DEVELOPING A CLINICAL LINGUISTIC FRAMEWORK FOR PROBLEM LIST GENERATION FROM CLINICAL TEXT

by

Danielle Lee Mowery

BS Biological Sciences, Univ. of Pittsburgh, 2006

MS Health & Rehabilitation Sciences, Univ. of Pittsburgh, 2008

MS Biomedical Informatics, Univ. of Pittsburgh, 2010

Submitted to the Graduate Faculty of

the Department of Biomedical Informatics in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2014

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF BIOMEDICAL INFORMATICS

This dissertation was presented

by

Danielle Lee Mowery

It was defended on

May 30th 2014

and approved by

Wendy Webber Chapman PhD, Chair, University of Utah

Titus Karl Ludwig Schleyer DMD PhD, Clem McDonald Professor, Indiana University

Shyam Visweswaran MD PhD, Assistant Professor, University of Pittsburgh

Janyce Marbury Wiebe PhD, Associate Professor, University of Pittsburgh

Stephane M. Meystre MD PhD, Assistant Professor, University of Utah

Dissertation Director: Wendy Webber Chapman PhD, Chair, University of Utah

DEVELOPING A CLINICAL LINGUISTIC FRAMEWORK FOR PROBLEM LIST GENERATION FROM CLINICAL TEXT

Danielle Lee Mowery, PhD

University of Pittsburgh, 2014

Regulatory institutions such as the Institute of Medicine and Joint Commission endorse problem lists as an effective method to facilitate transitions of care for patients. In practice, the problem list is a common model for documenting a care provider's medical reasoning with respect to a problem and its status during patient care. Although natural language processing (NLP) systems have been developed to support problem list generation, encoding many information layers - morphological, syntactic, semantic, discourse, and pragmatic - can prove computationally expensive. The contribution of each information layer for accurate problem list generation has not been formally assessed. We would expect a problem list generator that relies on natural language processing would improve its performance with the addition of rich semantic features.

We hypothesize that problem list generation can be approached as a two-step classification problem - problem mention status (**Aim One**) and patient problem status (**Aim Two**) classification. In **Aim One**, we will automatically classify the status of each problem mention using semantic features about problems described in the clinical narrative. In **Aim Two**, we will classify active patient problems from individual problem mentions and their statuses.

We believe our proposal is *significant* in two ways. First, our experiments will develop and evaluate semantic features, some commonly modeled and others not in the clinical text. The annotations we use will be made openly available to other NLP researchers to encourage

future research on this task and other related problems including foundational NLP algorithms (assertion classification and coreference resolution) and applied clinical applications (patient timeline and record visualization). Second, by generating and evaluating existing NLP systems, we are building an open-source problem list generator and demonstrating the performance for problem list generation using these features.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

Over the last seven years, I have had the pleasure to learn from and befriend so many wonderful people. Each individual has inspired and taught me to a better person, personally and professionally, in their own ways. I'm so fortunate to have met you all. Thanks for being there for me!

First and foremost, I would like to give thanks to the University of Pittsburgh Department of Biomedical Informatics Training program for the opportunity to learn an appreciation and love for natural language processing and biomedical informatics. Indeed, I have learned from the very best! I would like to thank my committee - Dr. Janyce Wiebe, Dr. Titus Schleyer, Dr. Shyam Visweswaran, Dr. Stephane Meystre, and Dr. Wendy Chapman - for their insightful feedback and continuous encouragement throughout my training. In particular, my mentor and friend, Dr. Wendy Chapman, for believing in me and for providing me with more opportunities than any graduate student could hope for. You have been tremendous!! I would also like to thank Dr. Brian Chapman for challenging me as well as showing great patience and understanding throughout the process. Dr. Pam Jordan and Dr. Henk Harkema for always keeping an open door and receptive ear. Thanks to Melissa Saul for introducing me to the biomedical informatics field and providing practical guidance. Many thanks to Jon Lustgarten, Himanshu Grover, Richard Pelikan, Holly Berty, Jeannie Irwin, Tanja Bekhuis, Rich Wilson, Saeed Zadeh, Zach Landis-Lewis, Kevin McDade, Matt Stokes, Katrina Romagnoli, Toni Porterfield, John Dowling, Melissa Tharp, Thankam Thyvalikakath, Marc Clayton, Wei Wei, and countless others for making my time there enjoyable and memorable.

Late in my studies, I had the opportunity to study at the University of California San Diego, Department of Biomedical Informatics. During my three years there, the faculty, students, and staff accepted me as though I was one of their own. I extend my gratitude to Cindy Wong, Abhishek Kumar, Myoung Lah, Gail Moser, Mindy Ross, Michele Day, Seena Farzaneh, Son Doan, Adela Grando, Aziz Boxwala, Rob El-Kareh, Staal Vinterbo, and again, so many more for their kindness. In particular, I would like to thank Dr. Mike Conway for his guidance and support. This adventure also gave me the opportunity to work with other trainees and learn to become a collaborative researcher. Within my first two months there, I met Sumithra Velupillai, a visiting graduate student from Stockholm University. Our collaborations have brought me so much professional growth, exciting adventures, and great fun. Many thanks to Maria Skeppstedt and Aron Henriksson as well. I am so grateful for your friendships. Within that time, I also had the opportunity to work with many of the good folks at the University of Utah, Department of Biomedical Informatics. I especially want to thank Brett South, Chris Leng, Shuying Shen, and Brad Adams for their support and generosity.

Much of my work has come to fruition due to funding, resources, and forums provided by several institutions. I would like to extend my appreciation to the National Library of Medicine, National Institute of Health, and National Institute of Dental and Craniofacial Research for their financial support. Also, many thanks to the Shared Annotated Resources (ShARe), Conference and Labs of the Evaluation Forum (CLEF), and open-source communities for making datasets, tools, and other resources available for this work and the greater natural language processing community. A big thank you to the American Medical Informatics Association, the Journal of Biomedical Informatics, and the Biomedical Natural Language Processing workshop for giving me a voice in the research community.

Throughout this time, my friends and family have been my personal cheerleaders and sources of strength. Thank you Charles for reminding me that there's more to life than my research career. I'm especially thankful for my mother, Sandy, and sisters, Dana, Sharon, Susan, and Kelly, for always believing in me and supporting my dreams without hesitation

or question. Finally, my dog, Zoey, has been my constant companion for years, by my side while I struggled and burned the midnight oil to complete this work. On paper I rescued her, but in my heart she rescued me. This work is dedicated to her. ♡

1.0 INTRODUCTION

Clinical narratives serve as a rich source of information describing a detailed account of a patient's clinical state over time. These accounts include problems experienced, tests completed, and treatments administered throughout the patient's life leading up to and including the current care encounter. Clinical and biomedical applications have been developed for clinical narratives that apply natural language processing (NLP) approaches to unlock these descriptions of problems, tests, treatments from free text and encode these accounts into a structured form. This structured form can be leveraged by inferencing algorithms for a variety of clinical and biomedical use cases including sentinel event identification (adverse-drug event detection for safety prevention [1]), clinical trial recruitment (smoking status identification for asthma studies [2]), quality assurance (quality measures of colonoscopy procedures for patient care [3]), and public health (syndromic classification for biosurveillance [4]). We propose to use NLP to generate a problem list from clinical narratives and evaluate how information encoded from clinical narratives contributes to accurate problem list generation.

Problem lists are summaries of a patient's clinical problems and their current status. When the problem list does not match the patient's clinical status, the patient is at risk for adverse drug events (treated for non-existing problems) or missed care opportunities (not treated for existing problems) [5]. NLP has been used to aid problem list generation in the clinical domain by proposing missed problems for inclusion in the structured problem list [6], evaluating the consistency of medication lists in the patient record [7], and visualizing problem changes over time in radiology images [8]. Over the last decade, researchers have developed automated problem list generators using NLP – each system using different types of features (morphological, syntactic, semantic, and discourse) [6, 8, 9]. Feature encoding

requires developing and running many NLP modules, increasing the likelihood of generating errors that propagate throughout the automated problem list generator reducing its accuracy. Prior to developing these complex NLP modules, it's necessary to understand the contribution of features for this task and estimate the best possible accuracy for a system given these features. In this thesis, we propose to focus on semantic features and evaluate the contribution of this information with respect to problem list generation. We will generate an active patient problem list derived from problem mentions and their semantic features encoded from the clinical text.

Our long term hypothesis is that *both semantic and discourse features will be necessary for accurate problem list generation.* Our short term hypothesis for this thesis is that *rich semantic features improve the accuracy and precision of active problem list generation over problem list generation without rich semantic features.* We will address this hypothesis in two aims:

Aim One: Build and evaluate a problem mention status generator for clinical narratives. In particular, we will develop classifiers that predict a problem mention status based on semantic features derived from descriptions in the clinical text.

Hypothesis: *Problem mention status classification using rich semantic features will have higher accuracy than problem mention status classification without rich semantic features.*

Aim Two: Build and evaluate a patient problem status generator for clinical narratives. In particular, we will develop classifiers that predict active patient problems based on semantic features derived from **Aim One** and define new semantic and discourse features that could potentially improve accuracy.

Hypothesis: *An active patient problem list generated using rich semantic features will have higher precision than an active patient problem list generated without rich semantic features.*

We will conclude with a discussion about how additional semantic and new discourse features could improve problem list generation. We will use manually corrected, feature annotations whenever possible to evaluate the contribution of this information without noise generated from the NLP modules. Our approach is grounded using a linguistic discourse framework to encode semantic and discourse features (Webber et al. [10]).

2.0 BACKGROUND

Electronic medical records have become an important tool for documenting patient-specific information relevant to providing care. However, a patient record can contain many references to problems that no longer require management. One tool used to reduce the cognitive burden of tracking *current, active* problems from *past, resolved* problems is the problem list.

2.1 PROBLEM LISTS FROM CLINICAL NARRATIVES

The Problem list is a tool for care providers to help facilitate clinical reasoning of a patient's active problems in the problem-oriented medical record (POMR), first developed by Lawrence Weed in the late 1960s [11]. In recent years, health care regulatory institutions – Center for Medicare and Medicaid Services (CMS), Joint Commission (JCAHO), Health Level Seven (HL7), etc. – have advocated the use of problem lists in their incentive and standard programs [12, 13]. As part of the Electronic Health Record Incentive Program, CMS defined 25 objectives used to demonstrate meaningful use of adopted health information technology including the Core Measure 3 objective that states to maintain an up-to-date problem list of current and active diagnoses in addition to historical diagnoses relevant to the patients care [12]. JCAHO defined the Standard IM 6.40 to help improve staff communication using a summary list of all significant diagnoses, procedures, allergies and medications (pg 22) [13]. The HL7 Personal Health Record System Functional Model PH.2.5.1 defines the problem list as a broad set of problems including diagnosis, symptoms, hypotheses and any other problems of interest to the care provider including characterizations indicating problem status including acute, chronic, resolved, historic, and recurrent [13]. Other credible source

definitions for problem lists from regulatory organizations are summarized in *Appendix A: Definitions of Problem Lists from Authoritative Sources* [13].

Problem lists can be generated using two types of data - structured and unstructured - from electronic medical records. Structured data elements can be represented using a drop down list, checkboxes, radio buttons, etc. In contrast, unstructured data elements can be represented using free text fields or clinical narratives. Benefits of using structured data include a predictable, structured format and consistent, conceptual meaning that is important for use of this data by systems downstream. However, structured problem lists may not contain a complete list of relevant problems from a patient encounter. In a pilot study assessing the overlap of problems from structured and unstructured fields, an estimated 50% of problems were only found in the unstructured text reports suggesting many problems could be missing and might need to be considered for problem list generation [14]. Clinical narratives are useful for problem list generation because of the rich details recorded about the patient's problems and related events that would normally be difficult to aggregate from only structured data fields, dispersed throughout the patient's medical record. However, these rich detailed accounts are captured with loose structure which makes encoding these descriptions so difficult. Variable lexical expressions (*dizziness* is synonymous to *vertigo*), ambiguous abbreviations (Does MI mean *myocardial infarction* or *mental illness*?), spelling errors (*haemocyte* vs. *hemocyte*), and telegraphic constructions (*pt +ive for H1N1*) make normalizing information from clinical narratives challenging. NLP tools can be used to regularize and map information from clinical text into a structured problem list.

2.2 LINGUISTIC KNOWLEDGE FOR NATURAL LANGUAGE PROCESSING

Like human understanding of language, natural language processing systems leverage a variety of linguistic information to extract meaning from text.

Morphological: Words are constructed using one or more morphemes [15]. Medical terminologies use Latin and Greek lemmes (base root forms) in conjunction with prefixes and suffixes to construct words representing problems and procedures. Orthographic rules are enforced to ensure acceptable word constructions are generated from constituent parts e.g., *afebrile* = *a* meaning **without** and *febrile* meaning **fever**. Derivations, inflections, compounding, and cliticization are morphological process for generating word variations.

Syntactic: Words are arranged into meaningful linguistic units, phrases and clauses, using categorical tags. Part of speech is a categorical tag associated with a lexeme (lexicography), such as noun, verb, and adjective. There are several commonly used part-of-speech tag sets. For example, the Penn Treebank tag set has 45 tags, whereas the Brown Corpus tag set has 87 tags [15]. Phrases and clauses can be combined into sentences governed by grammar rules that define which combinations are legal. Three types of syntactic ideas for combining words in sentences are constituency, grammatical relations, and subcategorization and dependency. Parsing structures a sentence into a linguistic structure such as a string, tree, or network. Two types of parsing are shallow parsing which generates phrases from individual words and deep parsing which relates phrases to each other. For example, a **noun phrase (NP)** is composed of a **Proper-Noun** like *Radiology*, signifying a unique department.

Semantic: Words, phrases, and clauses convey meaning by describing entities and events. Lexical semantics define the meaning of a concept from its lexicon or dictionary of words. Different words can convey similar meaning (synonymy) e.g., *Addison's disease* is equivalent to *adrenocortical insufficiency*. One word can convey many meanings (polysemy) e.g., *discharge* can be release of a patient from care or a substance from an abscess. Context is important for disambiguating these cases and selecting the correct sense (word sense disambiguation). Disambiguating the meaning of a word or phrase can involve mapping to a standardized vocabulary, terminology, or ontology. For instance, the Unified Medical Language System (UMLS) Specialist Lexicon contains over 300,000 words and over 550,00 lexical variants that map into semantic concepts within the UMLS Metathesaurus [16, 17, 18]. The

Metathesaurus contains over one million concepts of 135 semantic types. Compositional semantics entails deriving the meaning of a concept from the encompassing and surrounding sentences.

Once mapped, linguistic knowledge captured by morphological and syntactic structures about entities and events is linked to non-linguistic knowledge of the world to perform tasks accurately through use of meaning representations (pg 545) [15]. Meaning representations use various formats - first-order logic, semantic networks, and templates - to represent entities and events. Meaning representations serve as constituents to various frameworks - frames, models, and scripts - used to describe the expected roles of entities and events in the world (pg 617) [15]. Several tools and frameworks are available for generating meaning representations and describing entities and events role in the world. For instance, WordNet is a lexical resource that contains sense relations for English words including synonymy, hypernymy, hyponymy, and meronymy [19]. Recently, a similar resource was developed for clinical text, Medical WordNet [20]. Medical WordNet consists of two lexical resources, FactNet and BeliefNet. Medical FactNet describing "true beliefs" held by medical experts and Medical BeliefNet describing "general beliefs" held by non-medical experts about medical phenomenon. The true power of these lexical resources is realized once the representations are integrated with a semantic role network such as FrameNet or Propbank. FrameNet is a framework that captures entities and events as frame elements with attributes and relationships describing their role in a real-world scenario [21]. FrameNet II has an estimated 6,100 fully annotated lexical units, 825 semantic frames, and about 135,000 annotated sentences [22]. Another framework is Propbank that uses verb-driven, predicate-argument representations instead of FrameNet's semantic frames [23]. Propbank contains 20 thematic labels for over 4,500 frame sets defined in the framework.

Discourse: Phrases, clauses, and sentences form semantic units structured as collocated, coherent groups [15]. Discourse structures convey how each semantic unit relates to a previously introduced semantic unit using linguistic context. Semantic units are grouped using low- and high-level discourse structures. Low-level relations enforce meaningful construc-

tions between ideas. Meaningful relations can include temporal order relations e.g., "pain *after* fall.", and coherence relations e.g., "medication overdose *explains* his lethargy". In the general and biomedical domains, the Penn Discourse Treebank and Biomedical Discourse Treebank are two models of such relations from text. For the clinical domain, the UMLS Semantic Network contains 54 possible semantic relationships between semantic concepts from the Metathesaurus. High-level discourse structures like sections describe the subject matter of grouped semantic units. For example, the PAST MEDICAL HISTORY section of a clinical text groups events that occurred in the past.

