyrolactone | GBL |
| | | Gamma-butyrolactone |
| Variations of "of" | MRI of the spine | MRI of spine |
| | | Spine MRI |
| Variations of "in" | Muscle cramps in the calf | Calf muscle cramps |
| General clean up | Anterior 2/3 of tongue | Anterior of tongue |
| | Amylo-(1,4,6)-transglycosylase | Amylo transglycosylase |

### 4.1.3 Filtering annotations

During the development of the system it was decided to keep some additional metadata on the annotations. This was done with the aim of providing additional data to analyze and better understand some of the common factors among annotations that were correctly selected as well as the ones that were incorrectly selected. Out of the metadata fields stored, the ones that proved to have an impact were the vocabulary source (e.g. Snomed, RxNorm, ICD, etc.) used to generate that annotation and the semantic type (e.g. Laboratory Procedure, Food, Enzyme, etc.). By analyzing these metadata fields it was possible to pinpoint which vocabulary sources and semantic types performed the worst individually, as well as combinations of them.

The main concern with this method was that it could create a situation of overfitting the solution to this particular dataset, making the solution too specific for this dataset rather than a general solution that can be applied to other clinical notes in the future. To prevent

this from happening some measures were taken: the analysis was first done through 5-fold cross validation on the training dataset and only afterwards, using the data from the training dataset and applying it to the test dataset. The results from it were compared to the ones resulting from the previously done 5-fold cross validation done on the training dataset to confirm that it performed similarly.

After analyzing the performance results of each vocabulary on the training dataset it was decided to set a threshold for the minimum precision allowed, which means that all annotations with a vocabulary source with a precision below this threshold value and a set minimum occurrence amount were discarded. The reason for the minimum occurrence requirement was that it's not possible to infer that the source performs poorly overall for this sort of clinical texts with such limited data. The goal being to only filter particularly poor performing ones, as those are more likely to perform equally on a different clinical note.

### 4.1.4 Merging of annotations

Finally, in the last stage of the NER system, measures were taken to mitigate one fault in the cTAKES system. While analyzing the resulting annotations, it was noticed that the annotator was rather inaccurate at recognizing compound concepts, something that was also documented in the work of Monica Agrawal et al. [32]. A compound concept is a concept formed by simpler/smaller concepts, for example "left leg pain" is a compound concept formed by the concepts "left", "leg" and "pain". While in these cases cTAKES did identify the individual concepts accurately and even some less complex compound concepts like "left leg" and "leg pain", it failed to identify the compound concept as a whole. In order to fix this, annotations that had overlapping words, for example, using the previous given example of "left leg pain", the word "leg" is in both the "left leg" and "leg pain" annotations, and as such it would be merged into a single annotation.

## 4.2 Extract Transform Load

The final stage of the employed system is the Extract, Transform, Load (ETL), which is the name given to a process that consists of data extraction from one or more data-sources, its transformation and cleansing in order to make it optimized for reporting and analysis and, finally, loading it into a data storage or data warehouse. A crucial requirement of ETLs is that the process must be repeatable, so that it can be rerun whenever the source data is updated. The goal of the ETL process in this context is to convert the concepts extracted from the discharge summaries, standardize them to the OMOP-CDM and use them to populate a database following the same common data model (CDM) schema. A simplified overview of this process can be seen in Figure 4.3. Although, typically the process involves extracting from multiple data sources, in this case, the data is from a single source: the output of the NEN system.

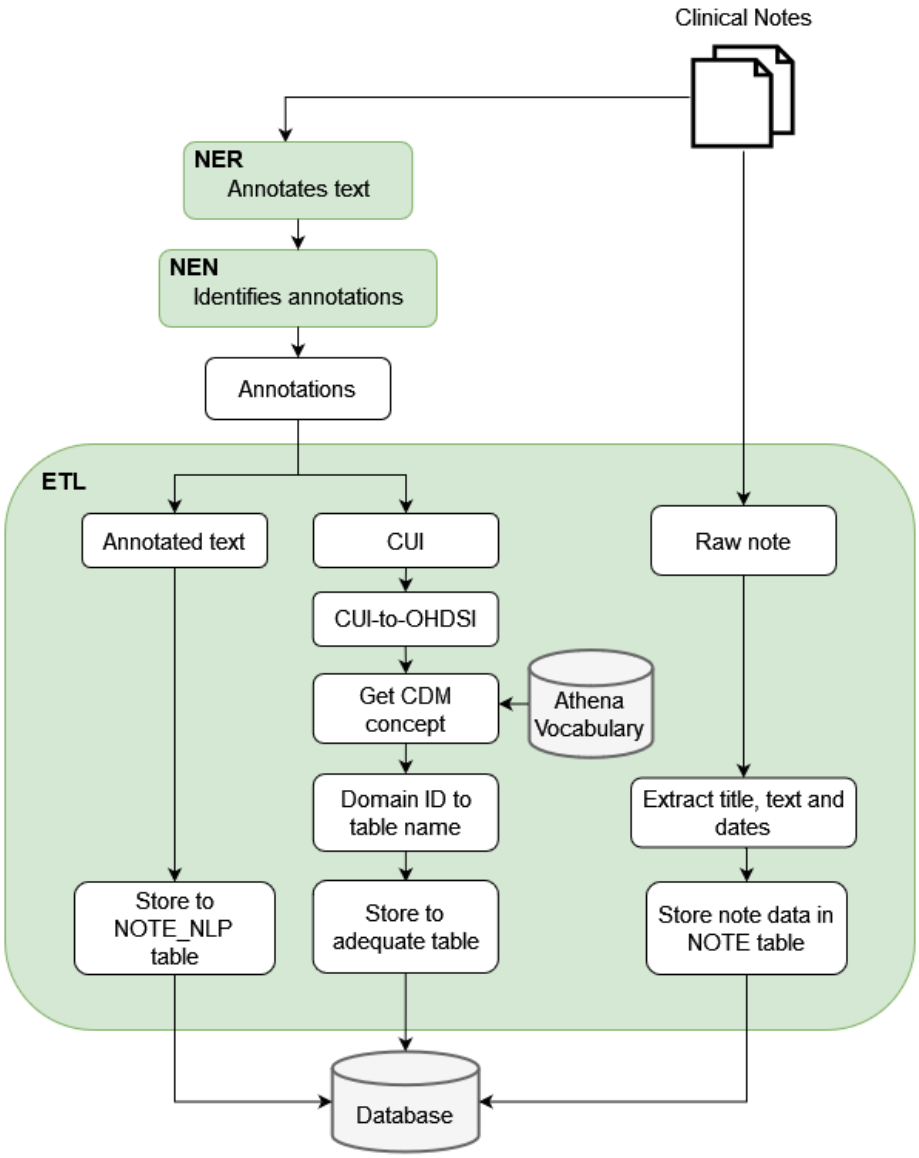


Figure 4.3: Simplified overview of the employed ETL system.

