A Medication Extraction Framework for Electronic Health Records

by

Andreea Bodnari

S.B., Worcester Polytechnic Institute (2010)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2012

Author . . . .
Department of Electrical Engineering and Computer Science
August 30, 2012

Certified by . . . .
Peter Szolovits
Professor of Computer Science and Engineering- MIT CSAIL
Thesis Supervisor

Certified by . . . .
Özlem Uzuner
Assistant Professor of Information Studies- SUNY, Albany
Thesis Supervisor

Accepted by . . . .
Professor Leslie A. Kolodziejski
Chair of the Department Committee on Graduate Students

A Medication Extraction Framework for Electronic Health Records

by

Andreea Bodnari

Submitted to the Department of Electrical Engineering and Computer Science
on August 30, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

This thesis addresses the problem of concept and relation extraction in medical documents. We present a medical concept and relation extraction system (*medNERR*) that incorporates hand-built rules and constrained conditional models. We focus on two concept types (i.e., *medications* and *medical conditions*) and the pairwise *administered-for* relation between these two concepts. For medication extraction, we design a rule-based baseline $medNERR_{med}^{greedy}$ that identifies medications using the UMLS dictionary. We enhance $medNERR_{med}^{greedy}$ with information from topic models and additional corpus-derived heuristics, and show that the final medication extraction system outperforms the baseline and improves on state-of-the-art systems. For medical conditions extraction we design a Hidden Markov Model with conditional constraints. The conditional constraints frame world knowledge into a probabilistic model and help support model decisions. We approach relation extraction as a sequence labeling task, where we label the context between the medications and the medical concepts that are involved in an administered-for relation. We use a Hidden Markov Model with conditional constraints for labeling the relation context. We show that the relation extraction system outperforms current state of the art systems and that its main advantage comes from the incorporation of domain knowledge through conditional constraints. We compare our sequence labeling approach for relation extraction to a classification approach and show that our approach improves final system performance.

Thesis Supervisor: Peter Szolovits
Title: Professor of Computer Science and Engineering- MIT CSAIL

Thesis Supervisor: Özlem Uzuner
Title: Assistant Professor of Information Studies- SUNY, Albany

Acknowledgments

My MIT journey has been a wonderful opportunity for professional and personal development. This journey had its own struggles and occasional failures, but those hardships helped model a stronger character and a more thorough scientist. I would like to bring thanks to all the special people who believed in me and my potential, and also to those who showed skepticism.

I would like to thank my advisors, prof. Peter Szolovits and prof. Özlem Uzuner, for their support and guidance, and for giving me the freedom to explore new horizons. I would also like to thank Rohit Joshi for taking the time to involve in rewarding spiritual and scientific conversations.

I owe a great deal of appreciation to the CSAIL faculty and administrative personnel who helped me schedule my RQE examination and submit my PhD proposal in such a short time period.

Last but not least, I would like to thank my wonderful friends, Cosmin Gheorghe and Towa Matsumura. Cosmin, thank you for all the support, care, and life lessons. Thank you Towa for unconditionally being there for me.

Meae familiae, ignoscite mihi relinquenti.
Omne opus meum dedicatum est vobis, magno cum amore, ex toto corde meo.

Per donum Dei, credo.
Fortitudo mea de caelis est. Deo adiuvante, non timendum.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electronic medical record (EMR) systems have become ubiquitous in the USA, and are steadily increasing in popularity in other countries as well. The prospective value of the information stored in such EMR systems is endless: it helps better monitor the patient's hospital stay, analyze treatment efficiency, predict health status and disease occurrences. Jensen et al. [22] discuss the value of the data stored in the narratives of medical records. They emphasize that EMR data can be used for personalized medicine, disease correlation identification, and together with genetic data it can help reveal genotype-phenotype relationships. Because clinical narratives have a free-text form, data is presented in an iregular structure and cannot be immediately extracted by automated medical systems. Natural language processing (NLP) techniques like named-entity extraction, relation recognition, and coreference resolution are often employed for leveraging the data of interest from EMRs.

Nevertheless, developing NLP techniques for clinical data is a challenging task. Factors that contribute to this challenge are the specialized lexicon, the terse language, the frequent misspellings, and the lack of common structure across EMR systems and across institutions. In addition, it is cumbersome to maintain an up-to-date medical NLP system as the medical field is very dynamic, with a rapidly increasing lexicon due to the frequent medical discoveries. Other issues are the lack of documentation for medical synonymy relationships and the abundant use of acronyms and abbreviations. Performing abbreviation disambiguation becomes a challenge in itself as there are numerous unrelated medical concepts that can be mapped to one given abbreviation (e.g., *P.O.D.* can be disambiguated into

20 valid medically-related terms like *place of death*, *perception of dyspnea*, *progression of diseases*). In addition, creating gold standards for the clinical domain is a more difficult task when compared to the general domain. Firstly, granting human annotators access to medical records is a sensitive problem due to patient confidentiality concerns. Secondly, the medical lexicon used in the narrative of EMRs varies significantly depending on institution and medical department (e.g., medical records from a teaching hospital tend to contain names of new medications used only in clinical trials, while regular hospitals use more standard medication names; also the language used in a pathology report is different from the language used in a surgical report).[2] Because novel medical concepts are often underrepresented, there is a high probability that the gold standard will not cover certain classes of medical terms; trying to constantly update the gold standard in order to create a representative set can become an cumbersome task. In contrast, even though the general domain (i.e., newswire) also adopts novel concepts, those terms are better represented as every mass-media institution tends to discuss the fresh news where those novel concepts are mentioned. Consequently, it is much easier to create a representative gold standard for the general domain.

We believe that the clinical domain is in need of adaptive NLP systems that expect little to no human intervention and are able to incorporate clinical knowledge from external sources. By using probabilistic learning techniques that incorporate world knowledge and declarative constraints, such systems will consequently be up-to-date with the evolution of the medical domain. Automated systems that have little interaction with humans would reduce the risk of data breach and consequently better comply with patient confidentiality guidelines.

In this work we address the problem of named-entity recognition (NER) and relation extraction for medical documents. Our medical named-entity and relation extraction system (*medNERR*) focuses on two concept types: *medications* and *medical conditions*. It extracts the pairwise *administered-for* relation defined when a medication is administered for a given medical condition. We develop the *medNERR* medication extraction module using domain-derived heuristics, topic modeling, and external knowledge sources. For the medical condition concepts we design a supervised probabilistic model with conditional constraints. The conditional constraints incorporate world knowledge on top of a probabilistic model learnt from observed data. This world knowledge is included in the form of hand-built soft rules and guides the model learning by allowing access to information that cannot be expressed through training data, or information that was not available in the training data. We model our solution to the relation extraction task using a sequence labeling approach, where we aim to identify the context between a medication and a medical condition involved in an administered-for relation. The context labeling is performed by a probabilistic model with conditional constraints. We show that domain knowledge expressed through conditional constraints contributes to the improvement of medical NLP systems. We also show that the sequence labeling approach for relation extraction performs better when compared to a classification approach. Overall, our relation extraction system represents an improvement over current state-of-the-art medical NLP systems.

1.0.1 Real World Applications

Concept and relation extraction systems for medical documents present several real-world applications. Concept extraction systems can be used as independent resources or as input to other NLP applications:

1. Pharmacotherapy is the most commonly used therapy for diseases. In order to guarantee safe, appropriate, and economical use of medications, the health professional requires immediate access to a patient's past and ongoing list of medications and the reasons for medication administration. Yet, the medications and their reason for administrations are stored as free-text inside medical documents and it is time consuming to read through collections of documents in order to retrieve the information of interest. Some of the medical record systems facilitate the automatic retrieval of lists of medications for a given patient but they still cannot retrieve the reasons for medication administration. In order to obtain the reason for medication administration the health professional would still have to relate back to the free-text narrative. Automated medication extraction system can help with preparing and displaying the reason for medication administration in a more efficient and accurate form.

2. Medical condition extraction systems are extremely valuable in the emergency department (ED). Most patient cases seen in the ED require immediate assistance, and the health care provider cannot take the time to read through the medical records in order to determine what additional medical conditions the patient might have. Patients with serious heart problems might not be recommended for anesthesia, or might require a specific dosage of anesthetic. Being aware of similar restrictions imposed by ongoing medical conditions is extremely important in order to guarantee that a patient's health is not aggravated by the choice of treatment. Because of the risk associated with trusting automated NLP systems, such systems would provide with a link to document paragraph from which the medical concept was extracted. Thus, health care providers can examine the specific paragraph instead of reading the entire document in order to reach the paragraph of interest.

Medical relation extraction systems are also valuable towards a series of applications:

1. Not all drug side-effects can be determined before a drug is approved for clinical use. If a patient starts taking medication X and then develops unexpected symptom Y, we can draw the preliminary conclusion that symptom Y is a side-effect of medication X. If many such scenarios are encountered and documented in EMRs, then we can stipulate that symptom Y is a side-effect for medication X.

2. Relation extraction systems can be used to better target and monitor treatments. For example, such systems could infer with a certain confidence rate the medication effectiveness on patients. Keeping track of the symptoms presented before and after medication administration, as well as the recurrence of medical conditions, are important factors in treatment targeting.

3. Because medical documents have a free-text form, it is often tedious and time consuming to read through collections of such documents. A relation extraction system could be used to generate a summary of a patient's history of medical conditions and administrated treatments. Having such a summary available could improve quality of care and diagnosing accuracy, as health professionals would have access to an immediate overlook over patient's health history.

1.1 Contributions

Our proposed work represents an improvement over current state of the art in both concept and relation extraction. We show that medical knowledge coupled with conditional constraints are important factors in the performance of a medical NLP system. We present for the first time the effectiveness of combining probabilistic models with conditional constraints on the medical domain. We also improve on current approaches to relation extraction by designing a sequence labeling solution instead of a pairwise classification solution. The sequence labeling approach focuses on the text structure and on identifying the actual context of the relation, while the pairwise classification approach aims to distinguish between pairs of concepts based on a set of features.

1.1.1 Thesis structure

We begin in Chapter 2 with the problem description and an overview of related work. In Chapter 3, we discuss the system architecture. In Chapter 4 we present the experimental methods and evaluation strategy, followed by the results and discussion in Chapter 5. We present a proposal for future work and conclusions in Chapter 6.

Chapter 2

Background and Related Work

2.1 Problem Overview

In this work we address the problem of medication-related information extraction from medical documents. Our task if two-fold: initially identify the medication-related concepts and then perform relation extraction on the identified concepts. The concepts of interest include medications (i.e., any biological substance for which the patient is the experiencer) and medical conditions (i.e., any illness, injury, disease, or disorder experienced by the patient). The pairwise relation we capture (i.e., "administered-for") is established between a medication – medical condition pair, and specifies that the medication was administered for the given medical condition. The administered-for relation must be mentioned in the narrative of the medical record and cannot be inferred or assumed based on prior medical knowledge.

$$\texttt{Patient with }\underbrace{\texttt{depression}}_{prob_1}\texttt{ and }\underbrace{\texttt{chronic kidney disease}}_{prob_2}$$

$$\texttt{is considered for }\underbrace{\texttt{hemodialysis}}_{med_1}\texttt{; he is continued on}$$

$$\underbrace{\texttt{Caltrate plus D}}_{med_2}\texttt{ and }\underbrace{\texttt{multivitamin}}_{med_3}\texttt{ and was started on }\underbrace{\texttt{advil}}_{med_4}$$

$$\texttt{for }\underbrace{\texttt{his pain}}_{prob_3}\texttt{ .}$$

Figure 2-1: An example of a sentence from a patient discharge summary with labeled concepts. Note: *med* represents a medication and *prob* represents a medical condition.