Pragmatic: In order to fully understand or comprehend the patient's story, one must place the meaning being conveyed about the patient in the context of what is known using situational context. This context can be world or domain knowledge that must be integrated to understand the patient's case. Obscure statements conveying implicit information in the clinical narrative may necessitate knowledge of the report type and information commonly documented in its sections e.g., the sentence "pt *drinks* and *smokes* regularly" in an adult social history is likely documenting the patient's frequency of alcohol consumption and cigarette intake [24]. This situational context could also come from other reports or structured data fields dispersed throughout the electronic medical record. To date, this information layer is still left largely unexplored.

Humans use these information layers to understand language.

2.3 CLINICAL NLP AND MACHINE LEARNING

NLP has been an active field of study since the 1950's [15, 24, 25]. NLP modules were developed for encoding morphological, syntactic, and semantic information layers focusing on rule development for syntactic and semantic parsing [25]. Two approaches widely used include finite state automata and context-free grammars [15]. However, constructing rules with adequate coverage for all possible grammar construction scenarios was challenging and

tedious. Following the late 1980s, researchers developed more robust rules by attributing probabilities to each rule e.g., probabilistic context-free grammars. Statistical approaches made NLP systems able to handle tasks under uncertainty and adaptable to new domains. Statistical approaches that use feature information from data to train a machine learner to build a classification model for a task is called machine learning [26]. In NLP, a variety of linguistic information can be encoded as features used to train the machine learner. Types of machine learning such as reinforcement, semi-supervised, unsupervised, and supervised learning have different learning parameters. For instance, an unsupervised learner accepts linguistic features as an input, but does not know the label of the output. On the other hand, a supervised machine learner accepts linguistic features as input and knows the label of the output. Challenges to using machine learning for an NLP task include 1) deciding which linguistic knowledge to use, 2) defining how to represent the feature in an input vector, 3) defining the output label to predict, and 4) selecting a machine learning approach. When the input and output label for a task is known, a supervised learning approach can be used to build an NLP system quickly with high performance.

2.3.1 Supervised Machine Learners

Three types of supervised machine learners are rule-learning, probabilistic, and discriminative learners. Decision Tree (DT), a rule learner based on inductive learning, generates a predictive model that learns a sequence of the most informative features that maximize the split distinguishing one output class label from another [27]. Information measures are used to define informativeness like information gain. The sequence of features is constructed using recursive partitioning. During each recursive cycle, one feature is deemed as more informative than the others, the feature is added as a variable node to the tree, some feature values become decision branches accompanied by the most probable output label, and other feature values become an attachment point for the next informative feature [27, 28, 26]. This creates a tree-like structure. Advantages of DT include a simple representation for interpretation and simple conversion to a probabilistic classifier (Naïve Bayes tree) [29]. Disadvantages of DT include generation of large trees with many, uninformative distal nodes and inflexible

Boolean rules that do not work with uncertainty. Naïve Bayes (NB), a probabilistic learner based on Bayes' theorem, generates a predictive model using all features to assign the most likely output label given the features [30, 31]. The most likely class is determined from posterior probabilities assuming a strong independence assumption between features and prior probability using average class label estimates from training data. Advantages of NB include tolerance of a large set of features and compensation for class imbalances using prior probabilities. Disadvantages of NB include low classification performance due to violation of the independence assumption. Support Vector Machine (SVM), a discriminative learner, projects input features into a n-dimensional space and defines a linear model with a hyperplane decision boundary to predict one of two output labels [32, 33]. The hyperplane selected optimizes the distance between two classes; the support vectors define the margins between the closest examples of each class. Advantages of SVM include tolerance of a large set of features, low likelihood of over-fitting, and tolerance of spare data vectors. Disadvantages of SVM include a complex model and decision boundary that can be hard to deconstruct.

Although each approach can produce different performances, informative features can increase the likelihood of better classification. The informativeness of a particular feature can be unclear. Therefore, many NLP researchers provide a large number of different features to the learner. A disadvantage of this approach is that it leads to curse of dimensionality, sparseness of data, and misclassification from irrelevant features.

2.3.2 Feature Selection and Machine Learning

Feature selection methodologies can be used to select the most relevant features, thereby reducing the model's complexity, reducing the model's run-time, and increasing the model's generalizability [34]. There are three types of feature selection methods defined by their use during the learning process [34]: *filter*, *embedded*, and *wrapper*. For classification tasks, each method has advantages and disadvantages. Filter methods like Chi-square and Pearson's correlation compute an informative score, rank the features, and select the most relevant features independent of the classifier applied. However, filter methods may filter out fea-

tures without evaluating their relationship with a desired outcome measure. For instance, using a filter method would not guarantee the model optimizes accuracy. Additionally, inadequate features may be selected without proper thresholding and no consideration of feature interactions. Embedded methods like Random Forest and Weighted Naïve Bayes combine feature selection and classification to optimize the classifier accuracy and reduce the number of features used. However, the type of feature selection used is tightly coupled with the classifier. Wrapper methods like sequential forward selection or backward elimination apply search algorithms, training iterations, and cross validation (e.g., best-first, bidirectional search method) to determine the usefulness of individual and interacting feature subsets while optimizing the accuracy of the classifier. Machine learning coupled with feature selection can be used to learn semantic and discourse features useful for accurate problem list generation.

2.4 LINGUISTIC FRAMEWORK AND PROBLEM LIST GENERATION

Care providers use rich semantic information to describe signs and symptoms and discourse to describe their diagnostic reasoning [35]. These clinical descriptions can be modeled using a linguistic framework developed by Webber et al. [10].

2.4.1 Webber Linguistic Framework

Based on Webber et al.'s linguistic definition of discourse as "a means for speakers to relate many ideas conveyed within one sentence or among many such that their sum are greater than the whole [10]," a clinical narrative could be characterized as a discourse containing many diverse elements that contribute to the understanding of a patient's status, which is a sum greater than the whole. Webber defines four discourse structures, *eventualities, discourse relations, functions*, and *topics*, used to relate ideas in a narrative. Eventualities are descriptions of events and states. Discourse relations are low-level constructions between

eventualities that convey a particular semantic relationship. Functions are constructions that serve communicative roles for eventualities. Topics are segments of narrative that convey the "aboutness" of eventualities described in a passage. We can apply these discourse elements using NLP to clinical narratives, grouping them into two practical categories, semantic (*eventualities*) and discourse (*discourse relations*, *functions,* and *topics*) and develop problem list generators using supervised learning to assert the patient's status with respect to the problems mentioned. We expect that automated problem list generation will perform more accurately when integrating semantic and discourse features derived from NLP systems used by existing problem list generators than with semantic features alone. In order to identify potentially useful semantic and discourse features, we reviewed the linguistic and clinical NLP literature.

2.4.2 Semantics and NLP

Lexical semantics is defined as the use of semantic representations "to capture the meaning of linguistic inputs and represent them as entities and events and their relationship to the world as we understand it (pg 548) [15]." We must define a canonical form such that different linguistic units that mean the same thing have the same semantic representation describing its sense. A first step is encoding these linguistic inputs to a common representation or vocabulary.

2.4.2.1 Medical Concepts Eventualities used for clinical information extraction can be disorders, procedures, drugs, anatomy, temporal expressions, and other concepts. Words and phrases describing these eventualties can be mapped to concepts in a standard vocabulary such as the International Classification of Disease-9th version-Clinical Modification (ICD-9-CM) codes [36], Unified Medical Language System (UMLS) [16, 17, 37] and Systematized Nomenclature of Medicine–Clinical Terms (SNOMED-CT) [38].

ICD-9-CM is a collection of classification codes for disease and procedures developed

and maintained by both the National Center for Health Statistics (NCHS) and the Centers for Medicare and Medicaid Services (CMS) to support billing from health care institutions [36]. ICD-9-CM codes are universally used in the healthcare for billing, this classification system only represents subtypes of clinical concepts, not semantic relationships. The UMLS Metathesaurus, developed and distributed by the National Library of Medicine (NLM) is the largest thesaurus in the biomedical domain and serves as an knowledge resource consisting of 135 semantic types [16, 17] including *Signs or Symptoms* and *Diseases or Syndromes*. The NLM also distributes SNOMED-CT, a clinical terminology developed by the College of American Pathologist, as a subset of the UMLS Metathesaurus. It contains 269,864 current classes and 407,510 current names[18]. In an early satisfaction survey of the UMLS, SNOMED-CT, ICD-9-CM codes, and READ codes (codification system used in the UK), UMLS and SNOMED-CT performed better in capturing clinical content both with and without semantic modifiers over other coding systems [39]. More recently, SNOMED-CT has been evaluated for its coverage and practical use for encoding problem lists demonstrating reasonable to superior coverage of diagnosis and problem terms using SNOMED-CT at 84% [40], 88% [41], and 92% [42]. The NLM has also made available the Clinical Observations Recording and Encoding (CORE) subset of SNOMED-CT [38]. This CORE subset is designed to have high coverage by sampling the most frequent problems observed from seven institutions including Beth Israel Deaconess Medical Center, Intermountain Healthcare, Kaiser Permanente, Mayo Clinic, Nebraska University Medical Center, Regenstrief Institute and the Hong Kong Hospital Authority. The CORE subset is comprised of 14,000 terms covering 95% of problems from each institution listed [43]. It was observed that 92% of the most frequent terms were found in the UMLS, and 81% of the terms in the UMLS had associated SNOMED-CT codes. Similar coverage was observed in an independent study by Wright et al [44]. In a study of terms that could not be exactly mapped during the development of the CORE subset, these terms required additional semantic features like negation [45].

NLP tools solely developed for mapping terms to vocabularies include Metamap [46], IndexFinder [47], and KnowledgeMap [48]. Metamap, developed at the National Library

of Medicine, is one of the most widely used concept mapping tools in the biomedical domain. Metamap uses symbolic NLP to map text into the controlled vocabulary [37, 46, 49]. Indexfinder maps to clinical concepts by generating all permutation of word sets and filtering out irrelevant concepts using syntactic and semantic rules [47]. However, it is unclear whether mapping by Indexfinder is more accurate than Metamap. Knowledgemap, another concept mapping algorithm, has been compared against Metamap; however, its performance was only tested for curricular documents (slide presentations, textual outlines) and it is not open-source. For problem list generation, Metamap's reported performances are recall: 70%; precision: 90% (Meystre et al [50]), recall: 88%; precision: 66% (Solti et al [51]), and recall: 56%; precision: 56% (Sibanda et al [9]). Other concept mapping systems were also developed for encoding problems, tests, and treatments as part of the 2010 i2B2/VA Challenge [52] and 2007 and ICD-9-CM codes as part of the 2007 Computational Medicine Challenge [53]. The highest performing system for each challenge achieved an F-measure 85% and 89%, respectively.

2.4.2.2 Negation Concept mapping alone may prove insufficient for accurate problem list generation. A semantic representation must also describe features of the problem so that we understand how it conceptually relates to or differs from other similar problem mentions. For instance, these two mentions of cough, "*complains* of cough" and "*denies* cough", differ by at least one semantic feature - negation. Traditionally, negation is addressed by identifying pertinent negatives in which a event is being denied. Negation has been approached by most NLP systems (NegEx, NegFinder, NegExpander) in two steps - detecting negation cues and determining their scope. One of the most widely used negation algorithms for clinical texts is NegEx [54]. NegEx is a simple lexical-approach comprised of common negation ("*no* pain") and pseudo-negation ("*no change* in pain") regular expressions achieving performances of recall: 82% and precision: 85%. Average precision was reported as high as 97% for ten clinical text types (n=42,106 reports) [55]. NegFinder is a negation approach comprised of a lexical scanner and context-free grammar parser to identify negation terms and the direction of their scope achieving an average performance of recall: 95% and precision: 93% for a variety of clinical texts. NegExpander is a simple algorithm that identifies negation terms in

sentences ("*no* edema and bleeding"), expands their scope across conjunctions, and replaces the text with a tokenized noun phrase (NO_EDEMA and NO_BLEEDING). NegExpander achieves an average performance of recall: 90% and precision: 93% for 100 outpatient notes [56].

2.4.2.3 Certainty In clinical text, a problem is not always clearly negated. A problem can be mentioned as existing with some level of certainty by the writer. For example, in" *unlikely* pneumonia", pneumonia could be described as having a possible existence. Speculative language is used by care providers to "hedge" or lessen their confidence about an assertion such as the existence of a problem diagnosis or cause. Recognizing this phenonenom, NLP researchers have modeled the relationship between negation and certainty as a continuum ranging from definitely positive to negative with varying degrees between [57, 58]. NLP systems designed to detect uncertainty include MITRE's assertion classifier, StAC and pyConText. MITRE's CARAFE-based assertion classifier uses lexical and document features to train a conditional random field and rule-based classifier to assert a problem as *present*, *absent*, or *possible* [59]. MITRE's classifier achieves high performance for present (recall: 98% and precision: 94%) and absent (recall: 92% and precision: 95%) problems and moderate performance for possible (recall: 53% and precision: 77%) problems. StAC, a statistical assertion classifier, uses lexical and syntactic features and a support vector machine to classify a problem with these same labels [57]. StAC's highest reported performance was recall: 88% and precision: 90% on 1,954 de-identified radiology reports from the Computational Medicine Center dataset. pyConText, an extension of NegEx, uses regular expressions to identify uncertainty state cues ("*no definite evidence of* embolism") and selects the uncertainty cue/value ("no definite": probably negated existence) within scope of the problem mention target [58]. pyConText achieves a high performance of recall: 94% and precision: 93% for 658 CTPA reports.

2.4.2.4 Experiencer Family histories are commonly recorded to help care providers assess the patient's risk for a particular disease. NLP systems that detect whether the problem was experienced by the patient or someone else include ConText [60, 61], Hx [62],

and HITEX Family history extractor [63]. ConText is an extension of the NegEx algorithm. ConText assumes all problem mentions are experienced by the patient unless a trigger term occurs with in the problem's scope. This simple heuristic has resulted in perfect recall and precision for 120 reports of 6 different types [61]. Hx is an algorithm for detecting ancillary cancer histories that uses this same heuristic and a Dynamic-Window method to identify the relative or non-relative experiencer. A comparison of the Dynamic-Window and ConText algorithm resulted in similar performances on 300 history and physicals. Other successful algorithms have been reported [63]. A more challenging problem is asserting whether a problem is recent.

2.4.2.5 Temporal Grounding Both historical and recent problems can aid diagnosing; however, recent problems tend to be active, therefore requiring care management. Studies suggest that distinguishing historical problems is more difficult than identifying recent problems [60, 61, 64]. The ConText algorithm assumes all problem mentions are recent, unless otherwise stated. This assumption and features used by ConText achieves moderate performance (recall: 76% and precision: 75%) for predicting whether a problem is historical [61]. Temporal features like verb tense and aspect, temporal expressions and sections can help predict historical problems. Rule learners like Ripper and RL trained with temporal features have been shown to outperform ConText with improved recalls and precisions ranging from 9-10 and 23-28 points, respectively [64]. Other supervised learners like Naive Bayes, k-Nearest Neighbor, and Random Forests using similar features boost improved performance over ConText (recall: 80% and precision: 61%) in recall and precision ranging from 0-23 and 0-11 points, respectively [65]. Some NLP systems such as TN-TIES [66, 67] can assign the time of onset while other systems like TimeText [68, 69, 70] can infer the full duration of a reported problem.

2.4.2.6 Other Semantic Features As part of the 2010 i2B2/VA Challenge, participants developed classifiers to assert a variety of semantic features. Semantic features predicted by NLP systems included whether a patient problem was *present, absent, possible, hypothetical, conditional,* or experienced by *someone else* [52]. The most effective assertion

classifier had an F-measure of 94% [71]. Other types of semantic features can be extracted from care providers' descriptions of patient problems in clinical text. A temporal or modality feature denotes problems expressed in irrealis, conditional, and hypothetical statements [61, 72, 73]. One of the most widely used frameworks for capturing time and modality in the general domain is TimeML [72]. Recent efforts have been made to extend this framework to the clinical domain by adding degree and severity features for a given problem [73]. Other clinical frameworks include the CLEF annotation schema that includes an anatomical location and laterality feature [74]. In addition to semantic features, the CLEF framework also introduces the ability to link multiple mentions of the same problem throughout the narrative.

2.4.3 Discourse and NLP

When care providers read a clinical narrative, they update their understanding of the problem status by identifying, merging, and reconciling related descriptions of the same problem. Specifically, when humans read a narrative, they evoke an entity with its first mention. During each subsequent reference, the reader will access this entity from memory and update the entity's status in memory (pg 696) [15]. According to Webber's discourse model, the purpose of a discourse is to enable the writer to communicate to the reader his or her understanding of some situation, directing the reader in synthesizing a similar model [75]. We would expect an NLP system would need to mimic this behavior to accurately assert a patient problem's status.