### 4.2.1 Converting CUI to OMOP-CDM concepts

The first step of this conversion is finding the corresponding standardized CDM concepts for the concepts extracted from the discharge summaries. The concepts extracted in the discharge summaries were classified using UMLS CUI and, in order to achieve this conversion from the CUI to the standardized Common Data Model (CDM) concepts, two publicly available resources were used, the OHDSI-to-CUI mapping package created by Juan M. Banda [23][16] and the OHDSI vocabulary obtained from Athena [3].

The OHDSI-to-CUI mapping package is essentially a list of OHDSI concept ID and CUI pairs, which works as a dictionary, so by having one element of the pair it's possible to obtain the other. Athena is an OHDSI vocabularies repository which allows its users to query

the vocabulary or, alternatively, download this vocabulary, which contains the concepts used in the CDM, allowing the queries to be done locally rather than online. Thus, using the OHDSI-to-CUI mapping, it was possible to associate the CUIs extracted from the discharge summaries to standardized CDM concept IDs and consequently using these concept IDs to find the corresponding concept in the vocabulary downloaded from Athena. Additionally, in the process of developing this ETL, Athena's query feature allowed for some form of validation by manually comparing the resulting mappings.

### 4.2.2 Storing the converted concept

After finding the matching CDM concept for each extracted concept the next step was to populate the database tables with this now standardized data. Having now the standardized OHDSI concepts associated with the concepts annotated from the clinical text, the next logical step is storing them in a database that follows the OHDSI schema. The CDM version used was the 5.4, which has 39 database tables, of which, in this process, 9 will be populated (person, note, note_nlp, observation, condition_occurrence, drug_exposure, procedure_occurrence, device_exposure, measurement, specimen).

Each OHDSI concept is associated with a set of information fields related to it, such as its domain, vocabulary source, whether it's standard or not, as observable in Table 4.5. The first step towards storing a standardized concept is knowing the adequate database table to store it in. In order to achieve this, a mapping was done between all the possible concept domain ID values and table names. This mapping was created using a combination of OMOP-CDM documentations and the OHDSI online forums as the aspect of community behind this CDM is an important factor.

Table 4.5: Example of a concept belonging to the Condition domain in the OMOP-CDM format.

| **Field** | **Value** |
|---|---|
| concept_id | 4104316 |
| concept_name | Multiple lesions |
| domain_id | Condition |
| vocabulary_id | SNOMED |
| concept_class_id | Clinical Finding |
| standard_concept | S |
| concept_code | 300582001 |
| valid_start_date | 2002-01-31 |
| valid_end_date | 2099-12-31 |
| invalid_reason | NAN |

The final step is to populate the table fields. In essence, the data stored in these tables is the OHDSI concept code, which will indirectly provide the previously mentioned associated data related to that particular concept, the provenance of the data, and contextual data specific to the concept type. The fields in these tables follow specific rules in this model. Some fields are mandatory, as shown in the example presented in Table 4.6, which roughly translates to meaning that they can't be left empty, designated in most database languages as "NOT NULL". In cases where the value for a mandatory field isn't present in the original

data it should be filled with a single zero, which in this CDM represents that the value isn't present in the source data. These mandatory fields were the priority in being populated. Additionally, some of the non mandatory fields were also populated, specifically the fields pertaining to the source values. In this context the source refers to the identifier used to classify the concept prior to the CDM standardization, which in this case was the UMLS CUI.

Table 4.6: CONDITION_OCCURRENCE database table in the OMOP-CDM schema.

| **Field name** | **Type** | **Mandatory** |
|---|---|---|
| condition_occurrence_id | Integer | x |
| person_id | Integer | x |
| condition_concept_id | Integer | x |
| condition_start_date | Date | x |
| condition_start_datetime | Timestamp | |
| condition_end_date | Date | |
| condition_end_datetime | Timestamp | |
| condition_type_concept_id | Integer | x |
| condition_status_concept_id | Integer | |
| stop_reason | Text | |
| provider_id | Integer | |
| visit_occurrence_id | Integer | |
| visit_detail_id | Integer | |
| condition_source_value | Text | |
| condition_source_concept_id | Integer | |
| condition_status_source_value | Text | |

The process of storing data is done in a specific sequence in order to follow the specifications of the OHDSI schema. The reason for this is that each annotation must be linked to a medical note and each note must be linked to a person. As such, the first tables being populated are the "person", "note" and "note_nlp" tables. Every entry in the note table refers to a single clinical note that was processed and it contains mostly metadata about the note, such as the note's file name, the date of when it was analyzed, what kind of note it is (which in this case is "Discharge Summaries") and the patient to whom it belongs.

The person table was only populated on a superficial level for the basic functioning of the database. Every note, as well as each concept, must refer to a person. However the focus of the system being developed is the extraction of medical concepts, and some fields of the person table require a more specialized data extraction in order to extract personal information relating to the patient's sex and age. Some of these have to be indirectly deduced, and for this reason, while table entries for the person table were created, they were only used to associate a unique person identifier to notes and concepts, so the remaining mandatory fields were filled with zeros which is the CDM's standard for "not present in the source".

The note_nlp table contains every term extracted from the clinical notes, but unlike the other more specialized tables, this table contains the term in its raw, unprocessed state, including how it was originally typed in the note and its exact position on the clinical note's text where it was identified. Essentially, every extracted term is added both in the note_nlp table in its raw state and in its specific domain table after being converted to its standardized

form.