We perform information extraction on a corpus of de-identified medical discharge summaries. Figure 2-1 presents a sample sentence from a medical discharge summary with labeled concepts. The given sentence captures several medications (e.g., hemodialysis, Caltrate plus D) and medical conditions (e.g., chronic kidney disease, his pain). The administered-for relations occur between the med_1-$prob_2$ and med_4-$prob_3$ concept pairs. Some concepts do not participate in any relations (e.g., med_3, $prob_1$).

Our aim is to develop a medical NLP system that takes as input the narrative of a patient's medical record and outputs the medication and medical condition concepts, as well as the relations that occur between the two sets of concepts. Throughout this paper we refer to the medical conditions involved in an administered-for relation as the reasons for medication administration.

2.1.1 Data

We develop *medNERR* on the medical corpus released for the 2009 Informatics for Integrating Biology and the Bedside (i2b2) medication extraction challenge.[57] The i2b2 medication corpus contains de-identified medical discharge summaries from Partners Healthcare and was released under a data use agreement with approval by appropriate Institutional Review Boards. Gold standard data includes annotations for medications, reasons, and relations, as well as the narratives of the medical records. Not all medical condition concepts were annotated as part of the gold standard, but only the medical condition concepts that took part in an administered-for relation (i.e., reasons). The training data was annotated by University of Sydney and released following the i2b2 medication challenge. The test data was annotated with the help of challenge participants; each document was annotated by two annotators and disagreements were resolved by a third annotator. The i2b2 organizers compiled the team annotations and released the test ground truth during the medication challenge.[58]

A total of 114 discharge summaries were used for training, 31 for development, and 251 for testing. Table 2.1 presents the distribution of medications and reasons in the training, development, and test data. The number of medications and reasons are similar across the training and test sets: a total of 6520 medications are contained in the training set and 8942 in the test set, while a total of 1389 reasons are contained in the training set and 1637 in the test set. On average, the training set presented a higher number of medications and reasons per file (average of 57 medications and 12 reasons per file) when compared to the test set (average of 36 medications and 6 reasons per file). We believe the difference in the concept ratio between the training and test set is a consequence of the fact that the training set was prepared by one annotator, while the test set was annotated by two independent annotators and an arbitrator.

	Total	Average per Discharge Summary
Training		
Medications	6520	57.193
Reasons	1389	12.184
Development		
Medications	1466	47.290
Reasons	291	9.387
Test		
Medications	8942	36.625
Reasons	1637	6.522

Table 2.1: Distribution of the medication and reason concepts in the training, development, and test data.

2.2 Related work

2.2.1 General domain information extraction

The NER task was defined during the Sixth Message Understanding Conference (MUC-6).[1] NER involved the identification of concepts expressed in written natural language and the classification of those concepts into sets of predefined types. The NER task initially targeted proper names as concept types, the most common examples being persons, locations, and organizations. The three most frequent concept types were referred to as "enamex", a term also defined during the MUC-6 competition. Later research divided the enamex types into more specialized sub-categories like city, state, and country for the location concept type[16, 26] politician and entertainer for the person concept type.[17]

CONLL conferences introduced the miscellaneous concept types, which covered all entities outside the enamex class. The availability of the GENIA corpus[38] in the bioinformatics domain led to the generation of specialized concept types like protein, DNA, cell line and cell type.[46, 51, 42]

While initial approaches to NER were rule-based algorithms developed on linguistic intuition, more recent approaches used machine learning techniques and graphical models. More modern solutions to NER used supervised machine learning as a way to automatically induce rule-based systems or sequence labeling algorithms that rely on a collection of training examples. Common supervised learning methods for NER included hidden markov models (HMM),[6] decision trees,[45] maximum entropy models,[8] support vector machines (SVMs),[5] and conditional random fields (CRFs).[30] The greatest shortcoming of supervised machine learning methods was their dependency on annotated corpora. The unavailability of large annotated corpora led to the development of alternative machine learning methods like semi-supervised learning[39, 36] and unsupervised learning.[15, 34] For a more extensive review of the state of the art in NER see Nadeau et al.[35], Kim-Sang et al.[50], and Ratinov et al.[41]

Extracting relations between named entities is a crucial step towards the understanding of written natural language. A relation represents a function $r(\bullet) \rightarrow \{0, 1\}$ over a tuple $t = (c_1, c_2, .., c_n)$, where $c_k, k \in [1, n]$ are concepts extracted from a given text. Most relations are binary (i.e., *administered-for(pain, ibuprofen)*, *president-of(Obama, USA)*), but more complex n-ary relationships do exists in specialized domains (i.e., *point-mutation(codon, 12, G, T)* in the biomedical domain). Initial relation extraction systems approached the task as a binary classification problem. Higher-order relations were generally factorized into binary relations and represented on a graph.[31] Algorithms were developed for reconstructing the initial complex relation from maximal cliques in the graph. The advantage of reducing higher-order relations to their binary counterpart was the possibility of applying the same methods developed for binary relations.

The research community proposed several machine learning solutions for relation extraction. One of the commonly used methods is supervised classification. It can be designed as a feature-based[24, 62] or a kernel-based solution.[60, 11] The kernel-based solution presents an advantage over the feature based solution as it can efficiently explore large feature spaces without the need of explicit feature representation. Motivated by the lack of availability of annotated relation corpora, the research community started developing semi-supervised and unsupervised relation extraction methods. The Dual Iterative Pattern Relation Extraction (DIPRE)[10] and Snowball[3] systems are two semi-supervised systems that rely on a small set of annotated relations for training. KnowItAll[15] and TextRunner[14] are two self-trained binary relation extraction systems that address large scale relation identification. For an extensive review of relation extraction methodologies refer to Geetha et al.[19]

2.2.2 Medical information extraction

Early research initiatives in medical NLP approached the task of named-entity recognition. The resulting medical NLP systems were rule-based solutions targeted at the medication concept type. Sirohi et al.[48] analyzed the performance of a dictionary-based NLP system with respect to the effectiveness of the employed lexicon. Their best system performance was 96.9% specificity and 85.2% sensitivity on a corpus of 100 medical records, using a lexicon of medication names, generic names, abbreviated medication names, and filtering techniques. Levin et al.[27] developed an algorithm to extract medication names from anesthesia electronic health records and normalize them to medication concepts from the RxNorm database.[47] Their algorithm used open source spelling correction tools and medication lists, and achieved a best sensitivity of 92.2% and a best specificity of 95.7%. Burton et al.[12] explored the task of linking medications to their reasons and developed a dictionary-based system with a 67.5% sensitivity and 86% specificity. Another exploration on medical NER was performed by Breydo et al.[9] The authors developed an algorithm for detecting inactive medications from the narrative of medical records, where inactive medications represented those medications previously taken by the patient but discontinued at the time the medical record was documented. Their system was evaluated against a total of 297 outpatient notes, and reported 87.7% sensitivity and 95.2% specificity on notes with documented inactive medications. Jagannathan et al.[21] compared four commercial general domain NER systems on their performance in the medical domain. The results predicted by each commercial system were combined to generate a comprehensive system predictions set. Evaluation on the combined predictions showed good performance on the medication concepts (93.2% F-measure) and a lower performance on concepts like strength (85.3% F-measure), route (80.3% F-measure), and frequency (48.3% F-measure). Research initiatives have been taken towards the development of more comprehensive NLP systems for medical records. Examples are the clinical Text Analysis and Knowledge Extraction (cTakes) system[44] developed at Mayo Clinic, the i2b2 HITEX[61], the Medical Language Extraction and Encoding (MedLEE) system[18] and MedEx.[59]

The i2b2 competitions steered the interest of the research community towards information extraction from unstructured clinical notes and discharge summaries. The first i2b2 competition[56] tried to classify the smoking status of a given patient while the second competition targeted the obesity status of a patient.[54] The third i2b2 competition (i.e., the i2b2 medication challenge)[57] geared the interest of the medical NLP community towards the development of NLP systems for extraction of medication-related information from medical records. The i2b2 medication challenge gathered 20 teams from different institutions around the world. Each team participated with a clinical NER system targeting five concept types: medications, dosages, modes, frequencies, durations, reasons, and list/narrative. Most teams developed rule-based systems and only three of the participating systems were hybrid designs between rule-based and supervised approaches.[40, 28, 32] The hybrid systems employed various machine learning techniques like AdaBoost, SVM, and CRFs. The competition evaluated systems on their overall performance on medication information extraction (i.e., overall evaluation). The top ranking system was a hybrid one with an overall F-measure of 85.7%, followed by a rule-based system with an overall F-measure of 82.1%. The competition pointed out that despite their good performance on most concepts, clinical NLP systems were having difficulties obtaining a good performance on complex concepts like reasons and durations. The i2b2 medication challenge nurtured further development of medical NER and relation extraction systems. Halgrim et al.[20] developed a hybrid system that initially detected medical concepts using a cascade of classifiers and then linked those concepts into medication events through a set of heuristics.

The authors tested their system on the i2b2 medication corpus and obtained an overall F-measure of 84.1%. Subsequent i2b2 competitions brought the research community together for solving NLP tasks like relation extraction[53] and coreference resolution.[55]

Even though the medical NLP field has received more attention within the past couple of years, there are still gaps to be filled in certain medical NLP tasks. The 2009 i2b2 medication information extraction challenge showed that automated systems can process structured concepts like medications using large vocabularies, but face difficulties in interpreting more subjective concepts like frequencies and reasons for administration, for which context understanding is required. We take on the challenge of developing an automated system for medication and reason extraction. Our goal is to process textual information at a deeper level and couple world knowledge with probabilistic models for better model learning.

Chapter 3

System architecture

medNERR consists of a preprocessing module, a concept extraction module, and a relation extraction module (see Figure 3-1). The concept extraction module (*medNERR*$_{concept}$) retrieves the medication and the medical condition concepts from a given medical document. The relation extraction module (*medNERR*$_{relation}$) determines the presence of an *administered-for* relation between a medication and a medical condition concept. We further describe the architecture of each module: section 3.1 addresses the preprocessing module, section 3.2 handles the concept extraction module, and the relation extraction module is discussed in section 3.3.

In the remaining of this paper we use the term $medNERR$ to refer to the complete medical NLP system; we attach a subscript to refer to the specific system task, and a superscript to refer to the specific system configuration (i.e., $medNERR_{task}^{configuration}$). For example, $medNERR_{med}^{greedy}$ represents the medication extraction module of the $medNERR$ system using only the greedy configuration.