2.4.3.1 Discourse relations Throughout a clinical narrative, a writer represents his or her understanding about an entity's status using discourse relations [75]. In the general domain, several efforts exist to model discourse relations between entities and events, including the Penn Discourse Treebank (PDTB) [76] and the Biomedical Discourse Treebank (BioDRB) [77]. PDTB is a collection of 5 discourse relations types between events and entities in articles from the Wall Street Journal [76]. Annotation of the Wall Street Journal consists of over 40,600 tokens and 5 types of relations [76]. Other relation annotations include sense

and attribution annotations. This model was adapted to the biomedical domain as the BioDRB; it includes temporal, conditional, causal and other relationships [77]. In the clinical domain, the UMLS Semantic Network serves as the most widely used discourse relations framework; there are 54 semantic relationships of two types - hierarchical and associative. Hierarchical relationships are *isa* relations e.g., cough *isa* Sign or Symptom. Associative relationships are physical, spatial, functional, temporal, and conceptual relations. Common types of relationships used to describe a problem include anaphoric, causal, and temporal relations.

Anaphoric relations In text, linguistic expressions that refer to the same entity are called coreferential. Anaphoric relation is a type of coreferential relation in which the understanding of one linguistic expression depends on the previous expression [78, 79]. For instance in "The pain was mild. It became severe.", It corefers to the pain. Anaphoric resolution has been the focus of both non-medical and medical domains through shared tasks such as the Message Understanding Conference-6 [80] and i2B2/VA Challenge [81]. For the 2011 i2B2/VA Challenge, the Ontology Development and Information Extraction (ODIE) corpus was annotated with coreferring entity mentions including people, problems, tests, and procedures. The most effective NLP system for mention extraction and coreference resolution was a rule-based system that achieved an F-measure of 70% partial and 72% exact match [81] against the reference standard. In a study of 180 clinical reports of coreferring types from ODIE, the most prevalent anaphoric entities after removing patient mentions were anatomical sites (30%), disease or syndromes (30%), and signs and symptoms (16%) [79]. An NLP system may need to identify and resolve these anaphoric mentions to assert a problem's status accurately. Some clinical NLP systems have been developed for identifying anaphoric expressions in clinical narratives [82, 83]. CaRE, an NLP system for extracting entities and their relationships from discharge summaries, uses orthographic, morphologic, syntactic, semantic, and temporal features with a Decision Tree approach to learn whether any pair of entities are coreferring or not [83]. CaRE's coreference resolution module had an average F-measure of 95% (B-cubed) and 81% (MUC) among pre-annotated clinical entities and events. cTAKES, an NLP system built on UIMA for processing clinical texts, uses

lexical, syntactic, and semantic features and support vector machines with an RBF kernel to create chains for coreferring mentions [82]. The cTAKES coreference module was evaluated with automatically extracted entities and has an F-measure of 69% (B-cubed) and 35% (MUC). Common errors were sentence distance limitations and entity recognition failures.

Causality relations In addition to identifying coreferring entities, recording the causal effects of interventions can provide important information for understanding change of a problem's status. For instance, in "her headache was *resolved* by aspirin", the headache status should update to *resolved*. Causal relations describe the effect one entity or event has on another. Recent efforts have been made to automatically encode causal relations between events and entities described in newspaper and biomedical research articles. For instance, one recent study reported moderate performances (F-measure: 66%) on biomedical texts using a Charniak parser and abductive inference engine [84]. In clinical text, causal relationships are encoded to convey the effects between problems, tests, and treatments. For instance, a causal link can be automatically encoded or inferred between two problems such as a disease and a radiological finding [85]. In the 2010 i2B2/VA Challenge, causal relations captured between entities - problems, tests, and treatments - included *worsens, improves, and causes* [52]. A support vector machine trained using lexical, syntactic, and semantic features produced the highest performance (F-measure: 74%) classifying these relations [86].

Temporal relations Causal relationships between entities alone can lead the reader to incorrectly change a problem's status. A reader must consider the time order of events conveyed in clinical narratives. Temporal relations describe the order between two events in time. One of the most influential temporal frameworks was developed by James Allen in the 1980s [87]. According to Allen's temporal framework, events are modeled as intervals with a start, duration, and end. Allen proposes 13 interval relations such as *before and after*, *during and contains*, and *overlaps and overlapped-by*. These interval types signify the values for temporal links (TLINKS) between events and temporal expressions in the TimeML standard [72]. This framework was recently extended by a few groups to capture temporal relations between clinical events and temporal expressions [73, 88]. Clinical NLP

models [73] and systems [68, 69, 70, 89] commonly use only 3 of Allen's relations - *before*, *after*, and *during*. Several shared tasks have drawn the attention of both the general and medical NLP community to build temporal extraction and relation systems including previous SemEval/TempEval [90, 91, 92] and recent i2B2/VA challenges [93]. In the clinical domain, temporal ordering systems can be used to build patient timelines [94], generate record chronicles [89] and answer clinical questions [95]. Few systems have formally evaluated and reported the performance of their temporal relations modules on clinical texts [89]. The Clinical e-Science Framework (CLEF) demonstrator, is an information extraction system developed to integrate unstructured and structured record data into a patient chronicle [89]. In a study of 98 documents from 5 patients, 159 CLEF entities, 605 temporal expressions, and 201 TLINKS were marked [96]. The rule-based algorithm uses TLINKS and verb features to predict temporal order with moderate performance (recall: 59% and precision: 73%) representing a a good baseline for this task. Recent temporal ordering efforts have demonstrated better accuracies using trained BoosTexter to order topical segments (average accuracy: 78%) [97] and ranked support vector machines to order medical events (average accuracy: 82%) [98].

2.4.3.2 Topic When explicit discourse relations fall short, the document structure or format can help. For instance, writers tend to discuss the same problem in the paragraph or section which can aid coreference resolution algorithms [82]. In clinical texts, changes in temporality are strongly correlated with changes in topic [97]. Topics describe entities and events and what is said about them. These descriptions often manifest in collocated text segments [10]. In the general domain, topic segmentation or predicting subtopic changes within a document can be used to aid information retrieval, information extraction, and text summarization efforts [15]. High-level topic changes can be predicted using conventionalized document formats for domain-specific document types. For instance, biomedical articles follow Introduction, Methods, Results, Discussion, and Conclusion (IMRDC) and clinical texts follow Subjective, Objective, Assessment, and Plan (SOAP) [15]. In clinical texts, low-level or subtopic changes can often be identified using section headers. Section headers

provide useful information for encoding semantic information about problems such as time occurrence, problem experiencers, and anatomical locations. For instance, HISTORY OF PRESENT ILLNESS denotes events leading up to the current visit, FAMILY HISTORY denotes problems experienced by others, and PHYSICAL EXAM - NECK denotes the anatomical location of the problem. Section taggers that predict both explicit and implicit topic changes have been developed and evaluated for clinical text [99, 100]. SecTag [99], is a probabilistic section classifier trained for history and physical exams. An evaluation of SecTag demonstrated higher classification for all sections (recall: 99.0% and precision: 95.6%), major sections (recall: 98.6% and precision: 96.2%), and unlabeled sections (recall: 96.6% and precision: 86.8%). A rule and probabilistic hybrid approach [100] for radiology and urology reports predicts labeled and unlabeled sections with accuracies over 95%. When section headers are not explicitly marked in clinical texts, implicit changes in topic can be predicted using lexical cohesion-based approaches such as the TextTiling algorithm [15]. When text segments fall out of a section's scope, unlabeled segments can be classified using supervised learning approaches trained for conventionalized topic models such as a SOAP classifier [101].

2.4.3.3 Function Entities and events, their relations, and topical context are important components used to update what is known about problems to a reader. A writer also uses communicative functions to specify how the reader should integrate this information for coherent understanding and accurate status updates [10]. Functions convey the speaker's communicative intention between text segments describing how they relate to each other to convey a greater meaning in the discourse. Coherence is a type of function. Coherence relations have been explored in the general domain. Coherence relations have an important feature in Rhetorical Structure Theory (RST) [15]. RST consists of 23 rhetorical relations such as *Background*, *Elaboration* or *Contrast* used to describe how two text segments - a central nucleus and a dependent satellite - relate to each other. Using RST, the intentions of the writer can be annotated and the document constructed to show these relations as a tree. Some rhetorical relations from RST can be found in an annotated subset of the Penn Discourse Treebank [76] called the RST Discourse TreeBank [102]. Another notable function

is a speech act. Speech acts have been annotated as part of Dialog Act Markup in Several Layers (DAMSL) in dialogs from switchboard data [103]. The DAMSL scheme contains two types of dialog acts, forward and backward communicative functions. Both function types are organized as a hierarchy of classes. Forward communicative functions consist of 5 super classes such as *Statement, Influencing Addressee Future Action,* and *Performative*. Backward communicative functions consist of 4 super classes such as *Accept, Understand, Answer,* and *Information-Relation*. To our knowledge, no efforts have been made to annotate clinical text with these coherence relations or communicative acts. Rhetorical relations and speech acts can provide important predictors for when a reader should update their understanding of a problem. For instance, the rhetorical relation *correction* may signify a change or update for a patient's problem status.

Writers and readers use various semantic and discourse features to convey and understand a patient's clinical state. In order to generate an accurate problem list, an NLP system may need to integrate this information to identify a problem and predict it's status. The clinical NLP community has produced several symbolic NLP systems that encode a variety of these features including MedLEE [104], SAPHIRE [105], Symtext [106], MPLUS [107], HiTEX [63], and cTAKES [108]. A few researchers have developed NLP systems that encode and integrate features specifically for problem list generation.

This thesis will focus on encoding semantic aspects of this linguistic framework, specifically eventualities, for the task of developing a natural language processing system to generate problem lists from clinical text. We will leave encoding discourse relations, functions, and topics as these discourse features relate to problem list generation to future work. However, we will conclude with a discussion about how discourse features could improve active problem list generation based on our experiments with semantic features.

2.4.4 Natural Language Processing and Problem Lists

In the past decade, the use of natural language processing for problem list generation has drawn the attention of the NLP community [109, 51, 9, 8].

2.4.4.1 Meystre et al. Meystre et al. (University of Utah) developed a two-part Automated Problem List (APL) generator - a background and foreground application [109, 6]. The background module extracts problems and their contexts (document type, sections, negation, etc.) from cardiology and surgical notes using MetaMap, NegEx, and regular expressions. These features serve as input to a trained Bayesian Network that asserts whether a problem was present (probable or certain in the present or past) or not. The foreground application suggests these proposed problems to care providers for consideration into the "official", structured problem list labeled as *active, inactive, proposed*, or *error*. The background NLP system for extracting 80 types of cardiac and general medicine problems from 160 reports of different types achieved a recall of 90% and precision of 69%[50]. The system was evaluated using a prospective randomized control study in the intensive care and cardiovascular units at Intermountain Health Care. It was found to improve the sensitivity of the problem list (from 9% to 41%) and the timeliness of problem addition (from 6 days to 2 days) [6].

2.4.4.2 Solti et al. Solti et al. (University of Washington) validated the generalizability of the Meystre framework by developing a prototypical problem list generator for identifying explicit and implicit problems in cardiology progress notes [51]. Their system also uses MetaMap to extract problems including problems beyond the 80 problems studied by Meystre et al. Their evaluation is limited to confirming the sensitivity and precision results observed by Meystre el al. achieving a recall of 88% and precision 66%.

2.4.4.3 Sibanda et al. Sibanda et al. (Massachusetts Institute of Technology) developed the Category and Relationship Extractor (CaRE) system for extracting semantic in-

formation from discharge summaries [9]. The CaRE system uses lexical, syntactic, semantic, and discourse features as input to trained Support Vector Machines modules applied to subtasks of de-identification, semantic category recognition, assertion classification, and semantic relationship classification. Semantic categories annotated include *diseases, symptoms, substances, practitioners, dosages, tests, results,* and *treatments.* Assertions marked include *present, absent, uncertain, and alter-association.* Semantic relationships labeled between semantic categories were limited to relationships including only disease and symptom entities such as "Treatment *cures* disease" and "Treatment *administered for* disease. This information is extracted and presented in a proposed problem list which consists of extracted problems, their assertions, and care outcomes from related tests and treatments. The evaluation of this system is focused toward an assessment of the system's individual modules and a theoretical problem list format is proposed. Both the statistical semantic category recognizer and rule-based assertion classifier achieve F-measures above 90% for most categories. The statistical semantic relationship recognizer achieves an F-measure of 67%.

2.4.4.4 Bashyam et al. Bashyam et al. (University of California Irvine) developed a system for extracting problems from radiology reports and discharge summaries - organizing and visualizing these problems on radiographs [8]. To date, this is the most complex problem list generator consisting of both rule-based and probabilistic approaches (e.g., Bayes filter, entity extractor, and discourse analyzer) for extracting problems, their contexts, and coreferring relationships as input to a Bayesian belief networks. These problems are organized using four dimensions - causal, existential, temporal, and spatial – in the electronic medical record and visualized over DICOM images. The reported performance of select NLP modules - Bayes filter - recall: 96%, precision: 94%, entity extractor - recall: 87%, precision: 96%, and discourse analyzer- recall: 97%, precision: 97% - suggest a high performing solution to problem list generation.

Comparisons between systems is challenging. Each of these automated problem list generators vary by domain, input features, classification approach, and system evaluation. In

terms of domain, the Solti system focuses on cardiology cases, the Sibanda addresses covers emergency medical cases, the Meystre system addresses surgical and cardiology cases, and the Bashyam system covers radiology cases. Regarding feature inputs, the Solti system relies solely on semantic features, the Sibanda system trains mainly on lexical, syntactic, semantic and discourse features, the Meystre system utlizes lexical, semantic, and discourse features, and the Bashyam system integrates lexical, syntactic, semantic, and discourse features. With respect to approach, the Solti system uses Metamap only, the Meystre system trains a Bayesian Network, the Sibanda system trains a Support Vector Machine, and the Bashyam system uses a variety of Bayesian approaches. In terms of evaluation, the Solti system is assessed only through problem identification, the Meystre system is evaluated by end system output, the Sibanda system is assessed by individual subtasks, and the Bashyam system is evaluated by only select modules. Although several of these studies have evaluated the performance of semantic extraction modules used to produce the problem list, to our knowledge, none of these studies have evaluated the contribution of these semantic features and their relationship to accurate problem list generation.

For this thesis, we focused our investigation on the development and evaluation of semantic features. In **Aim One**, we built and evaluated classifiers for problem mention status generator for clinical narratives. In particular, we developed a problem mention (instance-level problems) classifier that predicts a problem mention's status based on semantic features derived from descriptions in the clinical text. We evaluated the contribution of each semantic feature to accurate problem mention status generation. In **Aim Two**, we built and evaluated a patient problem status generator for clinical narratives. In particular, we developed a patient problem (document-level problems) classifier that predicts active patient problems based on the underlying semantic features derived from Aim One. We leveraged these features to develop a rule-based approach that predicts active patient problems for the problem list. We conclude with a discussion of future efforts to develop classifiers that encode both semantic and discourse features as we described using the Webber framework.

3.0 AIM 1

In this chapter, we begin by asking a simple question "what active problems need to be managed for this patient?". Initial review of our tools suggested that a richer semantic model was necessary to automatically address this question. One non-trivial problem to understanding a patient's problem status is determining whether the problem is temporally relevant to the current encounter. We developed a rich temporal and semantic schema that we hypothesized could prove useful for determining when a problem was experienced and asserting a problem mention's status within the clinical narrative. The following papers describe these studies and our findings in more detail.

Aim One: Build and evaluate a problem mention status generator for clinical narratives. In particular, we will develop classifiers that predict a problem mention status based on semantic features derived from descriptions in the clinical text.

Hypothesis: *Problem mention status classification using rich semantic features will have higher accuracy than problem mention status classification without rich semantic features.*

3.1 DISTINGUISHING HISTORICAL FROM CURRENT PROBLEMS IN CLINICAL REPORTS – WHICH TEXTUAL FEATURES HELP?

This study was published as an abstract and manuscript in the proceedings of the *2008 and 2009 Biomedical Natural Language Processing workshops*, respectively [64, 110]. Permissions to use unspecified excerpts from this manuscript were obtained from the original publisher.

3.1.1 Motivation

We aimed to determine what features are useful for identifying active patient problems. Specifically, we manually annotated data to learn about the particular usefulness of these features in clinical text and to know how many annotators are needed to produce a reliable reference standard. Our long-term goal is to develop a system that encodes semantic and discourse features that would distinguish active problems from all other mentions. Semantic features we hypothesized were relevant predictors included experiencer, negation and temporality that could help determine whether a problem was experienced recently by the patient. Our lab's existing system ConText could assert whether a problem was affirmed or experienced by the patient, but less accurate at discerning recent problems from historical findings [60, 61]. Previous study suggests that ConText performs with moderate recall (76%) and precision (75%) for predicting historical findings across various report types. This result suggested that trigger terms and simple temporal expressions are not sufficient for the task of distinguishing current problems from historical findings. We hypothesized that more temporal features could improve ConText's performance.