The table fields adhere to a naming system that helps the developers better understand their contents. For instance every table has a PREFIX_id field, where PREFIX represents the name of the database table. This rule is showcased in the example provided in Table 4.6, which demonstrates field condition_occurrence_id in the condition_occurrence database table. These PREFIX_id fields store unique identifiers for each table entry. Every table has a person_id field , which is a unique id identifying the person in the database to whom the discharge summary relates to. Some tables have a "type" field, for instance, there is the a condition_type_concept_id in the example presented in Table 4.6 This "type" field is used to determine the provenance of the record (e.g. the concept was obtained from an EHR, insurance claim, etc) [24], which, in this case, a value of 32817 was used for that field, that means "EHR" in this model.

### 4.2.3 Creating, connecting and inserting data to the database

Although the act of storing the normalized data into the database has been discussed, the actual database hasn't been described yet. The database management system of choice was the PostgreSQL [95], which is a reliable, relational Open Source system. The creation of the database itself is a rather straightforward process using the publicly available Data Definition Language scripts from the OHDSI github page [10] one can automate the creation of the database, as well as setting it up in accordance with the OHDSI schema. Lastly, in order to connect, read and insert data to the database, the Psycopg2 [26] python module was used, which is a popular PostgreSQL database adapter.

## 4.3 Summary

The final implemented system comprises three components: NER, NEN and ETL, with each component outputting data to the next one's input. The NEN component being the one already tested and evaluated in chapter 3 so, for that reason, this chapter only focused on the NER and ETL components.

The approach while developing the NER system was to use an annotator, namely cTAKES, and a set of methods to improve its performance. The first method aimed to make the task of annotating easier by preprocessing the text in the notes. The main aspect of the text preprocessing was the translation of medical abbreviations into their actual meaning, since cTAKES was not recognizing the majority of the abbreviations used in the notes. The second method employed consisted in enhancing the dictionary used by cTAKES by inserting additional entries to it. The third and fourth methods targeted cTAKES' output by first merging overlapping annotations and afterwards filtering out annotations that originated from vocabulary sources that had lower performance on the training data.

The final component of the system is the ETL, whose goal is to convert the now identified annotations into standardized OMOP-CDM concepts. This conversion process is achieved through the usage of the OHDSI-to-CUI mapping package created by Juan M. Banda which makes it possible to translate the CUI used to identify the annotations into OMOP-CDM concepts. After the conversion process is completed the concepts are stored in a PostgreSQL database following the OMOP-CDM schema.

# Chapter 5

# Results and validation

This chapter will present and discuss the results obtained from the evaluation of the implemented system in the previous chapter. Specifically, it will go over the results of the Named Entity Recognition (NER) and Extract Transform Load (ETL) components of the system. The Named Entity Normalization (NEN) component won't be discussed in this chapter since it was already evaluated and discussed in chapter 3, in the context of the N2C2 challenge, which resulted in a 77.5% accuracy.

The two components will be evaluated and discussed separately due to the fact that the results expected from each of them as well as the metrics used to measure their performance being so disparate. The other reason being that the system is modular, as such separate evaluation allows to pinpoint the weak points of the system and focus the development on improving that specific component in the future.

## 5.1 Named Entity Recognition (NER) results

As pointed out through the course of this dissertation, the focus of a NER system is to extract relevant concepts in free text and, as such, the evaluation methods and metrics chosen must reflect that.

### 5.1.1 Dataset and evaluation method

The evaluation and testing was performed using the same data set as previously described in the N2C2 challenge chapter, the i2b2 2010 data set, which as explained earlier, contains a total of 100 annotated discharge summaries and is split evenly in training and test subsets. In both subsets it's provided the position on the text file of every relevant entity, making the data set suitable to evaluate the NER system on, as there is an expected result to compare it to.

In order to measure the performance of the system the annotations provided in the challenge were compared to the ones resulting from the use of the employed annotator, cTAKES. This was done by comparing the spans of the annotations which, in this case, refers to their start and end positions and, consequently, their length in the text of the clinical note. There were also some added considerations and modifications, as the focus was to evaluate cTAKES annotating capabilities and not simply how similar the resulting annotations are to the ones in the challenge. Some of the annotations provided in the challenge were stripped of words

that add no value to the medical concept, for instance, the word "the" in cases where the annotation is "the patient".

### 5.1.2 Discussion of the results from each step taken to improve the NER system

The initial test was performed by using only cTAKES in its default state to annotate the clinical notes, which resulted in a recall, precision and F-score of 60.4%, 53.25% and 56.44% respectively. The approach taken while developing this system was to first prioritize achieving a high recall (i.e. high amount matching annotations), with precision being the second priority (i.e. reducing false positive annotations). The logic behind this approach was that by achieving a high annotation count first, it would be possible to afterwards analyze the metadata in the annotations to filter out annotations that are more likely to be false positives, thus increasing the precision.

The preprocessing of the text in the notes is the first stage of the NER component, although it wasn't originally expected to improve the results, due to cTAKES already having a text preprocessing step in its pipeline, it surprisingly increased both the recall and precision, by 4.98 and 3.42 percentage points respectively. After further analysis this was determined to be mainly due to the step where abbreviations are replaced by their meaning in the text.

The filtering based on vocabulary sources improved the precision by 7.18 percentage points but it also reduced the recall of the system by 2.49 percentage points. The improvement to the precision was much greater than the loss of recall, which resulted in an overall improvement, increasing the F-score by 2.62 percentage points.

The dictionary additions is the only method that directly enhances the annotator being used, unlike the other methods which indirectly enhance its functionality. The additions were initially created using Casper [62], and afterwards some additional ones were created. These resulted in an increase of the recall and precision by 6.78 and 3.55 percentage points respectively.

Finally, the last step and the one that proved to be the most impactful, the merging of overlapping annotations. This step aimed at mitigating one negative aspect of cTAKES, which is its inaccuracy at recognizing compound concepts, as also pointed out in the work of Monica Agrawal et al. [32]. The merging resulted in a considerable improvement, increasing recall and precision by 8.81 and 2.53 percentage points respectively.