3.1 Preprocessing

Sentence recognition. We use an in-house built rule-based sentence recognizer to identify the set of sentences from a given document. We opted for a simple sentence recognizer in order to optimize the final time performance of our system. The rule-based

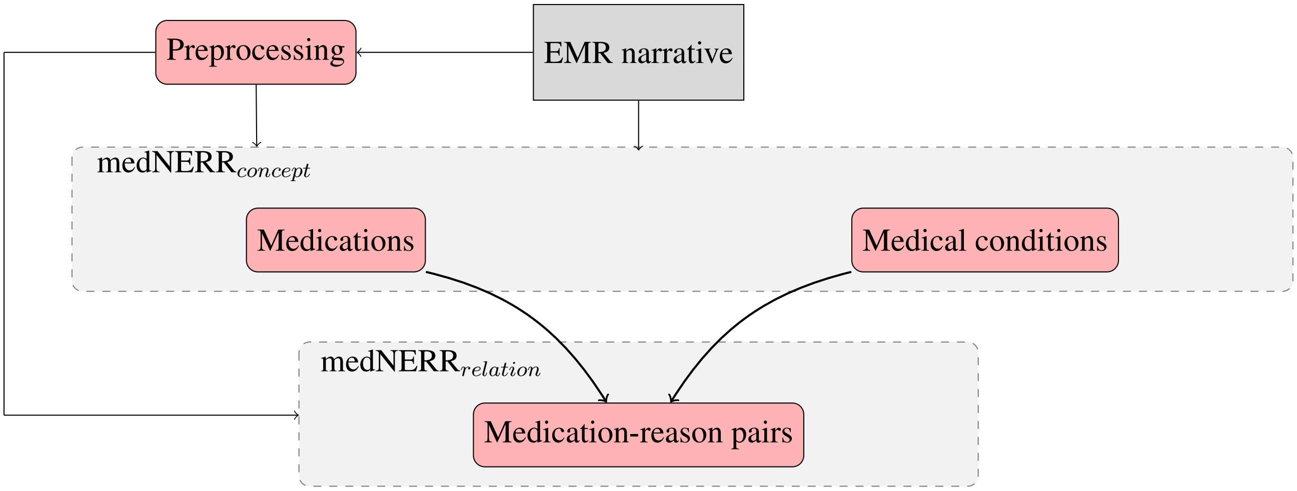

Figure 3-1: *medNERR* architecture

sentence recognizer delimits each sentence based on a set of punctuation signs (i.e., ".", "!","?"). The punctuation signs are not considered candidates for a sentence end if they occur within a word (e.g., "Ph.D.", "p.r.n.","M.D."). The sentence splitter outputs an ordered set of sentences, where the set is ordered based on the sentence occurrence order inside the given document. The rule-based sentence recognizer favors longer sentences by making a paragraph split based on punctuation signs alone. Because concepts are phrases spanning within a sentence and not across sentences, the sentence recognizer does not split the concepts of interest across multiple sentences.

Part-of-speech tagging. We apply the Stanford Log-linear Part-Of-Speech (POS) Tagger[25] in order to obtain the POS tags for each token in our corpus. We use the maximum entropy *left3words* model of the Stanford Tagger. The *left3words* model considers the three words to the left of the word being tagged as its context and is built on the Penn Treebank's Wall Street Journal sets 0-18. We run the tagger on a given document without performing any additional pre-processing. The tagger returns an ordered set of tokens and their associated part-of-speech tags.

Document section identification. We use an in-house built rule-based sectionizer to automatically identify systematic structures in medical records.[49] The sectionizer takes as input a predefined set of section header names and tries to identify whether the section header names occur within a given document. The set of section header names used in this research is included in Appendix A. The section headers were manually identified from a set of randomly sampled medical records. The sectionizer outputs a set of section headers and the narrative associated with each section header.

3.2 Concept extraction

3.2.1 Medication extraction

Our medication extraction system ($medNERR_{med}$) is composed of three sub-modules. The modules complement each other and together they produce a robust solution to medication extraction.

Module 1 ($medNERR_{med}^{greedy}$) We do not assume the existence of training data for medication extraction, but instead use the Unified Medical Language System(UMLS) database[7] to identify potential medication names. UMLS is a repository of biomedical vocabularies and contains a rich vocabulary of medications. We map text from medical documents to the UMLS database through the MetaMap software.[4] We discovered that using the UMLS database alone for medication identification is not sufficient, mainly due to two problems:

1. the UMLS database falsely classifies certain food names as medications; for example *"banana"* and *"grapefruit"* are classified as medication names.
2. MetaMap cannot account for the context in which a medication name is used; it consequently identifies some lab tests as medications because the specific lab test and medication share the same name. For example, the sentences *"patient's magnesium level was low"* and *"the patient was administered magnesium"* both contain the word *"magnesium"*, but in the first sentence the concept type is *lab test* while in the second sentence the concept type is *medication*. This distinction between the word *"magnesium"* being a lab test concept or a medication concept can be made only through context analysis.

We consider this module to be a greedy module ($medNERR_{med}^{greedy}$), as it gathers all possible medication names from a given document, without performing any filtering of false medication names.

Module 2 ($medNERR_{med}^{greedy+stingy}$) To address the problems encountered with *Module 1* we create a second module as a set of rules for removing the *"food"* medications and the *"lab test"* medications.

The $medNERR_{med}^{greedy+stingy}$ module contains a set of hand-built hard rules that describe several scenarios under which a phrase cannot be considered a medication. The rules are derived from observations on the development data. This module takes as input the medications identified by $medNERR_{med}^{greedy}$ and eliminates all medications that violate any of the rules; it is consequently *stingy* in terms of the phrases it accepts as medication candidates.

Module 3 ($medNERR_{med}^{greedy+neighbor}$) Because the rules introduced by *Module 2* are used as hard constraints, they remove some correct medication names. We decide to create a *neighbor model* in order to re-include some of the removed medications. In our *neighbor model* we hypothesize that many medications co-occur, and a high co-occurrence frequency translates into a higher confidence level for the medication extraction process. We use the output of $medNERR_{med}^{greedy}$ and $medNERR_{med}^{greedy+stingy}$ modules in order to include concepts with high confidence level.

This module relies on the *neighbor* property of medication tuples, and it is named after its hypothesis.

The architecture of the medication extraction system is depicted in Figure 3-2. We further describe each of the modules contained by the medication extraction system.

3.2.1.1 $medNERR_{med}^{greedy}$ module: UMLS medication mapping

We map the narrative of a medical document to the UMLS database using Metamap. For a given document d, the preprocessing module returns a set of sentences $S_d = \{s_1, s_2, .., s_l\}$, where l is the total number of sentences inside d. For each $s_i \in S_d$, $i \in [1, l]$ MetaMap

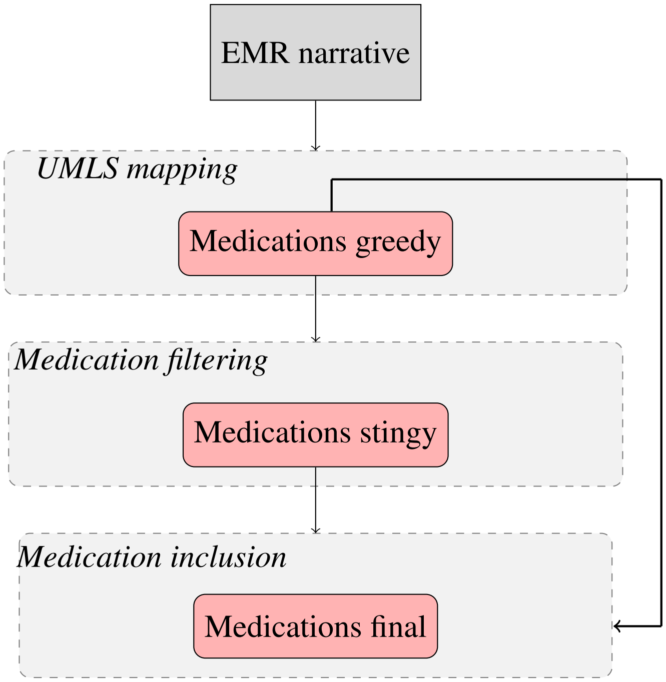

Figure 3-2: Medication extraction system architecture.

retrieves the set of highest scoring UMLS concepts. We filter the set of highest scoring concepts to only contain concepts with a medication-related semantic type. The medication-related semantic types are included in Table 3.1.

Abbreviation	Semantic type
antb	antibiotic
bacs	biologically active substance
clnd	clinical drug
orch	organic chemical
phsu	pharmacologic substance
strd	steroid

Table 3.1: UMLS semantic types used for selecting the medication concepts

The $medNERR_{med}^{greedy}$ module is separately applied to each document in our corpus. The final output of the module is the set of medication concepts:

$$M_{med}^{greedy} = \{m_{d_1,1}, m_{d_1,2}, ..m_{d_k,1}, m_{d_k,2}, .., m_{d_N,1}, m_{d_N,2}..\} \tag{3.1}$$

where d_k , $k \in [1, N]$ are documents in a corpus of length N and the index i in $m_{d_k,i}$ runs over all medication concepts found in document d_k.

3.2.1.2 $medNERR_{med}^{greedy+stingy}$ module: medication filtering

As previously mentioned, the medications set M_{med}^{greedy} contains some spurious medications concepts. These spurious concepts are generated because:

Issue 1 Metamap does not analyze the medical context in which a medication is encountered. It consequently cannot distinguish between medications and lab tests that have the same name (e.g., *magnesium* is both a medication and a lab test).

Issue 2 The UMLS database classifies specific foods as medications: phrases like *grapefruit*, *banana*, and *water* are consequently identified by Metamap as medications.

We propose a set of hand-built hard rules that incorporate solutions to ***Issue*** 1 and ***Issue*** *2* (see equation 3.2).

$$\mathcal{R} = \{R_k\}_{k=1}^{5} \textit{ where } R_k : m_{d_k,i} \rightarrow \{0, 1\} \tag{3.2}$$

Each rule represents a condition that must be verified by a medication concept. If any of the rules in equation 3.2 are violated by a given medication concept $m_{d_k,i}$, we remove the medication concept from the medications set M_{med}^{greedy}. The medication filtering rules are:

- R_1 : verify that concepts shorter than four characters are abbreviations. We hypothesize that all medication concepts are at least four characters long and concepts shorter than four characters are abbreviated forms of longer medication concepts. If a concept is shorter than four characters and is not contained in a verified list of medication abbreviations, we then classify it is an incorrect medication concept. Using the training dataset and the DailyMed database we create a list of medication abbreviations. We check that medication concepts of length less than four are included in our list of abbreviations.

- R_2 : verify that concept is located in a valid document section. Using the in-house built sectionizer we identify the set of sections present in a given document. We then require that our medication concept is located in a valid document section. Table 3.2 describes the list of invalid sections. The invalid sections were manually selected from the complete set of section headers described in Appendix A.

Section name	Description
entered dictated by attending	details about medical personnel
social history	details about social history
family history	details about family history
past surgical history	details about past surgical history
data laboratory data labs admission labs	report of lab result
allergies	list of known patient allergies
diet	details about patient's diet
physical examination	details about patient's physical examination

Table 3.2: Invalid document section headers.

- R_3 : medication does not end with colon. The health personnel reporting the patient's information tend to use medication names as section headers. In this scenario the medication name is not considered a valid medication concept. The R_3 constraint is specific to the corpus on which our system was developed.

- R_4 : medication is not classified as food according to Wikipedia. For each medication concept we perform a search on Wikipedia and retrieve the set of Wikipedia pages related to our concept. We check if one of the top 5 retrieved Wikipedia pages is classified under the *food*, *food and drink*, or *cuisine* categories.

