ABSTRACT

Title of dissertation: A SEMANTICALLY RICH COGNITIVE SEARCH ASSISTANT FOR CLINICAL NOTES

Clare Trahan Grasso
Doctor of Philosophy, 2017

Dissertation directed by: Professor Anupam Joshi
Department of Computer Science
and Electrical Engineering

There are many use cases in the medical industry and in research that require clinical information extraction from the narrative notes in electronic medical records. Significant advances have been made in recent years from using clinical text processing systems which rely heavily on the natural language processing. However, for text that is entered by the clinician at the point of care, where time efficiency is paramount, a shorthand style of text is used which is not amenable to this approach.

This research describes a novel approach that is robust to grammatically deficient text. It relies on techniques that are able to incorporate micro-contexts by taking into account scope, proximity, and location of multiple interdependent matched expressions.

The validity of this approach was established by employing it to create a semantically rich cognitive search assistant that runs in near real-time over the corpus of clinical notes from the Veterans Administration. The cognitive search

assistant was able to extract occurrences of pain events in the text with a positive precision of 84%, a positive recall of 94%, and an F-score of 89% at a rate of 0.31 seconds per note. The extracted results are saved in a semantic representation that permits a reasoning system to be incorporated to perform cognitively rich searches when used in conjunction with predefined medical ontologies.

The result is a semantically rich cognitive search assistant capable of near real-time structured search over clinical text that can be used in interactive applications such as clinical decision support.

A SEMANTICALLY RICH COGNITIVE SEARCH ASSISTANT FOR CLINICAL NOTES

By

Clare Trahan Grasso

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor Anupam Joshi, PhD, Chair/Advisor
Professor Timothy Finin, PhD
Professor Aryya Gangopadhyay, PhD
Professor Charles Nicholas, PhD
Professor Eliot Siegel, MD
Claudia Pearce, PhD

© Copyright by
Clare T. Grasso
2017

Dedication

In memory of my father who was always believed in me but didn't live quite long enough to see this to the end.

Dedicated to my wonderful family who have loved and supported me through it all.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible.

First and foremost I'd like to thank my advisor, Dr. Anupam Joshi, for allowing me the freedom to pursue this area and for taking everything in stride. Likewise, my colleagues at the eBiquity laboratory and the PhD lab, especially Jenn, Lisa, Prajit and Mahbubur, who were a constant source of help, support, and encouragement. I would also like to acknowledge all the support from the staff members that kept all the paperwork flowing and the administrative details in check.

I owe my deepest thanks to my family, Michael, Michelle, Tina, Peter, and JT, for giving their wholehearted love and support over these last five years. Many balls got dropped, but you loved me through it all.

I would like to acknowledge financial support from NSF Grant IIS-0910838 and by the Department of Defense through a supplement to IIP-0934364.

It is impossible to remember all, and I apologize to those I've inadvertently left out.

Lastly, thank you all and thank God!

Table of Contents

List of Tables viii

List of Figures x

List of Abbreviations xi

1 Introduction 1
1.1 Introduction . . . 1
1.1.1 What is the Problem? . . . 2
1.1.2 Approach . . . 4
1.2 Thesis Statement and Contributions . . . 5
1.3 Roadmap . . . 6

2 Background and Related Work 8
2.1 Overview . . . 8
2.1.1 Clinical Text . . . 8
2.1.2 Medical Nomenclature Systems . . . 9
2.1.2.1 UMLS . . . 11
2.2 Biomedical Entity Recognition . . . 12
2.2.1 Dictionary Lookup . . . 13
2.2.2 Pattern Matching . . . 13
2.2.3 ConText . . . 14
2.2.4 Natural Language Processing . . . 15
2.2.5 Machine Learning . . . 16
2.3 Clinical Text Processing Systems . . . 20
2.4 Clinical Decision Support Systems . . . 21
2.5 State of the Art in Clinical Information Extraction . . . 22
2.5.0.1 SemEval 2015 Task 14: Analysis of Clinical Text . . . 23
2.6 Summary . . . 26

3 Conceptual Model 28
3.1 Problem Genesis . . . 28
3.2 Comparison of Clinical Corpora . . . 29
3.3 Overview of Conceptual Model . . . 33

4 Semantic Design 38
4.1 Knowledge-Based Systems . . . 38
4.1.1 Knowledge Representation . . . 38
4.1.2 Description Logics . . . 39
4.1.3 OWL . . . 40
4.1.4 Rules . . . 41
4.1.5 Reasoners . . . 41
4.2 Semantic Structure of Existing Ontologic Resources . . . 42
4.3 Pain Ontology . . . 43
4.3.1 Requirements for Representation and Reasoning . . . 43
4.3.2 Implementation Environment . . . 46
4.3.3 Ontology Implementation Details . . . 47
4.3.3.1 High-level Concepts: . . . 48
4.3.3.2 Events . . . 49
4.3.3.3 Symptoms . . . 50
4.3.3.4 Pain Symptoms . . . 51
4.3.3.5 Summary . . . 52

5 Information Extraction 53
5.1 ConText Algorithm . . . 53
5.1.1 Limitations of ConText Algorithm . . . 55
5.2 ConText Algorithm - Extended . . . 57
5.2.1 Preprocessing of Corpus Text . . . 57
5.2.2 Pain . . . 59
5.2.3 Pain Severity . . . 61
5.2.4 Negation . . . 64
5.2.5 Anatomical Location . . . 65
5.2.6 Temporal Terms . . . 69
5.2.7 Onset . . . 72
5.2.8 Duration . . . 74
5.2.9 Pain Quality . . . 75
5.2.10 Pain Type . . . 78
5.2.11 Variability . . . 79
5.2.12 Time of Occurrence . . . 79
5.2.13 Modified ConText Graph . . . 80
5.2.14 Limitations . . . 81
5.2.15 Discussion . . . 81

6 Information Retrieval 83
6.1 Increasing Algorithmic Efficiency . . . 83
6.1.1 Clustering . . . 84
6.2 Implementation . . . 86
6.2.1 Elasticsearch . . . 87
6.2.1.1 Indexing the Corpus . . . 88
6.3 Experimental Methodology . . . 89
6.4 Results . . . 90
6.5 Summary . . . 92

7 Machine Learning 93
7.1 Overview . . . 93
7.2 Training and Test Set Development . . . 94
7.3 Processing Pipeline . . . 95
7.4 Classifiers . . . 96
7.4.1 Lab Results Classifier . . . 96
7.4.2 Medication List Classifier . . . 96
7.5 Implementation . . . 96
7.6 Discussion . . . 97

8 Clinical Decision Support 100
8.1 Overview . . . 100
8.1.1 Extraction . . . 100
8.1.2 Querying and Inference . . . 101
8.1.2.1 Clinference . . . 101
8.1.3 Runtime Efficiency . . . 103
8.2 Visualization of Pain Severity Events in Clinical Records . . . 103
8.2.1 Background . . . 103
8.2.2 Visualization . . . 104
8.2.3 Discussion . . . 106

9 Evaluation Method 108
9.1 Overview . . . 108
9.2 Corpus Description . . . 108
9.3 Annotating the Data . . . 109
9.3.1 Annotator Qualifications . . . 110
9.3.2 Preparing the Test Data for Annotation . . . 110
9.3.3 Annotator Training . . . 111
9.3.4 Annotation Coding . . . 111
9.3.5 Inter-Annotator Agreement . . . 112
9.3.6 Discussion of Annotations . . . 113
9.4 Summary . . . 115

10 Results 116
10.1 Overview . . . 116
10.2 Gold Standard . . . 116
10.2.1 Creating the Gold Standard from the Annotations . . . 116
10.3 Overview of Scoring Metrics . . . 117
10.3.1 Target Recognition Metrics . . . 118
10.3.2 Slot Attribute Metrics . . . 119
10.3.3 Per-Target Accuracy . . . 121
10.3.4 Overall Evaluation Metrics . . . 121
10.4 Results . . . 122
10.4.1 Target Recognition Results . . . 122
10.4.2 Slot Attribute Results . . . 123
10.4.3 Overall Evaluation Results . . . 123
10.5 Discussion . . . 124
10.5.1 Comparison with the State of the Art . . . 124
10.5.2 Sources of Error . . . 126
10.5.3 Analysis of Results . . . 127

11 Conclusion 130
11.1 Limitations . . . 133
11.2 Future Work . . . 133

A Appendix 136
A.1 Pain Ontology Definitions . . . 136

B Appendix 140
B.1 Official Annotation Guidelines . . . 140

Bibliography 143

List of Tables

2.1 List of features used in comparison 19
2.2 Comparison Of Classifiers For Identifying Semantic Class of Named Entities 19
2.3 Slot attributes for Task 2b 26

4.1 Logical Inference Properties 40

5.1 Pain Extraction Expressions 61
5.2 Numeric Pain Severity Regular Expression 62
5.3 Pain Severity Expressions 63
5.4 Negation Expressions 64
5.5 Variety of Anatomical Location Expressions 65
5.6 Location Extraction Expressions 70
5.7 Temporal Expressions 73
5.8 Onset Expressions 74
5.9 Duration Expressions 76
5.10 Pain Quality Expressions 77
5.11 Pain Type Expressions 78
5.12 Variability Expressions 80

6.1 Comparison Of Classifiers For Identifying Lines Of Pain 85
6.2 SINGLE ANCHOR TERM OF PAIN 86
6.3 ANCHOR TERMS OF *PAIN, LEVEL, QUALITY, SEVERITY, LOCATION, SCALE, ONSET* 86
6.4 ELASTICSEARCH TERMINOLOGY 88
6.5 FILTERS WITH NO STEMMING 90
6.6 FILTERS USED WITH STEMMING 90

7.1 Comparison Of Classifiers For Identifying Lab Reports 98
7.2 Comparison Of Classifiers For Identifying Medication Lines 99

9.1 Corpus Statistics 109
9.2 Inter-Annotator Agreement 113

10.1 Prevalence of Non-default Slot Attribute Values Over the Test Corpus. 118
10.2 Target Recognition Results 123
10.3 Slot Attribute Accuracy Results Using SemEval 2015 Metrics 124
10.4 Slot Attribute Accuracy Results Using Precision and Recall 124
10.5 Overall Evaluation Results 125

A.1 Patient Ontology Definition 136
A.2 Note Ontology Definition 136
A.3 Event Ontology Definitions 137
A.4 Certainty Ontology Definitions 137
A.5 Onset Ontology Definitions 137
A.6 Duration Ontology Definitions 138
A.7 Location Ontology Definitions 138
A.8 Severity Ontology Definitions 138
A.9 PainSeverity Ontology Definitions 139
A.10 PainSymptom Ontology Definitions 139
A.11 PainQuality Ontology Definitions 139
A.12 PainType Ontology Definitions 139

List of Figures

2.1 Example of a lexical entry in the SPECIALIST Lexicon. 10
2.2 Example of how the ConText algorithm determines a dependency between a target expression in the text *pain* and its modifier *does not have*. 15
2.3 Parsing of simple sentence. 17
2.4 Clinical Information Model for Pain. 24

3.1 Example of clinical text used in the SemEval 2015 challenge. 29
3.2 Example of clinical narrative taken from VHA Hospital VISTA EHR. 30
3.3 Example of nursing flow sheet from VHA Hospital VISTA EHR. . . . 32

4.1 SNOMED-CT entry for pain in the hallux. 43
4.2 NCI entry for pain and its child concepts. 44
4.3 Overall Structure of Composite Pain Ontologies. 48
4.4 Event and Certainty Ontologies. 50
4.5 Severity and PainSeverity Ontologies. 52

5.1 Examples of Context algorithm using targets and modifiers. 55
5.2 Example of a anatomical entry in FMA. 66
5.3 Example of an abbreviation entry for *llq* in NLM SPECIALIST Lexicon. 67
5.4 Underlying ConText Graph. 81

7.1 Example of a lab results that have been copied and pasted into the clinical note. 94
7.2 Example of a medication list. 95

8.1 Example of request sent by CDS system to the extraction engine. . . 104
8.2 SPARQL query for pain visualization data. 105
8.3 Pain Severity Events Visualization Results. 106

10.1 Gold Standard. 117

List of Abbreviations

API	application programming interface
BOW	bag-of-words
CDS	clinical decision support
CI	confidence interval
CIM	Clinical Information Model
CDS	clinical decision support
CRF	conditional random fields
cTAKES	clinical Text Analysis and Knowledge Extraction System
CUI	concept unique identifier
DL	description logic
EHR	electronic health records
EMR	electronic medical record
FMA	Foundation Model of Anatomy
HL7	Health Level 7
ISO	International Standards Organization
HIPAA	Health Insurance Portability and Accountability Act
i2b2	Informatics for Integrating Biology and the Bedside
ICD	International Classification of Diseases
IDF	inverse document frequency
IE	information extraction
IHTSDO	International Health Terminology Standards Development Organisation
IR	information retrieval
KB	knowledge base
KR	knowledge representation
NER	named entity recognition
NIH	National Institutes of Health
NLM	National Library of Medicine
NLP	natural language processing
OWL	Web Ontology Language
POS	part(s) of speech
RBF	radial basis function
RuleML	Rule Markup Language
SHARP	Strategic Health IT Advanced Research Projects
SNOMED-CT	Systematized Nomenclature of Medicine - Clinical Terms
SVM	support vector machine
SWRL	Semantic Web Rule Language
TF	term frequency
UMLS	Unified Medical Language System
VA	US Department of Veterans Affairs
VHA	Veterans Health Administration
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Chapter 1: Introduction

1.1 Introduction

Clinical documentation is the record of observations, impressions, plans and other activities arising in the course of patient care. It is intended to be an objective record of a patient‘s health history, physical findings, and diagnosis. This documentation is important for continuity of care, billing, insurance, legal proceedings, quality assurance, and research [1]. Because of the extreme diversity and heterogeneity of health data across health domains and institutions, the advent of electronic medical records (EMR) and computer-based documentation systems have brought with them a high value on using structured data input forms using standardized sets of codes to support machine readability and storage in backend databases [2].

However, clinical narrative notes remain the richest source of healthcare data in the clinical chart, providing highly nuanced insight into patient status, care, and treatment. Structured data using clinical codes and categories does not accommodate uncertainty or express a range of possible differential diagnoses. Unstructured data has been left largely unexploited because it is notoriously challenging to analyze automatically [3] [4].

While there are many uses for extracting this information from the EMR, there

are particularly compelling reasons for its use in clinical decision support systems (CDS). CDS systems can help practitioners minimize errors, improve quality, and increase efficiency in healthcare [5]. There is great promise in using natural language processing (NLP) to perform information extraction from the narrative notes in electronic medical records (EMR) for use in clinical decision support systems [3].

For CDS, it is not enough to determine that a particular medical concept appears in a patients record such as a simple text search would provide. Important contextual information that belongs to that concept, such as anatomical location, is also relevant. *Backache* will suggest a difference course of action than a *toothache*. Likewise, the onset and duration of a symptom may give important clues in a differential diagnosis.

Another critical element for CDS is that, in many clinical settings, the results must be available in near real-time for interactive use [5]. Key insights from the notes must be extracted and presented to the clinician quickly. A result that is returned after the patient leaves, may delay care. In the emergency department, grave decisions must be rendered very quickly in life-threatening situations, often with limited information given by a patient concerning their own medical history.

1.1.1 What is the Problem?

Current state of clinical text natural language processing systems: Currently, the most widely used clinical text processing systems [6] are based on the assumption that clinical notes employ a well-formed grammatical structure in the

narrative portions of the clinical documents. This syntactic structure can then be leveraged using statistical NLP [7] [8] and machine learning algorithms to determine dependencies between elements which may be used, for instance, to label atomic medical concepts appearing in the text to a normative reference terminology. They are also able to provide shallow semantics consisting of labeling medical events with their historicity ("history of"), negation ("does not have"), and uncertainty ("may have").

There are three major shortcomings in this approach.

Problem 1: For text that is entered by the clinician at the point of care, where time efficiency is paramount, a shorthand style of text is used that is heavily abbreviated and tends to ignore the rules of grammar, punctuation, and white space [9]. In a corpus of 1,200 notes coming from the US Veterans Health Administration (VHA), grammatically clean text constitutes only 5% of the total text leaving 95% of the text not amenable to those approaches. This is especially significant as the Veterans Health Administration is the largest healthcare provider in the US [10]. At present, there does not appear to be any extant literature describing research in labeling and extracting medical concepts from clinical text that have these characteristics.

Problem 2: While these systems are able to determine context based on grammatical dependencies in the text, they are not able to determine deeper conceptual dependencies that exist within the medical domain. For example, these systems are able to correctly identify tokens as numeric values, but they are not able to determine the semantics of what that number means. In the text "pain

severity: 10", the number 10 indicates the severity of pain being experienced, but these systems can only identify them as ordinary numbers.

Problem 3: These systems focus on labeling the narrative note for all syntactic and medical concepts at a very fine-grained level. This is appropriate for creating large databases of normalized health data that can be shared across institutions for open health initiatives, but is computationally complex and requires a great deal of post-processing to extract particular concepts and their related attributes. This complexity leads to runtime performance that cannot qualify as near real-time.

In addition, there is one final potential that is not being exploited. The biomedical world is rich in formal semantic ontologies that define not only the terms of the domain, but define the relationships between these terms as well. This, in turn, can be used with reasoning engines to perform inference over those terms and their relationships. For example, a search query for *arm pain* would return not only all instances of arm pain but all instances of *elbow pain*, *wrist pain*, and *hand pain* as well using the part-of relation. However, this is only possible if the extracted information is itself encoded in a structured semantic representation based on a well-defined ontology.

1.1.2 Approach

This research describes an approach that is robust to grammatically deficient text by not relying on grammatical structure but on the phrasal patterns that are prevalent in the medical domain. It relies on techniques that are able to incorporate

micro-contexts by taking into account scope, proximity, and location of multiple interdependent expressions in order to extract the relevant attributes of medical concepts. Expressions that rely on specialized lexicons and the results of other extraction algorithms are also accommodated. In addition, in order for the medical concept extraction to be useful in real clinical decision support systems, the extraction has been optimized for runtime efficiency in near real-time. Finally, a formal ontology was constructed so that a semantic representation of the extracted data may be used with an inference engine.

1.2 Thesis Statement and Contributions

The thesis of this research is: An approach that combines semantic and machine learning techniques can be used to extract medical concepts from clinical text for use in clinical decision support systems. Contributions of this research will include the following:

Medical Concept Knowledge Representation

A semantic knowledge representation for medical concepts that is modular and can be extended and shared with other applications.

Medical Concept Event Extraction

A novel approach that is robust to grammatically deficient text using techniques that are able to incorporate micro-contexts by taking into account scope, proximity, and location of multiple interdependent expressions.

Medical Concept Information Retrieval

Information retrieval techniques that can be used in conjunction with the knowledge representation to extract portions of the clinical narrative containing the concept. This part of the system focuses on filtering out all unneeded text in a way that is highly efficient and scalable in order to provide only that text which contains the concept into the next phase for the actual extraction.

Clinical Decision Support System

An API to make the results available for clinical decision support systems and for visualization, as well as a framework for storing and reasoning over the extracted data.

1.3 Roadmap

The remainder of this document proceeds as follows. Chapter 2 further describes the current production clinical text processing systems that are in wide use and the standardized medical terminology systems they are coupled with, followed by a brief overview of the underlying statistical natural language processing and machine learning algorithms that underlie these systems. The section ends with a description of state-of-the-art experimental systems including a description of their increased capabilities over production systems and results of their evaluation.

Chapter 3 describes the conceptual model employed in carrying out this research. It begins with a description of the original problem in its genesis. The section then goes on to describe how this problem and the difficulties surrounding it engendered a new approach to solving this problem.

Chapters 4 through 7 detail the methods used to implement the conceptual model to solve the problem.

Chapter 8 discusses how the validity of this approach was established by employing it to create a semantically rich cognitive search assistant that runs in near real-time over the corpus of clinical notes from the Veterans Health Administration. The system is able to extract medical concepts that are signs and symptoms along with their contextual attributes including location, severity, onset, duration, quality and type.

Chapters 9 and 10 explains the how the complete extraction algorithm was evaluated. Chapter 9 discusses the annotation methods that were used to create the gold standard. Chapter 10 describes the metrics used in the evaluation and concludes with the final results.

Chapter 2: Background and Related Work

2.1 Overview

2.1.1 Clinical Text

Meystre *et al.* define the difference between biomedical text and clinical text as follows: "[we] define biomedical text to be the kind of text that appears in books, articles, literature abstracts, posters, and so forth. Clinical texts, on the other hand, are texts written by clinicians in the clinical setting. These texts describe patients, their pathologies, their personal, social, and medical histories, findings made during interviews or during procedures, and so forth. Indeed, the term 'clinical text' covers the entire gamut of narratives appearing in the patient record."

This text is particularly challenging because of heavily overloaded and domain-specific short-hand terms and abbreviations, misuse of punctuation and whitespace, misspellings, and telegraphic sentence fragments. Additionally, there may be the inclusion of pseudo-tables containing lab results and vital signs, and preformatted templates that assist the clinician in data entry with fields to be filled in by the user, all of which tend to be highly idiosyncratic and institution-specific [11] [12] .

2.1.2 Medical Nomenclature Systems

Another major challenge is the large number of specialized terms that are used in this domain. Much effort has been invested in developing standardized biomedical nomenclature systems to define these terms. They contain terms to name anatomy, diseases and disorders, signs and symptoms, and clinical procedures. Currently there are over 180 different nomenclature systems defined in the biomedical domain constituting over seven million individual but sometimes overlapping terms.

While there is some conceptual overlap between lexicons, terminologies, and ontologies, there are key differences which will be emphasized for the purpose of this research.

Medical Lexicons

A *lexicon* is a dictionary of linguistic or factual elements in a specialized field. The National Library of Medicine (NLM) SPECIALIST lexicon is a large syntactic lexicon containing both general English and biomedical terms. Each lexical entry records the syntactic, morphological, and orthographic properties of the word. Currently there are 475,000 lexical items. It also contains the most common abbreviations for each term. [13].

Figure 2.1 shows a single entry from the SPECIALIST Lexicon. The *base* gives the normalized version of the term, *spelling* variations are listed, *entry* gives the identifier within the lexicon, *cat* gives the part of speech, and *variants* give morphological expressions of the base word.

```
base=hemoglobin
spelling_variant=haemoglobin
entry=E0031208
cat=noun
variants=uncount
variants=reg (plural: hemoglobins, hemoglobins)
```

Figure 2.1: Example of a lexical entry in the SPECIALIST Lexicon.

Medical Terminologies

A *terminology* contains a controlled vocabulary of terms that have a special meaning within a specific context. Terminology differs from lexicography in that it involves the study of concepts and/or conceptual systems and their labels, whereas lexicography studies words and their meanings. The terminology might or might not contain well-defined relationships between terms such as in a hierarchy of parent-child relationships. Terminologies are generally associated with library and information science.

The Systematized Nomenclature of Medicine - Clinical Terms (SNOMED-CT) claims to be, "the most comprehensive, multilingual clinical healthcare terminology in the world." As such, it has become the *de facto* standard in clinical text processing systems to enable meaning-based retrieval from clinical records for use in clinical decision support, reporting, and analysis applications. SNOMED-CT is owned, maintained and distributed by the International Health Terminology Standards Development Organisation (IHTSDO), a not-for-profit association of twenty-seven countries.

Medical Ontologies

An *ontology* represents knowledge as a hierarchy of concepts within a domain, using a shared vocabulary to denote the types, properties, and inter-relationships of those concepts. Ontologies are particularly helpful for resolving ambiguity between domains in entity recognition and for relation extraction. Most importantly, ontologies support reasoning and lend themselves to reuse.