After further analysis of the vocabulary filtering method it was observed that, within a certain range of threshold values, which will be explained next, for the filter strictness, it's possible to sacrifice recall for precision while keeping the F-score mostly unchanged, which can be observed in Figure 5.1. This wasn't possible when the vocabulary filtering method was initially introduced, however, it was possible after the implementation of the dictionary rewrite and merging of overlapping annotation methods, which vastly increased the amount of annotations, making the filtering increasingly effective. Before that increase in the amount of annotations, making the filter more strict resulted in a severe drop of the F-score. With this new development it was decided to go for a more precise system at the cost of some of the recall.

The actual values of the thresholds were initially defined as 0.3, 0.45 and 0.6, which were based on the recall distribution achieved using each vocabulary source individually. These values were defined in initial tests done to the NER system and, for that reason, don't represent the results currently obtained. However, in order to not create overfitting of the

solution to this particular data set, the scoring of each source, for the context of this filter, were preserved. A mapping was done between the vocabulary sources and the recall result obtained in those earlier tests. So, for clarity, those threshold values are now simply referred to as low, medium and high strictness.

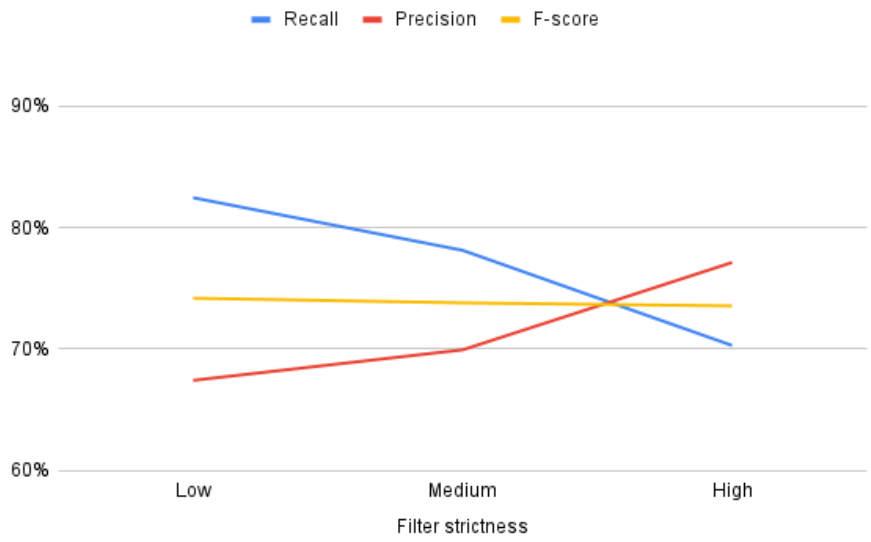


Figure 5.1: Graph demonstrating the effect of the vocabulary source filter's strictness levels on the recall, precision and F-score of the NER system.

### 5.1.3 Final results

As seen in Table 5.1, the system went from an F-score of 56.44% to 73.55% with the employed methods, having significant improvements in both recall and precision. Although each method improved the performance of the system, the methods that proved to have a bigger impact were the dictionary additions (both from Casper and the ones included in this work), and the merging of overlapping annotations.

Table 5.1: Evolution of the recall, precision and F-score values resulting from the implementation of each method to the NER system. The methods shown are cumulative, with each row including the methods described from the previous rows.

| **Method** | **Recall** | **Precision** | **F-score** |
|---|---|---|---|
| Basic cTAKES analysis | 60,04% | 53,25% | 56,44% |
| Text preprocessing | 65,02% | 56,67% | 60,56% |
| Filtering vocab. sources | 62,53% | 63,85% | 63,18% |
| Dictionary additions | 69,31% | 67,40% | 68,34% |
| Merge of annotations | 78,12% | 69,93% | 73,80% |
| Filter adjustment | 70,29% | 77,12% | 73,55% |

## 5.2 Extract Transform Load (ETL) results

The evaluation of the ETL component used a vastly different approach from the NER component. The conversion between the UMLS Concept Unique Identifier (CUI) and its corresponding OMOP-CDM concept ID was done using a pre-evaluated and validated mapping [23][16]. However, the mapping is not completely infallible and, after analysis it was found that a total of 7.92% of the annotations didn't have a corresponding concept ID, so, for that reason were discarded. It was also found that incorrect annotations were far more likely to not have a corresponding CDM concept ID than the correct annotations. Specifically, during the mapping process, 4.62% of the correct annotations were lost and 19.02% of the incorrect annotations were lost. This translated to a loss of recall by 3.25 percentage points but an increase of precision of 2.76 percentage points. For that reason the loss of F-score was minimal, at 0.65 percentage points, resulting in a final F-score of 72.90

The conversion itself after finding the corresponding CDM concept ID is a rather direct process. Once it's defined which database table each concept domain ID redirects to, the process becomes rather linear with no room for deviation. The validation of this process was done in two stages. The first stage consisted of manually verifying each possible domain and using the available OHDSI documentation along with some guidance from the OHDSI forums to make sure that the data being stored was being correctly selected. In the second and final stage, some extra precautionary verifications were done, such as selecting ten random concepts, still in their original form, of each domain and looking them up on Athena [3] to see if it matches their OMOP-CDM form, as well as querying the database to guarantee that the data was being stored properly.

## 5.3 Summary

This chapter provided an analysis regarding the results achieved with the NER and ETL components of the implemented system. The remaining component, NEN, wasn't discussed in this chapter since it was already tested and evaluated in chapter 3, where it was shown to achieve a 77.5% accuracy. The evaluation and testing was performed using the same data set as previously described in the N2C2 challenge chapter, the i2b2 2010 data set.

The NER component was evaluated based on its ability to recognize relevant concepts in the notes and its performance was measured using precision, recall and F-score. Employing a combination of methods to improve it, namely text preprocessing, cTAKES' dictionary improvements, vocabulary source filtering and merging of overlapping annotations, the system achieved a recall, precision and F-score of 70.29%, 77.12% and 73.55% respectively.

The evaluation of the ETL component used a vastly different approach from the NER component. The conversion between the UMLS CUI and its corresponding OMOP-CDM concept ID was done using a pre-evaluated and validated mapping. However, the mapping is not completely infallible and it was found that a total of 7.92% of the annotations didn't have a corresponding concept ID so, for that reason, they were discarded. That loss of annotations was mostly mitigated by the fact that incorrect annotations were far more likely to not have a corresponding concept ID than correct annotations. Lastly, some manual verifications were made to assure that the data was being stored in the proper database tables and fields.