- R_5 : in order to address *Issue 1* we require our system to differentiate between the context specific to a medication and the context specific to a lab test. We take advantage of the fact that medical documents contain a structured form with pre-defined sections; some of these sections contain only medication information while some sections are targeted to reporting lab test results. During the preprocessing step each document d in our corpus is sectionized and the relevant sections are identified. Across the entire corpus we create a set of medication sections and a set of lab test sections. We train a topic model over the set of medication sections and create the topic model TP_{med}. We then train a topic model over the set of lab test sections and create the topic model TP_{lab}. We use the Bound Encoding of the Aggregate Language Environment (BEAGLE)[23] algorithm for training our topic models. Figure 3-3 presents a graphical description of the proposed solution to *Issue 1*.

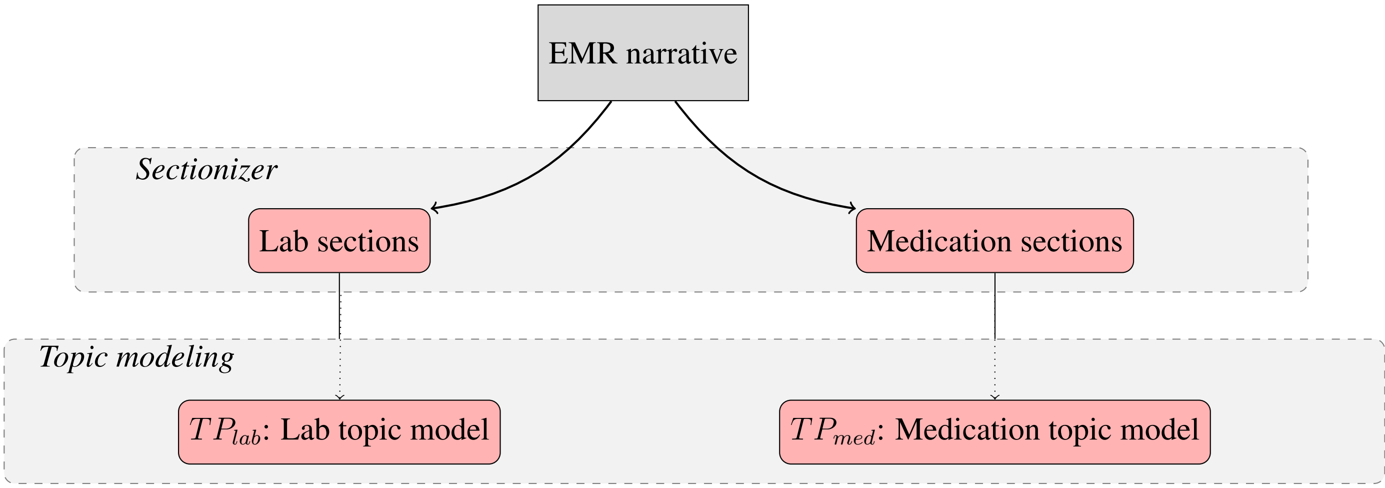

Figure 3-3: Topic model creation.

For constraint R_5 we then check that a given medication has higher probability of belonging to the TP_{med} than to the TP_{lab} topic model. Given a medication concept $m_{d_k,i}$, we compute:

1. $P(topic(m_{d_k,i}) = med|context)$ the probability of $m_{d_k,i}$ belonging to the medication topic model TP_{med} given the neighboring words (i.e., context)

2. $P(topic(m_{d_k,i}) = lab|context)$ the probability of the medication belonging to the lab topic model given the neighboring words

If $P(topic(m_{d_k,i}) = med|context) > P(topic(m_{d_k,i}) = lab|context)$ then $m_{d_k,i}$ belongs to TP_{med}, otherwise it belongs to TP_{lab}.

We iterate over all medications in set M_{med}^{greedy} and verify that they validate each of the R_1 – R_5 rules. The resulting filtered set of medications is:

$$M_{med}^{stingy} = \{m_{d_1,1}, m_{d_1,2}, ..m_{d_k,1}, m_{d_k,2}, .., m_{d_N,1}, m_{d_N,2}..\} \tag{3.3}$$

3.2.1.3 $medNERR_{med}^{greedy+neighbor}$ module: medication inclusion

The rules discussed in section 3.2.1.2 are hard constraints and eliminate some of the correct medication concepts. Yet, the hard constraints have the advantage of generating a set of medications with a low number of incorrect concepts. We consequently have a high confidence in the medications contained in the M_{med}^{stingy} set, and aim to re-include the filtered-out medications based on their relation to concepts already in M_{med}^{stingy}. We hypothesize that some medications are prescribed together in order to better treat the same medical condition (i.e., *heparin* and *warfarin* for anticoagulation, *glyburide* and *insulin* for type 2 diabetes). We use the medications in M_{med}^{stingy} as a reference set, and check whether there are medications in the set $M = M_{med}^{greedy} \backslash M_{med}^{stingy}$ that have a high probability of being prescribed together with an element in the reference set.

In order to compute the probability of medications being prescribed together we define a topic model TP over the entire corpus $\mathcal{C}$. We use the BEAGLE algorithm to generate the TP model. For each concept in the set M_{med}^{stingy} we generate the set N_{20}^{TP} as the set containing top 20 neighbors of a given concept according to TP. We chose the top 20 neighbors only, as the neighbor probability for the remaining neighbors was too low and in general introduced additional noise to the system without bringing significant contributions.

For each concept $m \in M$ we check whether it is contained in the set N_{20}^{TP} of a medication in M_{med}^{stingy}. If this condition is validated, we then re-include m as a medication. After we iterate over all concepts in M, we create the final set of medications:

$$M_{med}^{final} = \{m_{d_1,1}, m_{d_1,2}, .., m_{d_k,1}, m_{d_k,2}, .., m_{d_N,1}, m_{d_N,2}..\} \tag{3.4}$$

The final medication extraction system contains all of the three modules (i.e., greedy, stingy, and neighbor modules) and is referred to as $medNERR_{med}$ throughout the remaining chapters.

3.2.2 Medical condition extraction

We approach the medical condition extraction task as a sequence labeling task. We use a generative model to label sequences of tokens according to the Inside Outside Beginning (IOB) labeling scheme.[43] In our case the sequences of tokens are the sentences of each document in the corpus $\mathcal{C}$, as generated by the preprocessing module. Each token is labeled as B if it is the first token of a medical condition, I if it is a subsequent token of a medical condition, and O if it falls outside any medical condition. The sequence labeler is a HMM with conditional constraints.[13] The conditional constraints help encode expressive world and domain knowledge into the probabilistic model. They help model long distance relationships as well as first-order logic expressions, which cannot be integrated as system features.

The sequence labeler takes as input a sentence from our corpus and assigns an *I/O/B* label to each token of the given sentence. Our system uses the CogComp implementation of the constrained conditional model,[52] to which we add an additional set of features and a new set of conditional constraints. We describe the set of constraints in section 3.2.2.1 and the set of features in section 3.2.2.2. Both the constraints and the features are computed at the token level. The outcome of the medical condition extraction step is a set of medical conditions:

$$M_{medCond} = \{c_{d_1,1}, c_{d_1,2}, .., c_{d_k,1}, c_{d_k,2}, .., c_{d_N,1}, c_{d_N,2} ..\} \tag{3.5}$$

3.2.2.1 Constraints set

The constraints incorporate clinical and linguistic knowledge and are manually created. We specify below each of the constraints together with an explanation for the contribution they bring to the medical condition extraction system.

1. **Preposition signal**: specific linguistic prepositions signal an upcoming clarification (i.e., *claritin* **for** *the hay fever*) or justification (i.e., *cannot sing* **for** *I have a sore throat*). In the clinical domain such justifications and clarifications can represent

medical reasons for which a medication was administered. We target three prepositions (i.e., *for, of, to*) and hypothesize that these prepositions signal the beginning of a medical condition.

2. **Possessive pronouns**: according to the gold standard annotation guidelines the possessive pronouns are part of medical conditions (i.e., **her** *ongoing headache*). We thus require that the labeling for a medical condition includes the neighboring possessive pronouns.

3. **Treatment expressions**: medications are generally administered as treatment for specific medical conditions. We hypothesize that the word *treatment* would rarely be used inside a medical document unless it is followed by a target (i.e., *treatment* **for** *diabetes*). The treatment target is actually the medical condition of interest. We use the occurrence of the word stem *"treat"* as a signal for the beginning of a medical condition.

4. **Nearby medications**: the healthcare providers are required to document the patient's status as well as justify the administration or change in administration of treatment. We hypothesize that the occurrence of medications is a signal for the proximal occurrence of a medical condition. We use the medications set in order to determine where the medications occur within the medical record.

5. **Concept contains noun**: with few exceptions that refer to patient's generic symptoms (i.e., *feeling sick*, *looking pale*) most medical conditions represent a disease the patient is experiencing. In addition, the healthcare provider is less likely to prescribe medication based on symptoms alone, as the cause of the symptoms (i.e., disease) must be treated and not the symptom itself. Because disease names are classified as nouns from a grammatical perspective, we hypothesize that a medical condition must contain a noun.

6. **Concept adjective**: based on our intuition of what qualifies as a medical condition and based on the language of the medical records, we hypothesize that adjectives

alone (i.e., *sick, tired*) cannot represent a medical condition, but must occur together with another part of speech (i.e., *feeling sick, feeling tired*).

7. **Coordinating conjunction**: we aim to identify tuples of medical conditions based on coordinating conjunctions (e.g., *and, or*). We check whether coordinating conjunctions immediately follow a medical concept. If this scenario occurs, we consider that the conjunction is a signal for the beginning of another medical condition.

8. **Transition tokens**: we aim to identify the complete phrase for each medical condition. If the beginning of a medical condition was identified, we hypothesize that the medical condition cannot end unless a punctuation sign, a verb, or subordinating conjunction occurs.

9. **Concept start verb**: we hypothesize that a medical concept cannot begin with an active verb (i.e., verb in past tense, verb in 3rd person singular present, verb in past participle form, verb in base form).

3.2.2.2 Features set

The feature set used by $medNERR$ is inspired from the work of Halgrim et al.[20]

1. **Token value**: we include the value of the current token as a system feature.

2. **Distance from previous medication**: we compute the number of tokens between the current token and the previous medication concept. We use an upper threshold of 9 tokens if the previous medication is located more than 8 tokens away from current token.[20]

3. **Distance to next medication**: we compute the number of tokens between the current token and the next medication concept. We choose a standard distance of 9 tokens if the next medication is located more than 8 tokens away from current token.

4. **POS**: we include the part-of-speech tag of the current token. The POS tags are obtained from the preprocessing module.

5. **Lemmatized token**: we use the JWI Wordnet interface[29] in order to obtain the lemmatized form of the current token.

6. **Number check**: we verify whether the current token is a number (either integer or real).

7. **Token length**: we compute the character length of the current token. On average, the words in our corpus have a length of 4 characters. Based on empirical observations we notice that the longest words inside our corpus have a total length of 12 or more characters (e.g., hypertension, postmenopausal). We consequently use a maximum threshold of 12 for all the token lengths.

8. **Token line index**: we compute the offset of the token within the given document line.

9. **Nearby prepositions**: we check whether the token is preceded by a preposition.

10. **Time word**: we check whether the token contains any of the tokens *hour, week, day* or *year*.

11. **Spelled-out digit**: we check whether the current token is the English spelling of a digit.

12. **Is adjective**: indicates whether the current token is an adjective according to the POS tag. This is a binary feature that complements the POS feature.