The Foundation Model of Anatomy (FMA) ontology is a project of the Structural Informatics Group at the University of Washington. It is a formal, computer-based representation of classes, types, and relationships of the human body in a form that is navigable, parseable and interpretable by machine-based systems. Specifically, the FMA is a domain ontology that represents a coherent body of declarative knowledge about human anatomy. Currently it contains approximately 94,000 classes, over 170,000 terms, and over 2.4 million relationship instances from over 227 relationship types [14] [15].

FMA is open sourced and is licensed under a Creative Commons Attribution 3.0 unsupported License.

2.1.2.1 UMLS

In 1996 the Unified Medical Language System (UMLS) initiative was started at the National Library of Medicine (NLM) to unify medical vocabularies from various medical knowledge source so that they could be used across information systems [16].

The Metathesaurus is a terminologic resource within the UMLS environment. It is the Rosetta Stone of almost all of the medical terminologies, ontologies, coding systems, and other nomenclatures. It currently contains over 180 families of source vocabularies, 21 languages, 8.6 million terms, approximately three million concepts, and more than ten million relations [17] [18].

The UMLS Metathesaurus provides a mapping between all its covered terminologies and provides a common presentation of those terms. In doing so, it has assigned concept unique identifiers (CUI) for each concept which may encompass terms from multiple source terminologies which are essentially synonymous. The CUIs have become the standard identifier for medical terms across a very broad range in the biomedical domain. One very important functionality this provides is the normalization of terms such as *Addison Disease* in the biomedical literature and *Addison's disease* as it is used in clinical repositories.

2.2 Biomedical Entity Recognition

One of the most important tasks in clinical text processing is to recognize biomedical concepts in the text and to map them to a corresponding term in one of the standardized nomenclature systems. However, the exact expression of those terms may differ considerably in the text of clinical notes, and therefore causes difficulty in recognizing them.

Typically, once these terms have been identified in the text, they are *normalized* to a unique identification code coming from one of these standard reference

nomenclature systems. This also referred to as *entity recognition* whose meaning in the biomedical domain differs from that of the general NLP domain (in which case an entity is the name of any person, organization, *etc*).

2.2.1 Dictionary Lookup

Dictionary lookup employs general search techniques to match words in the text with terms in a nomenclature system. The time complexity for the lookup ranges from $O(1)$ for hash lookup, $O(logn)$ for indexed searches, and $O(n)$ for linear searches through the dictionary where n is the number of terms. As such, it can be relatively fast compared to other methods. Because most domain terms are covered, it provides a very high recall on individual words.

However, if the sequence of words in the text don't match the lexical representation defined in the dictionary, it can result in many false negatives. Likewise, false positives are generated for terms like *back* that appear in the dictionary, but have many other meanings in the general case. Dictionary lookup by itself is not generally effective.

2.2.2 Pattern Matching

Pattern matching uses regular expressions to identify specialized expressions in the text and combine them with other unknown expressions using wild card characters. Regular expression algorithms are implemented as character-based deterministic finite state automata whose time complexity is $O(n)$ in the length of the

string [19]. As such, they are relatively fast.

Their primary drawback is that regular expression patterns must be very specific in order to capture the exact sequences of characters in the text. For example, the authors in [20] created over 200 regular expression patterns to find the cardiac ejection fraction measurement in radiology reports in a quality assurance application.

2.2.3 ConText

Some patterns depend on the presence of other patterns and how they are co-located in the text. The co-location can be characterized by the proximity, scope, and direction of one expression in relation to another. By stipulating the main expression of interest as the target, and the dependent expression as the modifier, a dependency relation may be inferred between the target and the modifier. This allows specific dependencies in the text to be identified without relying on an underlying grammar.

The ConText algorithm [21] is heavily used in clinical text processing systems to find negation (*does not have*), certainty (*may have*), experiencer (patient or family member), historicity (e.g., *has a history of*) or whether the concept is hypothetical (e.g., *if the patient experiences pain...*). Figure 2.2 shows an example of this.

An implementation of this algorithm may be freely downloaded from `https://github.com/chapmanbe/pyConTextNLP`. The implementation comes with a knowledge base of clinical phrases, their semantic categories, and their regular expression

Text: ***The patient does not have pain.***
Target pattern: ***pain***

Modifier pattern: ***does not have***
Category: **Negation**
Direction: **Forward**

Figure 2.2: Example of how the ConText algorithm determines a dependency between a target expression in the text *pain* and its modifier *does not have*.

patterns. For example, *definite_negated* category has over 80 phrases defined, such as, *are ruled out*, *can be ruled out for*, *is negative*, and *free*.

These phrasal patterns are very specific and assume that the text uses clean underlying grammatical structure. It is also not able to directly leverage terms defined in a nomenclature system. However, its database base of patterns may be easily extended, and the algorithm is flexible so that different types of patterns, not originally envisioned by the authors, could be added.

2.2.4 Natural Language Processing

Natural language processing (NLP) consists of software and algorithms that are capable of analyzing unstructured textual information in order to understand human language within a specific context [7].

Statistical NLP relies on the observation that phrases share common distributional properties when the text follows underlying rules of grammar. Each word functions as a particular part-of-speech (POS) [22] within a sentence (e.g.,

noun,verb). Words are grouped into constituent phrases (e.g., *noun phrase, verb phrase*) [23]. Phrases combine to create sentences [24].

Once the grammar has been defined, statistical models can be built based on how words are used within sentences. The language model assigns probabilities to the words and the transitions between words and phrases based on the grammar. Parsing uses this model to discover how the words in a given sentence combine and relate to one another [25].

Parsing can be powerful tool in determining dependencies in the text by discovering their semantic scope. This is particularly important for negations of concepts, and for temporal expressions specified in the text. Figure 2.2.4 shows the results of processing a simple sentence through the Stanford CoreNLP [26] processing pipeline. The basic and enhanced dependency graphs show the effectiveness of these algorithms to determine the scope of expressions.

2.2.5 Machine Learning

Machine learning is used in clinical text processing to perform part-of-speech tagging, but more importantly, to identify and normalize sequences of words in the text that are diseases, disorders, signs, symptoms, anatomical locations, and medications. Two machine learning algorithms that have been heavily used in clinical text processing are support vector machines [27] and conditional random fields [28].

Support Vector Machines (SVM): are binary classifiers that use a bag-of-words approach to encode feature vectors in order to predict a label. In this

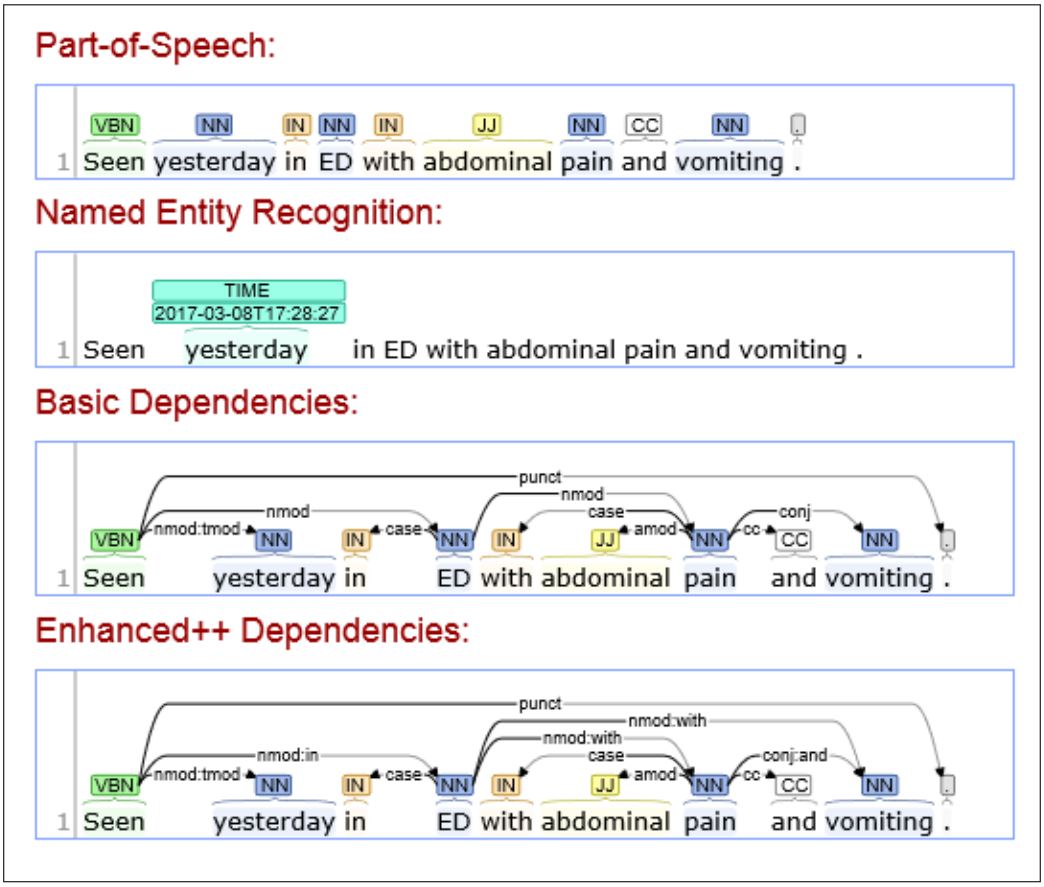

Figure 2.3: The results of NLP algorithms to identify parts-of-speech, recognize named entities, determine basic and enhance dependencies in the text

approach, every word in the data is represented as a feature in the vector during training. Each classification instance encodes any words that appear in that instance either as a binary indicator or as a normalized count over all words found in the instance. Random features such, as capitalization status or presence of certain words in a dictionary, may also be included in the feature vector. SVMs use a supervised training model that learns weights for each feature in order to assign any given feature vector instance to one of two predefined classes (labels).

The time complexity of SVMs for classification of instances is $O(n)$ in the length of the feature vector. However, the bag-of-words approach does not allow them to capture *sequences* of words in the text. Also, because SVMs are binary classifiers, a separate model must be trained for every label if there are more than two.

Conditional Random Fields (CRF): are discriminative factor graphs that are able to classify over multiple labels. In the context of clinical text processing, the input is a sequence of words. In a linear chain CRF, each word has its own state, but is also aware of the state of words in a window around it. The states are encoded as feature functions that compute an output based on the feature values that are fed into it. Using the chain, the CRF model is able to consider state-to-state transitions as well as feature-to-state dependencies. During supervised training, the model employs logistic regression to learn the weights for each of these feature functions in order to predict a label for the current word or sequence of words.

The time complexity of CRFs for classification is $O(n^3)$ although the training complexity is much higher at $O(n^7)$. CRFs are able to identify and classify multi-word sequences of arbitrary length in the text. However, they have a tendency to overfit the training data, and tend to fail if arbitrary words are embedded in a learned sequence.

Comparison of Approaches for Entity Recognition In 2008, Li *et al.* [29] presented a comparative study between SVMs and CRFs in a clinical text named entity recognition. In this study, they used these methods to identify medical concepts in clinical text and to predict one of four semantic classes: Disorders/Diseases,

Table 2.1: List of features used in comparison

Feature	**Description**
Dictionary lookup	True if the term appears in SNOMED-CT
Bag of Words	Set of unique words in the context
POS tags	Part of Speech tags of the context words
Capitalization	All upper case, all lower case, mixed case and initial upper case
Window size	Number of tokens representing context surrounding the target word
Distance	Proximity of the feature in regard to the target word
Number features	Presence or absence of related features

Table 2.2: Comparison Of Classifiers For Identifying Semantic Class of Named Entities

Approach	**Precision**	**Recall**	**F2**
Dictionary Lookup	85%	47%	60%
SVM	82%	49%	61%
CRF	96%	77%	86%

Signs/Symptoms, Anatomy, and Drugs. Table 2.1 shows the features used in the comparison.

Table 2.2 presents the results of the comparison between the SVM and CRF approaches. The line labeled Dictionary Lookup shows the results of using the CRF with *only* the dictionary lookup feature.

In the discussions of the results, the most discriminative features were the immediate context (the window of words surrounding a target word) and capitalization. The least discriminative features were the bag of words indicating that sequence matters, and POS features indicating that grammatical roles are not as important as context.

2.3 Clinical Text Processing Systems

Clinical text information extraction and annotation systems surveyed in this section predominantly incorporate one or more of the following approaches: pattern matching systems and/or rule-base systems, statistical NLP systems, and ontological systems.

cTakes - Mayo clinical Text Analysis and Knowledge Extraction System: cTakes was developed in conjunction with the Mayo Clinic, which is one of the largest integrated nonprofit medical group practices in the world [1].

cTakes' unique contribution is "an NLP system specifically tailored to the clinical narrative that is large-scale, comprehensive, modular, extensible, robust, open-source and tested at component and system levels...cTakes components are specifically trained for the clinical domain to create rich linguistic and semantic annotations. " [6] [25]

cTakes processing consists of a pipeline of tasks that runs within the UIMA framework [30] and is made of up the following components: sentence boundary detector; a tokenizer; a term normalizer based on the SPECIALIST Lexicon; an OpenNLP POS tagger retrained on clinical data; a shallow parser; and an NER annotator which maps concepts to a subset of the UMLS, SNOMED-CT and RxNORM terminologies where the look-up window consists of noun phrases. cTakes does not resolve ambiguities that result from identifying multiple terms in the same text span. Negation annotators implement the NegEx algorithm [31].

[1] `en.wikipedia.org/wiki/Mayo_Clinic`

The sentence boundary detector extends OpenNLPs supervised sentence detector tool to better handle periods, question marks, and exclamation marks. The tokenizer is a context-dependent process that merges tokens to create date, fraction, measurement, person title, range, roman numeral, and time tokens by applying rules (implemented as finite state machines) for each of these types.

"Performance of individual components: sentence boundary detector accuracy=0.949; tokenizer accuracy=0.949; part-of-speech tagger accuracy=0.936; shallow parser F-score =0.924; named entity recognizer and system-level evaluation F-score=0.715 for exact and 0.824 for overlapping spans, and accuracy for concept mapping, negation, and status attributes for exact and overlapping spans of 0.957, 0.943, 0.859, and 0.580, 0.939, and 0.839, respectively." [6]

2.4 Clinical Decision Support Systems

The term clinical decision support refers broadly to providing clinicians or patients with computer-generated clinical knowledge and patient-related information, intelligently filtered or presented at appropriate times, to enhance patient care. [32] It is estimated that at least 50% of the clinical information describing a patient's current condition and stage of therapy resides in the free-form text portions of the EHR. Notably, medical history and physical exams are included in this. Rindflesch *et al.* also determined that, when given the task of summarizing a patient's health status, physicians spent 50% of their time in the clinical notes portion of the patient record. [33] Additionally, even though patient data can be entered into a CDS

system manually, this would require clinicians to recognize the need for it and to have time to find and enter the requisite data. Two CDS features associated with improved patient care therefore are: automatically providing support as part of the workflow, and providing support at the time and location of decision making. Demner-Fushman *et al.* conclude that, "If CDS systems were to depend upon NLP, it would require reliable, high-quality NLP performance and modular, flexible, and fast systems." [11]

Because of these factors, the ability to extract data from free-form text — and put it into a form that clinical decision support systems can access and utilize — was identified as being one of the top ten grand challenges in clinical decision support. [32]

2.5 State of the Art in Clinical Information Extraction

Until recently, clinical text processing research has focused on entity recognition for biomedical concepts in text. However, these concepts don't stand by themselves — they have a surrounding context. This is essentially what the narrative note provides.

However, there are very large national and international efforts dedicated to providing a common format for health information content so that semantically interoperable information may be created and shared in health records. The two largest efforts in this area are the International Standards Organization (ISO) Health Level 7 (HL7) [2] and the Strategic Health IT Advanced Research Projects (SHARP)

[2] `http://www.hl7.org/implement/standards/`

Program [3].

Clinical Information Models (CIM) provide a common format for health information so that Semantically interoperable information may be created and shared in health records. These are **input** templates that define how clinical concepts can be combined using fill-in-the-box and drop-down lists. They provide a common data model between disparate clinical information systems and fit well into a relational database architecture. They relieve the need for natural language and enhance quality control.

CIMs associate clinical concepts with other information depending on its semantic type. However, the relations between the elements are implied by the data structure; there are no formal semantic relationships defined in the model. Figure 2.5 shows an element for pain from the Sharp model. The PatientIdentifier forms a link back to the patient, while the other attributes modify the pain event. Notice that the pain character attribute has predefined values from a value-set. No other values are allowed in this field.

2.5.0.1 SemEval 2015 Task 14: Analysis of Clinical Text

One of the most recent tasks to push the state of the art in clinical text processing was SemEval 2015 Task 14: Analysis of Clinical Text challenge [34]. "In addition to recognizing and normalizing named entities in clinical text, one fundamental task of clinical natural language processing is to identify the potential modifiers attached to specific named entities, such as negation and uncertainty. The

[3] `http://sharps.org/`

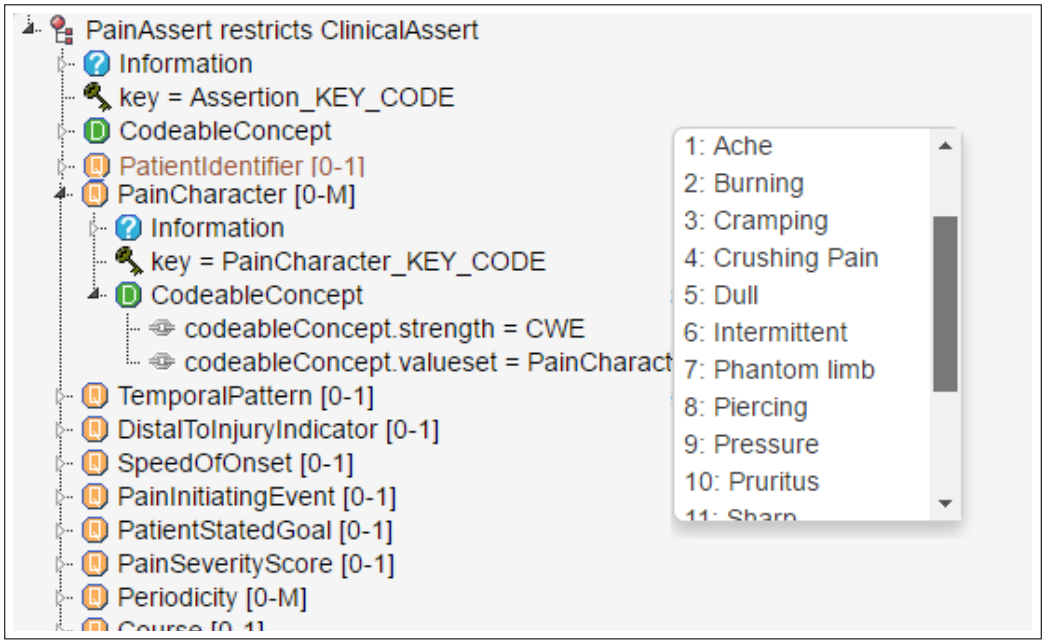

Figure 2.4: Key attributes are associated with the pain event such as location, onset, and character.

Analysis of Clinical Task is split into two tasks, one on named entity recognition, and one on template slot filling for the named entities. ”

In Task 1: Disorder Identification, the goal was to recognize the span of a disorder mention and to normalize it to a unique CUI in the UMLS/SNOMED-CT terminology in a set of clinical notes. Task 2 focused on identifying the normalized value for nine modifiers in a disorder mentioned in a clinical note: the CUI of the disorder (as in Task 1), as well as the potential attributes from the given clinical information model. In Task 2a, participants are given the disorders, and only the slots need to be extracted. In Task 2b, required both the disorders and the slot values to be extracted. Table 2.3 shows the slot values that were to be extracted from the text to populate the CIM and the results of the top performing team.

The data for the challenge came from the ShARe corpus containing clinical notes from the MIMIC II database and was manually annotated for disorder mentions and normalized to an UMLS Concept Unique Identifier (CUI) when possible. The notes consisted of radiology reports and discharge summaries. There were a approximately 88,000 disease/disorder terms to normalize against and 22,000 anatomical locations.

The evaluation for Task 1 consisted of calculating the precision, recall and F_2 score of the disorder mentions against the gold standard:

$$Precision = P = \frac{D_{tp}}{D_{tp} + D_{fp}} \tag{2.1}$$

$$Recall = R = \frac{D_{tp}}{D_{tp} + D_{fn}} \tag{2.2}$$

$$F_2 = \frac{2 * P * R}{P + R} \tag{2.3}$$

For Task 2, the evaluations was based on the F_2 score of the disorder mention along with the weighted accuracy of the slot values for that disorder. The weight for each slot was defined as:

$$\forall i 1..n_k, weight(s_k^i) = 1 - prevalence(s_k^i) \tag{2.4}$$

where $prevalence(s_k^i)$ is the prevalence in the corpus of the value k for slot i .

Thus, weighted per-disorder accuracy is defined as:

$$WA = \frac{\sum_{k=1}^{K} weight(gs_k) * I(gs_k, ps_k)}{\sum_{k=1}^{K} weight(gs_k)} \tag{2.5}$$

where, like above, gs_k is the gold-standard value of slot s_k and ps_k is the predicted value of slot s_k, and I is the identity function: $I(x, y) = 1$ if $x = y$ and 0 otherwise.

For Task 2b, only true positive disorders were used to compute overall accuracy. The final overall weighted score used to compute the rankings as computed as the F_2 score of the disorder recognition multiplied by the weighted accuracy (WA) of the slot filling.

Table 2.3: Slot attributes for Task 2b

Slot	**Description**	
Overall Score	$F_2 * WA$	80.8%
Disorder	sign, symptom, disease	92.6%
Negation	*denies, does not have*	97.6%
Subject	patient, family, other, donor	97.3%
Certainty	probable, definite	91.2%
Course	better, worse, resolved	89.9%
Severity	slight, moderate, severe, unmarked	93.9%
Conditional	*evaluated for*	89.9%
Location	anatomical location	86.4%
Location CUI	anatomical location CUI	81.9%

2.6 Summary

This chapter has described the difficulties associated with mining and extracting information from clinical text. There are well-defined medical nomenclatures systems used to provide their terms, definitions, relationships, and lexicography. However, these systems can also present a challenge due to their size. Other important components used in clinical text processing, such as, statistical NLP algorithms

were described. cTakes, a widely used end-to-end clinical text processing system that combined nomenclature resources with NLP was covered. Finally, the state-of-the-art in clinical entity recognition and slot filling was presented along with the results against which this research will be compared.

The following chapter describes how a particular problem at the Veterans Health Administration hospital that required clinical text processing to mine patient records in the EHR failed using these existing clinical text processing approaches. It then goes on to set forth a new approach to solve this problem.

Chapter 3: Conceptual Model

3.1 Problem Genesis

Physicians at the Veterans Administration (VA) in Baltimore were desirous of studying pain its affects on the care of veterans. Pain is a component in many different diseases and processes. At the VA, 80% of all ER visits have a primary complaint involving pain. However, data describing pain events in the patient record appears largely in the unstructured text, and the amount of text appearing in the clinical record for each patient is extensive. In the case of an appendicitis patient in the VA EHR system, a three-day stay in the hospital for a routine appendectomy produced almost 180 pages of text. By the 3rd day, a physician overseeing the patient would be required to peruse the entire 180 pages of text to extract facts of relevance to his clinical approach. This presented a great challenge for these physicians to research pain in this population. They were not able to unlock the data in the EMR to enable their research.

Additionally, there was great interest on the part of the physicians at the VA to use this electronic patient health data for clinical decision support systems. Radiologists expressed a desire for a smart search capability through the patient record in order to verify possible diagnoses when viewing an image. Others were interested in a clinical decision support system that could automatically extract pain

events from the patient record in order to quickly visualize the progression of pain after surgery.

3.2 Comparison of Clinical Corpora

The text used in the SemEval 2015 Task, while containing many abbreviations and non-sentences, was grammatically clean. In fact, discharge summaries and radiology reports (of which the corpus consisted) have long been used in clinical text processing research because they are written in a more formal style and have better grammatical structure than other types of notes [12]. Figure 3.1 shows an example of the text used in that challenge.