# Chapter 6

# Conclusion and future work

This chapter will present the conclusions of this dissertation, including a general analysis of the outcomes of the work carried out as well as potential future work to improve the developed work.

## 6.1 Conclusion

The goal of this dissertation was to develop a system capable of automatically processing clinical narratives by extracting relevant medical concepts from them and converting those same concepts into OMOP-CDM standardized concepts.

In order to better understand the subject and how to approach it, text mining and Natural Language Processing topics were studied and discussed, including frameworks that implement such technologies. Afterwards, in the development stage, in an attempt to approach this task in a more incremental way, the N2C2 2019 challenge was completed and described in chapter 3. This challenge served as an intermediary task, easing the development and testing of part of the system due to the fact it provided clear guidelines, metrics and a dataset to evaluate the performance of the developed work. Being a modular system, the component developed for this challenge was then implemented in the final system seamlessly.

The remaining system was described in chapter 4, where its conception, features, decisions taken to improve it, as well as its architecture were discussed. And finally, in chapter 5, the performance of the system was analyzed, by measuring the impact of each step taken to improve the system. Overall, the developed system fulfilled its objective of processing clinical texts, extracting concepts and converting the extracted texts to OMOP-CDM standardized concepts, as well as storing it in a database with a OMOP-CDM compliant schema, facilitating the usage of such data for observational studies.

## 6.2 Future work

Although the implemented solution fulfills the specified requirements, there are still some aspects of it that can be improved upon. Namely, the NER system could potentially be improved by checking the neighboring annotations of each annotation for potential merges. While the system currently merges overlapping annotations, there are still cases of annotations that should be merged but don't overlap. The extraction methods currently employed are focused on the extraction of medical concepts, and, for this common data model, which is

patient centric, some improvements should be made to derive better patient details from the free text.

The other improvements being towards the system's current lack of graphical interface. The system is currently operated through a set of scripts to run, either the full pipeline, or specific components. It would be an interesting proposal to make the system more autonomous and user friendly, possibly making it possible to use it remotely through a browser.

# Bibliography


[1] All you need to know about text preprocessing for nlp and machine learning. Online; accessed 25-July-2021. URL: https://www.freecodecamp.org/news/all-you-need-to-know-about-text-preprocessing-for-nlp-and-machine-learning-bc1c5765ff67.

[2] Apache uima. Online; accessed 20-June-2022. URL: https://uima.apache.org/.

[3] Athena – ohdsi vocabularies repository. Online; accessed 30-July-2022. URL: https://athena.ohdsi.org/search-terms/start.

[4] Atlas – a unified interface for the ohdsi tools. Online; accessed 20-October-2021. URL: https://www.ohdsi.org/atlas-a-unified-interface-for-the-ohdsi-tools/.

[5] becas. Online; accessed 12-May-2022. URL: http://bioinformatics.ua.pt/becas/.

[6] The book of ohdsi - common data model. Online; accessed 20-October-2021. URL: https://ohdsi.github.io/TheBookOfOhdsi/CommonDataModel.html.

[7] The book of ohdsi - design principles. Online; accessed 20-October-2021. URL: https://ohdsi.github.io/TheBookOfOhdsi/CommonDataModel.html#design-principles.

[8] Ctakes. Online; accessed 12-May-2020. URL: http://ctakes.apache.org/.

[9] Data standardization. Online; accessed 10-October-2022. URL: https://www.ohdsi.org/data-standardization/.

[10] Definition and ddls for the omop common data model (cdm). Online; accessed 25-August-2022. URL: https://github.com/OHDSI/CommonDataModel.

[11] Egas. Online; accessed 16-May-2022. URL: https://demo.bmd-software.com/egas/.

[12] Extracting information from text. Online; accessed 11-May-2020. URL: https://www.nltk.org/book/ch07.html.

[13] Hades - health analytics data-to-evidence suite. Online; accessed 20-October-2021. URL: https://ohdsi.github.io/Hades/.

[14] How to get started with nlp – 6 unique methods to perform tokenization. Online; accessed 25-July-2021. URL: https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/.

[15] Html entity. Online; accessed 30-July-2022. URL: https://developer.mozilla.org/en-US/docs/Glossary/Entity.

[16] Mapping from umls cui into the ohdsi vocabulary. Online; accessed 30-July-2022. URL: https://www.ohdsi.org/web/wiki/lib/exe/fetch.php?media=projects:workgroups:ohdsi_nlp_workgroup_umlstoohdsivocab.pdf.

[17] Medtagger. Online; accessed 12-May-2022. URL: https://github.com/OHNLP/MedTagger.

[18] Metamap. Online; accessed 12-May-2020. URL: https://metamap.nlm.nih.gov/.

[19] Neji. Online; accessed 16-May-2022. URL: https://github.com/BMDSoftware/neji.

[20] Nlp — how tokenizing text, sentence, words works. Online; accessed 25-July-2021. URL: https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/.

[21] Nlp – expand contractions in text processing. Online; accessed 25-August-2022. URL: https://www.geeksforgeeks.org/nlp-expand-contractions-in-text-processing/.

[22] Observational medical outcomes partnership (omop). Online; accessed 20-October-2021. URL: https://chime.ucsf.edu/observational-medical-outcomes-partnership-omop.

[23] Ohdsi concept_id to umls cui mapping package and mappings. Online; accessed 30-July-2022. URL: https://github.com/jmbanda/OHDSIconceptid2cui.

[24] Omop cdm v5.4 specifications. Online; accessed 30-July-2022. URL: https://ohdsi.github.io/CommonDataModel/cdm54.html.

[25] Omop common data model. Online; accessed 20-October-2021. URL: https://www.ohdsi.org/data-standardization/the-common-data-model/.

[26] Psycopg. Online; accessed 25-August-2022. URL: https://www.psycopg.org/.

[27] Truven marketscan datasets in cdm. Online; accessed 10-October-2022. URL: https://forums.ohdsi.org/t/truven-marketscan-datasets-in-cdm/4714/2.