3.1.2 Research Questions

We designed a study to determine 1) *which temporal features discern current problems from historical findings* and 2) *whether rich temporal features can discern current problems from historical findings*?

3.1.3 Methods

We randomly selected seven reports from 6 clinical report types (discharge summaries, surgical pathology, radiology, echocardiograms, operative gastrointestinal, and emergency department reports) dictated at the University of Pittsburgh Medical Center during 2007. These reports were pre-annotated for problems and a temporal label - *historical* or *recent* - by an infectious disease physician.

Which temporal features discern current problems from historical findings?

To address this question, we annotated the following four temporal features from clinical text:

Temporal Expressions: Temporal expressions are time operators like dates (*May 5th 2005*) and durations (*for past two days*), as well as clinical processes related to the encounter (*discharge, transfer*). For each problem, we annotated whether a temporal expression modified it and, if so, the category of temporal expression. We used six major categories from Zhou et al. [69] including: *Date and Time*, *Relative Date and Time*, *Durations*, *Key Events*, *Fuzzy Time*, and *No Temporal Expression*. These categories also have types. For instance, *Relative Date and Time* has a type *Yesterday*, *Today*, or *Tomorrow*. For the problem in the sentence "The patient had a stroke in **May 2006**", the temporal expression category is *Date and Time* with type *Date*. Statements without a temporal expression were annotated *No Temporal Expression* with type *N/A*.

Tense and Aspect: Tense and aspect define how a verb is situated and related to a particular time. We used TimeML Specification 1.2.1 forstandardization of tense and aspect where examples of tense include *Past* or *Present* and aspect may be *Perfective*, *Progressive*, *Both* or *None* as found in Saur, et al. [72]. We annotated the verb that scoped a problem mention and annotated its tense and aspect. The primary verb may be a predicate adjective integral to interpretation of the problem (Left ventricle *is enlarged*), a verb preceding the

problem (*has* hypertension), or a verb following a problem (Chest pain *has resolved*). In "her chest pain *has resolved*," we would mark "has resolved" with tense *Present* and aspect *Perfective*. Statements without verbs (e.g., No murmurs) would be annotated *Null* for both.

Trigger Terms: We annotated lexical cues that provide temporal information about a problem. For example, in the statement, "Patient has **past history** of diabetes," we would annotate "history" as *Trigger Term: Yes* and would note the exact trigger term. We used ConText's lexical cues [60, 61].

Sections: Sections are "clinically meaningful segments and topic labels which act independently of the unique narrative" for a patient [99]. Examples of report sections include *Review of Systems* (Emergency Department), *Findings* (Operative Gastrointestinal and Radiology), and *Discharge Diagnosis* (Emergency Department and Discharge Summary). We extended Dennys section schema with explicit, report-specific section headers not included in the original terminology. Similar to Denny, we assigned implied sections in which there was an obvious change of topic and paragraph marker. For instance, if the sentence "the patient is allergic to penicillin" followed the *Social History* section, we annotated the section as *Allergies*, even if there was not a section heading for allergies.

Specifically, two computational linguists and one biomedical informatician annotated the pre-annotated problems temporal expressions and trigger term features iteratively in groups of six (one of each report type). Between iterations, the three annotators resolved disagreements through discussion and updated our guidelines. We used Cohen's kappa, agreement taking account expected chance, to measure agreement for temporal features (Eq 3.1). C categorical labels compared using a C-by-C contingency table or confusion matrix can be used to represent agreements (diagonals) and disagreements (discordant pairs) for assigning an i instance annotation to a particular label. Observed Percent Agreement (A_O) is the proportion of instance annotations agreements. Beyond Observed Percent Agreement, Cohen's Kappa (k) takes into consideration expected agreement by chance (A_E). Expected agreement is computed from the marginal probabilities of the table or matrix representing

the product of the probabilities that each annotator will assign an instance annotation to that particular category. Cohen's kappa for temporal expressions and trigger terms by the final iteration was good at 0.66 and 0.69, respectively. Finally, the biomedical informatician annotated sections, verb tense, and aspect consulting with one of the computational linguists for unclear cases.

Cohen's (k) =

$$\frac{(A_O - A_E)}{(1 - A_E)} \tag{3.1}$$

Can rich temporal features discern current problems from historical findings?

Each problem was annotated as historical or recent. The presence (positive instances) and absence (negative instances) of the temporal label (historical or recent) were used to define true and false positives and true and false negatives between the reference standard and the automated classifier - see Table 1.

	Historical		**Recent**	
	Reference Standard	**Automated Classifier**	**Reference Standard**	**Automated Classifier**
True Positive (TP)	historical	historical	recent	recent
True Negative (TN)	recent	recent	historical	historical
False Positive (FP)	recent	historical	historical	recent
False Negative (FN)	historical	recent	recent	historical

Table 1: Definitions for Agreement and Performance Metrics

Since the goal of this study was to redesign the existing rule-based system, ConText, we experimented with supervised, rule learning approaches including Decision Tree (J48) [28], Repeated Incremental Pruning to Produce Error Reduction or Ripper (JRip) [111], and Rule Learner (RL) [112]. We used ten fold cross-validation to train each learner using temporal features encoded for problem mentions to predict whether a problem was historical or recent.

3.1.4 Results

The dataset contained 854 problem mentions with a distribution of 113 (13%) historical and 741 (87%) recent statuses.

Which temporal features discern current problems from historical findings?

The J48 Decision Tree algorithm learned 27 rules, six for predicting problems as historical and the remaining for classifying the problem as recent. The rules predominantly incorporated the trigger term and verb tense and aspect feature values. JRip learned nine rules, eight for classifying the historical temporal category and one "otherwise" rule for the majority class. The JRip rules most heavily incorporated the section feature. The RL algorithm found 79 rules, 18 of which predict the historical category. JRip and RL predicted the following sections alone can be used to predict a problem as historical: *Past Medical History*, *Allergies*, and *Social History*. Both J48 and RL learned that trigger terms like *previous*, *known*, and *history* predict historical. There was only one common, simple rule for the historical category found amongst all three learners: the trigger term *no change* predicts the historical category. All algorithms learned a number of rules that include two features values; however, none of the compound rules were common amongst all three algorithms.

Can rich temporal features discern current problems from historical findings?

We compared the performance of each rule learner - Decision Tree, Ripper, and Rule Learner - to ConText for 854 problem mentions from 42 reports. Table 2 shows the performance of each algorithm on the data set. Each rule learner demonstrated superior accuracy over the ConText baseline (92.4%) by Decision Tree: 1.6%, Ripper: 4.7%, and Rule Learner: 4.4% points. The RL algorithm outperformed all other algorithms in almost all evaluation measures. The RL scores were computed based on classifying the 42 cases (eight historical) for which the algorithm did not make a prediction as recent. ConText and J48, which exclusively relied on trigger terms, had lower recall for the historical category. All of the

rule learners out-performed ConText. JRip and RL showed substantially higher recall for assigning the historical category, which is the most important measure in a comparison with ConText, because ConText assigns the default value of recent unless there is textual evidence to indicate a historical classification. Although the majority class baseline shows high accuracy due to high prevalence of the recent category, all other classifiers show even higher accuracy, achieving fairly high recall and precision for the historical cases while maintaining high performance on the recent category.

Algorithm	**Accuracy (Both)**	**Recall (Historical)**	**Precision (Historical)**	**Recall (Recent)**	**Precision (Recent)**
ConText	92.4	73.2	70.1	95.3	95.9
J48	94.0	62.8	88.8	98.8	94.6
RL	96.8	82.2	97.8	99.7	97.5
JRip	97.1	83.2	94.0	99.2	97.5

Table 2: Performance of ConText, Decision Tree, Ripper, Rule Learner algorithms

3.1.5 Discussion

We evaluated which features predict whether a problem is historical or recent. Due to high prevalence of the recent category, we were especially interested in discovering temporal features that predict whether a problem is historical.

Which temporal features discern current problems from historical findings?

With one exception (*date* greater than four weeks prior to the current visit), temporal expression features always occurred in compound rules in which the temporal expression value had to co-occur with another feature value. For instance, any temporal expression in the category *key event* had to also occur in the *secondary diagnosis* section to classify the

problem as historical. For example, in "SECONDARY DIAGNOSIS: Status post Coronary artery bypass graft with complication of mediastinitis" the key event is the *coronary artery bypass graft*, the section is *secondary diagnosis*, and the correct classification is historical. Similarly, verb tense and aspect were only useful in conjunction with other feature values. One rule predicted a problem as historical if the problem was modified by the trigger term history and fell within the scope of a *present* tense verb with no aspect. An example of this is "The patient is a 50 year old male with *history* of hypertension".

Intuitively, one would think that a past tense verb would always predict historical; however, we found the presence of a past tense verb with no aspect was a feature only when the problem was in the *Patient History* section. Sometimes the absence of a verb in conjunction with another feature value predicted a problem as historical. For example, in the sentences "PAST MEDICAL HISTORY: *History* of COPD. Also diabetes" *also* functioned as a trigger term that extended the scope of a previous trigger term, *history*, in the antecedent sentence. A few historical trigger terms were discovered as simple rules by the rule learners: *no change*, *previous*, *known*, *status post*, and *history*. A few rules incorporated both a trigger term and a particular section header value. One rule predicted historical if the trigger term was *status post* and the problem occurred in the *History of Present Illness* section. This rule would classify the problem CABG as historical in "HISTORY OF PRESENT ILLNESS: The patient is...*status post* CABG. One important detail to note is that a number of the temporal expressions categorized as Fuzzy Time also act as trigger terms, such as *history* and *status post*both of which were learned by J48. A historical trigger term did not always predict the category historical. In the sentence "No focal sensory or motor deficits on *history*," *history* may suggest that the problem was not previously documented, but was interpreted as not presently identified during the current physical exam. Finally, sections appeared in the majority of JRip and RL historical rules: 4/8 simple rules and 13/18 compound rules. A few sections were consistently classified as historical: *Past Medical History*, *Allergies*, and *Social History*. One important point to address is that these sections were manually annotated.

Our results revealed a few unexpected observations. We found at least two trigger terms

indicated in the J48 rules, *also* and *status post*, which did not have the same predictive ability across report genres. For instance, in the statement "TRANSFER DIAGNOSIS: *status post* coiling for left posterior internal carotid artery aneurysm," *status post* indicates the reason for the transfer as an inpatient from the Emergency Department and the problem is recent. In contrast, *status post* in a Surgical Pathology report was interpreted to mean historical (e.g., PATIENT HISTORY: *Status post* double lung transplant for COPD.) In these instances, document knowledge of the meaning of the section may be useful to resolve these cases.

Another unexpected finding was that the trigger term *chronic* was predictive of recent rather than historical. This may seem counterintuitive; however, in the statement "We are treating this as chronic musculoskeletal pain with oxycodone", the problem is being referenced in the context of the reason for the current visit. Contextual information surrounding the problem, in this case treating or administering medication for the problem, may help discriminate several of these cases.

Can rich temporal features discern current problems from historical findings?

We assessed ConText in relation to the rules learned from manually annotated temporal features. J48 and ConText emphasized the use of trigger terms as predictors of whether a problem was historical or recent and performed with roughly the same overall accuracy. JRip and RL learned rules that incorporated other feature values including sections and temporal expressions, resulting in a 12% increase in historical recall over ConText and a 31% increase in historical recall over J48. Many of the rules we learned can be easily extracted and incorporated into ConText (e.g., trigger terms *previous* and *no change*). The ConText algorithm largely relies on the use of trigger terms like *history* and one section header, *Past Medical History*. By incorporating additional section headers that may strongly predict historical, ConText could potentially predict a problem as historical when a trigger term is absent and the header title is the only predictor as in the case of "ALLERGIES: peanut allergy". Although these sections header may only be applied to Emergency Department and Discharge

Summaries, trigger terms and temporal expressions may be generalizable across genre of reports. Some rules do not lend themselves to ConTexts trigger-term-based approach, particularly those that require sophisticated representation and reasoning. For example, ConText only reasons some simple durations like *several day history*. ConText cannot compute dates from the current visit to reason that a problem occurred in the past (e.g., stroke in *March 2000*). The algorithm performance would gain from such a function.

3.1.6 Conclusion

Although most problems in six clinical report genres are recent problems, identifying those that are historical is important in understanding a patient's clinical state. A simple algorithm that relies on lexical cues and simple temporal expressions can classify the majority of historical problems, but our results indicate that the ability to reason with temporal expressions, to recognize tense and aspect, and to place problem in the context of their report sections will improve historical classification. We learned that we can distinguish recent problems from historical problems with more temporal features. All rule learners outperform ConText in terms of overall accuracy, historical precision, and recent recall.

From this study, we concluded that richer features could help distinguish historical problems from recent problems. We hypothesized that richer semantic features could also help discern active problem mentions and other problem mention statuses.

3.2 SEMANTIC ANNOTATION OF CLINICAL EVENTS FOR GENERATING PROBLEM LISTS

Related publications were published as abstracts and manuscripts in the proceedings of the *2010 American Medical Informatics Association Annual Symposium Proceedings* [88], *2012 Biomedical Natural Language Processing Workshop* [113], and *2013 American Medical Informatics Association Annual Symposium (3rd place in Student Paper Competition)* [114]. Permissions to use unspecified excerpts from the manuscripts were obtained from the original publishers.

3.2.1 Motivation

From our previous study, we learned that adding more semantic features to a prediction model can improve a system's accuracy. We hypothesized that other semantic features could help distinguish active problems from other mentions. However, annotating many semantic features can be tedious and error-prone. We wanted to determine whether we could alleviate the burden of annotating many semantic features using pre-annotated default values. We hypothesized that adding more semantic features could help distinguish active problem status from other mention statuses.

3.2.2 Research Questions

We designed a study to 1) *develop an annotation model to encode problems and their semantic features that help identify patient problem mentions and their statuses*, 2) *determine whether annotators can annotate these semantic features and their statuses with high agreement and reliability*, and 3) *evaluate whether semantic features can accurately predict a problem status*?

3.2.3 Methods

We reviewed the linguistic literature for semantic features that might predict a problem status.

Can we develop an annotation model to encode problems and their semantic features that help identify problem mentions and their statuses?

In recent years, several annotation schemas were developed to model the clinical information contained in clinical reports, including CLEF [74] and i2B2 VA/Challenge [52]. Our aim was to develop a schema that integrates named clinical eventualities like problems and semantic features that are important for automatically identifying problem mentions and their statuses and filtering in only active problems that should be added to a patient problem list. We borrowed heavily from these existing schemas adding new elements when they did not already exist. We also aimed to align our annotated elements with current annotation initiatives in the NLP community including SHARP Common Type System [115] and ShARe Semantic Schema [116] to support the development of a generalizable NLP problem list generator applicable to data from different institutions and different report types. We applied Webber's discourse framework to define semantic annotations as *eventualities* and their attributes as features in Table 3. For each problem mention, one of the following status labels could be assigned: *Active*, *Inactive*, *Resolved*, *Proposed*, *Negated*, or *Other*.

Annotation Type	**Webber Elements**	**Annotation Features**
Semantics	*eventualities* (problems) and their semantic features	Problem, Experiencer, Existence, Aspect, Certainty, Intermittency, Change, Generalized/Conditional, Mental State, Relation to Current Visit, Historical
Status		*Active*, *Inactive*, *Resolved*, *Proposed*, *Negated*, *Other*

Table 3: Semantic features and problem mention statuses annotated

For this pilot study, we conducted the annotation of problem mentions and their statuses in three phases: Phase 1) Problem Mention Annotation, Phase 2) Problem Semantic Feature Annotation, and Phase 3) Problem Status Annotation.

Phase 1) Problem Mention Annotation: We defined a problem mention as all conditions represented as signs, symptoms, diagnoses, and test results. For instance, "Patient had minor [chest pain]PM.", chest pain is a problem mention. We randomly selected 1,557 de-identified, emergency department reports from the University of Pittsburgh Medical Center. These reports were annotated for problem mentions (instance-level) by a physician, including signs, symptoms, findings, and diagnoses according to the guidelines described in Chapman et al. [117, 118]. For this study, we considered these gold standard annotations and did not develop a system for problem mention boundary detection and normalization since we were only interested in predicting a problem mention's status.