> HISTORY OF PRESENT ILLNESS: This is a 61-year-old male with no significant past medical history who was seen initially in the Emergency Department on 05-22 after presenting with melena, a syncopal episode, and dyspnea on exertion.

Figure 3.1: Example of clinical text used in the SemEval 2015 challenge.

The corpus obtained from the Veterans Health Administration is strikingly different in character from that of the Share corpus. At the VA, clinicians write their notes while they are in the midst of patient care or after their shift is over. The focus of their efforts is to get all relevant documentation in the system in the shortest amount of time. This results in notes that are greatly abbreviated in a short-hand style that ignores many common grammatical rules. It is filled with sentence fragments, missing and creative use of punctuation and whitespace,

and many nonstandard and overloaded abbreviations. The clinical narrative in the patient history is also interspersed with structured and semi-structured text such as headers, lab results, and flow sheets that contain no grammatical structure at all.

Figure 3.2 shows a snippet of text from the VA corpus. Written in a grammatically correct way, the free-form text would read: "*The subject is a new patient and has no primary medical doctor. The patient states that in his past health history he was diagnosed with a 'mild, arithmic heart'. He broke his right wrist in 1991. He is a recovering drug addict, and has been clean for twelve years. The patient complains that he has had fluid on his left knee for one week in which he feels pressure, but no pain. Also, the patient complains of having blood in his stool for one year, and has lost twenty pounds in the past year. He has been using Citrocel. His family health history includes a father with colon cancer.*"

LOCAL TITLE: AMBULATORY/OUTPATIENT CARE NOTE
STANDARD TITLE: PRIMARY CARE NOTE

Pt identified by full name and full SS#:XXXXXXX,XXXXXX
Any Allergies:CODEINE
Any new or changed meds:NEW PT
Any hospitalization or surgury since last visit:SEEHX.

S:NEW PT.-HAS NO PMD. PAST HX.-DX. WITH "MILD ARITHIMIC HEART", BROKEN RIGHT WRIST 91, RECOVERING DRUG ADDICTION-CLEAN FOR 12 YRS. C/O FLUID ON LEFT KNEE X 1 WEEK. FEELS PRESSURE,NO PAIN. ALSO, C/O BLOOD IN STOOL X 1 YR . HAS LOST 20 LBS. IN PAST 1 YR . HAS BEEN USING CITROCEL. HX. COLON CA-FATHER .

Figure 3.2: Example of clinical narrative taken from VHA Hospital VISTA EHR.

This text snippet was tested on several systems which are built using statistical NLP models and a processing pipeline consisting of sentence segmentation, tokenization, POS tagging, phrase chunking, and word normalization, with the addition of concept annotation. The accuracy was poor with respect to all aspects of the pipeline, although some systems performed better than others.

For example, the clinical Text Analysis and Knowledge Extraction System (cTAKES) uses a language model trained on clinical texts and automatically extracts medical concepts from it [6]. It found most of the noun phrases, but only one of the verbs in the free-form text. It was not able to segment most of the sentences correctly due to the misuse of periods and missing white space. Many annotations of the anatomical features and medical concepts were mislabeled.

Another characteristic of the VA text is the use of nursing flow sheets. These are templated forms that are provided by the EHR to assist nurses in patient care documentation. They consist of a textual prompt with a fill-in-the-blank or drop-down-list response area. The text of these forms and the responses are then copied into the patient record. These templates are extremely institution specific and difficult to process [12]. Figure 3.2 shows a snippet from a pain assessment. In the case of a complete pain assessment flow sheet, a single pain event spans roughly 60 lines of text. Attributes of the pain event are recognizable in this text because they are collocated with other pain attributes — not because of sentence structure. Notice also, that in the line that begins *Pain:*, the word pain is just a section header, not an indication that a pain event has occurred. Its actual occurrence is dependent on the presence of valid values in the accompanying attributes.

Pain:
Patient reports pain during shift: Yes
Type of Pain:acute (<3 mos)
LOCATION:RT LOWER FLANK AREA
QUALITY:Aching
SCALE:3
PATIENT'S ACCEPTABLE LEVEL OF PAIN:3
Patient reports pain interfered with ability to perform ADLs:No
Alternative pain relief measures offered:PT'S FLANK AREA SUPPORTED BY PILLOWS

Figure 3.3: Example of nursing flow sheet from VHA Hospital VISTA EHR.

Only 5% of the text of the clinical notes coming from the VA VistA EHR is grammatically clean. The other 95% does not respond to the natural language pipeline or to other statistical techniques due to its lack of grammatical conformity.

However, there are some characteristics of the VA text that make it amenable to automated extraction. Although the clinical text does not follow formal rules of grammar, there are patterns and structure surrounding the raw features that relate to a particular concept. For pain, almost all occurrences of text in the form *8/10* that co-occur with *pain* are likely to be pain severity scores. Likewise, bigrams such as *denies pain* and *no pain* occur very frequently and completely characterize the patient's experience of pain at that point. Other words that describe the quality of the pain, such as *sharp*, and that appear in close proximity to the word *pain* have a high probability of indicating an actual pain event.

In line with the analysis by Li *et al.* [29], context is the most discriminative feature. Although capitalization features are largely lost, the contextual cues in the

text are retained. Phrasal structure and scope is even more microscopic, creating the leverage that enables the identification and extraction of medical concepts through the use of proximity, location, and scope.

3.3 Overview of Conceptual Model

This research addresses the short-comings of clinical text processing systems discussed in Chapter 2. The overall approach is vertical rather than horizontal. That is, current systems employ a horizontal process that recognizes every possible medical concept in the text, regardless of context, and matches it with an entry in a biomedical nomenclature. They are able to identify general attributes such as negation and historicity. However, these systems do not address the deeper semantics of those concepts and their particular attributes, only the fact that they appear. For example, in the text *pain 8/10*, these systems can correctly identify *pain* as a concept and label it with its correct CUI, but can only recognize *8/10* as a numeric fraction instead of as the severity level of the pain event. To extract semantically related attributes of particular concepts requires post-processing.

Instead, this research takes a vertical approach in which the semantics of the medical concept are well defined in advance along with all of its related attributes. In conjunction with this, local components are built which map from the lexical space to the semantic space by identifying both the phrasal cues that define the context and the lexical terms that define the values of the attributes. I am calling this a Lexico-Semantic approach. The goal, therefore, was to design a modular, composable,

semantic representation of medical concepts of the class *signs and symptoms* that can be used to perform semantic search and concept extraction from clinical records, that is as fine-grained and precise as possible.

In particular, this research focused on discovering and mapping the lexical and semantic patterns used in recording pain events in the patient record, and using them to build a Lexico-Semantic model for pain. Pain was targeted as a proof of concept because of the physicians' strong interest in it and because of its prevalence in human experience. Merriam-Webster defines pain as "a lack of well-being ... that ranges from mild discomfort ... to unbearable agony..." This results in a very wide variety of ways in which the concept of pain and its attributes may be expressed. Pain also has attributes beyond those of other symptoms; namely, the type of pain (*gas,cancer-related,surgical*), and its quality (*sharp, ache, burning, cramp*). Additionally, pain severity can be noted in the chart using a numeric scale (0-10), or it may be expressed with terms such as *little*, *annoying*, or *excruciating* which may or may not appear in any standard medical terminology. These expressions must be normalized to standardized terms such as *mild, moderate*, or *severe* if the data are to have practical use in analysis.

While the research system focused on pain, the same approach could be applied to other common symptoms, such as swelling which has similar characteristics. For example, the terms that express swelling (*swell, swelling, swollen, edema, oedema, turgescence, tumefaction*) are all easily added to the lexical database. Likewise, terms that describe the type of swelling (*superficial, local, localized, generalized, widespread, pitting, heat, cyclical, traumatic, inflammatory, congenital, neoplastic*)

are also easily added. Attributes of severity, location, onset, duration, and time all apply directly to swelling without modification or extension.

In contrast to other current approaches, this approach focuses on identifying a particular concept with all of its modifying attributes and returning it in a structured format that can be consumed by other applications. This is a common use case, for example, in monitoring particular health conditions in a patient record, patient cohort identification, disability status determination, interactive search, and clinical decision support, in which only a small set of concepts are deemed significant.

This approach is summarized by the following:

- Formally define the domain of the desired concept and encode it in a semantic representation.
- Discover the lexical context and cues surrounding its components.
- Map the lexical context cues to the corresponding components in the semantic domain.
- Use the mapping to guide the extraction of the concepts from the text.
- Package the components of each extracted event into the structured semantic representation so that it may be consumed in clinical decision support applications.

There are three major components needed to accomplish this.

Semantic The first is an ontological component which defines structural semantics and its relationships to other components. It may also codify domain knowledge.

For example, a pain severity of *0* is equivalent to a normalized severity level of *None* and a certainty of *Definite-Negated*.

Lexical The second component provides the mapping between the surface forms of the lexical expressions to their semantic counterparts.

Extraction Engine The third component is an extraction engine that interacts directly with the text. It uses the mapping component to recognize and translate concepts and attributes from the lexical domain into its structured representation in the semantic domain.

These triple-sets form composable building blocks. For instance, onset and duration can be reused as attributes of many other medical concepts that are of the type sign and symptom. The upper level mapping ontology defines the main medical concept of interest. It also imports the subontologies for its constituent attributes such as body location and severity.

Some mapping components contain a single lexical term such as *pain*. Other mappings require a recognition mechanism that employs complex regular expressions. Still other mappings require external sources. In this case, the mapping component employs resources such as terminologies, lexicons, and other ontologies. For example, the anatomical location mapping component contains references to terms from the Foundational Model of Anatomy (FMA) [15] and SNOMED-CT [35]. Still other mappings incorporate entirely separate algorithms such as temporal parsers.

Once the extraction is complete, the pertinent information is encoded as an individual instance in the ontology that contains the semantic representation of the

medical concept. This includes the reference to the document level information, normalized concepts identifiers, the original text found, and the line number and text span in which it was found in the note.

This research required five main thrusts in order to implement this approach. It included knowledge representation, entity recognition, information extraction, machine learning, information retrieval, and clinical decision support. Each of these is discussed in detail in Chapters 4 through 8.

Chapter 4: Semantic Design

4.1 Knowledge-Based Systems

Knowledge representation and reasoning focuses on methods for providing high-level descriptions of the world that can be used to build intelligent applications. In this context, "intelligent" refers to the ability of a system to find implicit consequences of its explicitly represented knowledge. Such systems are therefore characterized as knowledge-based systems. [36]

This section describes how knowledge representation and reasoning technologies were leveraged to produce a semantically rich cognitive search assistant [37] [38].

4.1.1 Knowledge Representation

Knowledge representation (KR) allows a model of the real world to be built using a set of symbols. By choosing a specific knowledge representation, the model is bound by the symbols in that system as to how it is able to express itself. This is called an "ontological commitment". For this reason ontologies are the heart of every knowledge representation [39]. They also provide a means to create new knowledge by manipulating existing knowledge through a process known as *reasoning* and

inference [40].

4.1.2 Description Logics

Formal logics represent knowledge that machines can interpret and then perform automated inference using sound, tractable reasoning methods. In general, a *description* is an expression in a formal language that defines a set of instances or tuples. A description logic (DL) is a syntax for constructing descriptions and a semantics that defines the meaning of each description. They are general purpose languages for knowledge representation and reasoning, and are suited for many applications. They are especially effective for domains where the knowledge is organized along a hierarchical structure based on the **is-a** relationship. This has motivated their use as a modeling language in the design and maintenance of large, hierarchically structured bodies of knowledge as well as their adoption as the representation language for formal ontologies [36]. They have two main components:

- **T-box: Terminological box** concepts, classes, relations
- **A-box: Assertional box** individuals, constants

Inference is the process of deriving new information from existing information. The purpose of properties (relations) in a DL is to enable inference. Table 4.1 shows the properties of logical inference with examples of their use.

DL languages are a subset of first order logic that are provably complete over a subsumption hierarchy. However, for tractability and decidability, its expressiveness must be restricted to subsets of that logic.

Table 4.1: Logical Inference Properties

Transitive Property
Infers a relationship between two unrelated nodes.
*if $a < b$ and $b < c$, then **infer** that $a < c$*
Functional Property
Infers that two nodes are the same, because each object instance may only have one value for that property.
*if Bob hasMother Maggie and if Bob hasMother Margaret, then **infer** that Maggie is the same individual as Margaret*
Inverse Functional Property
Infers that two nodes are the same. As the inverse of functional, each unique value of the property may belong to only one object.
*if Mike hasId 1234 and Michael hasId 1234, then **infer** that Mike and Michael are the same individual.*
Symmetric Property
Infers an additional relationship between two related nodes. *if Joe isMarriedTo Kathy then **infer** that Kathy isMarriedTo Joe*
Asymmetric Property
Prevents a symmetrical inference.
*if $a < b$ then **infer** that $b \not\leq a$*
Reflexive Property
Infers of an additional relationship of one node back to itself.
Peter knows Peter
Irreflexive Property
Prevents a reflexive inference
***infer** Bob is NOT the father of Bob, i.e., Bob cannot be his own father*

4.1.3 OWL

The Web Ontology Language (OWL) became a World Wide Web Consortium (W3C) [41] recommendation in February 2004 that implements a description logic.

The standardization of OWL sparked the development and/or adaption of a number of reasoners and ontology editors. OWL is extensively used in the life sciences community, where it has become a de facto standard for ontology development and data interchange [42].

4.1.4 Rules

Semantic Web Rule Language (SWRL) [43] is a language that combines OWL with the Rule Markup Language (RuleML). The rules are of the form of an implication between an antecedent (body) and consequent (head); when the conditions in the antecedent hold, then the conditions in the consequent must necessarily hold. From this, additional relations between objects are inferred.

4.1.5 Reasoners

Reasoning systems take as input a knowledge base (KB) (an ontology) consisting of a conceptual schema (T-box) and a set of individual instances (A-box). They are then able to answer queries regarding both the concepts (e.g., all concepts descending from *animal* such as *bird*) and the individuals (e.g., all individuals who possess the properties of a bird). The queries return both knowledge that has been explicitly stated in the KB, and additional knowledge that has been inferred by the reasoner [44].

4.2 Semantic Structure of Existing Ontologic Resources

There are several clinical nomenclatures including SNOMED-CT [35] and the National Cancer Institute Thesaurus [45]. Both of these are available in a flattened form that can be ingested into a relational database, and as a formal OWL-based ontology. These ontologies have hundreds of terms related to the concept of *pain*. However, that great number is a result of the use of pre-coordinated terms that results from an unnormalized (in the database sense) ontology.

Figure 4.2 shows an example of the taxonomy over the term *complaining of pain in toe (finding)*. Notice that most of the taxonomy is not related to pain, but rather it mirrors the anatomical taxonomy of the limb down to the toe even though these terms are already defined in the antomical portion of the nomenclature. The ontology has other relations defined that links the pain to its subject (experiencer) and its location.

Likewise, Figure 4.2 shows the entry for pain from NCI. Notice that all the direct descendents of the pain concept are made up of combinations of its modifying attributes including location (*abdominal*), severity (*annoying, agonizing*), type *breakthrough*, quality (*ache, crushing*) and duration (*acute*). This is because the ontology contains only **is-a** relations.

The use of pre-coordinated terms increases the size of the terminology, but limits the coverage. For example, SNOMED-CT has a concept defined for *chronic neck pain*, but not for *acute neck pain*. For this reason, SNOMED-CT is now moving to a post-coordinated structure that a defines relations between concepts

semantically rather than lexically. For the same reason, this research implemented a post-coordinated structure in its representation for the symptom (i.e, finding) of pain.

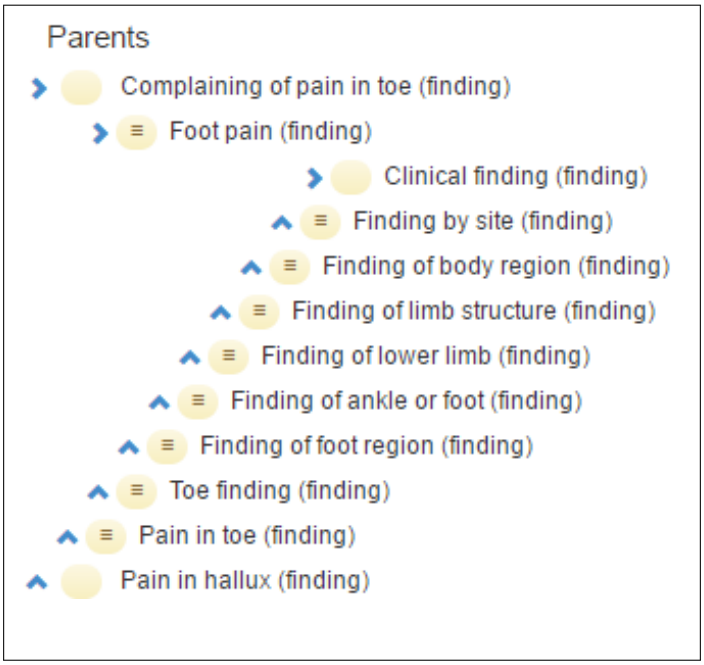

Figure 4.1: This entry shows the use of precoordinated terms over the anatomical taxonomy for toe.

4.3 Pain Ontology

4.3.1 Requirements for Representation and Reasoning

In modeling the domain of pain in the context of the clinical record, the following are required for its representation and reasoning:

Representation:

- Representation of concepts (classes, objects)

For example, the following concepts are needed: *patient, clinical note, clinical*

Figure 4.2: This entry shows the use of pre-coordinated terms over all the modifying attributes of pain.

events, symptom events, pain symptom events, onset, duration, anatomical location, pain type and *pain quality*.

- Representation of subsumption hierarchy over the concepts (**is-a** relations)

  For example, *a SeverityScoreNumeric* ***is-a*** *SeverityScore* ***is-a*** *Severity*. Concepts that are a child of another concept share all the restrictions and relations defined on their parent.

- Arbitrary relations (non **is-a** roles) between concepts

  Pain events must have arbitrary relations to their modifying attributes. For example, a *PainEvent* object ***hasSeverity*** relation with a *Severity* object.

- Arbitrary relations between concepts and literal data values

Pain severity may consist of either a literal string lexical value (*"some"*) or a literal numeric value (*3*).

- Restrictions on classes and equivalence between classes

  Equivalence allows the pain concept in this ontology to be tied to an equivalent pain concept in one or more existing clinical ontologies.

- Representation of individuals instances of concepts, their relations, and data values

  It is not enough to model and represent the structure of the domain. In cognitive search, individual data instances of the pain events must be individually extracted and transformed into the their semantic representation in the A-Box.

- The representation language must accommodate the dynamic composition of concept definitions.

  Because reasoning in a knowledge-based systems is polynomial in the size of the ontology [42], and interactive search requires near real-time response, the representation language most be able accommodate dynamic composition. This allows for the smallest possible ontology when specific search terms are specified.

Rules:

- Representation of inference rules over concepts, relations, individuals, and literal data values above and beyond the OWL axioms.

Specialized inference needs to occur over numeric severity scores in order to equate them to normalized severity concepts such as *SeverityNormMild*. While this is possible in the language representation, it cannot be handled by the reasoning engines using normal knowledge representation axioms. Instead, it can be accomplished using a rule processing inference engine.

Reasoner:

- Reason over concepts (classes), roles (relations), and individuals (instances).
- The reasoning must be sound and complete.
- Represent and reason over data property values (literals).
- The reasoner must support SWRL rules.
- The reasoner should support incremental reasoning.

4.3.2 Implementation Environment

The choice of a knowledge representation language in this research was guided by the expressiveness needed to represent the concepts in this domain, the representation languages used in existing biomedical ontology resources, current standards bodies recommendations, and by the available tools such as editors, query engines, and reasoners.

After analyzing these requirements, the following tools were chosen for the implementation:

- OWL/DL as the representation language for the ontologies

- SWRL as the representation language for the rules
- Pellet [46] as the reasoning system
- Jena [47] Java API framework for semantic web and Linked Data applications

Pellet was able to handle the expressivity requirements, and in particular, the ability to reason over the A-box with numeric and date literals. In addition, it was the only classifier that implemented incremental classifications [48] so that the reasoning process would not have to start from scratch each time new data is added to the model.

4.3.3 Ontology Implementation Details

A set of ontologies were created to capture the semantics of events, symptom events, and pain symptom events to enable cognitive search in the clinical record. Additional ontologies were created to model the attributes of the symptom events. These ontologies are fully normalized and can be composed dynamically using the import feature of OWL. This minimizes the footprint to just those symptoms and attributes that are required by the downstream application in order to keep the extraction and inference as light and agile as possible. Because of this, higher levels components are not aware of concepts or relations defined in lower level components.

In order to provide provenance and the ability to drill down, the location (span) of each extracted value in the text is maintained. Bindings map concepts to standardized terminologies whenever possible.

Appendix A contains the complete set of ontology definitions.

Figure 4.3.3 shows the ontologies that were developed and how they connect together in a graphical structure through their relation definitions. Each bubble is a separate concept. Data properties, such as *hasName* are not depicted. Each color represents a separate ontology component. One or more separate ontologies may combine into a single ontology using the OWL import feature.

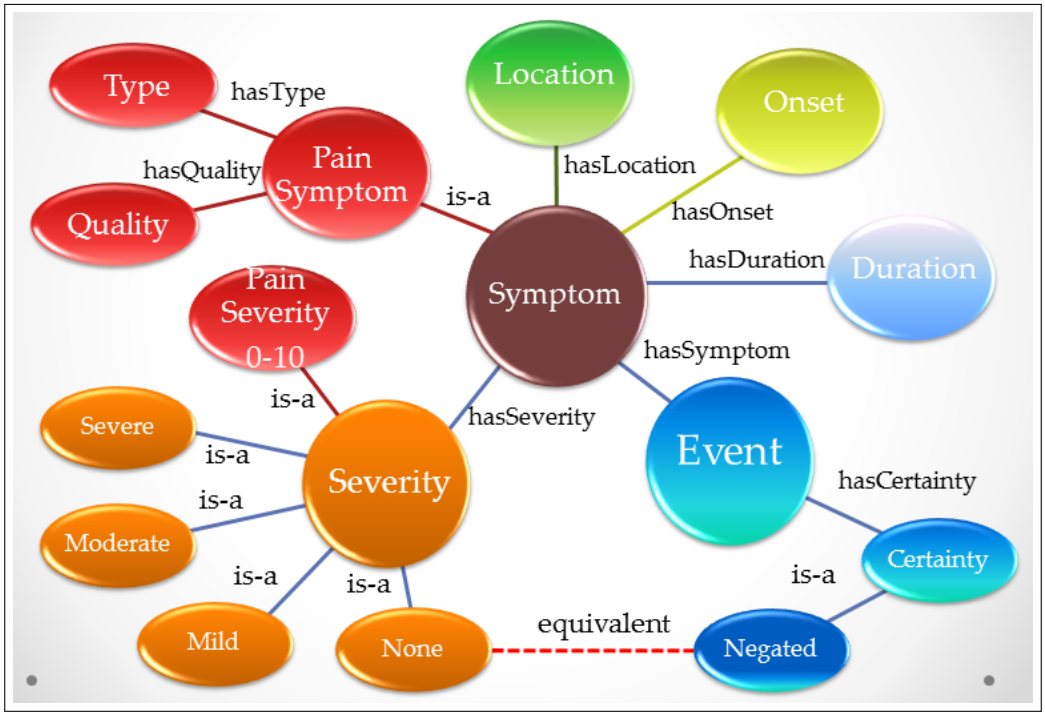

Figure 4.3: The set of ontologies created to model the pain symptom event. Each bubble is a separate concept. Each color represents a separate ontology. They are connected together in a graphical structure through their relation definitions.

4.3.3.1 High-level Concepts:

Patient: In the prototype system, patient attributes were kept to a minimum. In an EHR, this data would be stored in a back-end database. Because the research corpus has been de-identified according to HIPAA rules, it contains no per-

sonal information. Therefore, patient attributes were synthetically generated and modeled.