[28] What are the advantages of electronic health records? Online; accessed 25-July-2021. URL: https://www.healthit.gov/faq/what-are-advantages-electronic-health-records.

[29] What is a clinical narrative? Online; accessed 25-July-2021. URL: https://www.mghpcs.org/ipc/programs/recognition/Describing.asp.

[30] Steven Abney. Parsing by chunks. In *Principle-based parsing*, pages 257–278. Springer, 1991.

[31] Naveed Afzal, Vishnu Priya Mallipeddi, Sunghwan Sohn, Hongfang Liu, Rajeev Chaudhry, Christopher G. Scott, Iftikhar J. Kullo, and Adelaide M. Arruda-Olsonb. Natural language processing of clinical notes for identification of critical limb ischemia. 2017. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5808583, doi:10.1016/j.ijmedinf.2017.12.024.

[32] Monica Agrawal, Chloe O'Connell, Yasmin Fatemi, Ariel Levy, and David Sontag. Robust benchmarking for machine learning of clinical entity extraction. 2020. doi:10.48550/arXiv.2007.16127.

[33] João Rafael Almeida and Sérgio Matos. Rule-based extraction of family history information from clinical notes. 2020. URL: https://dl.acm.org/doi/abs/10.1145/3341105.3374000, doi:10.1145/3341105.3374000.

[34] Noha Alnazzawi, Paul Thompson, and Sophia Ananiadou. Mapping phenotypic information in heterogeneous textual sources to a domain-specific terminological resource. *PloS one*, 11(9):e0162287, 2016.

[35] Ruben Amarasingham, Laura Plantinga, Marie Diener-West, Darrell J Gaskin, and Neil R Powe. Clinical information technologies and inpatient outcomes: a multiple hospital study. 2009. doi:10.1001/archinternmed.2008.520.

[36] Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. 2010. doi:10.1136/jamia.2009.002733.

[37] Marta Atti, Fabrizio Pecoraro, Simone Piga, Daniela Luzi, and Massimiliano Raponi. Developing a surgical site infection surveillance system based on hospital unstructured clinical notes and text mining. *Surgical Infections*, 21(8):716–721, 2020. doi:10.1089/sur.2019.238.

[38] Facihul Azam, Aliyu Musa, Matthias Dehmer, Olli P. Yli-Harja, and Frank Emmert-Streib. Global genetics research in prostate cancer: A text mining and computational network theory approach. 2019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6383410/, doi:10.3389/fgene.2019.00070.

[39] Fisseha Berhane. Operations on word vectors. Online; accessed 30-July-2022. URL: https://datascience-enthusiast.com/DL/Operations_on_word_vectors.html.

[40] Ragnhildur I. Bjarnadottir and Robert J. Lucero. What can we learn about fall risk factors from ehr nursing notes? a text mining study. 2018. doi:10.5334/egems.237.

[41] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. 2004. doi:10.1093/nar/gkh061.

[42] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. 2016. doi:10.48550/arXiv.1607.04606.

[43] Alison Callahan, Nigam H. Shah, and Jonathan H. Chen. Research and reporting considerations for observational studies using electronic health record data. 2020. doi:10.7326/M19-0873.

[44] David Campos, Sérgio Matos, and José Luís Oliveira. A modular framework for biomedical concept recognition. 2013. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-281#citeas, doi:10.1186/1471-2105-14-281.

[45] Victor Castro, Caitlin Clements, Shawn Murphy, Vivian Gainer, Maurizio Slater, Jeffrey Weilburg, Jane Erb, Susanne Churchill, Isaac Kohane, Dan Iosifescu, Jordan Smoller, and Roy Perlis. Qt interval and antidepressant use: a cross sectional study of electronic health records. 2013. doi:10.1136/bmj.f288.

[46] Victor M. Castro, W. Kay Apperson, Vivian S. Gainer, Ashwin N. Ananthakrishnan, Alyssa P. Goodson, Christopher D. Herrick Taowei D. Wang, and Shawn N. Murphy. Evaluation of matched control algorithms in ehr-based phenotyping studies: A case study of inflammatory bowel disease comorbidities. 2014. doi:10.1016/j.jbi.2014.08.012.

[47] Victor M. Castro, W. Kay Apperson, Vivian S. Gainer, Ashwin N. Ananthakrishnan, Alyssa P. Goodson, Taowei D. Wang, Christopher D. Herrick, and Shawn N. Murphy. Evaluation of matched control algorithms in ehr-based phenotyping studies: A case study of inflammatory bowel disease comorbidities. 2014. doi:10.1016/j.jbi.2014.08.012.

[48] Jinying Chen, Jiaping Zheng, and Hong Yu. Finding important terms for patients in their electronic health records: A learning-to-rank approach using expert annotations. 2016. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5156821, doi:10.2196/medinform.6373.

[49] Qingyu Chen, Yifan Peng, and Zhiyong Lu. Biosentvec: creating sentence embeddings for biomedical texts. 2018. doi:10.48550/arXiv.1810.09302.

[50] Nancy Chinchor and Patricia Robinson. Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29, pages 1–21, 1997.

[51] Hyejin Cho, Wonjun Choi, and Hyunju Lee. A method for named entity normalization in biomedical articles: application to diseases and plants. 2017. doi:10.1186/s12859-017-1857-8.

[52] Hongjie Dai, Yen Ching Chang, Richard Tzong-Han Tsai, and Wen-Lian Hsu. New challenges for biological text-mining in the next decade. 2009. doi:10.1007/s11390-010-9313-5.

[53] Son Doan, Lisa Bastarache, Sergio Klimkowski, Joshua C Denny, and Hua Xu. Integrating existing natural language processing tools for medication extraction from discharge summaries. *Journal of the American Medical Informatics Association*, 17(5):528–531, 09 2010. doi:10.1136/jamia.2010.003855.

[54] David Ferrucci and Adam Lally. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004. doi:10.1017/S1351324904003523.

[55] Fern FitzHenry, Olga V. Patterson, Jason Denton, Jesse Brannen, Ruth M. Reeves, Scott L. DuVall, and Michael E. Matheny. Omop cdm for natural language processing: Piloting a va nlp data set.