3.3 Relation extraction

We hypothesize that the context of the *administered-for* relation has a systematic structure. Consequently, our system does not aim to classify pairs of medications – medical conditions as being in a relation, but instead focuses on identifying relation structures within a given document. We define the tokens between a medication and a medical condition that are involved in an administered-for relation as the relation structure (i.e., context). Even though most common solutions to the relation extraction task are pairwise classification approaches, we model our solution as a sequence labeling approach. We use a generative model in order to identify the relation context within a given sequence of tokens.

We design two scenarios to the relation context extraction task.

Scenario I ($medNERR_{relation}$): we assume that the relation context is a function of both the medical condition and the medication concepts. Given the medication and medical condition concepts as input, our system is required to identify the presence of a relation context with respect to the two concept types. This approach is described in detail in section 3.3.1.

Scenario II ($medNERR_{reason}$): we hypothesize that the medical condition concepts and the relation context are strongly correlated, and it would help with system performance if the medical condition concepts and the relation context are jointly predicted. Our system receives as input only the medication concepts, and is required to identify both the medical condition concepts and the presence of a relation context. This approach is described in section 3.3.2.

For both scenarios, we assume that there exists a relation context (i.e., at least one token) between the medication and the medical conditions involved in an *administered-for* relation. Our solution would fail to identify pairs of medication-medical conditions involved in an *administered-for* relation that are located adjacent to each other. For example, in the sentence "headache: aspirin, given to relieve", the medical condition "headache" is involved in an *administered-for* relation with the medication "aspirin", but our solution would not be able to correctly link the two concepts since there is no relation context between them.

3.3.1 Relation extraction with medical condition concepts

We use as a sequence labeler a HMM with conditional constraints. The algorithm used for token sequence generation is described in 3.3.1.1. Our system uses the CogComp implementation of the HMM, to which we added an additional set of features and an adjusted set of constraints. We further describe the set of constraints in section 3.3.1.2 and the set of features in section 3.3.1.3. Both the constraints and the features are computed at the token-level.

3.3.1.1 Token sequence generation

We adopt the IOB labeling scheme, where the B label represents the first word inside the relation context, the I label represents the remaining words inside the relation context, and the O label represents the words within the sequence of tokens located outside of a relation context. Our model takes as input a sequence of tokens and labels each token with an *I/O/B* label. We aim to label the entire corpus, and not just the sequences of tokens located between the medication and medical condition concepts. In order to generate the sequences of tokens to be labeled, we iterate over all the sentences in our corpus $\mathcal{C}$: if a relation context spans over k sentences, we then join the ordered set of k sentences into a single sequence of tokens, otherwise the sentence itself is included as a sequence of tokens. We join together specific sentences in order to avoid having a relation context span over two different token sequences. Given the sample sentences in 2-1, our algorithm would generate two sequences of tokens, as described in 3-4. The B label follows immediately after a medication or a medical condition concept, and the last label of a relation context (either I or B, depending on the length of the relation context) occurs before a medication or a medical concept.

Sequence 1 $\underbrace{\texttt{Patient}}_{O}\ \underbrace{\texttt{with}}_{O}\ \underbrace{\texttt{depression}}_{O}\ \underbrace{\texttt{and}}_{O}\ \underbrace{\texttt{chronic}}_{O}\ \underbrace{\texttt{kidney}}_{O}\ \underbrace{\texttt{disease}}_{O}$
$\underbrace{\texttt{is}}_{B}\ \underbrace{\texttt{considered}}_{I}\ \underbrace{\texttt{for}}_{I}\ \underbrace{\texttt{hemodialysis}}_{O}\texttt{;}$

Sequence 2 $\underbrace{\texttt{he}}_{O}\ \underbrace{\texttt{is}}_{O}\ \underbrace{\texttt{continued}}_{O}\ \underbrace{\texttt{on}}_{O}\ \underbrace{\texttt{Caltrate}}_{O}\ \underbrace{\texttt{plus}}_{O}\ \underbrace{\texttt{D}}_{O}\ \underbrace{\texttt{and}}_{O}$
$\underbrace{\texttt{multivitamin}}_{O}\ \underbrace{\texttt{and}}_{O}\ \underbrace{\texttt{was}}_{O}\ \underbrace{\texttt{started}}_{O}\ \underbrace{\texttt{on}}_{O}\ \underbrace{\texttt{advil}}_{O}\ \underbrace{\texttt{for}}_{B}\ \underbrace{\texttt{his}}_{O}\ \underbrace{\texttt{pain}}_{O}\ \texttt{.}$

Figure 3-4: An example of two sequences of tokens to be labeled. The sequences are generated based on the sample sentences in 2-1.

3.3.1.2 Constraints set

1. **Transition tokens**: we aim to identify the complete phrase for each relation context. We hypothesize that the relation context cannot end unless a medication or a medical condition concept is encountered inside the token sequence. If the relation context started before a medication then the transition can occur only when a medical condition concept is encountered; if the relation context started before a medical condition concept then the transition can occur only when a medication concept is encountered.

2. **End transition**: we check that the last label of the relation context is not the last token of the token sequence. This requirement is necessary because we need either a medication or a medical concept to be present immediately after the relation context ends.

3. **Concept start token**: we check that the first label within a relation context token sequence is a *B* label and that *I* labels do not occur without being preceded by a *B* label.

4. **Medical concept label**: we check that all medication and medical concepts tokens are labeled with an *O* label.

5. **Preposition signal**: we hypothesize that the occurrence of the preposition *with* in the proximity of a medication is an indicator for the beginning of a relation context.

6. **Medical condition present**: because the relation context requires the occurrence of both a medication and a medical condition concept, we check that the beginning of a relation context does not occur unless there is a medical condition concept in the proximity.

7. **PRN abbreviation**: we hypothesize that the abbreviation *PRN* (i.e., pro re nata / as needed) is a strong indicator for the beginning of a relation context. We thus require that the occurrence of the token *PRN* coincides with the beginning of a relation context.

8. **Relation patterns**: we notice that specific linguistic structures better capture the relation context. Table 3.3 presents the most common structures encountered in the training dataset. We use the occurrence of any of the structures presented in table 3.3 as a signal for the beginning of a relation context.

9. **Medical condition occurrence**: we hypothesize that most medical conditions mentioned inside a patient report are accompanied by a treatment specification. We use this hypothesis in the scenarios when a medical condition is not preceded by a medication; in this scenario we consider that the relation context begins at the token to the right of the medical condition concept.

10. **Punctuation separator**: we consider the occurrence of a punctuation token as the single token interposed between a medical concept and a medical condition concept as a signal for the occurrence of a relation context.

11. **Preposition proximal to medical condition**: we consider the scenario in which a preposition precedes a medical condition as a signal for the occurrence of a relation context.

Relation pattern
for relief of
for treatment of
for the treatment of
for the control of
indicated for
as treatment of
for prevention of

Table 3.3: Relation context patterns.

3.3.1.3 Features set

We use the same feature set as described in section 3.2.2 plus the additional feature:

1. **Medical condition concept**: we check whether the current token is part of a medical condition concept.

3.3.2 Relation extraction without medical condition concepts

We use a HMM with conditional constraints as a sequence labeler. The algorithm used for token sequence generation is described in section 3.3.2.1. Our system uses the CogComp implementation of the HMM supplemented with an adjusted set of constraints and an additional set of features. We further describe the set of constraints in section 3.3.2.2 and the set of features in section 3.3.2.3. Both the constraints and the features are computed at the token-level.

3.3.2.1 Token sequence generation

We adopt an adjusted variant of the IOB labeling scheme. We use a B_R label to represent the first word inside the relation context, a I_R label to represent the remaining words inside the relation context, a B_{MC} label to represent the first word inside a medical condition concept, a I_{MC} label to represent the remaining words inside a medical condition concept, and the O label to represent the words within the sequence of tokens located outside both of a relation context and a medical condition concept. Our model takes as input a sequence of tokens and labels each token with an $I_R, I_{MC}, O, B_R, B_{MC}$ label. In order to generate the sequences of tokens to be labeled, we iterate over all the sentences in our corpus $\mathcal{C}$: if a relation context spans over k sentences, we then join the ordered set of k sentences into a single sequence of tokens, otherwise the sentence itself is included as a sequence of tokens. We join together specific sentences in order to avoid having a relation context span over two different token sequences. Given the sample sentences in 2-1, our algorithm would generate two sequences of tokens, as described in 3-5. The B_R label follows immediately after a medication or a medical condition concept, and the last label of a relation context (either I_R or B_R, depending on the length of the relation context) occurs before a medication or a medical concept.

3.3.2.2 Constraints set

The $medNERR_{reason}$ system combines the set of constraints used by the $medNERR_{medCond}$ system (see 3.2.2) with a set of two additional constraints. The constraints inherited from

Sequence 1 $\underbrace{\texttt{Patient}}_{O}\ \underbrace{\texttt{with}}_{O}\ \underbrace{\texttt{depression}}_{B_{MC}}\ \underbrace{\texttt{and}}_{O}\ \underbrace{\texttt{chronic}}_{B_{MC}}\ \underbrace{\texttt{kidney}}_{I_{MC}}\ \underbrace{\texttt{disease}}_{I_{MC}}$
$\underbrace{\texttt{is}}_{B}\ \underbrace{\texttt{considered}}_{I}\ \underbrace{\texttt{for}}_{I}\ \underbrace{\texttt{hemodialysis;}}_{O}$

Sequence 2 $\underbrace{\texttt{he}}_{O}\ \underbrace{\texttt{is}}_{O}\ \underbrace{\texttt{continued}}_{O}\ \underbrace{\texttt{on}}_{O}\ \underbrace{\texttt{Caltrate}}_{O}\ \underbrace{\texttt{plus}}_{O}\ \underbrace{\texttt{D}}_{O}\ \underbrace{\texttt{and}}_{O}$
$\underbrace{\texttt{multivitamin}}_{O}\ \underbrace{\texttt{and}}_{O}\ \underbrace{\texttt{was}}_{O}\ \underbrace{\texttt{started}}_{O}\ \underbrace{\texttt{on}}_{O}\ \underbrace{\texttt{advil}}_{O}\ \underbrace{\texttt{for}}_{B}\ \underbrace{\texttt{his}}_{B_{MC}}\ \underbrace{\texttt{pain}}_{B_{MC}}\ \texttt{.}$

Figure 3-5: An example of two sequences of tokens to be labeled. The sequences are generated based on the sample sentences in 2-1

the $medNERR_{medCond}$ system help with the correct identification of the medical condition concepts, meanwhile the two additional constraints are targeted at relation context extraction. The constraints targeted at relation context extraction are:

1. **Relation patterns**: we notice that specific linguistic structures better capture the relation context (see Table 3.3). We use the occurrence of any of those structures as a signal for the beginning of a relation context.

2. **Preposition signal**: we hypothesize that the occurrence of the preposition *"with"* in the proximity of a medication is an indicator for the beginning of a relation context.

3.3.2.3 Features set

The $medNERR_{reason}$ system uses only two features:

1. **Token value**: we include the value of the current token as a system feature.

2. **Medication concept**: we check whether the current token is part of a medication concept.