Note: The *Note* concept models a clinical note. Attributes were defined based on the header information that appeared in each note of the research corpus and those of the MIMIC II database [49].

4.3.3.2 Events

Semantically, a clinical event covers a broad range. It may be a *symptom/finding* event as in this research, as well as a *disease/disorder* event. However, it may be other types of events that occur in a clinical context as well; for example, moving a patient from one floor in the hospital to another. As such, in the clinical text processing community generally associates two attributes with an event: the date/time, and the certainty of whether it actually occurred. Other attributes that are commonly modeled are: historicity (*history of*), subject (*patient,family member,guardian*), conditionals (*if this happens*). As this research did not focus on these general attributes, they were not included in the model.

Therefore, the two concepts that were modeled are events and certainty status. The **Certainty** ontology accommodates concepts of affirmation (*Existence, Negation*) and certainty *(Definite, Probable). Existence* and *Negation* are disjoint, as are *Definite* and *Probable*. The other four classes (*CertaintyDefiniteExistence, CertaintyProbableExistence, CertaintyDefiniteNegated*, and *CertaintyProbableNegated*) are convenience classes that reflect the pre-coordinated categories used in the extended ConText information extraction algorithm as described in Chapter 5. Figure 4.3.3.2

shows the graphical structure of these concepts.

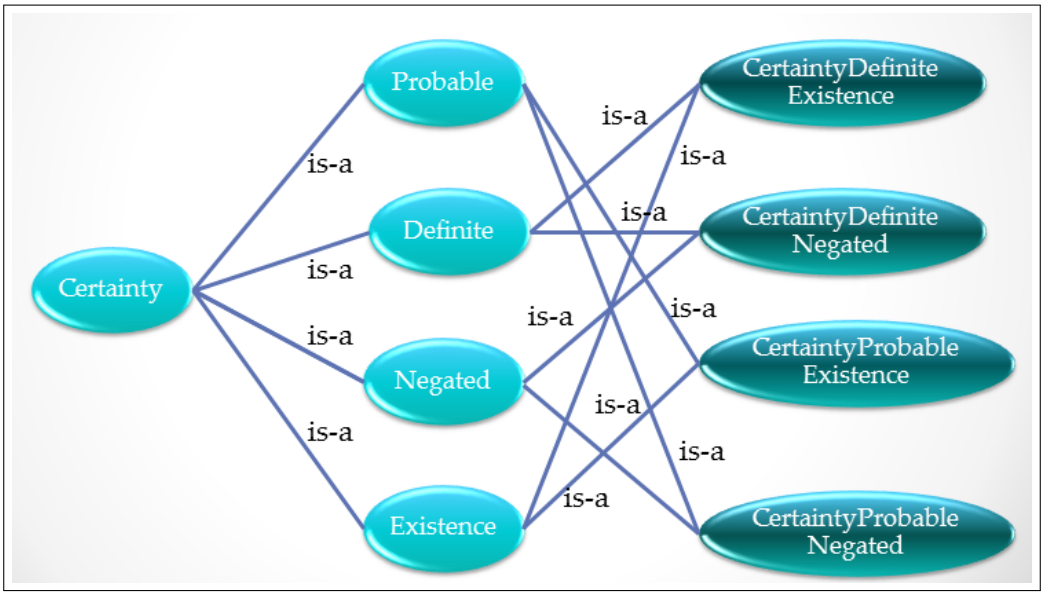

Figure 4.4: Events may be associated with their affirmation/negation and certainty, i.e., definite/probable.

4.3.3.3 Symptoms

As can be seen in Figure 4.3.3, a symptom may be associated with zero or more attributes: *Location, Onset, Duration*, and/or *Symptom*. Location, onset, and duration have minimal requirements consisting of a single string-valued data property. For onset and duration, this property may specify a temporal element (*yesterday*) or a situational element (*during exertion*). Location may have one or more anatomical locations specified (*both hands*).

Severity has a more complex structure because it must accommodate both numeric and lexical values. In addition, each of those values must be normalized to a semantic subclass of: *None, Mild, Moderate, Severe*. Another important facet is that

a severity of *None* is semantically the same thing as having a certainty of negated; so *SeverityNormNone* has been declared equivalent to *CertaintyDefiniteNegated*.

However, there are some symptoms, such as pain, that use a specialized set of values to indicate severity. One such is the $0 - 10$ pain scaled used in this corpus whose numeric range is strictly defined. Although it is a subclass of *Severity*, it is defined in a separate PainSeverity ontology under the design assumption that higher-level ontologies should not be burdened with concepts that are not needed in the general case.

Figure 4.3.3.3 shows the PainSeverity ontology. General severity concepts are visible because the Severity ontology was imported. The additional *PainScale_0_to_10* concept is declared as a subclass of *SeverityScoreNumeric* from the Severity Ontology. Notice also, highlighted in blue, the equivalence relation between *SeverityNormNone* and *CertaintyDefiniteNegated* as described above.

4.3.3.4 Pain Symptoms

Pain is a type of symptom. Therefore, the *PainSymptom* class is a subclass of *Severity* and shares all of its properties and relations. However, pain has two additional attributes that must be modeled: pain type (*cancer-related, surgical*) and pain quality (*sharp, ache*). Functionally, it ensures that *PainType* and *PainQuality* can only relate to a *PainSymptom* and not to a general *Symptom*. *PainType* and *PainQuality* consist only of their relation to *PainSymptom* and the pain expression used in the text.

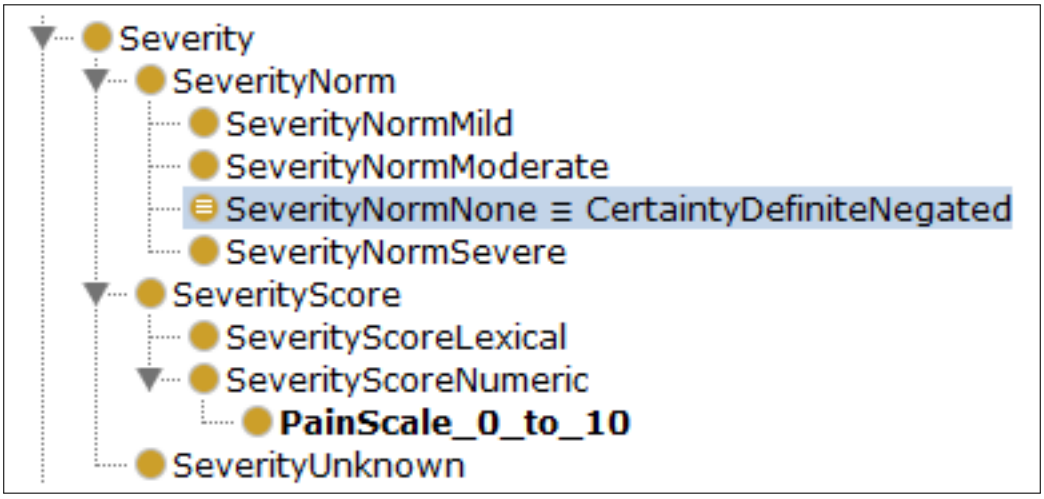

Figure 4.5: *PainScale_0_to_10* in PainSeverity has been declared as a subclass of *SeverityScoreNumeric* in the Severity ontology. An equivalence relation between *SeverityNormNone* and *CertaintyDefiniteNegated* is also shown highlighted in blue.

4.3.3.5 Summary

A set of fully normalized ontologies were designed to model pain symptom events in the clinical notes of a patient record taken from an EHR. These ontologies are modular and dynamically composable so that only the particular attributes that are desired in a clinical decision support application are used, allowing for the footprint to be as small as possible. This has performance implications in both information extraction and in reasoning.

Chapters 5, 6, and 7 detail how occurrences of pain events in the VA corpus are extracted in conjunction with the semantic model described here. Chapter 8 shows how the semantic model coupled with the extracted events can be used within a reasoning system to provide queries that can drive a clinical decision support system.

Chapter 5: Information Extraction

5.1 ConText Algorithm

The original ConText algorithm [21] extended its predecessor, NegEx [50], a stand-alone negation detection system. NegEx was successful, providing a 95% precision and 78% recall on determining whether medical concepts were negated or not when tested on 1,000 sentences taken from discharge summaries. The researchers' conclusion was that, "a simple regular expression algorithm for determining whether a finding or disease is absent can identify a large portion of the pertinent negatives from discharge summaries." It has become the *de facto* standard for negation detection in clinical text and has been incorporated into cTakes [6] and the National Library of Medicine's MetaMap [51]. One particularly important aspect of it in regard to this research, is that this algorithm did not rely on syntax or conformance to an underlying grammar in the text.

The ConText algorithm extended NegEx to include three more contextual attributes: hypthetical (*if he develops ...*), historical (*history of*), and experiencer (*patient, other*). It also extended the scoping mechanism. In NegEx, scope was defined by a hard window of characters around the targeted term. ConText allows the scope to be defined by semantic/lexical cues, such as the end of a sentence or the presence of conjunctives in the text (*he did not have dyspnea,* ***but*** *did have ...*).

Each phrasal cue is assigned to a semantic category that is defined in the supplied ConText lexical item database, or may be defined by the application using it.

The ConText algorithm defines two sets of phrasal patterns (i.e., items):

- **Targets**: patterns that recognize the main medical concepts of interest in the text.
- **Modifiers**: patterns that recognize attributes that modify those targets. Modifiers can look either backward or forward in the sentence for a target to attach to.

ConText comes with a database of target and modifier items. Each item contains a regular expression, a semantic category, and a direction in the case of modifiers. These expressions have been well tested on grammatically clean text.

Figure 5.1 shows two examples of how the algorithm works on a line of text from the VA corpus. The target term is pain; the scope for all target terms is the entire line. The *mild* modifier is defined to look forward. It finds the target *pain* within its scope, and the modifier rule is applied that attaches the modifier to the target. If *mild* had appeared after *pain* in the sentence, the rule would not have applied. The numeric term of *6* works similarly, but looks backward.

As the algorithm finds targets, its attaches modifiers to them by building a directed graph. As used in this research, each sentence constitutes an entire document, and a tree is formed in which the sentence is the root. Each target expression is a child of the sentence, and each modifier is a child of its target. Conjunctive/terminating expressions are children of the modifier whose scope they are

ending. However, more generally, this algorithm could be used to build a knowledge graph of a more complex structure containing notes, sections within notes, etc.

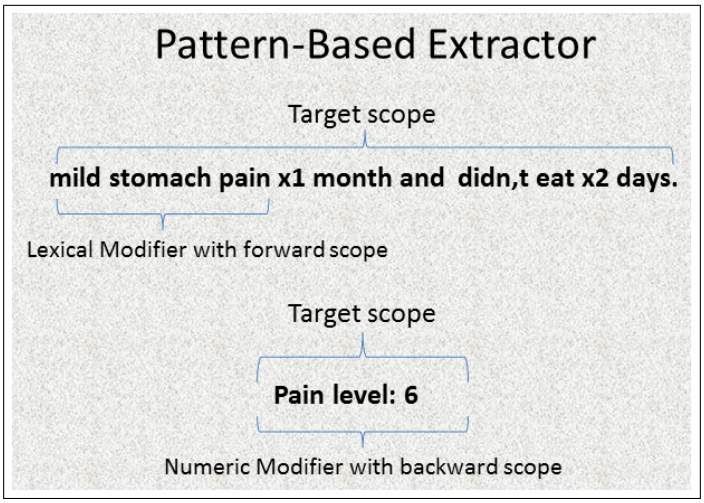

Figure 5.1: Two examples of how the ConText algorithm works on a line of text from the VA corpus.

Procedure 5.1 shows how the algorithm works, one sentence at a time.

5.1.1 Limitations of ConText Algorithm

The ConText algorithm is limited in the following ways.

1. The algorithm is limited to those cues and concepts which can be specified in a regular expression.

   (a) It is not able to find attributes such as anatomical locations which may need to rely on an external lexicon or other terminology resource.

   (b) It is not able to find attributes that are based on a more complex recognition algorithm such as temporal expressions which require a more sophisticated date/time parser.

Procedure 5.1 ConText Algorithm

```
for each sentence in note do
  for each target pattern do
    Save each target occurrence in sentence
  end for
  for each modifier pattern do
    Save each modifier occurrence in sentence
    Set scope of that occurrence based on its direction and on the presence of
    terminating punctuation or expressions
  end for
  for each target occurrence do
    for each modifier occurrence do
      if target occurs within modifier's scope then
        Index and return the target and modifier occurrences
      end if
    end for
  end for
  Build bidirectional context graph between occurrences of the targets and their
  modifiers
end for
```

2. The algorithm assumes that all of the modifiers are applicable to all of the targets. When additional types of semantic categories are added to the pattern database, it is not able to restrict which semantic target types are allowed to combine with semantically compatible modifier types; rather, each modifier can attach to any target based only on scope. For example, it allows an anatomical target to be attached to a temporal modifier.

3. It only operates within the scope of a single sentence on a single line. If the sentence is broken into separate lines (as in the VA corpus), it will not consider any scope beyond the `newline` character. In the nursing flow sheets in the VA corpus, target concepts may have modifiers that appear over a span several lines.

4. It assumes that all modifiers and targets are defined in advance and read in when the algorithm starts. For more complex semantic types, modifiers and targets may need to be created dynamically.

5. The item database of targets and modifiers contain expressions that are useful in grammatically clean text; relatively few of them are appropriate for expressions used in the VA corpus.

Despite these limitations, the basic algorithm is very flexible and has been extended in this research to address these limitations and accommodate a much deeper and richer semantics.

5.2 ConText Algorithm - Extended

The ConText algorithm was extended to accommodate the deep semantics of pain in the context of grammatically ill-formed clinical notes in order to be used in a semantically rich cognitive search assistant and for clinical decision support applications.

The following sections detail the extensions that were made by examining each of the following semantic components: pain, severity, location, onset, duration, pain quality, pain type, variability, and time.

5.2.1 Preprocessing of Corpus Text

As all the patients and their notes are contained in a single file in the research corpus, the text is broken up into its individual notes. This is done by scanning the text for note headers. This is important because the header contains the date/time

that the note was created. Many events that are recorded in the note, therefore, are relative to this time. The header format of these notes is specific to the VA and is differs based on whether it is a clinical note, a radiology report, or a lab report. Therefore, it is assumed that this process would be performed separately from the search and extraction process. Notes from other institutions would need to have their own header pre-processing code.

Other than breaking up the text into individual notes, the only preprocessing performed on the text is to convert it to lower case. It was discovered during the course of this research that punctuation (although grammatically non-conformant) provides a strong discriminative signal, and so it is kept.

The VA text is strongly line-oriented, and it is processed one line at a time. In most cases the newline characters have strong semantic meaning. The only case when this is not true is in narrative text. However, in the case of the short-hand style narratives, these `newline` characters provide a backstop for the scoping mechanism in the algorithm. There are, however, two concessions made in this regard. The algorithm is allowed to "peek" around the corner by appending two word tokens from the next line onto the end of the current line. The second concession comes when it is determined that a modifier on the current line is looking forward to a target on the next line. In this case, the text containing the modifier expression is prepended onto the following line.

5.2.2 Pain

Even though pain is addressed in this research as a single semantic concept, it incorporates many expressions both in the general sense and as a semantic parent to more specific pain concepts such as *myalgia*. In order to provide as much coverage as possible, terms expressing pain and its characterization were scraped from a wide variety of sources including SNOMED-CT, NCI Thesaurus, medical dictionaries, clinical text, and medical web-sites.

Many of these terms fit very well with the existing ConText architecture using regular expressions. Regular expressions accommodate word morphologies easily. For example, one term can express *pain, painful* and *painfree* using the regular expression `pain(full|free)?`.

Notice however, that the occurrence of *free* in *painfree* indicates a special case in which the negation modifier occurs as part of the target term. There are other special cases as well. Terms that might be regarded by the lay person as synonyms for pain, such as *ache* or *cramp* were annotated by the professional clinicians as being a pain quality modifier. However, in the text, they are used as synonyms and thus they are both targets and modifiers at the same time. Another special case are compound words indicating pain such as *headache* or *backache*. In this case, the location modifier is included as part of the pain/qualifier target term. In the case of *myalgia*, the location modifier is implicit in the meaning of the term coming from its Greek root of *mys* for muscle and *algos* for pain. Another issue was the use of abbreviations. Two of the most common were for *chest pain* (CP, C/P) and

headache (HA, H/A).

For these reasons, the ConText algorithm that relied on a predefined set of regular expressions with clean separation between targets and modifiers proved too limiting. Because it was non-trivial and computationally complex to automatically data-mine these terms and their anatomically related values from SNOMED-CT at runtime, a specialized pain lexicon was developed. Pain terms along with their morphologies, abbreviations, associated anatomical locations, and CUIs were included in it. These terms are read in during initialization and used to create a dictionary (i.e., hashmap) data structure that is kept in memory for the duration of the extraction.

When processing a line of clinical text, a regular expression was created from each term in the specialized lexicon. If that term was found, a target item was created dynamically and an occurrence of that target was added to the underlying context graph. In addition, if the pain term had an associated anatomical location, a modifier item was created dynamically and an occurrence of the modifier was added to the graph.

Pain terms may occur in the text without actually indicating a pain event in the patient record. For example, *Pain:* may occur as a section heading. For this reason, pain target term occurrences that do not have any modifiers attached are not included in the results.

In addition to recognizing and extracting appropriate pain terms, other phrasal patterns involving pain had to be filtered out. There are many different facets involved in treating a patient with pain, but these are not needed or desired in the

results. For this reason, an additional category called `PseudoPain` was created. This includes expressions such as *pain management regimen* or *instruct patient to report pain*. If a pain target appears within the span of these phrases, it is ignored.

Table 5.1 shows a few of the results from the pain extraction algorithm.

Table 5.1: Pain Extraction Expressions

Pain Terms	**Text**
True Positives	
pain	patient states he/she does have **pain**
chest pain	denies cough/sob/**cp**/
headache	denies any focal weakness or **headache**
ha (headache)	Denies any f/c, no NS, no **ha**/sz/dizziness,
dysuria	denies cough, diarrhea or **dysuria**
False Positives	
myositis	raising the question of a **myositis**. (*hyothetical*)
sore	explained including but not limited to **sore** throat, (*hypothetical*)
False Negatives	
pain	dnied any **pain** (*denied is misspelled*)
pain	**pain** on right leg raise (*leg appears on following line*)
pain	-simethicone for gas **pain** (*filtered out as a medication line*)
pain	c/o **pain**7-8/10 . fs=142 @ 6am/124 (*missing white space after the word pain*)

5.2.3 Pain Severity

Severity modifiers in general are lexical expressions, although they may also be numeric. Lexical expressions were scraped from various sources and included: *some, a lot, minor, little, not too bad, agonizing, annoying, crushing, debilitating, excru-*

ciating, heavy, high, high amount, high levels, intense, intractable, mild, minimal, moderate, negligible, unremittent, same, severe, significant, tolerable, torturing, unbearable and *unpleasant*. These terms were each associated with a normalized value of: *Mild, Moderate* or *Severe*.

In the case of pain severity, there are several different scoring methods in use. However, the most common one for adults, and the one used in the VA corpus, was the $0 - 10$ pain scale.

Numeric quantities are prevalent in clinical text. While it is relatively easy to find numbers ranging from 0 to 10, it is much more difficult to find those numbers while filtering numeric values not related to pain severity. Recognizing this score was accomplished using a regular expression – albeit a complex one: \b(?<!/x)(\d(\.5)?(?!\.)—10)(?!\))(-(\.5)?(?![\.)])—10)?(?:/10)(?!/)\b. Numeric pain severities looking backward are slightly different.

Figure 5.2 breaks down the regular expression for pain severity and explains how it works. Figure 5.3 presents some of the positive and negative results.

Table 5.2: Numeric Pain Severity Regular Expression

Regular Expression	**Meaning**
`\b`	must begin on a word boundary
`(?<!/)`	cannot be preceeded by a forward slash
`(\d(\.5)?|10)`	a single digit, possibly followed by *.5* or a *10*
`(?!/)`	not followed by another slash
`(?:/10)`	followed by */10*
`(?!/)`	but not followed by another / (as a in a date)
`\b`	ending on a word boundary

Table 5.3: Pain Severity Expressions

Pain Severity	**Text**
True Positives	
little	reports very **little** abdominal pain this am
tolerable	abdominal pain beginning abruptly at 630a which was **tolerable**
some	diarrhea as well as **some** abdominal pain.
significant	no **significant** pain issues
mild	patient is in **mild** discomfort
7	current pain level:**7**
0	pain score(scale 0-10): **0**
7	scale:7
3	stated pain level is **3**/10
3	pain score at discharge: **3**
6	how bad is the pain at its worst/most? **6**
2	how bad is the pain at its best/least? **2**
4	pain:4
0	patient rates their current level of pain at: **0**
0	pain score:**0** (02/05/2010 08:30)
8.5	s: pain: **8.5**/10 b hands, b shoulders
False Positives	
2	6) onset (when did the pain start?): **2** days ago (*2 is attaching to pain*)
2	mmt: shs **2**-/5 (*physical therapy strength measure*)
1	alternative pain relief measures offered: **1** tab of percocet (*1 is attaching to pain*)
1	location **1**: (*flow sheet is asking for the first pain location*)
False Negatives	
None	

5.2.4 Negation

Negation is closely related to severity. In fact, having a negated occurrence of an event is equivalent to asserting a severity of *None*. It is also equivalent to asserting a certainty of CertaintyDefiniteNegated or CertaintyProbableNegated. As this research did not distinguish between definite or probable, only the actual negation status was extracted. Negation has the distinction of being the only modifier that distributes over multiple targets: *denies headache/chest pain/shortness of breath.*

Figure 5.4 presents some of the positive and negative results.

Table 5.4: Negation Expressions

Negation	**Text**
True Positives	
denies	pt alert x 4 **denies** any pain
no	pt with **no** c/o pain at this time. await for ct tech.
denies	19:00 pt **denies** any pain
no	cv: **no** chest pain
no	denies any f/c, **no** ns, no ha/sz/dizziness
False Negatives	
denies	pt **denies** fever, chills, shortness of breath, chest pain. (*denies appears on previous line*)
denies	he **denies** pain, only discomfort. (*denies is attaching to both pain and discomfort*)
0, no	SCALE:**0** no **pain** (*gold standard error*)
False Negatives	
None	

5.2.5 Anatomical Location

Anatomical locations were the most difficult of all the attributes due to the large set of terms in the FMA (70,000) and the extreme and varied use of abbreviations. Also, unlike in the SemEval 2015 challenge, one phrase could entail several locations. Table 5.5 shows a few of the ways in which two different locations, *lower right abdomen* and *both hands and shoulders*, appeared in the VA corpus.

Table 5.5: Variety of Anatomical Location Expressions

lower right abdomen	**both hands and shoulders**
abdominal in the rlq	bilateral hands and shoulders
rt lower flank area	b shoulder, b hand
RLQ abd	b/l hands, shs
rt llq	b/l ue
r side	hands, shs 2/2

It is obvious from these examples, that a simple dictionary lookup into an existing anatomical terminology would not suffice. Instead, a further extension to ConText was built based on that which was used for pain. In this case, a full formal anatomical ontology was used as the source of terms - the Foundational Model of Anatomy [15]. By using an ontologic source for the terms, this source could also be used in the reasoning engine to perform inference over the anatomical terms. For example, searching for *upper extremity pain* would return all occurrences of not only hands and shoulders, but elbows and fingers as well.