[56] Tracy D Gunter and Nicolas 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):e383, 2005.

[57] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. 2012. doi:10.1016/C2009-0-61819-5.

[58] David Hand and Peter Christen. A note on using the f-measure for evaluating record linkage algorithms. *Statistics and Computing*, 28(3):539–547, 2018. doi:10.1007/s11222-017-9746-6.

[59] Marcus Hassler and Günther Fliedl. Text preparation through extended tokenization. *WIT Transactions on Information and Communication Technologies*, 37, 2006.

[60] Marti Hearst. What is text mining. *SIMS, UC Berkeley*, 5, 2003.

[61] Jane Herwehe, Wayne Wilbright, Amir Abrams, Susan Bergson, Joseph Foxhood, Michael Kaiser, Luis Smith, Ke Xiao, Amy Zapata, and Manya Magnus. Implementation of an innovative, integrated electronic medical record (emr) and public health information exchange for hiv/aids. 2011. doi:10.1136/amiajnl-2011-000412.

[62] Kristina M Hettne, Erik M van Mulligen, Martijn J Schuemie, Bob JA Schijvenaars, and Jan A Kors. Rewriting and suppressing umls terms for improved biomedical term identification. 2010. doi:10.1186/2041-1480-1-5.

[63] George Hripcsak, Jon D Duke, Nigam H Shah, Christian G Reich, Vojtech Huser, Martijn J Schuemie, Marc A Suchard, Rae Woong Park, Ian Chi Kei Wong, Peter R Rijnbeek, Johan van der Lei, Nicole Pratt, G Niklas Norén, Yu-Chuan Li, Paul E Stang, David Madigan, , and Patrick B Ryang. Observational health data sciences and informatics (ohdsi): Opportunities for observational researchers. PMID:26262116.

[64] Thomas H. McCoy Jr, Victor M. Castro, Ashlee M. Roberson, Leslie A. Snapper, and Roy H. Perlis. Improving prediction of suicide and accidental death after discharge from general hospitals with natural language processing. 2016. doi:10.1001/jamapsychiatry.2016.2172.

[65] Ning Kang, Erik M van Mulligen, and Jan A Kors. Comparing and combining chunkers of biomedical text. *Journal of biomedical informatics*, 44(2):354–360, 2011.

[66] Anne Kao and Steve Poteet. *Natural language processing and text mining*. Springer Science & Business Media, 2007.

[67] Martijn G Kersloot, Francis Lau, Ameen Abu-Hanna, Derk L Arts, and Ronald Cornet. Automated snomed ct concept and attribute relationship detection through a web-based implementation of ctakes. *Journal of biomedical semantics*, 10, 2019. URL: https://www.ncbi.nlm.nih.gov/pubmed/31533810, doi:10.1186/s13326-019-0207-3.

[68] Ellen Kim, Samuel M Rubinstein, Kevin T Nead, Andrzej P Wojcieszynski, Peter E Gabriel, and Jeremy L Warner. The evolving use of electronic health records (ehr) for research. In *Seminars in radiation oncology*, volume 29, pages 354–361. Elsevier, 2019.

[69] Ajay Kulkarni, Deri Chong, and Feras A. Batarseh. 5 - foundations of data imbalance and solutions for a data democracy. In Feras A. Batarseh and Ruixin Yang, editors, *Data Democracy*, pages 83–106. Academic Press, 2020. URL: https://www.sciencedirect.com/science/article/pii/B9780128183663000058, doi:https://doi.org/10.1016/B978-0-12-818366-3.00005-8.

[70] Robert Leaman, Ritu Khare, and Zhiyong Lu. Challenges in clinical natural language processing for automated disorder normalization. 2015. doi:10.1016/j.jbi.2015.07.010.

[71] Hongfang Liu, Suzette J. Bielinski, Sunghwan Sohn, Sean Murphy, Kavishwar B. Wagholikar, Siddhartha R. Jonnalagadda, K.E. Ravikumar, Stephen T. Wu, Iftikhar J. Kullo, and Christopher G Chute. An information extraction framework for cohort identification using electronic health records. 2013. PMID:24303255. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3845757/.

[72] Sijia Liu, Liwei Wang, Donna Ihrke, Vipin Chaudhary, Cui Tao, Chunhua Weng, and Hongfang Liu. Correlating lab test results in clinical notes with structured lab data: a case study in hba1c and glucose. *AMIA Summits on Translational Science Proceedings*, 2017:221, 2017.

[73] Filipe Lucini, Flavio Fogliatto, Giovani da Silveira, Jeruza Neyeloff, Michel Anzanello, Ricardo Kuchenbecker, and Beatriz Schaan. Text mining approach to predict hospital admissions using early medical records from the emergency department. *International Journal of Medical Informatics*, 100:1–8, 2017. URL: https://www.sciencedirect.com/science/article/pii/S1386505617300011, doi:https://doi.org/10.1016/j.ijmedinf.2017.01.001.

[74] Yen-Fu Luo, Weiyi Sun, and Anna Rumshisky. Mcn: A comprehensive corpus for medical concept normalization. 2019. doi:10.1016/j.jbi.2019.103132.

[75] Sérgio Matos, David Campos, Renato Pinho, Raquel M. Silva, Matthew Mort, David N. Cooper, and José Luís Oliveira. Mining clinical attributes of genomic variants through assisted literature curation in egas. 2016. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4897594/, doi:10.1093/database/baw096.

[76] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298, 1978.

[77] Ruslan Mitkov. *The Oxford handbook of computational linguistics*. Oxford University Press, 2022.

[78] I. C. Mogotsi, Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval. *Information Retrieval*, 13(2):192–195, Apr 2010. doi:10.1007/s10791-009-9115-y.

[79] Behrang Mohit. *Named Entity Recognition*, pages 221–245. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-45358-8_7.

[80] Robert G. Brooks Nir Menachemi. Reviewing the benefits and costs of electronic health records and associated patient safety technologies. 2005. doi:10.1007/s10916-005-7988-x.

[81] Damien Nouvel, Maud Ehrmann, and Sophie Rosset. *Named entities for computational linguistics*. John Wiley & Sons, 2016.