Chapter 4

Evaluation

4.1 Evaluation metrics

4.1.1 Concept extraction

We evaluate the concept extraction task using the precision (P), recall (R), and F-measure (F) metrics. We refer to the concepts identified by the system as the *system concepts*. The metrics evaluate the system performance in terms of:

1. **true positives** (tp): total count of system concepts that match the gold standard concepts

2. **false negatives** (fn): total count of gold standard concepts that are not identified by the system

3. **false positives**(fp): total count of system concepts that do not match the gold standard concepts

The P and R measures are defined as :

$$P = \frac{tp}{tp + fp} \tag{4.1}$$

$$R = \frac{tp}{tp + fn} \tag{4.2}$$

while the F-measure is the harmonic mean of the P and R:

$$F = \frac{2 * P * R}{P + R} \tag{4.3}$$

We consider two scenarios for computing the match between a given system concept and a gold standard concept:

- **exact match**: the ordered tokens set of the system concept is identical to the ordered tokens set of the gold standard concept and the system concept is located at the same position within the medical record as the gold standard concept.

- **inexact match**: the ordered tokens set of the system concept overlaps partially or totally the ordered tokens set of the gold standard concept. The system concept contains at least one token located at the same position as one of the gold standard tokens.

We compute the P, R, and F metrics under the two above-mentioned scenarios.

4.1.2 Relation extraction

The metrics used for relation extraction are identical to the evaluation metrics used during the i2b2 medication extraction challenge, as detailed by Uzuner et al.[54] The i2b2 metrics evaluate the system performance on reason extraction, where the reasons are the medical condition concepts linked to a medication. The evaluation performed by the i2b2 system for the reason concepts is equivalent with an evaluation on the *administered-for* relation extraction. The i2b2 metrics report system performance in terms of the P, R, and F measures but they compute those measures at the reason and reason-token level. The reason and reason-token P, R, and F measures are defined as:

$$P_{reason} = \frac{\textit{count reasons correctly extracted by the system}}{\textit{count reasons extracted by the system}} \tag{4.4}$$

$$R_{reason} = \frac{\textit{count reasons correctly extracted by the system}}{\textit{count reasons in the gold standard}} \tag{4.5}$$

$$F_{reason} = \frac{2 * P_{reason} * R_{reason}}{P_{reason} + R_{reason}} \tag{4.6}$$

$$P_{reason-token} = \frac{\textit{count correctly extracted tokens from each reason in the system output}}{\textit{count reason tokens in the system output}} \tag{4.7}$$

$$R_{reason-token} = \frac{\textit{count correctly extracted tokens from each reason in the system output}}{\textit{count reason tokens in the gold standard}} \tag{4.8}$$

$$F_{reason-token} = \frac{2 * P_{reason-token} * R_{reason-token}}{P_{reason-token} + R_{reason-token}} \tag{4.9}$$

4.1.3 Relation extraction statistical significance

In order to determine whether our relation extraction system is statistically significant from other relation extraction systems, we use approximate randomization tests.[37] Given two relation extraction systems S_1 and S_2, we record their predictions on the same test set as the $Pred_{s_1}$ and $Pred_{s_2}$, respectively. We evaluate each of the system predictions against the gold standard, and compute $D_{s_1-s_2}$ as the difference between the F_{reason} score obtained on the $Pred_{s_1}$ set and the F_{reason} obtained on the $Pred_{s_1}$ set. We then generate pseudo-outputs by swapping at 0.5 probability elements between the $Pred_{s_1}$ and $Pred_{s_2}$ set. We evaluate the new pseudo-prediction sets against the gold standard and compute $D_{s_1-s_2}^{pseudo}$ as the difference between the F_{reason} obtained on each pseudo-prediction set. If $D_{s_1-s_2}^{Pseudo} \geq D_{s_1-s_2}$ we increase a counter i by an order of one. We repeat the generation of pseudo-prediction sets and their evaluation N times, where $N = 10,000$ times. We compute the final p-value as $p = (i+1)/(N+1)$. If the p-value is smaller than $\alpha = 0.05$, we conclude that the difference in performance between the two systems is statistically significant.

We apply a conservative statistical correction (the Bonferroni correction[33]) to adjust for multiple comparisons. The Bonferroni corrected α was set to 0.016 for three comparisons, obtained by dividing the initial α value by the number of comparisons.

4.2 Experiments

4.2.1 Concept extraction

4.2.1.1 Medication extraction

We design a series of experiments in order to evaluate how individual components of the medication extraction system contributed to the final system's performance. The experiments are as follows:

Experiment 1 ($medNERR_{med}$): we evaluate the performance of the entire medication extraction system.

Experiment 2 ($medNERR_{med}^{greedy}$): we evaluate the performance of the medication extraction system containing only the greedy module.

Experiment 3 ($medNERR_{med}^{greedy+stingy}$): we evaluate the performance of the medication extraction system containing the greedy and stingy modules.

Experiment 4 ($medNERR_{med}^{greedy+neighbor}$): we evaluate the performance of the medication extraction system containing the greedy and neighbor modules.

4.2.1.2 Medical condition extraction

We design a series of experiments in order to evaluate the medical condition extraction system. We use the term $medNERR_{medCond}$ to refer to the complete medical condition extraction system, as described in section 3.2.2. We evaluate:

1. how the constraints contribute to system's performance

2. how the system performance changes when gold standard(gs) medications are used to generate the feature set versus when the system medications are used.

The experiments are as follows:

Experiment 1 (*no constraints+system concepts*): we evaluate the performance of the medical condition extraction system without the set of constraints. The feature set is generated using the system concepts.

Experiment 2 (*constraints+system concepts*): we evaluate the performance of the medical condition extraction system with the set of constraints. The feature set is generated using the system concepts.

Experiment 3 (*no constraints+gs concepts*): we evaluate the performance of the medical condition extraction system without the set of constraints. The feature set is generated using the gold standard concepts.

Experiment 4 (*constraints+gs concepts*): we evaluate the performance of the medical condition extraction system with the set of constraints. The feature set is generated using the gold standard concepts.

4.2.2 Relation extraction

We design a series of experiments in order to evaluate the relation extraction system. We apply the experimental settings to both the $medNERR_{relation}$ relation extraction system and to $medNERR_{reason}$ relation extraction systems. The two relation extraction systems are described in section 3.3.1 and section 3.3.2, respectively. We investigate:

1. how the constraints helped with relation extraction
2. how the performance of the concept extraction system influenced the performance of the relation extraction system. We specifically evaluate the scenarios when the medication and medical condition concepts are generated by *medNERR* and the scenario when the medications and the medical conditions are provided as gold standard.

The experiments are as follows:

Experiment 1 (*no constraints+system concepts*): we evaluate the performance of the relation extraction system without the set of constraints. The feature set is generated using the system medications and the system medical conditions.

Experiment 2 (*constraints+system concepts*): we evaluate the performance of the relation extraction system with the set of constraints. The feature set is generated using the system medications and the system medical conditions.

Experiment 3 (*no constraints+gs concepts*): we evaluate the performance of the relation extraction system without the set of constraints. The feature set is generated using the gold standard medications and the gold standard medical conditions.

Experiment 4 (*constraints+gs concepts*): we evaluate the performance of the relation extraction system with the set of constraints. The feature set is generated using the gold standard medications and the gold standard medical conditions.

4.2.3 Relation extraction statistical significance

We compare the relation extraction system against the best performing system from the 2009 i2b2 challenge, as well as against a relation extraction system based on a cascaded classification approach developed by Scott et al.[20]

Chapter 5

Results and discussion

In general, *medNERR* displayed better performance results on the development data compared to the test data. We attribute this behavior to the fact that the training and development data were annotated by one human annotator, while the test data were annotated by two independent annotators and one arbitrator.

In the remaining of this chapter we only report system results on the test data, but the system results on the development data are included for reference in Appendix B.

5.1 Medication extraction

5.1.1 Results

Table 5.1 presents the results of the medication extraction system across all 4 experimental settings on the test data. System performance varies across the match type, with better performance for inexact match: .902 F-measure for inexact match vs. .837 F-measure for exact match on test data. The best performance on medication extraction comes from the complete *medNERR* system. Each of the *medNERR* modules perform worse than *medNERR*. The best *medNERR* module is the $medNERR_{med}^{greedy+stingy}$ with a .882 F-measure for inexact match and .803 F-measure for exact match on test data. The second best *medNERR* module is the $medNERR_{med}^{greedy+neighbor}$ module, with a performance of .804 F-measure for inexact match and .734 F-measure for exact match. The $medNERR_{med}^{greedy}$ module performs worst overall (.756 F-measure for inexact match and .685 F-measure for exact match), but has the best recall out of all the modules (.962 recall for inexact match and .871 recall for exact match). Our medication extraction system does not manage to outperform the best medication extraction system from the i2b2 challenge or the medication system of Scott et al.

	Inexact match			Exact match		
System	P	R	F	P	R	F
$medNERR_{med}$	0.917	0.886	**0.902**	0.813	0.863	**0.837**
$medNERR_{med}^{greedy}$	0.622	0.962	0.756	0.536	0.871	0.685
$medNERR_{med}^{greedy+stingy}$	0.859	0.905	0.882	0.783	0.824	0.803
$medNERR_{med}^{greedy+neighbor}$	0.701	0.944	0.804	0.640	0.862	0.734
Best i2b2 system	-	-	-	-	-	0.884
Scott et al.	-	-	-	0.926	0.871	0.898

Table 5.1: Medication extraction- system performance results on test data. Bolded results represent the best system F-measure under given match type.

5.1.2 Discussion

The performance results of the medication extraction system show that simple mapping to a dictionary of medication names is not a sufficient solution for a well-performing system. Additional system extensions like context analysis and incorporation of world knowledge are needed in order to reduce the rate of false positives. Each of the $medNERR$ medication extraction modules contributed to performance improvements. The $medNERR_{med}^{greedy+stingy}$ module helped with the removal of falsely identified medication names. Yet, it eliminated a fraction of the correctly identified medication names: for inexact match it contributed a 7% improvement in terms of precision, but caused a 2% drop in recall, while for exact match it contributed a 6.4% improvement in precision but caused a .9% drop in recall. The $medNERR_{med}^{greedy+neighbor}$ module helped improve the system recall rate while maintaining the same precision level: the module contributed with a .5% improvement in recall for inexact match and a 1% improvement in recall for exact match.

In general, the $medNERR_{med}^{greedy}$ module failed to correctly identify new medications not yet incorporated into UMLS (e.g., “avadia”, “celondin”, “clonopin”, “gentamycin”) and abbreviated medication names (e.g., “abx”,“bp meds”, “ctx”, “cvp”). It incorrectly predicted as medications food-related words (e.g.,“alcohol”, “ethoh”, “coffee”,“grapefruit”), lab test names (e.g., “albumin”, “amylase”, “calcium”, “cholesterol”, “glucose”), some abbreviations (e.g., “appt”, “bid”, “bun”), and other miscellaneous words (e.g., “co2”, “o2”, “oxygen”, “component”, “drug”, “duration”, “fluids”). Because of the filtering rules, the $medNERR_{med}^{greedy+stingy}$ module eliminated a fraction of the correctly identified medications (e.g., “cefuroxime”, “dobutamine”, “labetalol”); at the same time it removed incorrect medication names like food (e.g., “etoh”, “alcohol”, “grapefruit”), lab test names (e.g., “albumin”, “alkaline phosphate”, “cholesterol”), incorrect abbreviations classified as medications (e.g., “appt”, “bid”, “bun”). The $medNERR_{med}^{greedy+neighbor}$ module helped with the elimination of false medication names originating from generic phrases like “all medications”, “clot”, “color”, “medications”, “drug”. On the downside, it removed some correct medication names that had a more generic form (e.g., “cardiovascular medication”, “diabetes medications”, “sedating medications”). The final medication extraction system inherited the missed medication names from $medNERR_{med}^{greedy}$; additional missed medication included generic medications (e.g., “diabetes medication”, “medications”) as well as medication abbreviations not contained in the list of medication abbreviations (e.g., “nph”, “ntg”). $medNERR_{med}$ incorrectly identified as medication names some lab result phrases (e.g., “calcium”, “iron”, “lithium level”), some patient attributes (e.g., “pain-free”, “non-insulin-dependent”,“heparinized”), and miscellaneous medical terms (e.g., “o2”, “fluids”, “oxygen”).