Three lexical resources were created. The first created a specialized lexicon of each individual word that occurred in the FMA. As there were terms in the text that were not present in FMA, such as *both, bilateral, body, side, generalized*; these

terms were added. The list also had to be expanded to include acquired locations, such as *surgical site, PICC site, site of incision.* One interesting note, is that while the FMA lists terms as nouns (*abdomen*), the text often uses the adjectival forms instead (*abdominal pain*). Fortunately, FMA entries often include "synonyms" that contain the adjectival forms of the entry. For this reason, words from the synonyms were included in the specialized lexicon. The lexicon was also extended to include plural forms as they are largely absent from FMA. For example, there are entries for *left hand* and *right hand* but not *hands.* Figure 5.2 shows an entry from the FMA.

[**Term**]
id: FMA:10429
name: Wall of abdomen proper
synonym: "Abdominal wall" EXACT []
is_a: FMA:20357 ! Subdivision of abdomen
relationship: constitutional_part_of FMA:61680 ! Abdomen proper
relationship: regional_part_of FMA:259054 ! Wall of abdomen

Figure 5.2: Example of a anatomical entry in FMA.

Next, in order to handle abbreviations, the NLM Specialist Lexicon [52] was consulted. The Specialist Lexicon contains entries for a wide variety of abbreviations used in the biomedical community. For example, the entry for *pt* contains sixty different expansions of which two are used in the VA corpus: *patient* and *physical therapy* as seen in *pt agrees with pt plan.* For each abbreviation entry, each one of its expansions was compared against every term in the FMA to find which abbreviations were associated with anatomical locations. For example, the abbreviation

llq expands to *lower limit of quantification* and *left lower quadrant*. In this case, *left lower quadrant* matched the *Left lower quadrant of abdomen* entry in FMA. Each match was added to the specialized anatomical lexicon. When complete, the anatomical lexicon contained roughly 10,300 unique tokens.

Figure 5.3 shows the *llq* entry from the Specialist Lexicon.

base=LLQ
entry=E0690666
cat=noun
variants=metareg
acronym_of=lower limit of quantification
acronym_of=lower limit of quantitation
acronym_of=left lower quadrant—E0699706

Figure 5.3: Example of an abbreviation entry for *llq* in NLM SPECIALIST Lexicon.

As this resulted in the inclusion of many general terms, such as *bed*, a stoplist was created to filter out words that created too many false positives in the location extraction results. When the FMA word list is read in during initialization, the stopwords are removed from the list. One additional list was created for words that were a valid part of an anatomical term, but could not stand on their own. For example, *left arm* would be a valid expression, but not *left*.

Anatomical abbreviations are highly ambiguous in the corpus. However, they tend to occur in close proximity to target terms. For this reason, a different scoping mechanism is used consisting of a hard window in the number of tokens around a target term. However, each multi-word location expression is combined into a single

token from which the window is calculated. Empirically, a window of 4 had the best balance between precision and recall results on this corpus. In this case, target terms may be pain targets (*painful*) or locations indicators from nursing flow sheets (*Location:*).

The location extraction begins by breaking the line into character-based tokens along with commas and plus signs which have important semantics in this context. This list of tokens is correlated and indexed to a list of all the target terms that were found using the normal pattern matching technique. Next, the algorithm iterates through the list of words to find strings that consist of words that exist in the specialized anatomical lexicon. The lexicon is expanded with additional connector words such as *and, of,* and *the*. However, some expressions must be filtered out explicitly, such as, *blood pressure*, *side effects*, *pressure ulcer*, and *surgical site*. The strings are trimmed to remove connector words that begin or end the string. Strings that are made up entirely of terms that cannot stand alone (*right, left*) are ignored.

After this, the distance (measured in number of tokens), is calculated to the target terms. If the anatomical expressions occurs within that window, a new modifier item is created dynamically and an occurrence of that modifier is added to the underlying context graph.

However, there is a scenario in which the anatomical expression could function as a target instead of a modifier. In the following text from a physical therapy note, *8/10 both hands*, the pain target term is never explicitly stated, it is only implied. Therefore the algorithm has to accommodate this by allowing the location to act as a target term for the severity score.

Because the ConText algorithm is not able to determine which modifiers are semantically appropriate for a given target, the algorithm must be extended to accommodate this. For example, in the following text *Location of patient prior to admission: home*, the *Location* target, gets attached to an onset modifier (*prior to admission*). In this case, the algorithm is extended to prune modifiers of location targets from the underlying context graph that are not anatomical terms. Likewise, when an anatomical term functions as a target, it is only allowed modifiers that are a severity expression. This pruning mechanism forms a semantic grammar based on the relations defined in the ontology.

Table 5.6 shows a few of the results from the location extraction algorithm.

5.2.6 Temporal Terms

Like pain and location, temporal terms could not be widely recognized using regular expression patterns. For this, an external algorithm had to be incorporated for parsing dates and times. Because the extraction was performed in Python, the `parsedatetime.py` module was used [53].

However, this module was not able to read some common date/time expressions used in this corpus. For example, the use of @ before a time expression was common, as in *ONSET: Feb 21,2011@08:00* or *no c/o pain or discomfort @0200.* White space in these expressions might also be missing, thereby interupting the pattern. For this reason, `parsedatetime.py` was also extended to accommodate these nuances.

Table 5.6: Location Extraction Expressions

Location	**Text**
True Positives	
left lq	chief complaint:left lq pain
right lower quadrant abd	1day h/o right lower quadrant abd pain a/w chills
rlq abd	episodes of rlq pain which generally last a couple of hrs
urinary tract	denies cough, diarrhea or dysuria (*pain on urination*).
throat	he denies any dysphagia/odynophagia/no reflux (*pain on swallowing*)
chest (pain)	Denies cough/SOB/CP/
drainage site	pain over drainage site worse this am
rt lower flank area	location:rt lower flank area
bilateral posterior head	he also has experience new bilateral posterior ha.
drain site right abdomen	worsening pain around drain site right abdomen
r side	4) location (point to where it hurts): r side
tip of the penis and rt side	soreness in his tip of the penis and rt side,s/p i/d
bilateral hands and shoulders	MSK: + joint pain in bilateral hands and shoulders R >L
b hands and shoulders	s: pain: 6/10 b hands and shoulders.
shs (shoulders)	pt reported pain in shs
shs and hands	7/10 pain in shs and hands
Incomplete True Positives	
right lower	right lower extemity: pain (*extremity misspelled*)
abdominal, rlq and right	abdominal pain in the rlq and right flank (*flank appears on next line*)
False Negatives	
fingers	fingers fully 2/2 to reports of increased pain (*out of the window*)
neck	neck: denies pain (*unknown reason*)

In addition, `parsedatetime.py` was able to find textual expressions of times such as *this morning*, but not terms used in the corpus such as *this am*. Likewise, it finds *last night* but not *overnight*. `parsedatetime.py` also returned false positives for expressions in the corpus such as interpreting the list number *1a.* as *1:00 am*, *129/73 blood pressure* as *1:29*, *kangaroo 924* as *9:24*, and *1/2 cup* as *Jan 2*. ICD codes such as *(icd-9-cm 719.47)* were especially problematic.

As a result, a lexicon approach was implemented for temporal expressions in addition to using `parsedatetime.py`. Terms added to the specialized temporal lexicon include: *suddenly, abruptly, slowly, gradually, currently, recently, at, on, shift, session, arrival, present, since, prior, am, a.m.* and *overnight*.

In order to filter out false positives on numeric expressions that were known not to be temporal (e.g., ICD codes and severity scores), those expressions were disguised by replacing them with tildes (~) before the text was analyzed by `parsedatetime.py`.

The extraction algorithm for temporal expressions was very similar to that of location expressions except that temporal expressions were not restricted to a window. After filtering out known numeric expressions, the text was analyzed by `parsedatetime.py`. The results were saved, and the spans of the recognized expressions from its results were disguised to prevent further processing on them in the remainder of the algorithm. Next, the text was tokenized and *a, of, the, in, within, to, end* were added to the temporal lexicon as connector words. Strings of temporal tokens were identified in the text. Strings consisting only of non-standalone words were ignored. However, temporal items were not created except in the context of the onset, duration, and time attributes which are described in the following sections.

One important restriction had to be put into place for temporal expressions. The use of numerics in the corpus was highly ambiguous and created a large number of false positives. For this reason, temporal terms were only allowed to look backward, not forward. As a results, almost all false negatives in the final results were due to temporal expressions that occurred to the left of the target term in the text.

Table 5.7 shows a few of the results from the temporal extraction algorithm.

5.2.7 Onset

Onset answers the question: When did the symptom first start?

Recognizing onset expressions in the text relied on both regular expression items and the temporal expression algorithm. The regular expression items could catch arbitrary expressions occurring immediately after the trigger terms. For example, the regular expression `\bprior to .+?\b` could recognize expressions such as *prior to admission*. Likewise, `\b(after|s/p)\b.*(surgery|procedure(s)?)\b` could catch expressions such as *patient has had abdominal pain* ***s/p his procedures***.

In the second case, words in the specialized temporal lexicon were labeled as marking the beginning of a temporal phrase that indicated the onset of the pain. Words in this set included: *onset, began, started, s/p, status post, beginning, until*. During the temporal extraction process, if the extracted temporal expression begins with one of these words, it is assigned to the semantic category of *onset*. In this case, an onset item is created, and that modifier is added to the underlying context graph

Table 5.7: Temporal Expressions

Temporal Terms	**Text**
True Positives	
last 24 hours	no pain now and no pain within the last 24 hours
at this time	patient states he/she does have pain at this time
last night	no c/o abd pain since d/c last night
hrs	7) duration (how long have you had the pain?)(hrs,day,wks,mos,yrs): hrs
today	pain free today
12/16/2010 14:02	pain score:2 (12/16/2010 14:02)
during shift	patient reports pain during shift: no
overnight	no abdominal pain overnight
on arrival	no c/o pain on arrival.
this am	temp this am 100.2- pt denies pain
9 pm	medicated x1 for pain about 9pm.
False Positives	
currently 7/10	right sided abdominal pain, currently 7/10 (*7/10 is a pain severity score not a date*)
98966	hc pro phone call 5-10 min (98966). (*attached 98966 to min as an anatomical term*)
2/10	pain now at 2/10 -pt states he is ok for now- (*2/10 is a pain severity score*)
yesterday	patient with h/o of abdominal discomfort, ? appendicitis based on ct scan done yesterday (*yesterday should attach to ct scan*)
False Negatives	
this am	this am, reported pain (*this am appears before pain*)
current	denies any current abdominal pain (*current appears before pain*)

where it attaches to a pain target. In nursing flow sheets, onset can also appear as a target term, such as, *Onset: feb 21,2011@08:00*. The temporal expression would

then attach to the onset target term instead.

Table 5.8 shows a few of the results from the onset extraction algorithm.

Table 5.8: Onset Expressions

Onset Expressions	**Text**
True Positives	
feb 21,2011@08:00	onset: feb 21,2011@08:00
beginning abruptly at 630a	constant abdominal pain beginning abruptly at 630a
2 days ago	6) onset (when did the pain start?): 2 days ago
after surgery	6) onset (when did the pain start?): after surgery
prior to admission	denies exertion cp or sob prior to admission
Incomplete	
s/p	soreness in his rt side,s/p i/d (*should include i/d as the name of a procedure*)
False Positives	
prior to admission	location of patient prior to admission: home
started on	no c/o pain.started on clears. (*started on applies to clears not pain*)
False Negatives	
until recently	pt denies any abd pain until recently (*until recently appears on the next line*)
for years	for years he has had a pain in certain positions (*for years appears before pain*)

5.2.8 Duration

Duration answers the question: When the symptom is present, how long does it last?

The extraction of duration is very similar to that of onset. Regular expres-

sions that were added to the ConText lexical item database included terms that were mined both from the corpus and from SNOMED-CT. These included: *during exertion*, *during sleep*, *during [...] exercise* and *during [...] motion* where ... may include adjectives such as *moderate*.

Words added to the specialized temporal lexicon the indicate the beginning of a duration expression include: *for, ongoing, chronic, acute* and *which*. However, there was disagreement between the annotators whether chronic/acute were terms that indicated **duration** or the **type** of pain.

Once duration modifiers were recognized in the text, they were added to the graph and attached to the nearest target term. Like onset, duration could also appear as a target term in the context of a nursing flow sheet, as in, *Duration:*.

Table 5.9 shows a few of the results from the duration extraction algorithm.

5.2.9 Pain Quality

Pain quality answers the question: What does it feel like? An extensive search was made through terminologies, clinical text, web sites, and other resources to find as many of these expressions as possible. The quality of the pain can vary greatly, and even then there is a long tail. For example, in the VA corpus, a patient describes his pain as, "beltlike burning sensations around his waist".

Recognition of pain quality expressions was handled entirely by regular expression patterns. Items were added to the modifier database for: *aching, burning, colicky, deep, cramping, crushing, cutting, dull, electric, gnawing, gripping, itch-*

Table 5.9: Duration Expressions

Duration Expressions	**Text**
True Positives	
hrs	7) duration (how long have you had the pain?)(hrs,day,wks,mos,yrs): hrs
acute	8) type of pain:surgical, acute (<3 mos)
during active motion	s: pt reports pain in his shoulders and hands during active motion.
for weeks	complains of hand pain, reports for weeks.
chronic	additionally, chronic pain in b/l hands,
for several weeks	7/10 b shoulders and hands for "several weeks now".
ongoing	his ongoing b/l hand and shoulder pain
chronic	//hand/shoulder pain b/l - chronic.
1 weeks	duration of complaint: 1 weeks
Incomplete	
which generally last a couple of hrs	episodes of rlq abd pain which generally last a couple of hrs to 6hrs (*missing whitespace in 6 hrs*)
False Positives	
chronic	gi: no abdominal pain, + chronic constipation (*chronic should attach to constipation*)
4	states pain med effective, but doesn't last. duration now decreased to q 4 (*duration applies to the length between doses*)
1 week	+ skin lump or papule, worsening, duration longer than 1 week (*algorithm is initially fooled that 1 is a severity score, and attaches duration to it*)
18 minutes 11 seconds	encounter duration: 18 minutes 11 seconds nurse name: (*same as above*)
False Negatives	
5 m	progressively increasing over the last 5 m (*doesn't recognize M as a month*)

ing, killing, prickling, pulling, pulsating, [non]-radiating, sharp, shocking, smarting, splitting, sore, tenderness, throbbing, tightening along with their morphologies.

However, even though these terms are considered modifiers, some of them are used as synonyms for pain, and thus act like targets in the text. Therefore, the following terms were also added to the target database of expressions: *discomfort, cramp, hurt, knot, pang, rack, spasm, tingle.* In nursing flow sheets, the following may also appear as target terms: *pain character, quality, quality of pain.*

Table 5.10 shows a few of the results from the quality extraction algorithm.

Table 5.10: Pain Quality Expressions

Pain Quality Expressions	**Text**
True Positives	
dull, ache	5) quality of pain (what does it feel like?) dull/ache
throbbing	quality:throbbing
aching, cramping	quality:aching, cramping
Incomplete	
False Positives	
tender	abd soft non-tender peg tube noted (*non-tender*)
False Negatives	
pressure	quality:pressure,
burning	a burning sensation in his pelvis (*a burning sensation is really a pain quality*)
tingling	he denies numbness and tingling (*tingling is used as target*)

5.2.10 Pain Type

Pain type answers the question: What is the pain due to? A search through terminologies, clinical text, web sites, and other resources returned relatively few terms. These included: *cancer-related, malignancy, neoplastic disease, metastases, breakthrough, gas, heartburn, inflammatory, mechanical, obstetric, labor pains, rebound tenderness, referred, superficial, post-operative* and any term ending in *-omy* as in *episiotomy.*

These terms responded well to regular expressions patterns. As with the other attributes, pain type could appear as a target item in nursing flow sheets as *Type of pain:* or simply as *Type:*.

However, it should be noted that while the annotators categorized *chronic* and *acute* as durations, in the nursing flow sheets they appear as a pain type.

Table 5.11 shows a few of the results from the quality extraction algorithm.

Table 5.11: Pain Type Expressions

Pain Type Expressions	**Text**
True Positives	
surgical	8) type of pain:surgical, acute (¡3 mos)
gas	abdominal discomfort from "gas"
cancer-related	type:cancer-related
False Positives	
surgical	abd discomfort @ old surgical site (*surgical is part of location*)
False Negatives	
None	

5.2.11 Variability

Variability answers the question: Is the pain constant, intermittent, getting better or worse?

This term was coined by the annotators themselves. In the SemEval 2015 challenge, this attribute was called Course but was more restrictive and was used only in the context of how the disease was progressing.

As with the other attributes, terms were scraped from terminologies and other resources. These terms reponded well as regular expression items. Terms included: *continue, rare, occasional, constant, increased, constant* and *intermittent*.

Target forms in the nursing flow sheets appear as: *If intermittent, how long does the pain last:* and *Is the pain constant:*.

Table 5.12 shows a few of the results from the quality extraction algorithm.

5.2.12 Time of Occurrence

Extracting the time of the occurrence was challenging. Temporal phrases were extracted as described above. However, determining when these expressions should attach to the targets was difficult. If a phrase was not identified as being an onset or a duration, it was treated as a time attribute. However, because of the large number of false positives, it was restricted to only looking backwards.

The results are the same as those listed in Section 5.2.6.

Table 5.12: Variability Expressions

True Positives	
intermittent	intermittent right flank pain.
continues	continues to complain of abdominal pain in the rlq
increased	now with increased abdominal pain surrounding drain
increasing	post perinephric drain placement with increasing abdominal pain
yes	is the pain intermittent:yes
yes	is the pain constant:yes
continue	his shoulders and hands continue to feel "sore."
False Positives	
continue	continue to trend creatinine/hct pain score: (*continue is attaching to pain score: on the next line*)
rare	pt without pain, + passing gas, rare bm (*should attach to bm instead*)
False Negatives	
variable	if intermittent, how long does the pain last(seconds, minutes, hrs, daysvariable (*missing white space and punctuation*)

5.2.13 Modified ConText Graph

Figure 5.2.13 shows in textual form the underlying ConText graph after processing one line of text. There are two targets: *discomfort* and chest pain *c/p*. The *discomfort* target is modified a two locations and an onset. The pain type attribute is incorrect. The *c/p* target is negated (*denies*) and has a location generated the by compound pain term lexicon.

```
Line: 12      abd discomfort @ old surgical site, onset this am, denies sob, denies c/p, pt.   thought process
 TARGET:  cid000001 -- PHRASE: 'discomfort' -- pain -- '' -- Span:(5, 15) Scope:[0, 97]
----MODIFIED BY:  cid000012 -- PHRASE: 'abd' -- fma_word -- forward -- Span:(1, 4) Scope:[4, 5]
----MODIFIED BY:  cid000013 -- PHRASE: 'surgical site' -- fma_word -- backward -- Span:(22, 35) Scope:[5, 22]
----MODIFIED BY:  cid000008 -- PHRASE: 'surgical' -- pain_type -- backward -- Span:(22, 30) Scope:[5, 22]
----MODIFIED BY:  cid000010 -- PHRASE: 'onset this am' -- onset -- backward -- Span:(37, 50) Scope:[5, 37]
TARGET:  cid000002 -- PHRASE: 'c/p' -- pain --  -- Span:(71, 74) Scope:[0, 97]
----MODIFIED BY:  cid000007 -- PHRASE: 'denies' -- definite_negated_existence -- forward -- Span:(64, 70) Scope:[70, 78]
--------MODIFIED BY:  cid000004 -- PHRASE: '.   ' -- conjunction -- terminate -- Span:(78, 82) Scope:[0, 97]
----MODIFIED BY:  cid000003 -- PHRASE: 'chest' -- fma_word -- backward -- Span:(71, 74) Scope:[35, 71]
```

Figure 5.4: The results of ConText algorithms to identify pain targets *discomfort* and *c/p* with their related attributes.

5.2.14 Limitations

While the algorithm was successful in extracting temporal terms, it did not attempt to convert those terms into an actual date/time. Given that these expressions are generally given relative to the date/time of the note in which they are recorded, this should, in fact, be possible. However, the terms can be non-exact, as in *days*. A theory for how to handle this would need to be decided.

5.2.15 Discussion

The ConText algorithm was extended to include the following modifying attributes: severity, location, onset, duration, pain quality, pain type, variability, and event time. While some of these are specific to pain, such as quality and type, the others are more general. Onset, curation, variability, and event time are applicable to medical concepts that are diseases/disorders and to symptoms/findings. Therefore, when coupled with other concept recognition algorithms, they may be useful

in a much larger context.

Chapter 6: Information Retrieval

For use in clinical decision support, the extraction algorithm must be not only accurate but fast, with near real-time performance. For this reason, information retrieval technologies [54] and other efficiencies were used to speed up basic search 10-fold.

Keyword search, pattern matching rules, and machine learning classifiers are computationally efficient and are effective at identifying concepts. Ford, *et al.* [55] noted that there was no clear difference in case-detection algorithm accuracy between rule-based and machine learning methods of extraction. One important attribute of this approach is that keyword search and pattern matching rules also work well with grammatically unsound text as seen in Chapter 5.

6.1 Increasing Algorithmic Efficiency

The original approach used in [56] applied the ConText algorithm by reading each line of text in the note, finding every target pattern in the line, finding every modifier pattern in the line, and then determining whether that <`target, modifier`> combination belonged together following the rules of proximity, scope, direction, and semantic relatedness. The algorithmic complexity is $O(l*t*m)$ where

l is the number of lines of text, t is the total number of target patterns, and m is the total number of modifier patterns.

There were several computational efficiencies that could be pursued at this point. First, each target and modifier belongs to one semantic category such as *pain severity*, *negated existence*, or *probable existence*. Because there are hundreds of modifiers, and some of the target/modifier category combinations are not compatible, the algorithm was changed. Initially only target patterns for the medical concept of interest are considered. Then only those modifiers that are appropriate for the targets that have been found are loaded and checked.

6.1.1 Clustering

More importantly, large efficiencies can be gained by only processing those lines of text that contain the concept that you are interested in. In the case of pain, only 1% of the total lines of text in the corpus refer to this concept. However, the question remained as to how to determine which lines those were.

Initial investigations experimented with supervised machine learning algorithms. Each line in the text that contained a pain target or modifier was labeled as positive. An extensive grid search was performed using various combinations of 19 features with the following classifiers: SVM [57] with linear, polynomial, and RBF kernels, Naive Bayes [58], Logistic Regression [59], and K-Nearest Neighbor [60]. Naive Bayes returned the highest recall at 88% with a precision of 15%. Logistic regression returned the highest precision at 87% and a recall of 55%. None of the

Table 6.1: Comparison Of Classifiers For Identifying Lines Of Pain

Classifier	**Positive Precision**	**Positive Recall**	**F2**
Multinomial Nave Bayes	17%	88%	29%
Linear SVM	80%	64%	70%
SVC-RBF	58%	77%	66%
SVC-POLY	36%	80%	49%
LogReg	76%	62%	68%
K-NN	76%	38%	49%

classifiers returned an F-score above 70%. Table 6.1 shows the classifier results.

Clustering experiments carried out on the corpus revealed that all the desired target/modifier/values for pain occurred within a very small window around one or more anchor terms, even when the pain event was spread out over several lines of text. Experiments were carried out to compare the effect of clustering by lines, by terms, the optimal window size, and the optimal anchor terms. Priority was given to finding the clustering approach that would maximize recall and minimize computation complexity.

The best performing clustering algorithm for both recall and complexity was based on defining one or more anchor terms and returning a window of lines around it. Extraction was then carried out only on those lines. For example, if all the lines in the text are numbered sequentially, and line number x contained the word *pain*, a window of $x - N$ lines above and $x + N$ lines below would be included in the result set. Tables 6.2 and 6.3 show the results of varying the window size around the anchor term(s). Precision and recall in this case are based on the number of lines in the corpus that actually contain data concerning pain mentions that should be processed during the extraction phrase.