Phase 2) Problem Semantic Feature Annotation: Next, we modeled semantic features that would help answer the following questions: Who experienced the problem (Experiencer)? Was the problem asserted as being present (Existence)? What phase is the problem occurring in (Aspect)? How certain is the physician that the problem exists (Certainty)? Was the problem intermittent in nature (Intermittency)? Did the problem change (Change)? Was the problem expressed in a generalized or conditional manner (Generalized-Conditional)? Did the writer express personal feelings or beliefs in postulating the problem (Mental State)? When did the problem occur relative to the current visit (Relation to CV)? Whether the problem began more than 2 weeks before the current visit (Historical)? For each semantic feature, we provided a default value. For each problem mention in the text, we asked annotators to mark the corresponding semantic feature value representing the context of the said problem mention by either keeping or changing each semantic feature's default value. We provide an example problem mention annotation for each **semantic feature** and give its *definition* and possible values including **bolded values** as the encoded values applied to the example sentence and starred values* as the default values provided to annotators [114]:

Experiencer: *who is experiencing the problem.*

Ex. The patients mother had breast cancer.

Experiencer: **other**, patient*

Existence: *whether a problem was present or not in the context of the mention.*

He denies chest pain.

Existence: **no**, yes*

Change: *whether there is variation in degree or quality of the problem*

Ex. She has had recurrent episodes of viral meningitis.

Change: unmarked*, changing, unchanging, decreasing, increasing, worsening, improving, **recurrence**

Intermittency: *whether the problem is episodic in nature.*

Ex. White female who complains of maroon stools two times.

Intermittent: unmarked*, **yes**, no

Certainty: *the amount of certainty expressed about whether a problem exists or not;* Note: this value is coordinated with Existence value.

Ex. I have no suspicion for bacterial infection.

Certainty: unmarked*, **high**, moderate, low; Existence: **no**, yes

Mental State: *whether an outward thought or feeling about a problem is mentioned.*

Ex. It seems to me there is some active GI bleeding.

Mental State: **yes**, no*

Generalized/Conditional: *whether a problem is described in a non-particular or conditional context.*

Ex. The patient has chest pain at rest.

Generalized/Conditional: **yes**, no*

Relation to Current Visit: *position of the problem time interval to the current encounter.*

Ex. Past medical history: Chronic Obstructive Pulmonary Disease.

Relation to Current Visit: **before**, meets_overlaps*, after

Historical: *whether a problem started greater than 2 weeks before current visit.*

Ex. Past medical history: Chronic Obstructive Pulmonary Disease.

Historical: **start >2 wks**, start <2 wks, not clear*, not applicable

Can annotators annotate semantic features and statuses with high agreement and reliability?

We validated the annotation model by conducting a pilot annotation agreement study. We generated a reference standard of problems and their semantic features using the following approach. We randomly selected (n=35) of the original 1,557 reports from Phase 1. We used 5 emergency department reports to develop our guidelines. For each problem mention, we annotated the semantic feature values describing the problem in the context of the sentence. Using 30 emergency department reports, a final reference standard of semantic feature values for each problem was generated independently and adjudicated using consensus review by a computational linguist and a biomedical informatician. Disagreements were discussed and resolved with two computational linguists. We tested annotators understanding and ability to annotate the model. We recruited medical and non-medical students. We trained all subjects together on how to annotate assertions using annotation software (Protege 3.3.1 with the Knowtator plugin) and guidelines. Subjects were provided pre-annotations of problems from emergency department reports (n=283 problems). These

problem annotations contained a preset majority label for each semantic feature (default value). Subjects were instructed to change the default values using the context of the sentence containing the problem mention (default+annotator). For each semantic feature, we tested agreement using average Cohen's kappa values and reliability using a generalizability coefficient.

Cohen's Kappa: Inter-annotator Agreement (IAA) is the a reliability measure that evaluates the degree to which two annotators agree with each other. We computed IAA using Cohen's Kappa (Eq. 3.1) explained by Hripcsak and et al. [119] and Artstein and Poesio [120]. Kappa coefficients greater than 0.70 are considered acceptable.

Generalizability Coefficient: Relative Generalizability Coefficients are measures used to indicate how many annotators are necessary to reliably annotate each category based on the inter and intra- annotator variance. The assumption is that as you add annotator annotations you may reduce these variances and converge toward a higher, hypothetical annotation performance. Similar to Kappa, the closer to 100 the better the performance; performances above 0.70 are acceptable. Since the problem annotations were pre-spanned each possible attribute of a problem can be ranked and represented as a numeric. For example, Certainty feature values can be encoded as 1=unmarked, 2=low, 3=moderate, and 4=high. For each semantic feature, we calculated the Relative Generalizability Coefficient explained by Hripcsak and et al. [119] and graphed the hypothetical gains in performance over time based on the annotations generated by annotators.

Phase 3) Problem Status Annotation: Finally, for each problem mention in the text, we defined status definitions that would help physicians generate a problem list of mentioned problems and their statuses, and prioritize other problem mentions based on their relevance to the patient encounter and current state of health. We trained two biomedical informaticians (post doctorates) to annotated each problem mention with a status label (below). One domain expert (physician), adjudicated (Adj) the disagreements, creating the final reference standard. We measured inter-annotator agreement using Cohen's kappa (Eq

3.1). According to our model of problem lists, a mention of a problem mention can have one of six possible status labels:

Active (A): *a problem mention occurring with high certainty within the patient with an onset within two weeks of the admission and being actively managed during the current episode of care.*

Inactive (I): *a problem mention chronically experienced by the patient, but not being managed during the current episode of care.*

Proposed (P): *a problem mention being considered as occurring or diagnosed with less than high certainty.*

Negated (N): *a problem mention being denied or that never occurred.*

Resolved (R): *a problem mention that occurred during the current episode of care, but was either successfully treated or culminated on its own.*

Other (O): *any other problem mention not classified with the five previous status labels.*

Can semantic features accurately predict a problem status?

Using the reference standard generated for semantic features of problems, we conducted a proof of concept study to evaluate the informativeness of the semantic annotations when predicting a problem mentions status. In order to evaluate the informativeness of the semantic annotations when predicting a problem mentions status, we split the dataset into training (70%) and test (30%). Using Weka 3.6.8, we selected three supervised learning classifiers Decision Tree [28], Naïve Bayes [31], and Support Vector Machine [33] to predict a problem mention status. We used problem and aspectual phase attributes as input features.

We evaluated the semantic input features using a feature selection study. Using 10-fold cross validation and the training set, we implemented a best-first, bidirectional search method and Accuracy evaluation metric to learn the informativeness of each semantic input features for each classifier. We report the proportion of folds that identified each attribute as informative to classification on the training set. We built a classifier using the full training set and applying only the input features observed as useful in one or more training folds to classify unseen problem mention statuses on the held out test set. We report the performance of the classifier for both training and test sets using Accuracy, Area under the Receiver Operating Curve (ROC), Recall, Precision, and F1 score. In order to evaluate the informativeness of each semantic feature individually, we trained a Support Vector Machine using 10-fold cross validation. Finally, also we conducted an ablation study (leave-one-semantic-feature-out) using this system to evaluate how much performance dropped for each status label and it's eliminated feature.

3.2.4 Results

In this section, we report results of our annotation study and of our problem mention status classification study.

Can annotators annotate semantic features and statuses with high agreement and reliability?

We assessed the average IAA between annotators for annotating semantic features for each pre-annotated problem mention. In Figure 1 , the average kappa agreement between an annotator and the reference standard for problem mentions varied from low kappa for Intermittency: 0.39±0.1 and Generalized or Conditional: 0.46±0.3 to moderate kappa for Magnitude before Current Visit: 0.5±0.2, Certainty: 0.52±0.1, Units before Current Visit: 0.56±0.2, Mental State: 0.59±0.2, Change: 0.63±0.1, Relation to Current Visit: 0.64±0.1 to high kappa for Existence: 0.8±0.1 and Experiencer: 1.0±0 [114]. For aspectual phase mentions, annotators correctly identified an average of 21 matches with the reference stan-

dard. Annotators achieved high kappa for Phase Type: 0.96±0.3 (not shown).

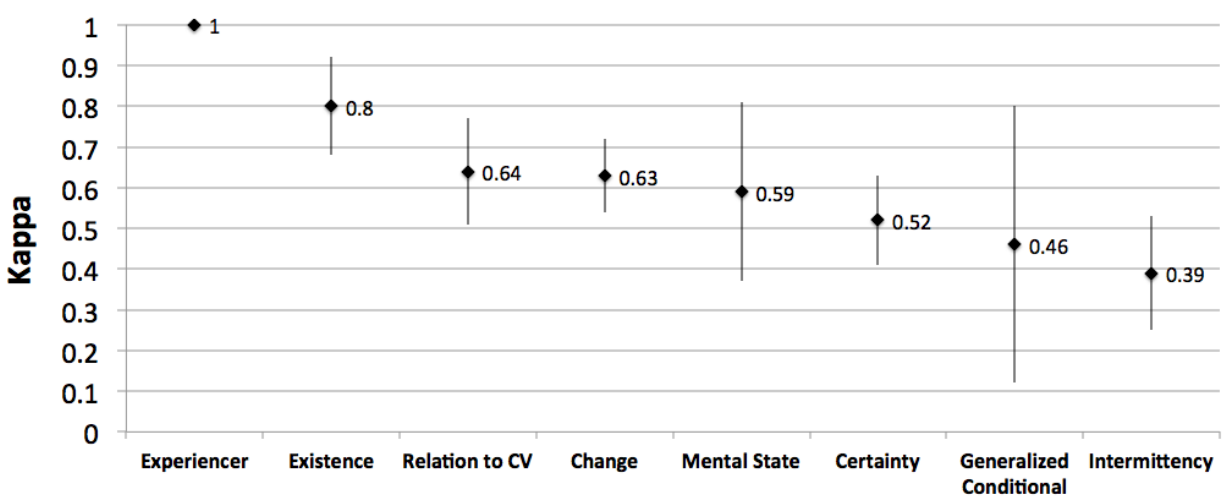


Figure 1: Average kappa agreement with standard deviation for each semantic feature

We assessed the reliability of each semantic feature and projected the number of annotators necessary to produce a hypothetical increase in the reliability using the generalizability coefficient. Figure 2 demonstrates the projected increase in reliability as the number of annotators increases. Semantic features are labeled below the number of annotators needed to reach a coefficient over 0.70. Using a generalizability coefficient above 0.70, we determined we would need the following number of annotators to obtain reliable annotations for each category: 1 (Experiencer and Negation), 2 (Certainty, Change, Mental State, Relation to CV), 4 (Intermittency), and 8 (GeneralizedConditional)[121].

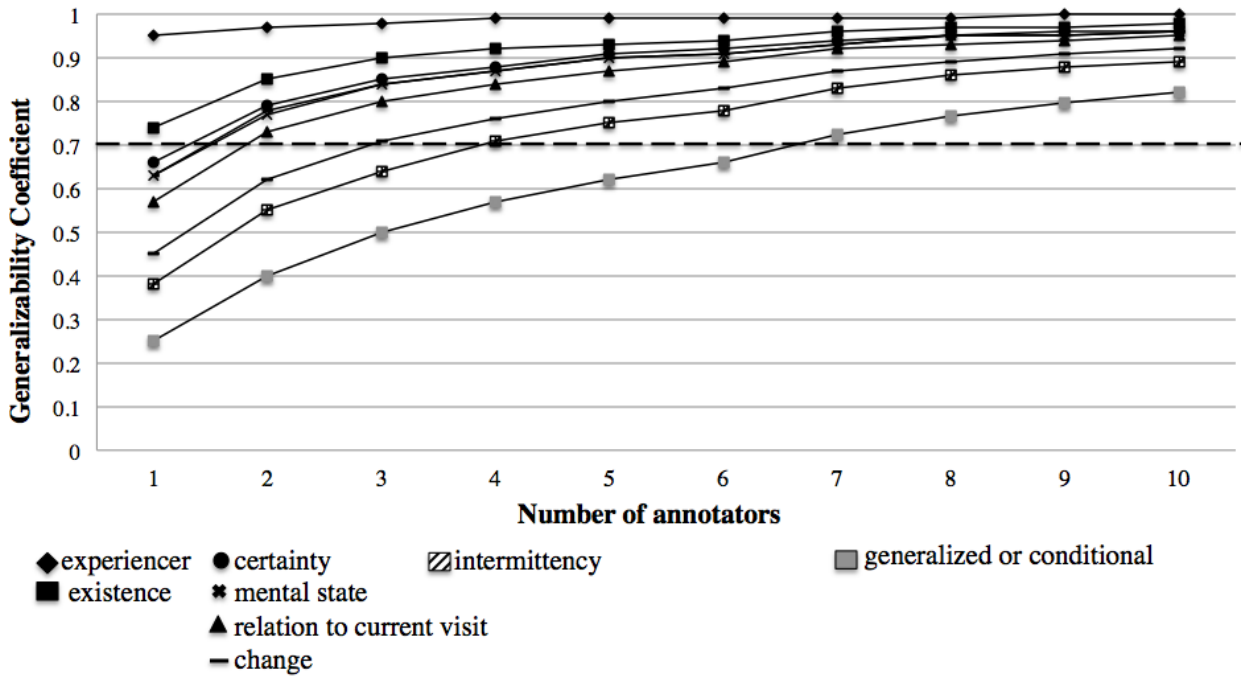


Figure 2: Generalizability coefficients for each semantic feature produced by n number of annotators (actual and hypothetical)

We evaluated the pairwise agreement between annotators for asserting a problem mention's status. Kappa pair-wise agreement was A1-A2 (23.6%), A1-Adj (33.4%), and A2-Adj (77.3%) [114]. The most prevalent status was Active among A1, A2, and Adj annotators in Table 4. The majority of disagreements between A1-A2 were Inactive/Active.

	Active (A)	**Inactive (I)**	**Proposed (P)**	**Resolved (R)**	**Negated (N)**	**Other (O)**
A1	110 (39%)	101 (36%)	5 (2%)	29 (2%)	31 (11%)	7 (2%)
A2	198 (70%)	28 (10%)	7 (2%)	0 (0%)	28 (10%)	22 (8%)
Adj	181 (64%)	21 (7%)	7 (2%)	22 (8%)	26 (9%)	25 (9%)

Table 4: Distribution of status labels for each annotator

Can semantic features accurately predict a problem status?

We evaluated the relevance of each semantic feature for asserting the status of a problem mention. We observed problem semantic features, Experiencer and Existence, are consistently 100% informative for asserting a problem mention's status among classifiers [114]. Naïve Bayes and Support Vector Machine determined all features relevant for at least 1 fold. In contrast Decision Tree, only determined 5 of 10 features relevant.

Problem Attributes	**Decision Tree**	**Naïve Bayes**	**Support Vector Machine**
Experiencer	10 (100%)	10 (100%)	10 (100%)
Existence	10 (100%)	10 (100%)	10 (100%)
Change	0 (0%)	8 (80%)	10 (100%)
Intermittency	0 (0%)	3 (30%)	4 (40%)
Certainty	7 (70%)	8 (80%)	10 (100%)
Mental State	0 (0%)	2 (20%)	9 (90%)
GeneralizedConditional	0 (0%)	1 (10%)	3 (30%)
Relation to Current Visit	0 (0%)	8 (80%)	10 (100%)
Historical	0 (0%)	6 (60%)	6 (60%)
Aspectual Phase	1 (10%)	8 (80%)	10 (100%)

Table 5: Count (%) of Folds/10 that an attribute was determined as relevant.

Our training set of 198 (70%) problem mentions had a distribution of *Active* 127 (64%), *Inactive* 15 (8%), *Proposed* 6 (3%), *Resolved* 15 (8%), *Negated* 18 (9%), and *Other* 17 (9%); our test set of 85 (30%) problem mentions had a distribution of *Active* 54 (64%), *Inactive* 6 (7%), *Proposed* 2 (3%), *Resolved* 7 (8%), *Negated* 8 (9%), and *Other* 8 (9%). All classifiers outperformed a majority class baseline (*Active*: 64% Overall Accuracy) in Table 6. For Weighted Average Accuracy, the test set was between 4-9 points lower than the training set among classifiers. For Weighted Accuracy and F1 Score, Support Vector Machines demonstrated higher performance over Decision Tree and Naïve Bayes. Performances were higher for *Active* and *Negated* and lower for *Inactive* and *Resolved* among classifiers.