5.2 Medical condition extraction

5.2.1 Results

Table 5.2 describes the results of the medical condition extraction system. The system had a better performance when provided with gold standard medication concepts, compared to when it had to predict the concepts itself. For example, $medNERR_{medCond}$ showed an F-measure of .317 for inexact match when using system medications and an F-measure of .359 for inexact match when using gold standard medications. Our system performed best when it made use of the conditional constraints, regardless of the type of medication concepts (i.e., system medications or gold standard medications) it was provided with. When provided with system medications, the performance dropped from .317 F-measure for inexact match with conditional constraints to .123 F-measure for inexact match without conditional constraints. A similar change in performance was noticed when gold standard medications were provided: the system F-measure dropped from .359 for inexact match with conditional constraints to .127 for inexact match without conditional constraints. The system performance varied based on the match type, with inexact match resulting in better performance results. When system medications and conditional constraints were used, the system evaluated at .317 for inexact match and .265 for exact match, while when gold standard medications and conditional constraints were used, the system evaluated at .359 for inexact match and .301 for exact match.

		Inexact match			Exact match		
Concepts	Constraints	P	R	F	P	R	F
GS	No	0.067	0.854	0.125	0.055	0.697	0.102
GS	Yes	0.233	0.790	**0.359**	0.195	0.661	**0.301**
System	No	0.066	0.847	0.123	0.054	0.692	0.100
System	Yes	0.201	0.750	0.317	0.168	0.628	0.265

Table 5.2: Medical condition extraction- system performance results on test data. Bolded results represent the best system F-measure under given match type.

5.2.2 Discussion

The medical condition extraction system does not perform as well as the medication extraction system. This behavior is mainly attributed to the subjectivity and complexity of the medical condition concept type. When compared to the medication concepts, the medical condition concepts are less structural and systematic, and can include complex linguistic phrases (e.g., “feeling under the weather”). Another reason for this behavior is the fact that the gold standard for the medical conditions contained only the medical conditions that were involved in an *administered-for* relation. Within each medical document there were additional medical conditions not involved in such relations that our system identified. Those additional medical conditions were treated as false positives at evaluation time, thus producing small precision results.

Our system showed good results in terms of recall (e.g., .750 recall for inexact match using system concepts and conditional constraints and .628 recall for exact match using system concepts and conditional constraints). As shown in table 5.2, the conditional constraints helped with the overall system performance, contributing on average to a 20% increase in F-measure regardless of the type of medications (i.e., system medications or gold standard medications) the system was provided with. In general, the conditional constraints reduced the number of false positives (e.g., for inexact match evaluation, precision increase from .066 with system concepts and no conditional constraints to 0.201 with system concepts and conditional constraints). The medical condition extraction system did not show too much variation in performance when provided with different sets of medication concepts. The gold standard medication set contributed to a 4% performance increase when compared to the system medication set under the conditional constraints setting, but did not help boost performance when no constraints were used by the relation extraction system.

When $medNERR_{medCond}$ was provided with conditional constraints, it usually missed fraction of the medications that were correctly predicted by $medNERR_{medCond}$ without constraints. For example, $medNERR_{medCond}$ with conditional constraints would miss medications that contained only adjectives (e.g., "afib", "hypertensive") or had a long structure (e.g., "mild concentric left ventricular hypertrophy", "increased shortness of breath"). Yet, $medNERR_{medCond}$ with constraints would correctly predict medical conditions that were rarely encountered inside the training and test set (e.g., "septicemia", "acidosis", "bradyarrhythmia", "costochondritis", "gaseousness","heparin-induced thrombocytopenia"). Since none of the medical conditions predicted by $medNERR_{medCond}$ with constraints and not by $medNERR_{medCond}$ without constraints were actually encountered in the training set, we conclude that the conditional constraints actually helped the medical condition extraction system correctly identify correct concepts based on context alone. The constraints also helped reduce the number of falsely identified medication concepts; the presence of constraints helped remove incorrect concepts that related to patient vitals (e.g., "a blood pressure", "a heart rate", "a glucose level") or patient treatment (e.g., "antibiotic regimen", "stool softener"). The constraints did help identify additional disease names (e.g., "anemia", "angina", "bipolar disorder"), but because those additional disease names were not included in the gold standard our evaluation treated them as falsely identified medical condition concepts.

5.3 Relation extraction

5.3.1 Results

Table 5.3 presents the performance of the relation extraction system. Performance results varied based on the concept type, inclusion of constraints, and match type. The $medNERR_{reason}$ system performed better than the $medNERR_{relation}$ system when provided with system concepts: $medNERR_{reason}$ evaluated at .606 F-measure for inexact match with constraints and .527 F-measure for inexact match without constraints, while $medNERR_{relation}$ evaluated at .333 for inexact match with constraints and .266 for inexact match without constraints. When provided with gold standard concepts, the $medNERR_{relation}$ system performed better, with a .758 F-measure for inexact match with constraints, while $medNERR_{reason}$ obtained a 0.576 for inexact match with constraints. The constraints helped improve system performance, for both the exact and inexact match evaluation settings. The $medNERR_{reason}$ obtained a .606 F-measure for inexact match with constraints and .527 F-measure for inexact match without constraints, and a .610 F-measure for exact match with constraints and .530 for exact match without constraints. $medNERR_{reason}$ outperformed the best i2b2 system and the Scott et al. system both when provided with gold standard and with the system concepts. The best i2b2 system evaluated at .457 F-measure for exact match, the Scott et al. system evaluated at .471 F-measure for exact match, while our system evaluated at .610 F-measure for exact match with system concepts and included constraints.

			Inexact match			Exact match		
System Name	Concepts	Constraints	P	R	F	P	R	F
Test								
$medNERR_{relation}$	GS	No	0.827	0.584	0.685	0.816	0.590	0.685
$medNERR_{relation}$	GS	Yes	0.927	0.672	**0.758**	0.928	0.679	**0.767**
$medNERR_{reason}$	GS	No	0.786	0.455	0.576	0.782	0.486	0.584
$medNERR_{reason}$	GS	Yes	0.836	0.547	0.661	0.833	0.559	0.669
$medNERR_{relation}$	System	No	0.210	0.374	0.266	0.186	0.310	0.233
$medNERR_{relation}$	System	Yes	0.297	0.379	0.333	0.281	0.349	0.311
$medNERR_{reason}$	System	No	0.746	0.407	0.527	0.737	0.414	0.530
$medNERR_{reason}$	System	Yes	0.795	0.489	**0.606**	0.748	0.515	**0.610**
Best i2b2 system	System	No	0.483	0.481	0.482	0.487	0.430	0.457*
Scott et al.	System	No	-	-	-	0.734	0.347	0.471*

Table 5.3: Relation extraction- system performance results on test data. Bolded results represent the best system F-measure under given match type and concepts type. Starred F-measure results are results statistically significant from $medNERR$ at Bonferroni corrected $\alpha = 0.016$.

5.3.2 Discussion

The $medNERR_{relation}$ and the $medNERR_{reason}$ systems specialize in different aspects of the relation extraction task. The $medNERR_{relation}$ system is specialized in identifying the presence of a relation between given medical concepts, and performs best when provided with perfect input (i.e., gold standard concepts). On the other hand, the $medNERR_{reason}$ system is able to determine both the presence of a relation and identify the medical condition concept to which a medication is related to. The $medNERR_{reason}$ system is not as sensitive to the concepts set given as input; its F-measure drops by 6% when provided with system concepts versus when provided with gold standard concepts. In contrast, the $medNERR_{relation}$ system is more sensitive to the input concept set, and drops in F-measure performance by 40% when provided with system concepts versus when provided with gold standard concepts. When evaluated as an end-to-end system (i.e., using only system concepts), $medNERR_{reason}$ performs best, with a .606 F-measure on inexact match and .610 F-measure for exact match. For both $medNERR_{relation}$ and $medNERR_{reason}$ the constraints set help improve performance. The inclusion of constraints benefits both with the reductions of false positive and with the identification of additional true positives.

For $medNERR_{reason}$, the constraints contribute to a 5% increase in precision and a 8% increase in recall when provided with system concepts and evaluated in the inexact match setting. For $medNERR_{relation}$, the constraints contribute to a 3% increase in precision and a 10% increase in recall when provided with system concepts and evaluated in the inexact match setting. Our system is able to outperform the state of the art in relation extraction by a 13% margin. It also outperforms the Scott et al. system that approaches relation extraction as a classification task.

In general, $medNERR_{relation}$ overly predicted the *administered-for* relations when not provided with conditional constraints. $medNERR_{relation}$ without constraints would incorrectly link medications to medical conditions based on their proximity within the document (e.g.,"aspirin"–"pain", "aspiring"–"headache", "colace"–"pain", "diuretics"-"worsening cardiac function"). It would also miss some of the correct relations (e.g., "tylenol"–"pain", "robitussin"–"cough") that were correctly predicted by $medNERR_{relation}$ with constraints. For $medNERR_{reason}$, the constraints helped reduce the rate of falsely identified medications (e.g., "clonazepam"–"esophagitis", "keflex"–"elevated bun/cr") and also helped identify additional relations (e.g., "steroids"–"arthralgias secondary"). When compared to $medNERR_{relation}$, $medNERR_{reason}$ was able to better predict relation like "flexeril"–"chronic low back pain", "epinephrine"–"proper pressure". The advantage of $medNERR_{reason}$ consisted in better context prediction (e.g., "epinephrine **to maintain** proper pressure", "the patient recently started Flexeril **to treat** chronic low back pain"). Both $medNERR_{reason}$ and $medNERR_{relation}$ managed to outperform the best i2b2 system and the system of Scott et al. due to their ability to link medications to medical conditions based on context alone.

Chapter 6

Future work

Even though our research helped improve performance on relation extraction for medical documents, there are still a series of improvements and taks that can be explored.

1. As mentioned in the discussion section, our $medNERR_{relation}$ system performs worse than the $medNERR_{reason}$ system as it relies on a clean set of concepts. In order to improve the performance of the $medNERR_{relation}$ system, we would need to improve on the medical concept extraction system. An additional set of hand-built constraints could be built to limit the set of possible medical conditions to the medical conditions that are involved in an *administered-for* relation.

2. Another future research step would be the reduction of the dependency on annotated data for the medical condition and relation extraction systems. Chang et al. have shown that the conditional constraints coupled with semi-supervised probabilistic models can perform equally well to the supervised probabilistic models. It would be beneficial to explore whether similar observations hold on the medical domain.

3. Our relation extraction solution assumes that the relation context occurs between the medication and medical condition concepts. We disregard the cases where the relation context is located outside the two concepts. Future work should investigate the performance of an adjusted solution that considers both the inside (between the medication and medical condition concepts) and the outside (outside the medication and the medical condition context) relation context.