Table 6.2: SINGLE ANCHOR TERM OF PAIN

Window (# of lines)	Positive Precision	Positive Recall
1	31%	91%
2	30%	95%
3	30%	96%
5	29%	97%
10	27%	97%

Table 6.3: ANCHOR TERMS OF *PAIN, LEVEL, QUALITY, SEVERITY, LOCATION, SCALE, ONSET*

Window (# of lines)	Positive Precision	Positive Recall
1	25%	98%
2	24%	98%
3	24%	98%
5	23%	98%
10	21%	98%

Table 6.2 uses only a single anchor term of *pain*. Table 6.3 uses the term *pain* as well as the names of the attribute modifiers that will be extracted. The final set of anchor terms was *pain, level, quality, severity, location, scale, onset*. These results indicate that, using these terms, it is only necessary to bring back one line above and one line below any line that contains at least one of those anchor terms to get 98% recall in the result set.

6.2 Implementation

To implement this clustering approach, information retrieval techniques were pursued. A survey of several information retrieval systems were evaluated for performance and flexibility. The system with the widest variety of search parameters was chosen. In addition, several additional filters were evaluated.

The description of this implementation begins with a discussion of the Elasticsearch structured information retrieval system that was chosen for this task, its unique characteristics, and how the clinical documents were ingested into the system. Section 6.3 describes how the experiments were set up to determine the recall and runtime performance characteristics of this approach with the results presented in Section 6.4.

6.2.1 Elasticsearch

Elasticsearch is an open-source search engine built on top of Apache Lucene [61], a full-text search-engine library. It is a distributed real-time structured document store, capable of generating real-time analytics over very large data sets to accommodate petabytes of structured and unstructured data. Every individual field in a document is separately indexed and searchable. A single field may have multiple indexes by changing tokenizing, stemming, and other text analyzer options. Multiple fields can be combined into a single index. Each field index generates its own set of tf-idf [62] statistics and can customize its own relevance scoring algorithm based on them.

Elasticsearch exposes its API through RESTFul web services using a JSON-encoded request body. All responses likewise are returned as serialized JSON objects.

In Elasticsearch, terminology may be confusing. For instance, the word *index* takes on different meanings depending on whether the term is being used in the

Table 6.4: ELASTICSEARCH TERMINOLOGY

Elasticsearch	**Relational Database**
Index	Database
Document type	Table
Document mapping	Table definition
Field	Column
Document	Row
Index a document	Insert a row
Re-index a document	Delete a row and re-insert it

context of Lucene, Elasticsearch, or a relational database. For the sake of clarity, Table 6.4 cross-references a few terms.

6.2.1.1 Indexing the Corpus

The line-oriented nature of this corpus is greatly amenable to leveraging Elasticsearch to filter out unneeded lines of text from the extraction process. Even though the complexity of the extraction algorithm only differs by a constant, there is still a significant reduction in the number of lines that need to be processed.

Three indexes were built in Elasticsearch. The *emr_patient* index contains patient demographic information. In this case the information was generated synthetically since the records were de-identified. It has a single document type: *patient*. The *emr_headers* index contains three document types, one for each basic note type in CPRS: *clinical_note, radiology_report, lab_report*. These documents types contain the meta-information about the notes, such as the date, time, title, and signers on the note. Finally, the *emr_note_lines* index treats every line of text as a separate document. It contains a single document type, *note_line*, which contains references back to the patient, the note, the line number within the note, as well as the text

of that line.

The entire corpus was indexed into the Elasticsearch cluster. The documents were ingested at a rate of 10,000 lines of text per 1.06 seconds (95% C: ±0.093) or 3.48MB of text in less than 11 seconds. For reference, this data was the equivalent of 2,343 pages of text, single-spaced.

6.3 Experimental Methodology

Experiments were performed with Elasticsearch that were similar to the clustering experiments above. An initial request was made to Elasticsearch with no filtering which simply returned all the lines in the set and then the extraction algorithm was performed. Next, the extraction was run using different filters detailed below. The results were evaluated by wall-clock time and by accuracy of the extraction algorithm after filtering. An initial baseline experiment was also run to determine the performance characteristics when reading in the text directly from local disk.

The following filters were compared:

- Filter 0: no filter, all lines used
- Filter 1: *pain*
- Filter 2: *pain level discomfort quality severity location scale onset*
- Filter 3: *pain cp c/p discomfort painful painfree ache cramp hurt pang knot pressure spasm sore soreness tingle severity scale*

Table 6.5: FILTERS WITH NO STEMMING

Filter	Mean (sec)	Precision	Recall	F2
0	80.79 (±1.8)	93%	81%	87%
1	8.17 (±0.31)	97%	72%	82%
2	10.18 (±0.63)	97%	78%	86%
3	12.30 (±0.06)	96%	79%	87%
4	12.47 (±0.16)	96%	80%	87%

Table 6.6: FILTERS USED WITH STEMMING

Filter	Mean (sec)	Precision	Recall	F2
0	84.85 (±3.3)	93%	81%	87%
1	8.02 (±0.21)	97%	72%	83%
2	10.44 (±0.55)	97%	78%	86%
3	12.47 (±0.14)	96%	80%	87%
4	12.58 (±0.35)	96%	80%	87%

- Filter 4: *pain cp c/p discomfort painful painfree ache aches cramps cramps cramping hurt hurts hurting pang pangs knot knots knotting pressure*

6.4 Results

These experiments were run on a commodity Intel i7 1.8 GHz processor. The Elasticsearch cluster was made up of a single node running locally.

Remarkably, the mean wall-time difference between retrieving all of the text directly from a disk file versus retrieving it from Elasticsearch over HTTP on localhost was insignificant at 0.945 seconds (95% C: ±1.26).

The results in Table 6.5 show the trade-off between speed and accuracy of the extraction results between the different filters. Table 6.6 shows results on the same set of filters, but with stemming turned on in Elasticsearch using the Porter stemmer.

According to standard definitions, precision, recall and F2 are defined as:

$$precision = TP/(TP + FP) \tag{6.1}$$

$$recall = TP/(TP + FN) \tag{6.2}$$

$$F2 = 2 * precision * recall/(precision + recall) \tag{6.3}$$

where TP is true positives, FP is false positives, and FN is false negatives.

These results show that an approximate 10x speedup between Filter 0 (no filter) and Filter 1 (*pain*) may be achieved if the application is able to suffer a 9% reduction in recall. Given an average note length of 198 lines, this algorithm is able to process notes at a rate of 24.7 notes/second (0.04 seconds per note) in this configuration. By way of comparison, a subset of notes, consisting of 7670 lines of text and 55 notes, was analyzed by cTakes. It took 9 mins 13 seconds (using the same hardware configuration) for a rate of 0.1 notes/second (10 seconds/note). By extrapolation, it would take over an hour and 15 minutes to process the same corpus used in the experiments presented here.

There was no significant difference in results for time or accuracy between stemming and no stemming.

The lower precision of the no filter (93%) configuration compared to the others is due largely to physical therapy notes. Physical therapists deal largely with three things: range of motion, strength, and pain. In these notes, pain is implied in the following text: *7/10 hands, shoulders* but not stated directly. Therefore, it is

bypassed by all the filters. Because this text is also the most difficult to extract from, the precision is higher in the filtered configurations than in the no filter configuration.

6.5 Summary

In the clinical NLP challenges, the focus is on the accuracy of the identification and extraction tasks. Teams do not report on runtime performance. However, given that many use the same NLP pipeline that is used in cTAKES, performance should be similar to that. The approach shows that for extracting a small set of symptoms, using information retrieval techniques to filter out large portions of text provides a significant performance advantage with relatively little loss in recall.

Chapter 7: Machine Learning

7.1 Overview

While machine learning algorithms were not effective in recognizing medical concepts in the text, there were important tasks for which they were effective. Figures 7.1 and 7.2 show portions of lab results and medication lists that appear in the medical record. While lab results are also stored in a structured format in the backend database, they are frequently copied and pasted into the note. In the lab results, there are many terms that appear in the anatomical lexicon. The numeric entries in the report, therefore, are seen as severity scores by the algorithm. Likewise, in the medication list, pain medications contain the word *pain* along with a number, such as *PO Q4H PRN WHEN NEEDED pain > 4*. This combination results in a large number of false positives for pain. It was not possible to filter these out using rules and/or regular expressions.

However, the text has marked distributional semantics as can be seen by a casual visual examination of the text. For this reason, statistical machine learning algorithms were incorporated to identify and filter out these types of reports. Figure 7.1 and 7.2 show examples of lab results and a medication list.

```
Lab results:
CBC: MODIFIED CBC (OUTPUT); BLOOD
Coll.  Date:  09/19/08 05:30 09/18/08 10:09
Test Name Result Result Units Range
WBC 6.1 5.0 K/MM3 4.8 - 10.8
RBC 5.10 4.79 M/cmm 4.7 - 6.1
HGB 14.0 13.3 L G/DL 14 - 18
HCT 43.1 40.4 L % 42 - 52
MCV 84.5 84.4 fL 80 - 100
MCH 27.4 27.7 uug 27 - 31
MCHC 32.4 32.8 gm/dL 32 - 37.5
RDW 14.6 H 14.6 H % 11.5 - 14.5
PLT 234 236 K/cmm 140 - 440
MPV 9.1 9.0 fL 7.4 - 10.4
```

Figure 7.1: Example of a lab results that have been copied and pasted into the clinical note.

7.2 Training and Test Set Development

A special subset of data taken from the training corpus was created from which to build and test the machine learning models. First, because the amount of text with lab results was small in comparison with the rest of the text, all of the lab results were copied into the subset. Next, the same amount of text from a variety of notes containing other clinical text and medication lists was also copied in to form a balanced set of data.

```
Active Inpatient Medications (including Supplies):

Active Inpatient Medications Status
================================================
1) ACETAMINOPHEN ORAL (160mg/5ml) LIQUID 650MG (20ML)
PO Q4H PRN WHEN NEEDED pain > 4
2) CALCITRIOL ORAL (1mcg/ml) SOLN,ORAL 0.25 MCG
(0.25ml) JT DAILY
3) CALCIUM CARBONATE ORAL (1250mg/5ml) 1250MG/5ML PO
TID
4) DOCUSATE SODIUM ORAL (50mg/5ml) 100MG/10ML PO BID
5) HEPARIN INJECTION INJ,SOLN 5000 UNITS/1ML SC Q8H
6) INSULIN REGULAR (HUMAN) INJ SLIDING SCALE SC Q4H FS
q4h.  0-69:  call HO; 70-200=0 units; 201-250=2
units; 251-300=4 units; 301-350=6 units; 351-400=8
units,; >400=10 units and call HO
```

Figure 7.2: Example of a medication list.

7.3 Processing Pipeline

A processing pipeline was set up to preprocess the text and then run a grid search over a selection of machine learning classifiers using a variety of possible parameters on each.

The Scikit-Learn [63] Python libraries were used. After the text was converted to lowercase, the `CountVectorizer` [64] was used to convert each line of text to a matrix of token counts. Next, `TfidfTransformer` [65] to transform the counts to a normalized term frequency representation [66].

Three-fold cross validation was used for training and testing. The model for the classifier with the best accuracy was saved.

7.4 Classifiers

7.4.1 Lab Results Classifier

The following classifiers were examined for the lab results: Naive Bayes [58], Decision Tree [67], Logistic Regression [59], and Support Vector Classifiers (SVC) [68] with polynomial and radial basis function (RBF) kernels [69]. A grid search was performed on each classifier, using whichever parameters it offered. Table 7.1 show the training parameters and results of four of these classifiers. Of all classifiers, the Multinomial Naive Bayes performed the best with an accuracy of 99.09%.

7.4.2 Medication List Classifier

The same methodology was used with the medication lists as was used with lab results. However, as can be seen in Figure 7.2, the strongest signal that the line containing $pain > 4$ is a medication line comes from the line above it. Therefore, this classifier depends on having two lines of text rather that one; samples sent in to the classifier are two lines concatenated together. For medication lists, Multinomial Naive Bayes provided the best results at 98.15%. Table 7.2 shows the training parameters and results of four of these classifiers.

7.5 Implementation

The pain extraction algorithm reads in the classifier model during initialization. As the algorithm processes one line at a time, each line is classified in each

of the classifiers. If either one of them returns a positive classification, the line is skipped.

7.6 Discussion

The results on the train/test set of data seem to be better than on the actual data, and tends to err with false positives. This had some effect on the final scores of the extraction results.

Table 7.1: Comparison Of Classifiers For Identifying Lab Reports

Multinomial Naive Bayes	**DecisionTree**
Best score: 0.9909 Best parameters set: clf__alpha: 0.1 clf__class_prior: None clf__fit_prior: True tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'	**Best score: 0.9853** Best parameters set: clf__criterion: 'gini' clf__max_depth: None clf__max_features: None clf__min_samples_leaf: 1 clf__min_samples_split: 2 clf__random_state: None tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'
LogisticRegression **Best score: 0.9894** Best parameters set: clf__C: 10.0 clf__class_weight: None clf__dual: True clf__fit_intercept: True clf__intercept_scaling: 0.5 clf__penalty: 'l2' clf__tol: 1e-06 tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'	**SVC-POLY** **Best score: 0.9498** Best parameters set: clf__C: 1.0 clf__cache_size: 1000 clf__coef0: 0.0 clf__degree: 2 clf__kernel: 'poly' clf__tol: 0.001 tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'

Table 7.2: Comparison Of Classifiers For Identifying Medication Lines

Multinomial Naive Bayes	**DecisionTree**
Best score: 0.9815 Best parameters set: clf__alpha: 0.01 clf__class_prior: None clf__fit_prior: True tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'	**Best score: 0.9702** Best parameters set: clf__criterion: 'gini' clf__max_depth: None clf__max_features: None clf__min_samples_leaf: 1 clf__min_samples_split: 2 clf__random_state: None tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'
LogisticRegression **Best score: 0.9800** Best parameters set: clf__C: 10.0 clf__class_weight: balanced clf__dual: True clf__fit_intercept: True clf__intercept_scaling: 10.0 clf__penalty: 'l2' clf__tol: 1e-06 tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'	**SVC-POLY** **Best score: 0.8899** Best parameters set: clf__C: 1.0 clf__cache_size: 1000 clf__coef0: 0.0 clf__degree: 2 clf__kernel: 'poly' clf__tol: 0.001 tfidf__norm: 'l2' tfidf__smooth_idf: False tfidf__sublinear_tf: False tfidf__use_idf: False vect__binary: False vect__max_df: 1.0 vect__min_df: 0.0 vect__ngram_range: (1, 2) vect__stop_words: None vect__token_pattern: '\b\w+\b'

Chapter 8: Clinical Decision Support

8.1 Overview

Once the information has been extracted from the clinical record, and converted into a structured semantic representation, it can be used in clinical decision support (CDS) applications. This chapter begins by describing how the overall architecture of how this semantically rich cognitive search assistant could be used with a CDS system. The chapter ends with an actual application that was requested by and created for physicians at the VA to monitor post-surgical pain.

8.1.1 Extraction

The CDS application sends a request to the extraction engine which specifies which targets and attributes to extract as well as which data source to extract from. The CDS also specifies what kind of representation the results should be returned as. Results may be returned as JSON (JavaScript Object Notation) [70] objects, or as an OWL/XML serialization.

The data is read in, and the required extraction components are activated to find those occurrences of the targets and attributes that appear in the text. As each

occurrence is found, it is saved in a data structure in memory.

Once the extraction is complete, results are packaged as specified in the initial request. The in-memory dictionary data structure that contains the results serialize easily into a character-based JSON representation. If a semantic representation is requested, the required ontologies are read in, and the extracted values are mapped into their semantic representation and serialized as OWL/XML. Results are sent back to the CDS application as JSON and/or OWL/XML according to the initial request.

8.1.2 Querying and Inference

If inference is desired, the CDS application sends the OWL/XML data to the inference system. The inference runs as a separate process that is exposed through a RESTFul API.

8.1.2.1 Clinference

Clinference is a Java application that takes requests to load ontology resources and to run queries on the data contained in them. Clinference uses the Jena Framework [47] to wrap the Pellet reasoner [46].

As stated in Section 4.3.2, Pellet was chosen as the best reasoning tool for this research. Pellet reads in the OWL/XML data which contains both the domain knowledge defined in the ontologies (T-Box) and the semantically represented result instances (A-Box, see Section 4.1.2). Pellet uses its inference engine to apply all

T-Box domain concepts, rules, and relationships, to the A-Box instances in order to generate any other inferred relations and knowledge. For example, inference is used to map the lexical and numerical severities to their normalized values; "minimal" would be mapped to *Mild* by inferring the **is-a** relationship on **SeverityNorm-Mild**, while numeric values of "6" would map to *Moderate* pain by using the SWRL rules (see Section 4.1.4) defined in the **PainSeverity** ontology (see Appendix-A) to infer an **is-a** relationship with **SeverityNormModerate**. If a pain event has a certainty that is negated ("no pain"), an inference is made that the pain severity is *None* because an equivalence between those **CertaintyDefiniteNegated** and **SeverityNormNone** was defined in the **Severity** ontology. Likewise, a query on a location of "abdomen" would return all result instances in which the text directly specified *abdomen* as well as instances in which any of its subparts such as *lower left quadrant of abdomen* were specified.

The results over the inferenced data are retrieved from Pellet via the SPARQL [71] query that was specified in the request. Results are returned as a set of tuples, similar to that which would be returned from a query from a relational database system, but serialized as JSON.

Once the results are received by the CDS application, they can be used as intended.

8.1.3 Runtime Efficiency

While the extraction algorithm was capable of handling several modifiers, this application only required pain, severity, and location. This allows extra efficiencies for runtime performance. Because the ontologies are designed as components, only those ontologies needed to represent clinical events, symptoms, pain symptoms, pain severity, and location are needed. Likewise, the information extraction engine will bypass processing any attributes that are not requested. In having less ontologic axioms and less data to reason over, the reasoning performance is also enhanced.

8.2 Visualization of Pain Severity Events in Clinical Records

8.2.1 Background

Physicians at the VA hospital in Baltimore wanted to quickly assess and monitor a patient's post-surgical pain using a visualization that contained as much pertinent information from the unstructured text in the patient chart as possible. A proof-of-concept software system was built that uses ontology-based semantic search to extract pain event mentions from a patient chart, store them in a structured semantic representation in a knowledge-base, run and inference engine over them to infer normalized values from expressions of severity in the text, and then query the knowledge-base to retrieve data necessary to visualize patient pain severity information [72].

8.2.2 Visualization

The application begins by defining the parameters of the request which include the patient id, the targets to extract, and any modifiers. In this case, the patient has had surgery for appendicitis, therefore the location would be set to *abdomen*. The purpose for which the system is built further defines the type of notes to filter in/out (*nursing*), which symptom to target (*pain*, which modifiers to include *pain severity*, and whether to send the results of the extraction to the reasoning system in OWL/XML format. The JSON serialized request is sent to the extraction system via the RESTFul API. Figure 8.1 shows an example of a request.

```
request = {"targets": {"pain": ["negation", "severity", "location"] },
"patientId" : "1",
"filtered" : "yes",
"inferredData": "yes",
"rolledUpData": "yes",
"rawData": "yes",
"patientData": "yes",
"headerData": "yes",
}
```

Figure 8.1: Example of request sent by CDS system to the extraction engine.

The extraction system reads the parameters, and performs the extractions. Once the process is complete, the results are encoded into a semantic representation combined with any ontologic components, and sent to Clinference in an OWL/XML serialization which the reasoning system is able to consume.

The application submits a SPARQL query to Clinference which forwards the request to Pellet. After the query completes, Clinference returns the location, severity, note type, note time, and both the extracted and normalized value of each pain severity event ordered by date and time as a JSON structured serialization. The JSON representation is the consumed by the CDS application which packages it for the graphics library in order to create the visual representation. Figure 8.2.2 shows an example of a SPARQL query made by the pain visualization application to the Clinference inferencing system.

```
SELECT  ?localTitle ?standardTitle ?noteDateTime ?locationExpr ?severityNorm
WHERE {
          ?patient            Patient:hasPatientId          ?patientid .
          ?patient            Note:hasNote                  ?note .
          ?note               Note:hasNoteDateTime          ?noteDateTime .
          ?note               Note:hasLocalNoteType         ?localTitle.
          ?note               Note:hasStandardNoteType      ?standardTitle.
          ?note               Event:hasEvent                ?symptomEvent.
          ?symptomEvent       Provenance:hasProvenance      ?provenance.
          ?provenance         Provenance:hasGlobalId        ?globalid.
          ?symptomEvent       Severity:hasSeverity          ?severity.
          ?severity           Severity:hasSeverityNormValue ?severityNorm.
          ?symptomEvent       Location:hasLocation          ?location.
          ?location           Location:hasLocationExpression ?locationExpr
```

Result | Messages

localTitle	standardTitle	noteDateTime	locationExpr	severityNorm
Ecs Nursing Triage Note	Emergency Dept Note	2011-02-21T17:10:00	abd	Unknown
Ecs Nursing Triage Note	Emergency Dept Note	2011-02-21T17:10:00	surgical site	Unknown
Ecs Nursing Triage Note	Emergency Dept Note	2011-02-21T17:10:00	chest	None
Ecs Nursing Triage Note	Emergency Dept Note	2011-02-21T17:10:00	epigastric	Moderate
Ecs-Boilerplate	Emergency Dept Note	2011-02-21T17:30:00	abdomen	Unknown
Ecs-Boilerplate	Emergency Dept Note	2011-02-21T17:30:00	abdominal	Mild
Ecs-Boilerplate	Emergency Dept Note	2011-02-21T17:30:00	?	None
Ecs-Boilerplate	Emergency Dept Note	2011-02-21T17:30:00	abdominal	Unknown
Ecs Nursing Flow Record	Emergency Dept Note	2011-02-21T18:34:00	chest	Moderate

Figure 8.2: The CDS application queries the Clinference system for both the raw extracted data and the inferred data.

The visualization shows pain severity over time. The horizontal dimension shows time as the number of hours that have passed since the date and time associated with the first note. The vertical dimension is made up of four stacked layers.

The top layer shows the anatomical location of the pain. In the second layer, pain severity is represented as a red circle which is centered horizontally over the time of the event occurrence. In order for the clinicians to ascertain the progression of pain as quickly as possible, the pain severity is represented in two different visual dimensions: 1) the size of the circle; and 2) the color of the circle. A small green circle indicates that the note explicitly indicated the absence of pain. If the existence of the pain event was indicated as being *Definite*, but the severity was not stated ("Patient states he does have pain at this time"), a question mark is displayed instead of the circle. The third layer contains text which gives more specific information about the context. This includes the actual date and time of the event, and the type of note it occurred in. The bottom layer is the actual pain value extracted from the clinical text.

Figure 8.3: For each event, the visualization includes the note type, date/time, actual severity value, normalized severity value, and location.

8.2.3 Discussion

The results in Figure 8.2.2 show the final visualization of the progression of pain for the appendicitis patient. Initially, the pain resolves and the patient is discharged. However, the patient is called back after a positive diagnosis of appendicitis for which he receives surgery. The immediate post-surgical pain is severe, and oscillates between none and moderate due to treatment with pain medication.

The patient is finally discharged with mild pain.

The two biggest problems in the visualization were 1) the inability to stretch or scroll the graph and 2) to adjust the visualization to the density of the pain severity event data. In the future, the use of graphical gaming libraries might address this and allow the clinician to interact with the image in order to scroll, zoom, or stretch the image. It might also allow the clinician to hover over a data point in order to see the actual text with the extracted values highlighted. A more complete user interface for search would allow the clinician to refine the search results and to filter on aspects such as date, symptom type, or note type. If multiple symptoms are displayed, it may also be possible to layer them while allowing the clinician to selectively overlay them with each other. Other important related data that might be displayed are medication events that are used to treat the symptoms.

Chapter 9: Evaluation Method

9.1 Overview

In order to evaluate the effectiveness of the extraction algorithm, a comparison against human domain experts must be made. The human experts recognize and codify the expressions of pain in the text along with its related attributes. This provides the baseline against which the algorithm is measured. However, human judgements themselves are open to error or differences in opinion. Therefore, the annotated results must be measured against one another to compare accuracy and/or agreement. This chapter describes the process used to prepared the data for annotation, the guidelines that were given to the annotators, and the measured agreement between the annotators. The annotations provide the gold standard from which the results of the extraction algorithm are measured in the following chapter.