Classifier	**Status**	**ROC**		**Rec**		**Prec**		**F1**	
		Train	Test	Train	Test	Train	Test	Train	Test
Decision Tree	A	80.9	83.8	93.7	90.7	77.8	77.8	85.0	83.8
	I	79.3	89.9	0.0	0.0	0.0	0.0	0.0	0.0
	P	82.0	67.8	33.3	50.0	50.0	50.0	40.0	50.0
	R	56.9	73.4	0.0	0.0	0.0	0.0	0.0	0.0
	N	99.0	98.1	100.0	100.0	0.9	72.7	94.7	84.2
	O	94.2	75.4	94.1	62.5	80.0	55.6	86.5	58.8
Wt. Ave		81.8	83.5	78.3	74.1	66.5	62.7	71.8	67.9
Naïve Bayes	A	85.6	84.1	89.0	83.3	77.9	75.0	83.1	78.9
	I	83.4	87.6	6.7	0.0	9.1	0.0	7.7	0.0
	P	95.2	98.8	16.7	0.0	50.0	0.0	25.0	0.0
	R	85.1	80.5	13.3	0.0	40.0	0.0	20.0	0.0
	N	99.4	100.0	94.4	100.0	89.5	72.7	91.9	84.2
	O	97.8	86.4	70.6	50.0	75.0	50.0	72.7	50.0
Wt. Ave		88.0	86.1	73.7	67.1	69.8	59.2	70.7	62.8
Support Vector	A	81.7	76.0	92.9	87.0	84.9	81.0	88.7	83.9
Machine	I	85.6	81.0	13.3	0.0	100.0	0.0	85.6	0.0
	P	99.3	99.1	83.3	50.0	71.4	50.0	76.9	50.0
	R	87.9	76.8	46.7	28.6	70.0	50.0	56.0	36.4
	N	99.7	99.1	100.0	100.0	90.0	72.7	94.7	84.2
	O	99.2	78.0	100.0	75.0	85.0	60.0	91.9	66.7
Wt. Ave		86.1	79.3	84.3	75.3	85.0	69.3	81.8	71.7

Table 6: Classifier performances on training and test data.

The full dataset of 283 problem mentions had a distribution of *Active* 181 (64%), *Inactive* 21 (8%), *Proposed* 8 (3%), *Resolved* 21 (8%), *Negated* 26 (9%), and *Other* 25 (9%). Baseline F1 was highly variable with performance as high as 89 for *Negated* and as low as 13 for *Inactive*. Figure 3 shows performance drops ranging from 4-100% (F1: baseline) for each status when a semantic feature was held out. For each held out feature, we report the most significant drops from baseline:

Active (F1: 83): *Existence* reduced performance by 4%.

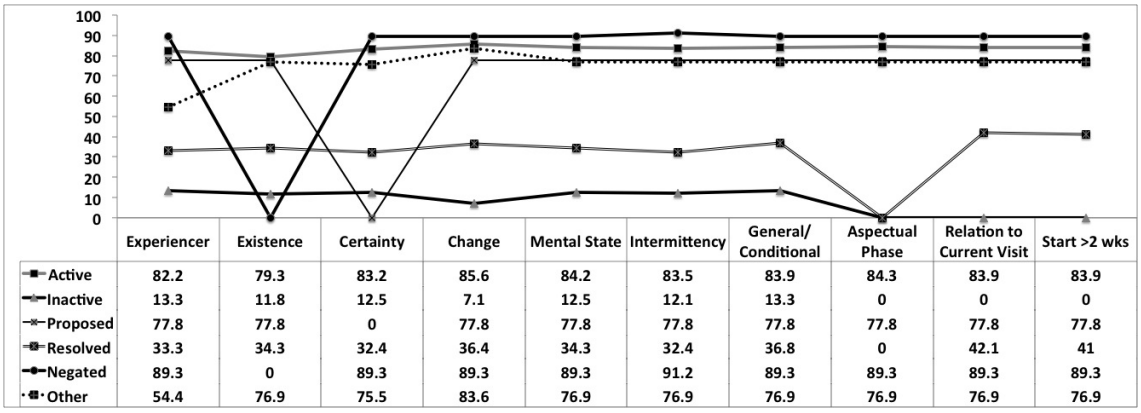
Inactive (F1: 13): *Change*, *Aspectual Phase*, *Relation to Current Visit*, and *Historical* reduced performance ranging from 45-100%.

Proposed (F1: 78): *Certainty* reduced performance by 100%.

Resolved (F1: 33): *Aspectual Phase* reduced performance by 100%.

Negated (F1: 89): *Existence* reduced performance by 100%.

Other (F1: 77): *Experiencer* reduced performance by 29%.



	Experiencer	Existence	Certainty	Change	Mental State	Intermittency	General/ Conditional	Aspectual Phase	Relation to Current Visit	Start >2 wks
Active	82.2	79.3	83.2	85.6	84.2	83.5	83.9	84.3	83.9	83.9
Inactive	13.3	11.8	12.5	7.1	12.5	12.1	13.3	0	0	0
Proposed	77.8	77.8	0	77.8	77.8	77.8	77.8	77.8	77.8	77.8
Resolved	33.3	34.3	32.4	36.4	34.3	32.4	36.8	0	42.1	41
Negated	89.3	0	89.3	89.3	89.3	91.2	89.3	89.3	89.3	89.3
Other	54.4	76.9	75.5	83.6	76.9	76.9	76.9	76.9	76.9	76.9

Figure 3: Ablation Study: F1 performance by status

3.2.5 Discussion

For this study, we aimed to 1) develop an annotation model to encode problems and their semantic features that help identify patient problem mentions and their statuses, 2) determine whether annotators can annotate these semantic features and their statuses with high agreement and reliability, and 3) evaluate whether semantic features can accurately predict

a problem status?

Can annotators annotate these semantic features and their statuses with high agreement and reliability?

We introduced an annotation schema for clinical information extraction of events for generating an accurate problem mention status. We learned that agreement for problem semantic features ranges from moderate to high. This observation is not surprising as the literature shows agreement suffers beyond 2 categories especially for less prevalent categories [122]. Indeed, a study of the CLEF schema reports moderate F1 scores in the 60s for entity and relationship annotations from clinical narratives [74].

Can semantic features accurately predict a problem status?

From our feature selection study, we learned that attributes like Experiencer, Existence, and Certainty are consistently more informative than other attributes for predicting mention status among classifiers. From our classification study, we observed that classifiers (Naïve Bayes and Support Vector Machine) that use rare occurring attributes like Change, Mental State, and Intermittency perform better than a classifier (Decision Tree) that does not use them. We suspect our classifiers performed poorly predicting status labels for *Inactive*, *Resolved*, and *Proposed* due to subtle differences in definition between status labels (*Inactive* and *Resolved*) and few instances in the dataset. In terms of comparable studies, like the i2B2 assertion classification, other researchers have demonstrated adding lexical, syntactic, section, and other semantic annotations can boost performance [59]. In future work, we plan to assess how annotating words and phrases as evidence representing the rationale for assigning a particular attribute value can help teach a supervised machine learner such as the Support Vector Machine how to automatically assign new problem annotation attributes reducing annotator efforts [123]. For example, in the sentence, "possible pneumonia", instruct the annotator to annotate the word *possible* as evidence of Certainty: Moderate.

For the ablation study, we learned which features are most informative for asserting a problem mention status by observing the largest drop in performance once a semantic features was held out. As a first step toward evaluating the usefulness of each semantic feature, we performed the study using a multi-class machine learner. This preliminary work suggests relationships between semantic attributes and predicting a problem mentions status. The initial findings are intuitive. Based on our status definitions and these observed general relationships, we have developed a rule-based approach to asserting a problem mention's status using the following rules:

1. **If** Existence: no, assign *Negated*.
2. **Else if** Certainty: moderate or low, assign *Proposed*.
3. **Else if** Change: NOT(unmarked) AND Relation to Current Visit: after, assign *Active*.
4. **Else if** Experiencer: other OR (GeneralizedConditional: yes OR Historical: N/A), assign *Other*.
5. **Else if** Relation To Current Visit: before OR Historical: greater than 2 weeks, assign *Inactive*.
6. **Else if** Existence: yes AND (Aspectual Phase: culmination OR Change: improved), assign *Resolved*.
7. **Else if** Relation to Current Visit: After OR Historical: N/A, assign *Other*.
8. **Else**, assign *Active*.

In future work, we will develop individual classifiers for each status and evaluate the predictive ability of each individual attribute value. We will expand our study and apply these rules to classify statuses for problem mention from other report types such as discharge summaries, radiology, electrocardiograms, and echocardiograms. These problem mentions and their statuses will be used to infer annotated document-level patient problem annotations and their statuses for this corpus. These studies will be discussed in next chapter.

3.2.6 Conclusion

We concluded that some semantic features can be more difficult to annotate than others. However, some features can be useful for predicting a problem mention's status from clinical text for more prevalent statuses. We apply rules learned from the ablation study to assert a problem mention's status in Chapter 4, Section 4.3.

In this study, we were able to discern the active problem mentions from other problem mention statuses. We identified semantic features necessary for asserting other statuses. We hypothesized that these other problem mention statuses could be used to filter non-*Active* problems from an active patient problem list.

4.0 AIM 2

In this chapter, we develop a patient problem list reference standard and evaluate how well existing clinical vocabularies cover patient problem concepts. Then, we will report on experiments leveraging problem mentions and their semantic feature annotations to generate an active patient problem list. Finally, we conclude with initial steps toward generating and evaluating new semantic and discourse features that could potentially help active patient problem list generation and visualization of problem mention descriptions from clinical text.

Aim Two: Build and evaluate a patient problem status generator for clinical narratives. In particular, we will develop classifiers that predict active patient problems based on semantic features derived from **Aim One** and define new semantic and discourse features that could potentially improve classification based on an error analysis of the clinical text.

Hypothesis: *An active patient problem list generated using rich semantic features will have higher precision than an active patient problem list generated without rich semantic features.*

4.1 GENERATING PATIENT PROBLEM LISTS FROM THE SHARE CORPUS USING SNOMED CT/SNOMED CT CORE PROBLEM LIST

Aspects of this study were published as a manuscript in the proceedings of the *2014 BioNLP Workshop* [124]. Permissions to use unspecified excerpts from this manuscript were granted from the original publisher.

4.1.1 Motivation

A patient problem list from a clinical document can be derived from individual problem mentions within the clinical document once these mentions are mapped to a standard vocabulary. In order to develop and evaluate accurate document-level inference engines for this task, a patient problem list could be generated using a standard vocabulary. Adequate coverage by standard vocabularies is important for supporting a clear representation of the patient problem concepts described in the texts and for interoperability between clinical systems within and outside the care facilities. In this pilot study, we report the reliability of domain expert generation of a patient problem list from a variety of clinical texts and evaluate the coverage of annotated patient problems against SNOMED CT and SNOMED **C**linical **O**bservation **R**ecording and **E**ncoding (CORE) Problem List.

4.1.2 Research Questions

The goals of this study are 1) *determine how reliably two domain experts can generate a patient problem list leveraging SNOMED CT from a variety of clinical texts* and 2) *assess the coverage of annotated patient problems from this corpus against the CORE Problem List.*

4.1.3 Methods

In this IRB-approved study, we obtained the manually-annotated **Sh**ared **A**nnotated **Re**source (ShARe) corpus originally generated from the Beth Israel Deaconess Medical Center [116] and stored in the **M**ultiparameter **I**ntelligent **M**onitoring in **I**ntensive **C**are, version 2.5

(MIMIC II) database [125]. This corpus consists of discharge summaries (DS), radiology (RAD), electrocardiogram (ECG), and echocardiogram (ECHO) reports from the **I**ntensive **C**are **U**nit (ICU). The ShARe corpus was selected because it 1) contains a variety of clinical text sources, 2) links to additional patient structured data that can be leveraged for further system development and evaluation, and 3) has encoded individual problem mentions with semantic feature annotations within each clinical document that can be leveraged to develop and test document-level inference engines. We elected to study ICU patients because they represent a sensitive cohort that requires up-to-date summaries of their clinical status for providing timely and effective care.

For this annotation study, two annotators - a physician and nurse - were provided independent training to annotate clinically relevant problems - *signs*, *symptoms*, *diseases*, and *disorders* - at the document-level for 20 reports. The annotators were given feedback based on errors over two iterations. For each patient problem in the remaining set, the physician was instructed to review the full text, span the a problem mention, and map the problem to a CUI from SNOMED-CT using the extensible Human Oracle Suite of Tools (eHOST) annotation tool [126]. If a CUI did not exist in the vocabulary for the problem, the physician was instructed to assign a "CUI-less" label. Finally, the physician then assigned one of five possible status labels - *Active*, *Inactive*, *Resolved*, *Proposed*, and *Other* - based on our previous study [114] to the mention representing its last status change at the conclusion of the care encounter. Patient problems were not annotated as *Negated* since patient problems are assumed negated at a document-level [109]. If the patient was healthy, the physician assigned "Healthy - no problems" to the text. To reduce the cognitive burden of annotation and create a more robust reference standard, these annotations were then provided to a nurse for review. The nurse was instructed to add missing, modify existing, or delete spurious patient problems based on the guidelines.

How reliably can two domain experts generate a patient problem list leveraging SNOMED CT from a variety of clinical texts?

We assessed how reliably annotators agreed with each other's patient problem lists using inter-annotator agreement (IAA). We evaluated IAA in two ways: 1) by problem CUI and 2) by problem CUI and status. Since the number of problems not annotated (i.e., ***true negatives (TN)***) are very large, we calculated F1-score as a surrogate for kappa [119]. F1-score is the harmonic mean of recall and precision, calculated from ***true positive***, ***false positive***, and ***false negative*** annotations, which were defined as follows:

true positive (TP) = the physician and nurse problem annotation was assigned the same CUI (and status)

false positive (FP) = the physician problem annotation (and status) did not exist among the nurse problem annotations

false negative (FN) = the nurse problem annotation (and status) did not exist among the physician problem annotations

$$\textbf{\textit{Recall}} = \frac{TP}{(TP + FN)} \tag{4.1}$$

$$\textbf{\textit{Precision}} = \frac{TP}{(TP + FP)} \tag{4.2}$$

$$\textbf{\textit{F1-score}} = 2\frac{(Recall * Precision)}{(Recall + Precision)} \tag{4.3}$$

We sampled 50% of the corpus and determined the most common errors. These errors with *examples* were programmatically adjudicated with the following **solutions**:

Spurious problems: *procedures*
solution: exclude non-problems via guidelines

Problem specificity: *CUI specificity differences*
solution: select most general CUIs

Conflicting status: *negated vs. resolved*
solution: select second reviewer's status

CUI/CUI-less: *C0031039 vs. CUI-less*
solution: select CUI since clinically useful

We split the dataset into about two-thirds training and one-third test for each report type. The remaining data analysis was performed on the training set.

What is the coverage of annotated patient problems from this corpus against the CORE Problem List?

We characterized the composition of the reference standard patient problem lists against two standard vocabularies SNOMED-CT and SNOMED-CT CORE Problem List. We evaluated the coverage of patient problems against the SNOMED CT CORE Problem List since the list was developed to support encoding clinical observations such as findings, diseases, and disorders for generating patient summaries like problem lists. We evaluated the coverage of patient problems from the corpus against the SNOMED-CT January 2012 Release which leverages the UMLS version 2011AB. We assessed recall (Eq 4.1), defining a TP as a patient problem CUI occurring in the vocabulary and a FN as a patient problem CUI not occurring in the vocabulary.

4.1.4 Results

We report the results of our annotation study on the full set and vocabulary coverage study on the training set.

How reliably can two domain experts generate a patient problem list leveraging SNOMED CT from a variety of clinical texts?

The full dataset is comprised of 298 clinical documents - 136 (45.6%) DS, 54 (18.1%) ECHO, 54 (18.1%) RAD and 54 (18.1%) ECG. Seventy-four percent (221) of the corpus was annotated by both annotators. Table 7 shows agreement overall and by report, matching problem CUI and problem CUI with status. Inter-annotator agreement for problem with status was slightly lower for all report types with the largest agreement drop for DS at 15% (11.6 points).

Report Type	**CUI**	**CUI + Status**
DS	77.1	65.5
ECHO	83.9	82.8
RAD	84.7	82.8
ECG	89.6	84.8

Table 7: Document-level IAA by report type for problem (CUI) and problem with status (CUI + status)

We report the most common errors by frequency in Table 8. By report type, the most common errors for ECHO, RAD, and ECG were CUI/CUI-less, and DS was Spurious Concepts.

Errors	DS	ECHO	RAD	ECG
SP	423 (42%)	26 (23%)	30 (35%)	8 (18%)
PS	139 (14%)	31 (27%)	8 (9%)	0 (0%)
CS	318 (32%)	9 (8%)	8 (9%)	14 (32%)
CC	110 (11%)	34 (30%)	37 (44%)	22 (50%)
Other	6 (>1%)	14 (13%)	2 (2%)	0 (0%)

Table 8: Error types by frequency - Spurious Problems (SP), Problem Specificity (PS), Conflicting status (CS), CUI/CUI-less (CC)

What is the coverage of annotated patient problems from this corpus against the CORE Problem List?

In the training set, there were 203 clinical documents - 93 DS, 37 ECHO, 38 RAD, and 35 ECG. The average number of problems were 22±10 DS, 10±4 ECHO, 6±2 RAD, and 4±1 ECG. There are 5843 total current problems in SNOMED-CT CORE Problem List. We observed a range of unique SNOMED-CT problem concept frequencies: 776 DS, 63 ECHO, 113 RAD, and 36 ECG by report type. The prevalence of covered problem concepts by CORE is 461(59%) DS, 36 (57%) ECHO, 71 (63%) RAD, and 16(44%) ECG. In Table 9, we report coverage of patient problems for each vocabulary. All reports have SNOMED CT coverage of problem mentions above 80%. After mapping problem mentions to CORE, we observed coverage drops for all report types, 24 to 36 points.