[82] Tiago Nunes, David Campos, Sérgio Matos, and José Luís Oliveira. Becas: biomedical concept recognition services and visualization. 2013. URL: https://academic.oup.com/bioinformatics/article/29/15/1915/265850, doi:10.1093/bioinformatics/btt317.

[83] Pascal Pfiffner. py-umls. Online; accessed 30-July-2022. URL: https://github.com/chb/py-umls.

[84] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[85] Preethi Raghavan, James L. Chen, Eric Fosler-Lussier, and Albert M. Lai. How essential are unstructured clinical narratives and information fusion to clinical trial recruitment? 2014.

[86] Anand Rajaraman and Jeffrey David Ullman. *Data Mining*, page 1–17. Cambridge University Press, 2011. doi:10.1017/CBO9781139058452.002.

[87] Ruth Reátegui and Sylvie Ratté. Automatic extraction and aggregation of diseases from clinical notes. In *International Conference on Information Technology & Systems*, pages 846–855. Springer, 2018.

[88] Ruth María Reátegui and Sylvie Ratté. Comparison of metamap and ctakes for entity extraction in clinical notes. 2018. URL: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-018-0654-2, doi:10.1186/s12911-018-0654-2.

[89] Anthony Rios. pymetamap: Python wraper for metamap. Online; accessed 30-July-2022. URL: https://github.com/AnthonyMRios/pymetamap.

[90] Alejandro Rodríguez-González, Roberto Costumero, Marcos Martinez-Romero, Mark Wilkinson, and Ernestina Menasalvas-Ruiz. Extracting diagnostic knowledge from medline plus: A comparison between metamap and ctakes approaches. 2018. doi:10.2174/1574893612666170727094502.

[91] AK Santra and C Josephine Christy. Genetic algorithm and confusion matrix for document clustering. *International Journal of Computer Science Issues (IJCSI)*, 9(1):322, 2012.

[92] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and

knowledge extraction system (ctakes): architecture, component evaluation and applications. 2010. doi:10.1136/jamia.2009.001560.

[93] João Silva, Rui Antunes, João Almeida, and Sérgio Matos. Clinical concept normalization on medical records using word embeddings and heuristics. 2020. doi:10.3233/SHTI200129.

[94] Michael Simmons, Ayush Singhal, and Zhiyong Lu. Text mining for precision medicine: bringing structure to ehrs and biomedical literature to understand genes and health. *Translational Biomedical Informatics*, pages 139–166, 2016.

[95] Gregory Smith. *PostgreSQL 9.0: High Performance*. Packt Publishing Ltd, 2010.

[96] NP Tatonetti, JC Denny, SN Murphy, GH Fernald, G Krishnan, V Castro, P Yue, PS Tsau, I Kohane, DM Roden, and RB Altman. Detecting drug interactions from adverse-event reports: Interaction between paroxetine and pravastatin increases blood glucose levels. 2011. doi:10.1038/clpt.2011.83.

[97] Eugene Tseytlin, Kevin Mitchell, Elizabeth Legowski, Julia Corrigan, Girish Chavan, and Rebecca S. Jacobson. Noble – flexible concept recognition for large-scale biomedical natural language processing. 2016. URL: https://link.springer.com/article/10.1186/s12859-015-0871-y, doi:10.1186/s12859-015-0871-y.

[98] Hongkui Tu, Zongyang Ma, Aixin Sun, and Xiaodong Wang. When metamap meets social media in healthcare: Are the word labels correct? In *Asia Information Retrieval Symposium*, pages 356–362. Springer, 2016. doi:10.1007/978-3-319-48051-0_31.

[99] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

[100] Sophia Y. Wang, Suzann Pershing, Elaine Tran, and Tina Hernandez-Boussard. Automatic extraction and aggregation of diseases from clinical notes. 2019. URL: AutomatedExtractionofOphthalmicSurgeryOutcomesfromtheElectronicHealthRecord, doi:10.1016/j.ijmedinf.2019.104007.

[101] Stephen John Matthew C. Wenceslao and Maria Regina Justina E. Estuar. Using ctakes to build a simple speech transcriber plugin for an emr. *Conference: the third International Conference*, 2019. URL: https://www.researchgate.net/publication/334903773_Using_cTAKES_to_Build_a_Simple_Speech_Transcriber_Plugin_for_an_EMR, doi:10.1145/3340037.3340044.

[102] Ian H Witten. Text mining, 2004.

[103] Yonghui Wu, Jeremy L. Warner, Liwei Wang, Min Jiang, Jun Xu, Qingxia Chen, Hui Nian, Qi Dai, Xianglin Du, Ping Yang, Joshua C. Denny, Hongfang Liu, and Hua Xu. Discovery of noncancer drug effects on survival in electronic health records of patients with cancer: A new paradigm for drug repurposing. 2019. doi:10.1200/CCI.19.00001.

[104] Yunqing Xia, Xiaoshi Zhong, Peng Liu, Cheng Tan, Sen Na, Qinan Hu, and Yaohai Huang. Combining metamap and ctakes in disorder recognition: Thcib at clef ehealth lab 2013 task 1. In *CLEF (Working Notes)*, 2013.

[105] Hua Xu, Son Doan, Kelly Birdwell, James D. Cowan, Andrew J. Vincz, David W. Haas, Melissa A. Basford, and Joshua C. Denny. An automated approach to calculating the daily dose of tacrolimus in electronic health records. 2010. PMID:21347153. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041548/.

[106] Hua Xu, Shane P Stenner, Son Doan, Kevin B Johnson, Lemuel R Waitman, and Joshua C Denny. Medex: A medication information extraction system for clinical narratives. 2010. URL: https://www.researchgate.net/publication/40907392_MedEx_A_medication_information_extraction_system_for_clinical_narratives, doi:10.1197/jamia.M3378.

[107] Shang-Ming Zhou, Ronan A. Lyons, Muhammad A. Rahman, Alexander Holborow, and Sinead Brophy. Predicting hospital readmission for campylobacteriosis from electronic health records: A machine learning and text mining perspective. *Journal of Personalized Medicine*, 12(1), 2022. URL: https://www.mdpi.com/2075-4426/12/1/86, doi:10.3390/jpm12010086.