9.2 Corpus Description

The corpus consists of portions of ten deidentified electronic medical records from patients that were downloaded from the VA's VistA [73]) electronic health system (EHS). Each patient has significant health problems involving pain such as cancer, kidney disease, and appendicitis. There are over forty different note types contained in the charts including triage, emergency department, surgery, radiology,

Table 9.1: Corpus Statistics

Patient Chart	Number Lines	Number Tokens	Unique Tokens	Number Notes	Note Types
Patient - Appendicitis	7,671	30,591	3,046	55	27
Patient - Syncope	3,095	13,927	2,116	15	12
Patient - Perirenal Abscess	28,424	118,332	4,893	229	48
Patient - Hypercalcemia	24,595	103,890	4,486	191	40
Subtotal	63,785	266,740		490	
Patient - Colon cancer	7,504	34,624	3,019	45	24
Patient - Pain	8,500	31,579	3,383	82	34
Patient - Anemia	5,762	24,858	2,593	32	17
Patient - Lung Cancer	7,824	34,255	3,071	52	25
Subtotal	29,590	125,316		211	
Total	93,375	392,056		701	

laboratory, nursing assessments, and physical therapy. Each patient record spans from several days to several months and contains 15 to 200 individual clinical notes. Each line of text contains up to 100 characters, and each line is broken at word boundaries. If this corpus was printed out single-spaced in a 10-point font with one inch margins, it would be approximately 1,500 hundreds pages.

There are a total of 93,375 lines of which the first 63,785 lines are being used in development; and the remaining 29,590 as the test set. For this research, each line of text is assigned a globally unique line id. Figure 9.1 shows the breakdown of these patient records.

9.3 Annotating the Data

Two sets of annotations were performed. The first set focused only on pain severity and its translation into a normalized form *None, Mild, Moderate, Severe*. The second set annotated all attributes of pain including severity. The evaluation

for the extraction is based on the second set of annotations.

The annotation methodology was based on those of SemEval 2015 Challenge as specified in its annotations guidelines [1]. Minor modifications were required due to differences in the characteristics of the text, especially as regards the nursing flow sheets in the VA corpus. However, the most significant change was the elimination of the normalized CUI for the disorder mention. Its lack of inclusion is a product of the limited resources available for annotations. Even if terms were restricted to the use of SNOMED-CT instead of the entire UMLS, SNOMED-CT is large and difficult to navigate for the uninitiated. It was unreasonable to required the annotators in this research to assume this burden.

9.3.1 Annotator Qualifications

The domain experts consisted of three clinical professionals. The first was a physician with 30 years experience. The second was a registered nurse with 35 years experience holding a masters degree in nursing and was adjunct faculty at the University of Maryland School of Nursing. The third was a head surgical nurse with 30 years experience. None of the annotators received any fee. Each annotator took roughly two hours to perform the annotations.

9.3.2 Preparing the Test Data for Annotation

The test set of data was the equivalent of 650 pages of clinical notes. As resources for annotation were extremely limited, it was unreasonable to expect the annotators to read through the entire test corpus. Large sections of the corpus

[1] blulab.chpc.utah.edu/sites/default/files/ShARe_Guidelines_CLEF_2013.pdf

were made up of lab reports or medication lists on which no extraction would be performed. There were also large portions of text that were duplicates of other portions. This was caused by clinicians addending previously entered notes. When a note is addended, the text of the original note is automatically copied into the addenda. In one case, a note was addended five times - with each addenda containing the text from all the previous addended notes five levels deep.

After all the extraneous text was removed, there remained 65 pages of notes. Each line of text was labeled with its unique identifier at the beginning of each line. The test set of notes was then printed out on paper - one set for each annotator.

9.3.3 Annotator Training

The clinicians had ever done this task before, and the nature of the task was quite foreign to them. A set of guidelines was drawn up to instruct the annotators in their task. I read through the guidelines with the annotators, and they were allowed to ask questions for clarification. Once they felt they understood the task, I assisted them in annotating the first page and answered any additional questions. For the remainder of the time, they worked by themselves. The official guidelines are shown in Appendix B.

9.3.4 Annotation Coding

Each annotator was given a yellow highlighter and pencil. The annotators highlighted any span of text that indicated pain or one of its attributes. Using the pencil, they labeled the highlighted span with its attribute type: P (*pain*), N (*negated*), S (*severity*), L (*location*), Q (*quality*), T (*type*), O (*onset*), D (*duration*).

There was some confusion on how to highlight temporal terms relating to the time of the occurrence, so there are no annotations for event time.

9.3.5 Inter-Annotator Agreement

The annotations were evaluated using Cohen's κ coefficient [74] which measures the agreement between two annotators. Agreement was measured between two of the annotators with the third annotator providing the tie-breaking value.

When calculating κ, the observed agreement A_o is calculated. This is simply the percentage of items that the annotators agree on divided by the total number of items.

$$A_o = \frac{items_agreed_on}{total_items} \tag{9.1}$$

Next, the expected agreement, A_e, calculates how much the annotators would agree if they simply made random choices in each separate category. The chance of annotator a_1 agreeing with annotator a_2 on nominal value c in category C is $P(a_1|c) \cdot P(a_2|c)$. The chance, therefore, of the annotators agreeing on any c is:

$$A_e = \sum_{c \in C} P(a_1|c) \cdot P(a_2|c) \tag{9.2}$$

κ is the average of A_0 and A_e:

$$\kappa = \frac{A_0 - A_e}{1 - A_e} \tag{9.3}$$

There are several scales used to interpret κ. In this research, the Landis and Koch [75] interpretation was used; that is: $0.0-0.2$ (*slight*), $0.2-0.4$ (*fair*), $0.4-0.6$ (*moderate*), $0.6-0.8$ (*substantial*), and $0.8-1.0$ (*perfect*).

Results of inter-annotator agreement for each pain attribute are shown in Table 9.2.

Table 9.2: Inter-Annotator Agreement

Attribute	A_0	A_e	κ
Existence	0.685	0.360	0.509
Severity	0.800	0.088	0.781
Location	0.708	0.083	0.681
Onset	0.697	0.170	0.635
Duration	0.651	0.167	0.581
Quality	0.681	0.097	0.647
Type	0.911	0.761	0.628
Variability	0.704	0.169	0.644

9.3.6 Discussion of Annotations

When the annotators agreed on a medical concept, they agreed completely and marked the same spans and gave them the same label. The difficulty in this task came in reading through 65 pages of medical notes in one sitting. It was very easy to simply miss occurrences. This was made obvious in that unmarked occurrences were very similar to occurrences marked previously by the same annotator. For example, *denied pain* was marked in many other instances as an occurrences of pain negation, but in the following text it was missed by two of the three annotators: *assumed care of patient upon return to floor at 0930hrs. vss. pt denied pain.*

As the task went on, the annotators became more fatigued, and missing annotations became more pronounced. Near the end, one of the annotators missed roughly half the annotations that were found by the other two.

Another case of missed annotations was caused by the nursing flow sheets. In the prompt, *7) Duration (How long have you had the pain?)(hrs,day,wks,mos,yrs):,*

the answer to the prompt appears on the following line. That value for pain duration was not noticed by any of the annotators. Also in the flow sheets, some values are repeated more than once. One of the annotators highlighted only one value instead of highlighting and labeling each mention.

One other cause of missed annotations was the use of abbreviations. In particular, headache is abbreviated in the notes as *ha*. When embedded in a narrative note, such as *brain with contrast done given new ha which shows multiple enhancing lesions*, it was easily missed.

There was clinical disagreement over the word *tender*. It is used to describe the results of palpation on the abdomen during the physical exam. It appears in the notes as *abdomen: soft, non-tender, without organomegaly or masses.* The experts generally agreed that this did not constitute a negation of pain occurrence. Tender, however, may also be regarded as a pain occurrence as in *Abd distended tender to right side*. It was left up to the experts to determine when tender constituted a pain occurrence or not. Agreement on this was low.

Using the Landis and Koch scale, the agreement over all the attributes in this annotation task ranges from moderate to substantial. In research efforts where more annotation resources are available, an iterative approach is taken wherein the annotators are allowed to look back over their annotations and check whether an occurrence was not marked due to clinical judgement or was simply an oversight on their part. If it was due to oversight, the annotation is changed. This procedure results in higher inter-annotator agreement scores and would have been beneficial in this task.

It was not possible to compare the inter-annotator agreement with the state-of-the-art SemEval 2015 task as it was not presented in the paper describing the task [34].

9.4 Summary

Three clinical professionals, with a combined experience of 90 years, annotated the test dataset by marking the spans and labeling the value of pain events and their modifying attributes in the medical records of several patients. These annotations are the ground truth used to evaluate the accuracy of the information extraction algorithm based on the clinical judgements of these domain experts. The next chapter discusses how the annotations were codified into a machine-readable gold standard, and how the information extraction algorithm performed when measured against it.

Chapter 10: Results

10.1 Overview

The evaluation of the extraction algorithm was designed to be as similar to that used by the SemEval 2015 challenge as possible. However, there were necessarily some modifications based on the differing characteristics of the two corpora; most notably due to the presence nursing flow sheets and the line oriented nature of the VA corpus.

10.2 Gold Standard

10.2.1 Creating the Gold Standard from the Annotations

Once the annotators had completed their task, the creation of the gold standard commenced. The text spans against which the extraction algorithm would be scored were derived from the highlighted phrases in the text. If the highlighted phrase was the value of a pain attribute, the label penciled-in next to the highlight was used to determine which attribute that value belonged to, and the value of that attribute was coded into the gold standard.

The gold standard consists of the set of pain targets (see Chapter 5) and any of its related attributes (modifying values). Targets could be references to pain (*pain, painful*), or other attributes that appear as targets in nursing flow sheets (*severity,*

location, onset, duration, quality, type). Each line could also contain more than one target (*patient denies cp/ha/sob*). In this case, there is a gold standard instance created for each target that share the same line id but have different spans.

Each gold standard instance, then, is made up of the line id and span (beginning and ending character indexes) of the target term along with the values for any attribute/modifier for that target. The final gold standard consisted of 426 <line id,span> targets. Figure 10.2.1 shows an example of the first few lines of the gold standard (minus attributes that did not fit on the printed page). Table 10.1 shows the prevalence over the test corpus of non-default values for each of the slot attributes.

Line Id	Text	Span		Negation	Severity	Location	Onset	Duration
63802	Duration of Complaint: 1 Weeks	0	8					1 weeks
63826	Pain level: 6	0	4		6			
63872	mild stomach pain x1 month and didn,t eat x2 days .	13	17		mild	stomach		
63932	Headache (ICD-9-CM 784.0)	0	8			head		
64243	15:15 assumed care of pt. pt alert x 4 denies any pain,	69	73	Negated	0			
63963	CURRENT PAIN LEVEL:10	8	12		10			
63966	LOCATION: stomach	0	9			stomach		
63967	QUALITY OF PAIN:Dull ache	0	15					
63970	ONSET: Nov 30,2010	0	6				Nov 30,2010	

Figure 10.1: The gold standard annotations of several targets and some of their attributes.

10.3 Overview of Scoring Metrics

The extraction algorithm processes the text and returns a set of predictions. Each prediction contains the line id of the occurrence, the span identifying the target text, the existence status, negation status, severity, location, onset, duration,

Table 10.1: Prevalence of Non-default Slot Attribute Values Over the Test Corpus.

Attribute	**Count**	**Prevalence**
Negation	160	38%
Severity	230	54%
Location	187	44%
Onset	12	3%
Duration	43	10%
Quality	26	6%
Type	19	4%
Variability	15	4%

quality, type, and event time. Each of these predictions is compared against the gold standard in order to score both the target spans and the attribute values.

In order to compare the results of this research with the state-of-the-art, the evaluations metrics defined for the SemEval 2015 challenge were used. However, metrics that involved disorder spans in SevEval 2015 were changed to target spans in this work. As slots and attributes are largely equivalent between the two, those two terms are used interchangeably from here on.

10.3.1 Target Recognition Metrics

One difference in scoring methodology involved the disorder mention in SemEval 2015 versus the target concept in this task. This is because the target in this research may be a disorder mention (*pain, painful*), or it could be a target term for one of the attributes (*Location:*) in a nursing flow sheet.

The target predictions are scored on their own apart from any modifying attributes. The scoring module reads each prediction and checks whether its line id and span match any in the gold standard. If it does, it is counted as a true positive; if not it is counted as a false positive. When all predictions have been processed,

any gold standard targets that have matched with a prediction are counted as false negatives. From this, precision, recall, and F_2 scores are calculated.

$$Precision = P = \frac{T_{tp}}{T_{tp} + T_{fp}} \tag{10.1}$$

$$Recall = R = \frac{T_{tp}}{T_{tp} + T_{fn}} \tag{10.2}$$

$$F_2 = \frac{2 * P * R}{P + R} \tag{10.3}$$

Where T_{tp} is the number of true positive, T_{fp} are the number of false positive, and T_{fn} are the number of false negative predictions made for a <line id, span> combination.

10.3.2 Slot Attribute Metrics

Unweighted Per-Slot Accuracy. Unweighted per-slot accuracy measures the algorithm's ability to extract the values of a particular slot. It is defined as the number of correct values for that slot across all gold standard instances. This is a modification of the SemEval 2015 metric which only counts values over true positive predictions. For slot k, the unweighted per-slot accuracy is defined as:

$$SlotAccuracy_Unweighted_k = \frac{\sum_{i=1}^{GS} I(gs_{k,i}, ps_{k,i})}{GS} \tag{10.4}$$

where GS is the number of gold standard instances, $gs_{k,i}$ is the gold-standard value of slot s_k at instance i, and $ps_{k,i}$ is the predicted value of slot s_k at instance i, and

I is the identity function: $I(x, y) = 1$ if $x = y$ and 0 otherwise.

Weighted Per-Slot Accuracy. Weighted accuracy was also calculated for each slot. Weighted accuracy was useful because some attributes have very few values specified in the set of gold standard instances. Adding the weight to the calculation gives a better indication of accuracy on an unbalanced set of data. In this research, the weight calculation based on prevalence was modified from that used in SemEval 2015 (see Equation 2.4). In SemEval 2015, slot values were normalized to a small number of terms such as *unmarked, slight, moderate, severe*. Prevalence was based on the ratio of each of these normalized terms in that slot. However, in this research, the task was to extract the actual values specified in the text. For this reason, prevalence was modified to be based on the number of gold standard instances that contained a non-default (usually empty) value for that slot. Prevalence for slot k is defined as:

$$Ps_k = \frac{\sum_{i=1}^{GS} I(gs_{k,i}, ds_{k,i})}{GS} \tag{10.5}$$

where GS is the number of gold standard instances, $gs_{k,i}$ is the gold-standard value of slot s_k at instance i, $ds_{k,i}$ is the default value of slot s_k at instance i, and I is the identity function: $I(x, y) = 1$ if $x = y$ and 0 otherwise.

The weight for slot k ($weight(gs_k)$) containing the default value is Ps_k while the weight for each slot k containing a value other than the default is $1 - Ps_k$.

Once the weight was calculated using the modified prevalence, the weighted accuracy for each slot was the same as that used in SemEval 2015, namely:

$$SlotAccuracy_Weighted_k = \frac{\sum_{k=1}^{GS} weight(gs_{k,i}) * I(gs_{k,i}, ps_{k,i})}{\sum_{i=1}^{TP} weight(gs_{k,i})} \tag{10.6}$$

where GS is the number of gold standard instances, $gs_{k,i}$ is the gold-standard value of slot s_k at instance i, $ps_{k,i}$ is the predicted value of slot s_k at instance i, and I is the identity function: $I(x, y) = 1$ if $x = y$ and 0 otherwise.

10.3.3 Per-Target Accuracy

Per-target accuracy measures the correctness of slot values in a particular prediction. Equations 10.7 and 10.8 calculate the accuracy of a single prediction p_i. **Per-target unweighted accuracy** is defined as:

$$PerTargetAccuracy_Unweighted_i = \frac{\sum_{k=1}^{K} I(gs_{k,i}, ps_{k,i})}{K} \tag{10.7}$$

Per-target weighted accuracy is defined as:

$$PerTargetAccuracy_Weighted_i = \frac{\sum_{k=1}^{K} weight(gs_{k,i}) * I(gs_k, ps_{k,i})}{\sum_{k=1}^{K} weight(gs_{k,i})} \tag{10.8}$$

where K is the number of slots, $gs_{k,i}$ is the gold-standard value of slot $s_{k,i}$ at instance i, $ps_{k,i}$ is the predicted value of slot s_k at instance i, and I is the identity function: $I(x, y) = 1$ if $x = y$ and 0 otherwise.

10.3.4 Overall Evaluation Metrics

Overall Per-Target Accuracy: The average of the per-target accuracies is the same as that used in SemEval 2015. It is calculated over all the true positive predictions in unweighted and weighted scores.

$$Overall_Accuracy_Unweighted = \frac{\sum_{i=1}^{TP} PerTargetAccuracy_Unweighted_i}{TP} \quad (10.9)$$

$$Overall_Accuracy_Weighted = \frac{\sum_{i=1}^{TP} PerTargetAccuracy_Weighted_i}{TP} \quad (10.10)$$

where TP is the number of true positive predictions.

Final Overall Combined Accuracy: The integration of the overall accuracy with the F_2 score of the target span identification gives the final overall combined accuracy. This is the final measure by which teams participating in the SemEval2015 challenge were ranked. It is calculated as both an unweighted and weighted scores. The combined score is defined as:

$$Combined_Overall_Accuracy_Unweighted = F_2 * Overall_Accuracy_Unweighted \quad (10.11)$$

$$Combined_Overall_Accuracy_Weighted = F_2 * Overall_Accuracy_Weighted \quad (10.12)$$

10.4 Results

10.4.1 Target Recognition Results

Table 10.2 shows the results of the extraction of target terms as defined by the metrics in Section 10.3.1. Examples of true positive, false positive, and false

negative extractions of the pain targets can be seen in Section 5.2.2.

Table 10.2: Target Recognition Results

Metric	**Results**
T	401
T_{tp}	376
T_{fp}	69
T_{fn}	25
$Precision$	84.49%
$Recall$	93.77%
F_2	88.89%

10.4.2 Slot Attribute Results

Table 10.3 shows the results of the information extraction algorithm for the slot attributes. The results are expressed using the metrics presented in Section 10.3.2. As the weighted metrics are based on the prevalence of non-default values over the entire corpus (both training and testing), those prevalences are shown in Table 10.1.

Additionally, because precision, recall, and F_2 are more widely used metrics, these are also presented in Table 10.4. However, in this case they are only computed over the set of true positive targets that were found. In each case, $true_positive + false_positive + false_negative = 401$, the number of true positive targets. Examples of true positive, false positive, and false negative extractions of the slot attributes can be seen in Sections 5.2.3 through 5.2.11.

10.4.3 Overall Evaluation Results

Table 10.5 shows the overall combined accuracy of the extraction algorithm over all targets and all slot values. Metrics are those defined in Sections 10.3.1 and

Table 10.3: Slot Attribute Accuracy Results Using SemEval 2015 Metrics

Attribute	**Overall Unweighted**	**Overall Weighted**
Negation	97.01%	96.81%
Severity	95.51%	95.21%
Location	95.76%	96.19%
Quality	98.25%	99.37%
Onset	99.00%	99.71%
Duration	94.76%	98.21%
Variability	99.25%	99.70%
Type	99.00%	99.43%

Table 10.4: Slot Attribute Accuracy Results Using Precision and Recall

Attribute	**Precision**	**Recall**	F_2
Negation	96.81% (364/376)	100.00% (364/364)	98.38%
Severity	96.50% (358/371)	98.62% (358/363)	97.55%
Location	96.51% (359/372)	98.90% (359/363)	97.69%
Quality	98.40% (369/375)	99.73% (369/370)	99.06%
Onset	99.47% (372/374)	99.47% (372/374)	99.47%
Duration	98.61% (355/360)	95.69% (355/371)	97.13%
Variability	99.73% (373/374)	99.47% (373/375)	99.60%
Type	99.20% (372/375)	99.73% (372/373)	99.47%

10.3.4.

10.5 Discussion

10.5.1 Comparison with the State of the Art

These results compare favorably with the current state-of-the-art in a similar task carried out as a SemEval 2015 Challenge to identify medical concepts in clinical text and to extract the values of any modifying attributes. Though similar, there

Table 10.5: Overall Evaluation Results

Metric	**Score**
Target Precision	84.49%
Target Recall	93.77%
Target F_2	88.89%
Overall Unweighted Slot Accuracy	97.14%
Overall Weighted Slot Accuracy	94.01%
Combined Overall Weighted Accuracy	83.56%

were notable differences.

- The VA corpus is characterized by text with very little underlying grammatical structure while the Share corpus notes are grammatically clean.
- The VA corpus contains 48 different types of notes, while the Share corpus is made of up 2 types of notes (radiology and discharge notes).
- The SemEval challenge targeted medical concepts of up to 88,000 types, while this research focused only on pain. However, due to the use of pre-coordinated terms in the standard terminologies, pain actually constituted hundreds of those terms.
- The SemEval challenge normalized disease/disorders and anatomical locations to CUIs; this research did not.
- The SemEval challenge normalized each attribute to a small set of terms. This research extracted the actual values.
- This research allowed for multiple values in a slot. For example, *abd discomfort @ old surgical site* would have two locations: *abd* and *surgical site*. Both

locations would have to be present to count the slot value as a true positive. SemEval only allowed for a single value.

- This algorithm is able to fill in anatomical locations that are part of compound terms, abbreviations, and implied by the term itself (*mylagia*).

- This algorithm was able to fill in pain targets that were not present in the text, but implied by the presence of its attributes (*7/10 hands*). SemEval assumed all targets were explicit.

10.5.2 Sources of Error

Sources of false positives. These were due to the development of the test set of notes as well as the annotations, and the pain targets of *tender* and *pressure*.

1. Many false positives were caused when the negating phrase *denies* occurred on the previous line. Therefore, the pain targets on the following line are listed as affirmed instead of negated.

2. The larger context is not known by the algorithm. For example, *The patient's goal is to be pain free* occurs in the context of an assessment interview. The algorithm has no concept of a patient's goals.

3. Certain portions of notes that contained pain occurrences were unknowingly left out of the test set of notes. Because it was unreasonable to ask the annotators to process 650 pages of notes, they were greatly abridged (see Section 9.3.2). Some of the notes were abridged too far.

4. The annotators themselves missed annotating some occurrences pain events (see Section 9.3.6).

Sources of false negatives. False negatives were caused by the following:

1. In the case of *tender*, there was disagreement between clinicians on when the term constituted a pain event or not. In the initial set of annotations (see Section 9.3), *tender* was considered a sign and not symptom. It was left out of the lexicon as a target term. The second set of annotations, on which the results were calculated, generally did consider the term to indicate a pain event and annotated them as such.

2. In the case of *pressure*, the term was so ambiguous over the entire that it was removed as a target term. However, it was annotated as a pain target/qualifier (*feels pressure, no pain*).

3. In order to filter out the word *pain* that was not used in the context of a pain event, the item was pruned from the ConText graph if it had no modifiers. However, in the **Problem List** portion of the note, the word *Pain* stands alone as a chronic condition. In this case, it should be kept with an implied modifier of *chronic*.

10.5.3 Analysis of Results

The conjecture put forth in Chapter 3 that there were phrasal patterns in the VA Corpus that were strongly discriminative over short distances was correct. Those short-hand phrasal patterns were discovered and successfully leveraged to identify

the presence of medical concepts related to pain and to extract their modifying values.