Report Type	Patient Problems	Annotated with SNOMED CT	Mapped to CORE
DS	2000	1813 (91%)	1335 (67%)
ECHO	349	300 (86%)	173 (50%)
RAD	190	156 (82%)	110 (58%)
ECG	95	77(81%)	43 (45%)

Table 9: Patient problem coverage by SNOMED-CT and SNOMED-CT CORE

4.1.5 Discussion

In this feasibility study, we evaluated how reliably two domain experts can generate a patient problem list and assessed the coverage of annotated patient problems against two standard clinical vocabularies.

How reliably can two domain experts generate a patient problem list leveraging SNOMED CT from a variety of clinical texts?

Overall, we demonstrated that problems can be reliably annotated with moderate to high agreement between domain experts (7). For DS, agreement scores were lowest and dropped most when considering the problem status in the match criteria. The most prevalent disagreement for DS was Spurious problems (8). Spurious problems included additional events (e.g., **C2939181**: *Motor vehicle accident*), procedures (e.g., **C0199470**: *Mechanical ventilation*), and modes of administration (e.g., **C0041281**: *Tube feeding of patient*) that were outside our patient problem list inclusion criteria. Some pertinent findings were also missed. These findings are not surprising given on average more problems occur in DS and the length of DS documents are much longer than other document types. Indeed, annotators are more likely to miss a problem as the number of patient problems increase. Also, status differences can be attributed to multiple status descriptions which are harder to track over a longer document. The most prevalent disagreements for all other document types were CUI/CUI-less in which identifying a CUI representative of a clinical observation proved more difficult. An example of Other disagreement was a sidedness mismatch or redundant patient problem annotation. For example, **C0344911**: *Left ventricular dilatation* vs. **C0344893**: *Right ventricular dilatation* or **C0032285**: *Pneumonia* was recorded twice.

What is the coverage of annotated patient problems from this corpus against the CORE Problem List?

We observed that DS and RAD reports have higher counts and coverage of unique patient

problem concepts. We suspect this might be because other document types like ECG reports are more likely to have laboratory observations, which may be less prevalent findings in CORE. Across document types, coverage of patient problems in the corpus by SNOMED CT were high ranging from 81% to 91% (9). However, coverage of patient problems by CORE dropped to moderate coverages ranging from 45% to 67%. This suggests that the CORE Problem List is more restrictive and may not be as useful for capturing patient problems from these document types. A similar report of moderate problem coverage with a more restrictive concept list was also reported by Meystre and Haug [109].

4.1.6 Conclusion

Based on this feasibility study, we conclude that we can generate a reliable patient problem list reference standard for the ShARe corpus and SNOMED CT provides better coverage of patient problems than the CORE Problem List. In the next section, we plan evaluate from each ShARe report type how well a patient problem list can be derived from the individual problem mentions.

Now that we have created a reference standard of patient problems for each clinical report, the next step is determining how accurately we can generate an active patient problem list leveraging manually annotated problem mentions using the semantic features described in Chapter 3, Section 3.2.

4.2 EXPERIMENTS GENERATING A PATIENT PROBLEM LIST FROM PROBLEM MENTIONS

4.2.1 Motivation

Our overarching hypothesis is that we can generate an active patient problem list (document-level) using the underlying problem mentions (instance-level) and their semantic features from the clinical text. In this pilot study, we test this hypothesis. To avoid confounding factors from automated problem mention generation, we rely on manually annotated problem mention annotations. We conclude with an error analysis and future work addressing new features.

4.2.2 Research Questions

The goals of this study are to determine 1) *how accurately can we identify patient problems from the report using manually annotated problem mentions and semantic features* and 2) *how precisely can we assert the status for the identified problems*? From our error analysis, we define new semantic and discourse features that could potentially help generate a more accurate and precise patient problem list.

4.2.3 Methods

Our hypothesis is that an active patient problem list from a clinical document can be derived from individual problem mentions and their semantic features.

For this study, we used the manually-annotated patient problem list for the ShARe corpus from Chapter 4, Section 4.1 as well as the manually-annotated problem mentions for the corpus. These annotated problem mentions represent any of the following UMLS semantic types: "Congenital Abnormality", "Acquired Abnormality", "Injury or Poisoning",

"Pathogenic Function", "Disease or Disorder", "Mental or Behavioral Dysfunction", "Cell or Molecular Dysfunction", "Experimental Model of Disease", "Anatomical Abnormality", "Neoplastic Process", or "Signs and Symptoms" [127]. These annotations were generated by two professional medical coders using consensus review. Each problem mention is encoded with semantic feature annotations based on the ShARe schema described on the ShARe/CLEF eHealth Challenge 2014 website [128]. The semantic features of the ShARe schema contain subtle differences to the Problem Mention schema from our study in Chapter 3, Section 3.2 [114]. In the following section, we describe how we mapped the ShARe schema semantic annotations to the Problem Mention schema to support generating the active patient problem list from these problem mention annotations.

How accurately can we generate the patient problem list with the ShARe problem mention annotations?

There are many potential approaches for leveraging the problem mentions in a report to generate a patient problem list. We used a simple approach of directly mapping each problem mention in the report to a patient problem in the problem list. For instance, if the report had one mention of Pneumonia, we add Pneumonia to the active problem list. We call this approach direct mapping and we would expect this approach to perform with fairly high sensitivity, but low precision. As we learned in the paper in Chapter 3, Section 3.2, semantic features of problem mentions can improve the precision of a problem list, because, for example, some problem mentions are experienced by someone other than the patient or are negated. Therefore, we implemented three filters to increase precision. The first filter assigns all problems as *Active*. The second filter removes negated problems (Filter *Negated*). The third considers non-*Active* statuses generated from all semantic features when removing problems from the problem list (Filter non-*Active*). We describe these three filters below:

1) Assign Active: This filter simply assigns all unique problem mention CUIs from the report as *Active* and adds them to the patient problem list.

2) Filter Negated: This filter assigns each problem mention one of two statuses *Negated* or *Active*. When the problem mention has the semantic feature Existence: no, the mention is assigned as *Negated*. The filter then removes these *Negated* problem mentions and assigns all the remaining unique problem mention CUIs as *Active*.

3) Filter Non-Active: This filter assigns each problem mention one of six status labels - *Active*, *Inactive*, *Resolved*, *Negated*, *Proposed*, *Other* - based on two processing steps, semantic feature transformation and status assertion rules, learned from our study in Chapter 3, Section 3.2 [114]:

3a) Semantic Feature Transformation: We mapped the ShARe schema semantic annotations into the Problem Mention schema semantic annotations by 1) *condensing classes and feature values*, 2) *expanding feature values*, 3) *eliminating classes*, and 4) *adding new classes*. A side-by-side comparison of each semantic feature can be found in Figure 4. We provide one example for each transformation approach:

3a.1) Condense classes and feature values For the ShARe Subject Class, we condensed the values from the ShARe schema to the Problem Mention schema values. For example, non-patient labels like Subject Class: family_member, donor_family_member, donor_other, null, or other were condensed to Experiencer: other.

3a.2) Expand feature values For the Uncertainty Indicator Class, we applied the pyConText lexicon and certainty cues from previous studies [113, 114] to expand the feature values of Uncertainty Indicator: yes or no to Certainty: high, moderate, low, or unmarked. Specifically, if the template contained Uncertainty Indicator: yes, we would apply a look up using the Uncertainty Indicator cue span against the pyConText lexicon. The certainty value associated with the cue was then assigned as the Certainty value. If no value was observed, a default was assigned as Certainty: high based assumption that most problem mentions are expressed with high certainty [113].

3a.3) Eliminating classes We eliminated the Intermittency Class from the Problem Mention schema since this feature was not useful for asserting a problem mention status.

3a.4) Adding new classes We added the Aspectual Phase Class by applying lexicon from the Problem Mention paper to assign each problem mention's feature as Aspectual Phase: initiation, continuation, culmination, or unmarked.

ShARe Semantic Features: Potential Values	**Definitions from ShARe guidelines: A span of text that ..**	**Problem Mention Semantic Features: Potential Values**
Negation Indicator: no, yes	indicates a disease/disorder was negated.	Existence: yes, no
Subject Class: patient, family_member, donor_family_member, donor_other, null, other	indicates who experienced a disease/disorder.	Experiencer: patient, other
Uncertainty Indicator: no, yes	indicates a measure of doubt into a statement about a disease/disorder.	Certainty: unmarked, low, moderate, high
Course Class: unmarked, changed, increased, decreased, improved, worsened, resolved	indicates progress or decline of a disease/disorder.	Change: unmarked, changed, increased, decreased, improved, worsened, recurrence Aspectual Phase: unmarked, initiation, continuation, culmination
Severity Class: unmarked, slight, moderate, severe	indicates how severe a disease/disorder is	--
Conditional Class: false, true	indicates conditional existence of disease/disorders under certain circumstances.	Generalized/Conditional: no, yes
Generic Class: false, true	indicates a generic mention of a disease/disorder	
Body Location: NULL, CUI, CUI-less	represents an anatomical location of these UMLS semantic types: Anatomical structure; Body location or region; Body part, organ or organ component; Body space or junction; Body substance; Body system; Cell; Cell component; Embryonic structure; Fully formed anatomical structure; Tissue	--
DocTime Class: before, after, overlap, before-overlap, unknown	indicates temporal relation between a disease/disorder and document authoring time	Relation to Current Visit: before, meets_overlaps, after
Temporal Expression: date, duration, time, none	represents any TIMEX (TimeML) temporal expression related to the disease/disorder: start date, duration, or end date of the disease/disorder.	Start Before the Current Visit: greater than 2 weeks, less than 2 weeks, notClear, n/a

Figure 4: Side-by-side comparison of the ShARe and Problem Mention semantic features

3b) Status Label Assertion: For each problem mention, we asserted one of six statuses - *Active*, *Inactive*, *Resolved*, *Negated*, *Proposed*, *Other* - based on the semantic feature transformation values using the following rules inferred from Aim 1:

1. **If** Existence: no, assign *Negated*.
2. **Else if** Certainty: moderate or low, assign *Proposed*.
3. **Else if** Change: NOT(unmarked) AND Relation to Current Visit: after, assign *Active*.
4. **Else if** Experiencer: other OR (GeneralizedConditional: yes OR Historical: N/A), assign *Other*.
5. **Else if** Relation To Current Visit: before OR Historical: greater than 2 weeks, assign *Inactive*.
6. **Else if** Existence: yes AND (Aspectual Phase: culmination OR Change: improved), assign *Resolved*.
7. **Else if** Relation to Current Visit: After OR Historical: N/A, assign *Other*.
8. **Else**, assign *Active*.

These rules each serve a specialized purpose. Rule 1 filters denied problem mentions such as "Denies cough". Rule 2 asserts proposed problems that may or may not have occurred with a significant amount of doubt. For example, Rule 3 maintains an active status for problems described as presuppositions like "if worsening pain return to ER" in which pain is still an active problem upon discharge. Rule 4 classifies problems experienced by someone other than the patient or problems that occur in a non-particular or conditional way. Rule 5 assigns problems that occurred in the past. Rule 6 assigns problems that had occurred at one time, but have resolved on their own or were successfully treated. Rule 7 assigns problems that might occur in the future. Otherwise, the problem mention is assigned as an *Active* problem.

Finally, for each unique problem mention CUI we asserted a patient problem status. For ECG, ECHO, and RAD, we assigned the problem mention CUI based on the last mention status in the list for all matched CUIs. For DS, we implemented a series of 11 report-specific rules that assert a status based on the position of 1 or more CUIs and their statues in a

problem mention list. We then filtered out all non-*Active* problem mentions.

For each filter (Assign Active, Filter Negated, and Filter Non-Active), we assessed how well we could identify active patient problems.

This study is comprised of two tasks leveraging these annotations to 1) identify each patient problem from the problem mentions in the report and 2) generate the active patient problem list using three filters. To evaluate how well problem mention annotations identify active patient problems, we applied F1 as a measure of IAA between the unique problem mention CUIs and patient problem CUIs. We applied an exact CUI match criteria. We defined a ***true positive***, ***false positive***, and ***false negative*** as follows:

true positive (TP) = the problem mention CUI and patient problem CUI was assigned the same CUI (and *Active* status)

false positive (FP) = the problem mention CUI (and *Active* status asserted by the rule) did not exist among the patient problem CUI (and *Active* status)

false negative (FN) = the patient problem CUI (and *Active* status) did not exist among the problem mention CUI (and *Active* status asserted by the rule)

Recall, ***precision***, and ***F1*** were calculated using Eq 4.1 - 4.3. We conducted an error analysis to quantify several types of errors that occurred and determine what additional information may be needed to perform the task accurately.

4.2.4 Results

In this section, we report how accurately we can generate the patient problem list from the ShARe problem mention annotations using direct mapping. We report how accurately we can generate an **active patient problem list** using three rules. We conclude with an analysis of common errors.

ShARe Corpus Characterization

For this study, we used manually-annotated problem mentions and their semantic features from the ShARe corpus. We maintained the training and test split described in Chapter 4, Section 4.1. By report type, the ShARe dataset contains 6538 DS, 979 ECHO, 136 RAD, and 576 ECG problem mentions and their semantic features. These mentions were used to develop semantic feature transformation and status label assertion rules to assert a problem mention's status in the clinical text. We performed an evaluation on unseen cases from the test split of the ShARe dataset which contains 2560 DS, 450 ECHO, 255 RAD, and 60 ECG problem mentions and their semantic features. From the SNOMED CT CUIs and these problem mention statuses, we generated patient problem lists by predicting a patient problem and whether its status was active.

How accurately can we generate the patient problem list with the ShARe problem mention annotations?

In Table 10, both training and test data have similar distributions of patient problem concepts for each report type. We observed most patient problem concepts occur in DS followed by ECHO and RAD. For test, using exact match criteria for evaluating whether a patient problem concept annotation could be generated from an annotated problem mention concept annotation, overall scores demonstrated moderate performance ranging from F1 of 53.7 to 72.1, recall of 60.0 to 68.8, and precision of 45.3 to 75.9. For each report type, recall values represent the highest accuracy achievable given this exact CUI matching criteria. Specifically, we could not identify 33% DS, 33% ECHO, 40% RAD, and 31% ECG of patient problems using this approach.

	Train				Test			
Patient Problems	**Ct (%)**	**F1**	**Rec**	**Prec**	**Ct (%)**	**F1**	**Rec**	**Prec**
DS - CUIs	1813 (77.3%)				774 (75.0%)			
CUIs		51.5	66.0	42.2		54.1	67.1	45.3
ECHO - CUIs	300 (12.8%)				140 (13.6%)			
CUIs		61.5	74.4	52.4		57.1	66.7	50.0
RAD - CUIs	156 (6.6%)				85 (8.2%)			
CUIs		49.5	60.3	42.0		53.7	60.0	48.6
ECG - CUIs	77 (3.3%)				33 (3.2%)			
CUIs		66.7	59.7	75.4		72.1	68.8	75.9
Total CUIs	2,346 (100.0%)				1,032 (100.0%)			

Table 10: Overall - unique problem mention CUI to patient problem CUI match

How precisely can we identify active patient problems using problem mentions with with three patient problem status rules?

The prevalence of *Active* patient problems by report are DS: 47.9% (train); 48.4% (test), ECHO: 93.0% (train); 93.6% (test), RAD: 87.9% (train), 68.2% (test), and ECG: 79.2% (train); 72.7% (test). For all report types, train and test show similar performance for F1 with DS and RAD performing the lowest (Table 11). The best recall values were achieved for all reports assigning all problem mention CUIs as *Active*; however, this rule achieved the lowest precision values. On the test set, precision increased with the Filter Negated rule. Peak precision values were achieved by the following rules: Filter Non-Active for DS, RAD, and ECG; Filter Negated for ECHO.

	Train				Test			
Active Patient Problems	**Ct**	**F1**	**Rec**	**Prec**	**Ct**	**F1**	**Rec**	**Prec**
DS - Active CUIs	869				375			
Assign Active		28.3	60.0	18.5		31.5	63.5	21.0
Filter Negated		34.0	59.6	23.8		35.6	62.4	24.9
Filter Non-Active		36.4	48.1	29.3		38.4	50.4	31.0
ECHO - Active CUIs	279				131			
Assign Active		59.9	75.3	49.8		55.2	66.4	47.3
Filter Negated		68.8	73.8	64.4		61.5	64.1	59.2
Filter Non-Active		64.3	64.2	64.4		54.6	51.9	57.6
RAD - Active CUIs	137				58			
Assign Active		46.0	60.6	37.1		44.2	62.1	34.3
Filter Negated		52.4	59.1	47.1		50.7	62.1	42.9
Filter Non-Active		52.2	49.6	55.7		56.6	55.2	58.2
ECG - Active CUIs	61				24			
Assign Active		67.2	67.2	67.2		67.9	75.0	62.1
Filter Negated		67.8	67.2	68.3		69.2	75.0	64.3
Filter Non-Active		74.1	65.6	85.1		75.0	75.0	75.0

Table 11: Performance generating an active patient problem list for each rule by report type

We observed that many errors are caused by false positives in which the problem mention CUI does not exist among the patient problem list CUIs (Table 12). When there is a one-to-one match between a single problem mention and single patient problem, the status labels are often mismatched e.g, problem mention: *Active* and patient problem: *Resolved*. In few cases, the status label appears in error e.g., *InactiveProposed* is concatenated.