4. We have proposed a different approach to the relation extraction task, and testing this approach on different corpora would help with its validation.

Chapter 7

Conclusion

In this thesis we presented a new approach to relation extraction from medical documents. We have shown the value of performing relation extraction as a sequence labeling task in contrast to a classification task. We have also explored for the first time the value of using conditional constraints to guide learning in a supervised probabilistic framework for the medical domain. Our relation extraction task significantly outperformed the state of the art, as a result of both the conditional constraints and the sequence labeling approach.

Appendix A

Medical record section header names

Document information	record
	admission date
	report status
	discharge date
	identification
Diagnoses	admit diagnosis
	principal discharge diagnosis
	principal diagnosis
	principal diagnoses
	secondary diagnosis
	secondary diagnoses
	discharge diagnosis
	chief complaint
	other medical issues considered at this time
	other diagnosis
	other diagnoses
	diagnoses
Allergies	allergies
	allergy

	potentially serious interaction possible allergy
History	brief history of physical illness history of present illness history of the present illness past medical history past surgical history social history social sh family history
Exams	review of systems admission exam physical examination on admission physical examination PE on admit PE on discharge assessment and plan plan physical examination on arrival consults
Procedures and labs	hospital course hospital course by problem hospital course by system course operations and procedures laboratory data labs laboratory data on admission

	studies admission labs studies at the time of adminission laboratory data on discharge data heme
Medications	medications rx on admit home meds meds current medications discharge medications medications on discharge drug history medications at rehab
Discharge	discharge condition disposition advanced directives conditions on discharge follow-up follow up follow up appointment medications on admission medications on discharge
Other	diet all pmh hpi medical service

	impression summary by system problem summary heent assessment
End of medical record headers	additional comments cc to-do list for pcp dictated by attending entered by

Table A.1: Medical record section header names

Appendix B

System results

	Inexact match			Exact match		
System	P	R	F	P	R	F
$medNERR_{med}$	0.899	0.941	**0.919**	0.870	0.910	**0.890**
$medNERR_{med}^{greedy}$	0.791	0.963	0.869	0.764	0.930	0.839
$medNERR_{med}^{greedy+stingy}$	0.848	0.947	0.895	0.818	0.914	0.863
$medNERR_{med}^{greedy+neighbor}$	0.856	0.956	0.903	0.829	0.926	0.875

Table B.1: Medication extraction- system performance results on development data. Bolded results represent the best system F-measure under given match type

		Inexact match			Exact match		
Concepts	Constraints	P	R	F	P	R	F
GS	No	0.076	0.817	0.139	0.062	0.668	0.114
GS	Yes	0.298	0.836	**0.439**	0.245	0.687	**0.361**
System	No	0.076	0.821	0.138	0.061	0.668	0.112
System	Yes	0.248	0.798	0.378	0.202	0.649	0.307

Table B.2: Medical condition extraction- system performance results on development data. Bolded results represent the best system F-measure under given match type

			Inexact match			Exact match		
System Name	Concepts	Constraints	P	R	F	P	R	F
$medNERR_{relation}$	GS	No	0.809	0.626	0.706	0.787	0.622	0.694
$medNERR_{relation}$	GS	Yes	0.903	0.661	**0.763**	0.917	0.680	**0.781**
$medNERR_{reason}$	GS	No	0.731	0.440	0.549	0.710	0.436	0.540
$medNERR_{reason}$	GS	Yes	0.798	0.551	0.652	0.774	0.553	0.645
$medNERR_{relation}$	System	No	0.270	0.422	0.330	0.235	0.371	0.288
$medNERR_{relation}$	System	Yes	0.355	0.433	0.390	0.341	0.412	0.373
$medNERR_{reason}$	System	No	0.711	0.412	0.522	0.674	0.398	0.501
$medNERR_{reason}$	System	Yes	0.745	0.508	**0.604**	0.714	0.498	**0.587**

Table B.3: Relation extraction- system performance results on development data. Bolded results represent the best system F-measure under given match type and concepts type

Bibliography

[1] *Message understanding conference - 6: A brief history*, 1996.

[2] AACM, 2012.

[3] Eugene Agichtein and Luis Gravano. Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, DL '00, pages 85–94, New York, NY, USA, 2000. ACM.

[4] A. R. Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pages 17–21, 2001.

[5] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 8–15, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[6] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pages 194–201, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[7] Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(Database issue):D267–D270, January 2004.

[8] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *In Proceedings of the Seventh Message Understanding Conference (MUC-7*, 1998.

[9] Eugene M. Breydo, Julia T. Chu, and Alexander Turchin. Identification of inactive medications in narrative medical text. In *AMIA Annual Symposium Proceedings*, pages 66–70, 2008.

[10] Sergey Brin. Extracting patterns and relations from the world wide web. In *Selected papers from the International Workshop on The World Wide Web and Databases*, WebDB '98, pages 172–183, London, UK, UK, 1999. Springer-Verlag.

[11] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[12] Matthew M. Burton, Linas Simonaitis, and Gunther Schadow. Medication and indication linkage: A practical therapy for the problem list? In *AMIA Annual Symposium Proceedings*, pages 86–90, 2008.

[13] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Structured learning with constrained conditional models. *Machine Learning*, 88(3):399–431, September 2012.

[14] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, December 2008.

[15] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134, June 2005.

[16] Michael Fleischman. Automated Subcategorization of Named Entities. In *ACL (Companion Volume)*, pages 25–30, 2001.

[17] Michael Fleischman and Eduard Hovy. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[18] C. Friedman, O. P. Alderson, H. J. Austin, J. J. Cimino, and B. S. Johnson. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174, March 1994.

[19] S. Geetha, G.S. Anandha Mala, and N. Kanya. A survey on informaton extraction using entity relation based methods. In *Sustainable Energy and Intelligent Systems (SEISCON 2011), International Conference on*, pages 882 –885, july 2011.

[20] SR Halgrim, F Xia, I Solti, E Cadag, and O Uzuner. A cascade of classifiers for extracting medication information from discharge summaries. *Journal of Biomedical Semantics*, 2(3), July 2011.

[21] V. Jagannathan, J. C. Mullett, G. J. Arbogast, A. K. Halbritter, D. Yellapragada, S. Regulapati, and P. Bandaru. Assessment of commercial nlp engines for medication information extraction from dictated clinical notes. *Journal of American Medical Informatics Association*, 78(4):284–291, 2009.

[22] Peter B. Jensen, Lars J. Jensen, and Soren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nat Rev Genet*, 13(6):395–405, June 2012.

[23] M. N. Jones and D. J. K. Mewhort. *Psychological Review*, 114:1–37, 2007.

[24] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ACLdemo '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[25] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, 2003.

[26] Seungwoo Lee and Gary Geunbae Lee. Heuristic methods for reducing errors of geographic named entities learned by bootstrapping. In *Proceedings of the Second international joint conference on Natural Language Processing*, IJCNLP'05, pages 658–669, Berlin, Heidelberg, 2005. Springer-Verlag.

[27] MA Levin, M Krol, AM Doshi, and Reich DL. Extraction and mapping of drug names from free text to a standardized nomenclature. In *AMIA Annual Symposium Proceedings*, pages 438–442, 2007.

[28] Z Li, Y Cao, and L Antieau. Extracting medication information from patient discharge summaries. In *Proceedings of the Third i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, 2009.

[29] Mark Finlayson, 2012.

[30] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[31] Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 491–498, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[32] SM Meystre, J Thibault, and S Shen. Description of the textractor system for medications and reason for their prescription extraction from clinical narrative text documents. In *Proceedings of the Third i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, 2009.

[33] Rupert G. Miller. *Simultaneous statistical inference*. McGraw-Hill New York,, 1966.

[34] Robert Munro and Christopher D. Manning. Accurate Unsupervised Joint Named-Entity Extraction from Unaligned Parallel Text. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, Jeju, Korea, July 2012. Association for Computational Linguistics.

[35] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[36] David Nadeau, Peter Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity advances in artificial intelligence. *Advances in Artificial Intelligence*, 4013:266–277, 2006.

[37] Eric W. Noreen. *Computer-Intensive Methods for Testing Hypotheses : An Introduction*. Wiley-Interscience, April 1989.

[38] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The genia corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 82–86, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[39] Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, AAAI'06, pages 1400–1405. AAAI Press, 2006.

[40] John Patrick and Lin Min. A cascade approach to extract medication event (i2b2 challenge 2009). In *Proceedings of the Third i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, 2009.

[41] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[42] Thomas C. Rindflesch, Lorraine Tanabe, John N. Weinstein, and Lawrence Hunter. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium in Biocomputing*, pages 517–528, 2000.

[43] Rachel E. O. Roxas, editor. *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation, PACLIC 22, Cebu City, Philippines, November 20-22, 2008*. De La Salle University (DLSU), Manila, Philippines, 2008.

[44] 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. *Journal of the American Medical Informatics Association : JAMIA*, 17(5):507–513, September 2010.

[45] Satoshi Sekine. Nyu: Description of the japanese ne system used for met-2. In *Proc. of the Seventh Message Understanding Conference (MUC-7*, 1998.

[46] Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine - Volume 13*, BioMed '03, pages 49–56, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[47] L Simon, M Wei, M Robin, G Vikraman, and N Stuart. Rxnorm: prescription for electronic drug information exchange, 2005.

[48] E. Sirohi and P. Peissig. Study of effect of drug lexicons on medication extraction from electronic medical records. In Russ B. Altman, Tiffany A. Jung, Teri E. Klein, A. Keith Dunker, and Lawrence Hunter, editors, *Pacific Symposium on Biocomputing*. World Scientific, 2005.

[49] Peter Szolovits. Finding structure in medical reports. `http://groups.csail.mit.edu/medg/projects/text/findstruct/FindStruct_doc.html`.

[50] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[51] Yoshimasa Tsuruoka and Jun'ichi Tsujii. Boosting precision and recall of dictionary-based protein name recognition. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine - Volume 13*, BioMed '03, pages 41–48, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[52] University of Illinois at Urbana-Champagne, 2012.

[53] Ö. Uzuner, B.R. South, S. Shen, and S.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.

[54] Ozlem Uzuner. Recognizing obesity and comorbidities in sparse data. *Journal of American Medical Informatics Association*, 16(4):561–570, Aug 2009.

[55] Ozlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian, and Brett South. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of American Medical Informatics Association*, 17:514–518, February 2010.

[56] Ozlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. Identifying patient smoking status from medical discharge summaries. *Journal of American Medical Informatics Association*, 15(1):14–24, Feb 2008.

[57] Ozlem Uzuner, Imre Solti, and Eithon Cadag. Extracting medication information from clinical text. *Journal of American Medical Informatics Association*, 17:514–518, June 2010.

[58] zlem Uzuner, Imre Solti, Fei Xia, and Eithon Cadag. Community annotation experiment for ground truth generation for the i2b2 medication challenge. *JAMIA*, 17(5):519–523, 2010.

[59] 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. *Journal of the American Medical Informatics Association*, 17(2):19–24, 2010.

[60] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, March 2003.

[61] Qing T. Zeng, Sergey Goryachev, Scott Weiss, Margarita Sordo, Shawn N. Murphy, and Ross Lazarus. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. *BMC Medical Informatics and Decision Making*, 6(1):30–39, July 2006.

[62] Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *ACL*. The Association for Computer Linguistics, 2005.