The specialized lexicons that were built to identify and extract pain target terms, anatomical locations, and temporal terms were also effective in identifying individual words as well as phrases. They also were able to be extended to associate pain terms with their locations, as well as filter out expressions that could not stand on their own. This had the advantage of not relying on a supervised method of machine learning, and was relatively easy to do based on existing terminology systems.

However, this in and of itself was not enough. Underlying the extraction of concepts is a grammar that mirrors the semantic structure of the pain ontologies. It is this concept level grammar that constrains the transitions of the phrases, which phrases are to be included, and how they interact to ensure that the phrases attach in the correct way.

The nursing flow sheets imposed their own unique grammar that was not well-suited to the ConText algorithm. In the prompt, *Patient states current pain due to surgical or other invasive procedure: No*, the pain targets and attributes are positive until they are negated as a separate expression at the end. The prompts in the flow sheets were very specific (although the did vary between VA facilities), and it was difficult to write patterns that would generalize to other institutions. More research is needed in this area.

Because the telegraphic style of the short-hand style text is fundamentally different than clean text, and imposes different algorithmic constraints, it would

be helpful to create a classifier to determine which style of text is currently being processed. When the text is clean, it would be safe to combine lines, perform sentence segmentation, and possibly incorporate other NLP techniques to determine the scope. It would also be helpful to integrate other attributes for events such as experiencer (*patient,family member*), historicity (*history of*), and hypotheticals (*if he has ...*).

This algorithm also would be more effective if the larger context of the pain expression were known. For example, knowing whether the type of note was a nursing note, a surgical note, or a discharge summary may help to determine whether the pain event is occurring during the period covered by the note or whether it was historical. Within the note, it would be helpful to know whether the pain term is occurring as part of the chief complaint, as part of a pain assessment, as part of a physical exam, as part of a patient assessment interview, or as part of a chronic problem list.

Chapter 11: Conclusion

The goal of this research effort was to prove the thesis that an approach that combines semantic and machine learning techniques can be used to extract medical concepts from clinical text for use in clinical decision support systems.

As was seen in Chapter 2, the extraction of medical events in the clinical records is fraught with difficulty. Much of this is due to a large set of formal terms in the biomedical domain. However, with the availability of formal terminologies and the use of NLP and machine learning algorithms, much progress has been made. Current state-of-the-art systems are able to identify occurrence of diseases/disorders in the text and retrieve related attribute data with an overall cumulative weighted accuracy of 80.8%.

However, as was seen in Chapter 3, the difficulty of performing this kind of extraction was magnified in the text of the VA corpus in which a shorthand style of text is used. This text is heavily abbreviated and tends to ignore the rules of grammar, punctuation, and white space. In addition, the use of nursing flow sheets spread the occurrence a single pain event over many lines of text, and requires the targeting of several different types of phrasal expressions.

This research began at the request of physicians at the VA to find occurrences of pain events in the textual notes of patient records. Chapter 4 showed that defining

a formal set of ontologies to model these events, and encoding the extraction results in a structured semantic representation, allows deductive inferencing technologies to reason over the data by combining it with existing biomedical ontologies to produce a semantically rich cognitive search assistant.

Processing the VA corpus to find occurrences of these pain events resulted in a novel approach to clinical information extraction that is robust to grammatically deficient text. It relies on techniques that are able to incorporate micro-contexts by taking into account scope, proximity, and location of multiple interdependent matched expressions. Chapter 5 described how an existing algorithm was extended in order to implement this approach.

In order for this extraction capability to be suitable for clinical decision support, it was necessary that it provide near real-time performance. The algorithm was refined for runtime efficiency. Also, by experimenting with clusters of terms related to the medical concept being extracted, Chapter 6 showed how it was able to leverage these terms in conjunction with the elasticsearch engine to provide a 10x speedup for a processing rate of 24.7 notes/second (0.04 seconds per note).

Some of the text in corpus, such as lab results and medication lists, needed to be filtered out. Chapter 7 relates how the positive distributional characteristics of these types of text are conducive to machine learning algorithms. The chapter described the train/test dataset that were created from the corpus, and that in both cases the Multinomial Naive Bayes had the best accuracy at 99% on the lab reports and 98% on the medication lists.

Chapter 8 established the practical use of this research by presenting a clini-

cal decision support system requested by the physicians at the VA to monitor the progression of post-surgical pain. The application specified the patient and the targets/modifiers, and the algorithm was able to extract the pain events recorded in the patient chart (180 pages of text), and present a visualization of those events. The entire process took 7 seconds.

The evaluation and accuracy of the extraction algorithm, laid out in Chapters 9 and 10, established that the overall cumulative weighted accuracy of 85.33% was greater than current state-of-the-art system for a similar task (80.8%).

As shown by the above, this research validates the thesis that an approach that combines semantic and machine learning techniques can be used to extract medical concepts from clinical text for use in clinical decision support systems. The novel contributions of this research include:

1. A set of formal ontologies to embody a semantic knowledge representation for medical concepts that is modular and can be extended and shared with other applications.

2. A novel approach that is robust to grammatically deficient text using techniques that are able to incorporate micro-contexts by taking into account scope, proximity, and location of multiple interdependent expressions.

3. Information retrieval techniques that can be used in conjunction with the knowledge representation to extract portions of the clinical narrative containing the concept. This allowed the extraction algorithm to be highly efficient and scalable.

4. An API exposed through the REST interface to make the results available for clinical decision support systems and for visualization, as well as a framework for storing and reasoning over the extracted data.

11.1 Limitations

This research does not attempt to normalize precoordinated expressions such as (*severe pain*) to UMLS CUIs. However, it is able to recognize CUIs for individual terms of *severed* and *pain* and use them to create a post-coordinated expression. Because this is the direction that not only SNOMED-CT and ICD-10 are going, and that the use of precoordinated does not make sense when used in a reasoning system, there are no future plans to do this.

The Foundational Model of Anatomy (FMA) was the only anatomical ontology that was able to successfully be converted from its native OWL/XML format into RDF/XML (or any other format) to ingest into Pellet via the Jena Framework. As FMA is a very large ontology, it takes roughly 15 minutes to check for consistency on load. All efforts to convert the smaller anatomical subset of SNOMED-CT to a format the Jena could read were unsuccessful. This limitation must be overcome for the system to be useful outside a research lab.

11.2 Future Work

As this research only handled pain and swelling, it could be extended to not only to other sign/symptoms but diseases/disorders as well. The specialized lexicons that were built for location may work very well for this also.

This approach assumed a non-standard grammar. However, consultant notes are generally much cleaner grammatically. Applying a classifier to determine which type of text is being processed would make the algorithm better suited to handle all styles of text by using the appropriate tools for each.

The nursing flow sheets are difficult, but they have their own type of grammar which is usually of the form: *prompt: response*. There could be a way to discover more general semantic patterns that are useful at other institutions.

While the algorithm was successful in extracting temporal terms, it did not attempt to convert those terms into an actual date/time. Given that these expressions are generally given relative to the date/time of the note in which they are recorded, this should, in fact, be possible. However, the terms can be non-exact, as in *days*. A theory for how to handle this would need to be decided.

Also, the narrative notes may expression a progression of pain events. For example, in the text *Co of midepigastric constant abdominal pain beginning abruptly at 630a which was tolerable thru the day, but gradual grew worse*, there are two pain events: the first started at 6:30 and was tolerable through the day; and the second started sometime near the end of the day and was more severe than tolerable (i.e., the severity turned from mild to moderate). The algorithm is not able to capture these sequences

The extraction algorithm may also perform better if it was known what section of the note the pain mention occurred in. For example, the History of Present Illness may contain both current, historical, and progressive expressions of pain. The Physical Exam portion is always current, but also reflect pain that is the result of

palpations. Pain Assessments are current, but reflect many aspects of pain treatment as well as current pain status.

While these current limitations are significant, it seems that with more effort in this area of research, the majority of occurrences can be found and processed appropriately.

Chapter A: Appendix

A.1 Pain Ontology Definitions

Table A.1: Patient Ontology Definition

Concepts			
Taxonomy	*Patient*		
Relations		**Domain**	**Range**
Object Property			
Data Property	***hasPatientId***	*Patient*	string
	hasPatientFirstName	*Patient*	string
	hasPatientLastName	*Patient*	string
	hasPatientDescription	*Patient*	string
	hasPatientGender	*Patient*	string
	hasPatientRace	*Patient*	string
	hasPatientBirthDate	*Patient*	xsdDateTime

Table A.2: Note Ontology Definition

Concepts			
Taxonomy	*Note*		
Relations		**Domain**	**Range**
Object Property	***hasNote***	*Patient*	*Note*
Data Property	***hasNoteId***	*Note*	string
	hasTitle	*Note*	string
	hasSubTitle	*Note*	string
	hasNoteType	*Note*	string
	hasNoteDateTime	*Note*	xsdDateTime

Table A.3: Event Ontology Definitions

Concepts			
Taxonomy	*Event*		
Relations		**Domain**	**Range**
Object Property	***hasEvent***	*Note*	*Event*
Data Property	***hasEventDateTime***	*Event*	xsdDateTime
	hasEventTimeExpression	*Event*	string

Table A.4: Certainty Ontology Definitions

Concepts			
Taxonomy	*Certainty* *Existence* *Negation* *Definite* *Probable* *CertaintyDefiniteExistence* *CertaintyProbableExistence* *CertaintyDefiniteNegated* *CertaintyProbableNegated*		
Relations		**Domain**	**Range**
Object Property	***hasCertainty***	*Event*	*Certainty*
Data Property	None		

Table A.5: Onset Ontology Definitions

Concepts			
Taxonomy	*Onset*		
Relations		**Domain**	**Range**
Object Property	***hasOnset***	*Symptom*	*Onset*
Data Property	***hasOnsetExpression***	*Onset*	string

Table A.6: Duration Ontology Definitions

Concepts			
Taxonomy	*Duration*		
Relations		**Domain**	**Range**
Object Property	***hasDuration***	*Symptom*	*Duration*
Data Property	***hasDurationExpression***	*Duration*	string

Table A.7: Location Ontology Definitions

Concepts			
Taxonomy	*Location*		
Relations		**Domain**	**Range**
Object Property	***hasLocation***	*Symptom*	*Location*
Data Property	***hasLocationExpression***	*Location*	string

Table A.8: Severity Ontology Definitions

Concepts			
Taxonomy	*Severity* *SeverityNorm* *SeverityNormNone* *SeverityNormMild* *SeverityNormModerate* *SeverityNormSevere* *SeverityScore* *SeverityScoreLexical* *SeverityScoreNumeric*		
Relations		**Domain**	**Range**
Object Property	***hasSeverity*** ***hasSeverityNorm***	*Symptom* *Symptom*	*Severity* *SeverityNorm*
Data Property	***hasSeverityScoreLexicalValue*** ***hasSeverityScoreNumericValue***	*SeverityNorm* *Symptom*	string decimal
Equivalence	*SeverityNormNone* ≡ *CertaintyDefiniteNegated*		

Table A.9: PainSeverity Ontology Definitions

Concepts		**Subclass of**	
Taxonomy	*PainScale_0_to_10*	*SeverityScoreNumeric*	
Relations		**Domain**	**Range**
Object Property	***hasPainSeverity***	*PainSymptom*	*PainScale_0_to_10*
Data Property	***hasPainScale_0_to_10_Value***	*PainScale_0_to_10*	decimal: $[0, 10]$

Table A.10: PainSymptom Ontology Definitions

Concepts		**Subclass of**	
Taxonomy	*PainSymptom*	*Symptom*	
Relations		**Domain**	**Range**
Object Property	None		
Datat Property	None		

Table A.11: PainQuality Ontology Definitions

Concepts		**Subclass of**	
Taxonomy	*PainQuality_0_to_10*		
Relations		**Domain**	**Range**
Object Property	***hasPainQuality***	*PainSymptom*	*PainQuality*
Data Property	***hasPainQualityExpression***	*PainQuality*	string

Table A.12: PainType Ontology Definitions

Concepts		**Subclass of**	
Taxonomy	*PainType_0_to_10*		
Relations		**Domain**	**Range**
Object Property	***hasPainType***	*PainSymptom*	*PainType*
Data Property	***hasPainTypeExpression***	*PainType*	string

Chapter B: Appendix

B.1 Official Annotation Guidelines

High-Lite Any Text In The Note That Indicates Any Of These Aspects Of A Pain Event in a Patients Record

1. PAIN:
   (a) Is it present?
   (b) Include medical terms that constitute pain, e.g., *myalgia*
      i. write in the anatomical location that it corresponds to
2. NEGATIONS
   (a) Is it specifically negated (eg, denied)
3. SEVERITY
   (a) How severe is the pain?
   (b) Write in: none, mild, moderate, severe (or 0,3,6,9)
   (c) High-lite the terms in the text indicate that severity
   (d) What is the range of the pain severity (at its worst, at its best) ?
   (e) If not known - write "unknown"
4. LOCATION
   (a) High-lite text that indicates the location of the pain
   (b) Mark with letter L
   (c) There may be more than one pain event in one sentence - for different anatomical locations - mark all of them
   (d) If location is abbreviated in the text, write in the correct term
5. QUALITY OF PAIN
   (a) High-lite text that indicates the quality of the pain (eg, burning, etc)

(b) Mark with letter Q

6. TYPE OF PAIN
   (a) High-lite text that indicates the quality of the pain (eg, surgical, etc)
   (b) Mark with letter T
7. TIME OF EVENT
   (a) High-lite text that indicates the time of the pain occurence
   (b) Is there an absolute time recorded for that event?
   (c) If time of pain just relative to the date/time of the note, high-lite the time of the note.
8. ONSET
   (a) High-lite text that indicates the time of onset of the pain
   (b) Mark with letter O
   (c) Write down which date(s) on the calendar would that correspond to
9. DURATION
   (a) High-lite text that indicates the duration of the pain
   (b) Mark with letter D
   (c) Write down how many hours/days/weeks would you sat that that corresponds to
   (d) Is the pain Chronic / Acute ?
10. VARIABILITY
   (a) High-lite text that indicates whether the pain is constant / intermittent
11. WHAT NOT TO ANNOTATE:
   (a) How well controlled the pain is - just affirm that pain was present
   (b) How well the patient is able to cope with pain
   (c) How well the patient is able to verbalize pain (or not)
   (d) If pain level is acceptable
   (e) What does the pain keep you from doing (ADLs, etc)
12. MISTAKES
   (a) Put an X through any highlights that you made but changed your mind on
13. ABBREVIATIONS

(a) Write down on this sheet any abbreviations you use in your annotations along with their non-abbreviated form(s)

14. Pain Scale

    (a) On a scale of 0-10, assuming 0 is no pain, what numeric values constitute:

        i. Mild:

        ii. Moderate:

        iii. Severe:

Bibliography

[1] S. Rosenbloom, W. Stead, J. Denny, D. Giuse, N. Lorenzi, S. Brown, and K. Johnson. Generating clinical notes for electronic health record systems. *Applied Clinical Informatics*, 1(3):232–243, 2010.

[2] C.J. McDonald. The barriers to electronic medical record systems and how to overcome them. *J Am Med Inform Assoc*, 4(3):213–221, 1997.

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

[4] Angus Roberts, Robert Gaizauskas, Mark Hepple, George Demetriou, Yikun Guo, Ian Roberts, and Andrea Setzer. Building a semantically annotated corpus of clinical texts. *J. of Biomedical Informatics*, 42(5):950–966, 2009.

[5] R.A. Greenes, editor. *Clinical Decision Support The Road Ahead*. Elsevier Inc, 2007.

[6] 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*, 17(5):507–513, Sep 2010.

[7] Reda Chouffani. Five areas in which healthcare big data, nlp will affect patients. `http://searchhealthit.techtarget.com/news/2240183624/Five-areas-in-which-healthcare-big-data-NLP-will-affect-patients`, 2013. Accessed June 2014.

[8] Computational linguistics.

[9] B. Anderson, I.D.J. Bross, and N. Sager. Grammatical compression in notes and records: Analysis and computation. *American Journal of Computational Linguistics*, 2(4):68–81, 1975.

[10] Nation's largest healthcare system pledges involvement in healthy hospital iniative. https://archive.epa.gov/epapages/newsroom_archive/newsreleases/66a4a31db7c1ae178525703d0067d18b.html. Accessed February 2017.

[11] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald. What can natural language processing do for clinical decision support? *Journal of Biomed Informatics*, 52(5):760–772, 2009.

[12] S. M. Meystre, G. K. Savova, K. C. Kipper-Schuler, and J. F. Hurdle. Extracting information from textual documents in the electronic health record: a review of recent research. *IMIA Yearbook of Medical Informatics*, pages 128–44, 2008.

[13] Umls specialist lexicon. http://www.nlm.nih.gov/pubs/factsheets/umlslex.html. Accessed June 2014.

[14] C. Rosse and J.L.V. Mejino. *Anatomy Ontologies for Bioinformatics: Principles and Practice*, volume 6. Springer, 2007.

[15] Foundational model of anatomy. http://si.washington.edu/projects/fma. Accessed April 2017.

[16] N. Sager, M.S. Lyman, C. Bucknall, N.T. Nhn, and L.J. Tick. Natural language processing and the representation of clinical data. *J Am Med Inform Assoc*, 1(2):142–160, 1996.

[17] Umls metamap. http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/. Accessed June 2014.

[18] O. Bodenreider. The unified medical language system (umls): Integrating biomedical terminology. *Nucleic Acids Research*, pages D267–D270, 2006.

[19] Ken Thompson. Regular expression search algorithm. *Communications of the ACM*, 11(6):419–422, 1968.

[20] J. H. Garvin, Scott L DuVall, B. R. South, B. E. Bray, and D. Bolton. Automated extraction of ejection fraction for quality measurement using regular expressions in uima for heart failure. *J Am Med Inform Assoc*, 19(5):859–866, 2012.

[21] Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy W. Chapman. Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, 42(5):839–851, Oct 2009.

[22] Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. Equations for part-of-speech tagging. In *In Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 784–789, 1993.

[23] Steven Abney and Steven P. Abney. Parsing by chunks. In *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers, 1991.

[24] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *In Proc. IntL Conf. On Language Resources And Evaluation (LREC*, pages 449–454, 2006.

[25] L. W. DAvolio, T. M. Nguyen, S. Goryachev, and L. D. Fiore. Automated concept-level information extraction to reduce the need for custom software and rules development. *J Am Med Inform Assoc*, 18(5):607–613, 2011.

[26] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[27] Taku Kudo and Yuji Matsumoto. Chunking with support vector machines. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[28] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[29] Dingcheng Li, Karin Kipper-Schuler, and Guergana Savova. Conditional random fields and support vector machines for disorder named entity recognition in clinical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, BioNLP '08, pages 94–95, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[30] Unstructured information management architecture (uima). `http://uima-framework.sourceforge.net/`. Accessed June 2014.

[31] P.G. Mutalik, A. Deshpande, and P. M. Nadkarni. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the umls. *J Am Med Inform Assoc*, 8(6):598–609, 2001.

[32] D. F. Sittig, Wright, J. A. Osheroff, B. Middleton, J. M. Teich, J. S. Ash, and et al. Grand challenges in clinical decision support. *J Biomed Inform*, 41(2):38792, 2008.

[33] D. Reicherta, D. Kaufman, Benjamin Bloxhamb, Herbert Chase, and Nomie Elhadad. Cognitive analysis of the summarization of longitudinal patient records. In *AMIA 2010 Symposium Proceedings*, page 667671, 2010.

[34] N. Elhadad, S. Pradhan, S.L. Gorman, S. Manandhar, W. Chapman, and G. Savova. Semeval-2015 task 14: analysis of clinical text. In *Proc. SemEval 2015*, pages 303–310, Denver, CO: ACL, 2015.

[35] Snomed-ct. `http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html`. accessed Jan 2017.

[36] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA, 2003.

[37] Open health nlp (ohnlp) consortium. `http://www.ohnlp.org`. Accessed June 2014.

[38] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, The OBI Consortium, N. Leontis, P. Rocca-Serra, Alan Ruttenberg, S.-A. Sansone, R. H. Scheuermann, N. Shah, P. L. Whetzel, and S. Lewis. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25:1251 – 1255, 2007.

[39] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.

[40] Grega Jakus, Veljko Milutinovic, Sanida Omerovic, and Saso Tomazic. *Concepts, Ontologies, and Knowledge Representation*. Springer Publishing Company, Incorporated, 2013.

[41] World wide web consortium. `http://www.w3.org`. Accessed February 2017.

[42] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. Owl 2: The next step for owl. *Web Semant.*, 6(4):309–322, November 2008.

[43] Semantic web rule language. `http://www.w3.org/Submission/SWRL/`. accessed Jan 2017.

[44] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In *Proceedings of the Third International Joint Conference on Automated Reasoning*, IJCAR'06, pages 292–297, Berlin, Heidelberg, 2006. Springer-Verlag.

[45] National cancer institute thesaurus. `https://ncit.nci.nih.gov/ncitbrowser/`. accessed Jan 2017.

[46] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semant.*, 5(2):51–53, June 2007.

[47] Jena: a java framework for building semantic web and linked data applications. https://jena.apache.org/. accessed Jan 2017.

[48] Sunitha Abburu. A survey on ontology reasoners and comparison. *International Journal of Computer Applications*, 57(17):33–39, 2012.

[49] M. Saeed, C. Lieu, G. Raber, and R. Mark. Mimic ii: A massive temporal icu patient database to support research in intelligent patient monitoring. volume 29, pages 641–644, 2002.

[50] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform*, 34(5):301–10, 2001.

[51] A. R.Aronson and F. M. Lang. An overview of metamap: historical perspective and recent advances. *J Am Med Inform Assoc*, 17(3):229–236, 2010.

[52] The specialist lexicon. http://lexsrv3.nlm.nih.gov/Specialist/Home/index.html. Accessed June 2014.

[53] parsedatetime.py: Parse human-readable date/time strings. https://github.com/bear/parsedatetime. accessed Jan 2017.

[54] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *An Introduction to Information Retrieval - Online Edition*. Cambridge University Press, 2009.

[55] Elizabeth Ford, John A Carroll, Helen E Smith, Donia Scott, and Jackie A Cassell. Extracting information from the text of electronic medical records to improve case detection: a systematic review. *Journal of the American Medical Informatics Association*, 23(5):1007–1015, Sep 2016.

[56] C. Grasso, A. Joshi, and E. Siegel. Beyond ner: Towards semantics in clinical text. In *Proc. ISWC-BDM2I 2015*, 2015.

[57] C. Cortes and V. Vapnik. Support vector networks. *Mach. Learn.*, 20:273–297, 1995.

[58] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[59] STROTHER H. WALKER and DAVID B. DUNCAN. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.

[60] N. S. Altman. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3):175–185, 1992.

[61] lucene.apache. https://lucene.apache.org/core/. Accessed Jan 2017.

[62] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. volume 24, pages 513–523, 1988.

[63] scikit-learn. http://scikit-learn.org/stable/. accessed Jan 2017.

[64] sklearn - countvectorizer. http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html. accessed June 2014.

[65] sklearn - tfidftransformer. http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html#sklearn.feature_extraction.text.TfidfTransformer. accessed Jan 2017.

[66] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1(4):309–317, October 1957.

[67] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

[68] Chih-Chung Chang and Chih-Jen Lin. Training support vector classifiers: Theory and algorithms. *Neural Comput.*, 13(9):2119–2147, September 2001.

[69] Jooyoung Park and Irwin W Sandberg. Approximation and radial-basis-function networks. *Neural computation*, 5(2):305–316, 1993.

[70] Introducing json. http://www.json.org/. accessed Jan 2017.

[71] Sparql query language for rdf. http://www.w3.org/TR/rdf-sparql-query/. Accessed Jan 2017.

[72] C. Grasso, A. Joshi, and E. Siegel. Visualization of pain severity events in clinical records using semantic structures. In *Proc. IEEE-ICSC 2016*, 2016.

[73] Worldvista homepage. http://worldvista.org/. Accessed February 2017.

[74] Jacob Cohen. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968.

[75] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 1977.