	Train			Test		
ReportType	**NPP**	**OOM**	**SLE**	**NPP**	**OOM**	**SLE**
DS	1630	320	9	620	505	8
ECHO	201	67	2	92	38	6
RAD	130	11	0	54	26	0
ECG	15	3	0	7	2	1

Table 12: Some observed error types: NPP=No patient problem, OOM=One-to-one mismatch, SLE=Status label error

4.2.5 Discussion

In this study, we evaluated by report type, how accurately we can generate the patient problem list from the ShARe problem mention annotations based only on CUIs. We reported how accurately we can we identify the patient problem and how precisely we can assert an active status using problem mentions with the three rules.

How accurately can we generate the patient problem list with the ShARe problem mention annotations?

Ideally, we would like to recover all patient problems using explicit problem mentions. However, for this dataset and exact CUI matching, we determined that at best we can achieve moderate recalls. We hypothesize that we could reach higher recall levels using the UMLS hierarchy "is-a" relationships to assert more general or specific CUIs. For example, when to assert **C0032290**: *Aspiration pneumonia* rather than the more general **C0032285**: *Pneumonia*. However, these problem mention descriptions can be dispersed throughout the clinical text; therefore, merging and reconciling conflicting descriptions can be challenging and may require anaphoric relations to identify coreferring expressions representing the problem concept and rhetorical relations to either merge new or correct existing information. For example, in "Pt has pneumonia. It appears to be caused by aspirating food." an anaphoric relation is needed to link coreferring expressions pneumonia and It as well as a rhetorical relation like Elaboration to denote the type of pneumonia is aspiration from the description. In a future study, we will run cTAKES coreference resolution module, annotate rhetorical relations, and quantify how frequently this phenomenon occurs and what features can be used to correctly assert the more general or specific patient problem. In our error analysis, we observed that most problem mentions do not enter the patient problem list. This can be be partially explained by the ShARe annotation scope in which annotations do not include quantitative expressions of problems such as "Temperature: 101.3" which represents **C0015967**: *Fever*. Some problem mentions are missing within the most prevalent patient problems in each report type. For instance, we want to address missing problem mention

annotations for **C0015967**: *Fever* in RAD and **C3164445:** *Abnormality of aortic valve* in ECHO, and **C0018800:** *Cardiomegaly* in ECG which occur within the top 5 most prevalent patient problems to obtain significant improvements in recall. On the other hand, perhaps some problems or findings shouldn't be added because they lack clinical relevance to the current visit or are represented by a broader disease concept. In future studies, we will have our domain experts qualify and quantify the frequency of all types of errors at a more detailed level.

How precisely can we identify active patient problems using problem mentions with three patient problem status rules?

For this study, we aimed to generate an active patient problem list using semantic features annotations. We observed that *Active* is the majority class for all report types, but most prevalent for ECHO, RAD, and ECG producing higher recall and precision values than DS. For DS, filtering non-*Active* patient problems was most challenging. We suspect this is because 1) prevalence of non-*Active* patient problems is much higher than other report types, 2) status changes are described more frequently throughout DS than other clinical reports, and 3) DS report structure is not always chronological, hence the last problem mention description is not likely the final status change. For instance, many DS end with the Discharge Diagnosis which enumerates the patient diagnoses for the visit, but not likely report that they were successfully treated or had controlled all the underlying symptoms. This information is usually found in the Hospital Course or potentially in the Condition on Discharge sections. Integrating section knowledge with Filter Status rules may improve performance. Although precisions ranged from poor (DS) to moderate (ECHO, RAD) to high (ECG), precision gains were made adding negation and status filtering over a majority class *Active* baseline demonstrating that we can generate a more precise active patient problem list leveraging semantic features. From our error analysis, we observed that the second most prevalent error occurs when there is a one problem mention to one patient problem CUI match, but the status labels do not match. For example, the problem mention **C0007282:** *Carotid artery stenosis - Active* and patient problem **C0007282:** *Carotid artery stenosis -*

Resolved are in mismatch. In this case, the carotid stenosis was successfully treated with a right ICA stent without complication. Identifying whether a CUI for a problem-specific eventuality such as a treatment or procedure with a causal relation describing an improved status may prove useful for inferring the patient problem status. Other sources of error could be in the semantic feature transformation and status label assertion rules. For instance, we may not have one or more cues necessary to assert one or more values of Certainty or Aspect Phase. In a follow up study, we will manually annotate the problem mention study values and statuses for each problem mention. We are actively developing the pyConText algorithm to encode these semantic feature values and assert the status label. Using these annotations and pyConText's output, we will conduct a formative evaluation on the training set to quantify how frequently these processing steps are the source of performance degrade throughout this pipeline. In few cases, we observed concatenated status labels. We believe these were introduced during the annotation process using the annotation tool.

4.2.6 Conclusion

From this study, we conclude that the ShARe annotations and our rules provide moderate performance generating an accurate patient problem list. More semantic and new discourse features may improve over these baseline approaches.

5.0 FINAL CONCLUSIONS AND FUTURE WORK

From this thesis work, we learned many new things and have discovered new directions for improving patient problem list generation. We learned from our studies that generating an accurate and precise patient problem list can require identifying, normalizing, and integrating several pieces of semantic information. In particular, we learned 1) distinguishing historical problems from recent problems requires both document and instance-level features, 2) distinguishing active problem mentions from other problem mention statuses improves using richer semantic features, 3) generating a reliable patient problem list remains challenging and patient problem concept coverage may be better using SNOMED CT, and 4) generating an active patient problem list using rich semantic features will have higher precision than an active patient problem list generated without rich semantic features.

5.1 DISTINGUISHING HISTORICAL PROBLEMS FROM RECENT PROBLEMS REQUIRES BOTH DOCUMENT AND INSTANCE-LEVEL FEATURES.

In our first study, we learned that in order to accurately assign a problem as historical or inactive, both discourse (sections) and semantic (temporal expressions) features beyond simple lexical terms (trigger terms) must be identified and applied in a prediction model. In terms of generating an active problem list, we will experiment with filtering out historical sections and sentences containing historical temporal markers from document processing to reduce processing time and false positives. However, historical problems are not the only source of false positives when generating an active patient problem list.

5.2 DISTINGUISHING ACTIVE PROBLEM MENTIONS FROM OTHER PROBLEM MENTION STATUSES IMPROVES USING RICHER SEMANTIC FEATURES.

In our second study, we observed that about 8% of problem mentions in reports are inactive. This 8% contributes to the 36% of problem mentions in reports that are non-active problems. In order to correctly predict other non-active problem mentions including proposed, negated, resolved, and other problem mentions, richer semantic features must be annotated. Understanding how well these semantic features can be annotated and how important they are for predicting a problem mention's status is a first step toward validating the need for these features in our framework. We investigated how well annotators could annotate more fine-grained attributes like Certainty and Historical. We observed that Certainty and Historical are more difficult to reliably annotate than other semantic features, such as Experiencer and Existence. In future work, we will conduct an annotation study in which annotators mark the "rationale" or evidence for each attribute value and investigate potential factors that contribute to annotator disagreements [123]. As a first step toward evaluating the usefulness of each semantic feature, we performed an ablation study using a multi-class machine learner. This preliminary work suggests relationships between problem attributes and predicting a problem mentions status. In future work, we will develop individual classifiers for each status and evaluate the predictive ability of each individual attribute value. We will also evaluate whether some problem mention concepts are correlated with a particular problem mention status or attribute value. For example, whether a problem concept is more likely to be assigned a proposed status or non-default Certainty value [129].

5.3 GENERATING A RELIABLE PATIENT PROBLEM LIST REMAINS CHALLENGING AND PATIENT PROBLEM CONCEPT COVERAGE MAY BE BETTER USING SNOMED CT

In the third study, we learned that generating a consistent and reliable patient problem list between two clinicians can be more difficult for some report types (discharge summaries) over others (echocardiograms, electrocardiograms, and radiology reports) due to issues in problem representation, problem granularity, and missing patient problems. In future work, we will study how annotators decide which CUI to assign a patient problem, which granularity is sufficient for a patient problem, and which patient problems should make the patient problem list using semi-automated approaches combined with interactive search and visualization. We also observed that although coverage of patient problems for this corpus appears high for all report types using SNOMED CT, patient problems are not as well represented in SNOMED CT CORE Problem List. However, in practice, there is a tradeoff between being expressive enough, but not being too large to search for a suitable concept. This pilot study is a first step in evaluating the effect of reducing the size of a clinical vocabulary (SNOMED CT) and the potential implications of whether the reduced vocabulary (SNOMED CT CORE Problem List) could sufficiently support a practical, clinical problem in terms of concept mapping. However, other tradeoffs should be evaluated for implementing a problem list generation or recommendation system such as algorithmic search times identifying a sufficient concept or clinician search times identifying a missed problem or modification times correcting incorrect problem suggestions.

5.4 GENERATING AN ACTIVE PATIENT PROBLEM LIST USING RICH SEMANTIC FEATURES WILL HAVE HIGHER PRECISION THAN AN ACTIVE PATIENT PROBLEM LIST GENERATED WITHOUT RICH SEMANTIC FEATURES

In the final study, we observed that patient problem lists in specialty reports (echocardiograms, electrocardiograms, and radiology reports) can be generated more precisely than patient problem lists in general summary reports (discharge summaries). When implementing a problem list recommendation system, we can leverage this performance difference in deciding how we provide information to clinicians in a useful way and scope who to target. One potential design is implementing a patient problem recommendation system for generalist from specialist reports. This implementation could have higher impact bridging the information gap and adding findings not mentioned in the discharge summaries which might already contain an overview of most patient problems from the entire medical record, an area not evaluated in this current work. In terms of natural language processing and specialty reports, problem mentions (instance-level) were shown to have better direct mapping to patient problems (document-level) than for discharge summaries. Potential areas of future work based on these annotations include visualizing a summary of these problem mention changes for the resulting active patient problems. User studies could help illuminate the best output for the patient recommendation system. For instance, just a problem concept? problem with status? or problem with status and attributes? etc. Other areas of future work include investigating how new discourse features such (SOAP) and document structure (order and diagnostic lists) can improve the accuracy of the patient problem list as well this visualization of the problem status throughout time and the narrative. For a complete list of proposed discourse features for the Clinical Linguistic Framework, see the Appendix A. For experiments building an automated SOAP classifier for clinical text, see Appendix B.

6.0 CONTRIBUTIONS

This thesis is innovative and provides several contributions including the development and examination of a new framework (Clinical Linguistic Framework) and resources (ShARe dataset).

6.1 NEW FRAMEWORK

We proposed a new clinical linguistic framework that encodes semantic features according to an existing linguistic framework by Webber et al. [10]. We included new semantic features not leveraged by other problem list generators including fine-grained certainty levels, temporality, and modality features not addressed by the preceding systems. We have shown progress evaluating the informativeness and integration of semantic features to assert an accurate patient problem list using this framework and SNOMED CT/SNOMED CT CORE Problem List. From our error analysis, we gained a deeper understanding of which discourse features may be useful for problem list generation, providing insight for strategic computing. This knowledge gives guidance to developers seeking to encode informative features for problem list generation and has provided new directions for continued research in this area.

6.2 NEW RESOURCES

We developed a new resource for problem list generation adding a document-layer of patient problem annotations to the openly, available ShARe corpus [116, 128]. We have defined a problem, baseline approach, and common resource that can be leveraged and extended by the greater clinical NLP community. It is our hope that further progress can be made for this clinical problem and better informatics solutions can be engineered based on our discoveries to improve both patient care and outcomes.

APPENDIX A

EXTENDED CLINICAL LINGUISTIC FRAMEWORK

We hypothesize new semantic and discourse features could improve accurate patient problem list generation. These proposed discourse features and the implemented semantic Clinical Linguistic Framework to generate active patient problem lists can be found in Table 13.

Annotation Type	**Webber Elements**	**Annotation Features**
Semantics	*eventualities* and their semantic features	Problems, Treatments, Tests Experiencer, Existence, Aspect, Certainty, Change, Generalized/Conditional, Relation to Current Visit
Discourse	*discourse relations* *functions* *topics*	Causal, Temporal, Anaphoric Rhetorical Sections, SOAP classes
Status		*Active*, *Inactive*, *Resolved*, *Proposed*, *Negated*, *Other*

Table 13: Clinical Linguistic Framework with Semantic and Discourse Features

There are several opportunities we plan to investigate for constructing an automated patient problem list generator using the ShARe corpus. Through our error analysis, we observed that many patient problems did not have a problem mention annotation. We plan to increase the number of problem mention annotations by generating regular expressions from the grounding patient problems from the training data using the pyConText algorithm. For instance, the quantitative expression "Temperature: 101.3" which represents **C0015967**: *Fever* can be encoded with a regular expression as a problem mention. For this dataset we can formatively evaluate how well the algorithm detects problem mention boundaries and normalizes the CUI value based on the existing 2013 ShARe/CLEF eHealth Challenge Task 1 annotations for which the best boundary detection system for problem mentions achieved F1 of 75 (80 precision and 71 recall) and normalization accuracy of 59. We can also use the 2014 ShARe/CLEF eHealth Challenge Task 2 annotations to evaluate how well we can predict the semantic features from the ShARe schema. We can build on this representation by adding the Problem Mention schema as an additional semantic meta-layer for each problem mention. We will continue to evaluate the effect of our proposed rules and learn new rules that incorporate knowledge about the document format including functions, topics, and discourse relations from our proposed Clinical Linguistic Framework as well as new semantic annotations representing treatments and tests. In terms of correctly asserting active patient problem statuses, we have begun machine learning experiments based on the training data to learn whether our rules for status assertion should include knowledge about the type of problem. For instance, should some findings be left out of the patient problem list (e.g., an allergy) or more likely to acquire one label over another (more often *Proposed* rather *Active*).

Finally, we will experiment with creating visualizations with the annotations and conduct a usability study to determine how best to organize and display the patient problem list leveraging semantic and discourse features. For instance, is the list best generated using the original SOAP structure? or using a hierarchical display showing each problem finding's relationship to the greater disease? or a timeline showing the evolution of the problem mention with related eventualities like treatments and tests over time?

APPENDIX B

SOAP CLASSIFIER

Through our literature review and study analyses, we defined several potential new semantic and discourse features that might improve the accuracy of an active problem list. As a first step to structuring these problem mentions and other potential problem-specific eventualities such as related treatments and tests, we developed a discourse topic classifier, a SOAP classifier, based on Dr. Lawrence Weed's original patient problem list framework.

B.1 BUILDING AN AUTOMATED SOAP CLASSIFIER FOR EMERGENCY DEPARTMENT REPORTS

This study was published as an original research article in the *2010 American Medical Informatics Association Annual Symposium Proceedings* [130] and *Journal of Biomedical Informatics* [101]. Permissions to use unspecified excerpts from this manuscript were granted from the original publisher [131].

B.1.1 Motivation

We realized problem mentions are described using different contexts to convey the care provider's medical decision making. Multiple descriptions of a patient's problem progression

over time can be visualized using the original SOAP structure described by Weed [11]. We chose to capture the intention of the writer from the narrative using this discourse feature to provide situational context for each potential problem mention identified in the clinical text. In future work, we would like to investigate how the SOAP framework can be used to organize the symptoms (S: *subjective*), signs (O: *objective*), reasonings (A: *assessment*) and treatments (P: *plans*) mentioned in the report relative to coded problems (numbered problems with this supporting clinical information) into a problem-oriented SOAP note from the clinical free-text.

B.1.2 Research Questions

We designed a study to determine 1)*whether the SOAP framework could be annotated with high agreement by annotators*, and 2) *determine the types of features that support successful automated SOAP classification*?

B.1.3 Methods

We began our study with a review of the medical literature to define the intention of a writer composing a clinical text. We selected the Lawrence Weed's SOAP framework from the problem-oriented medical record (POMR), a general framework used by care providers in the medical field to document their medical decision making. In the POMR record, active problems are numbered. For each problem, the clinician lists four kinds of information (S) subjective, (O) objective, (A) assessment and (P) plan. The clinician starts *subjective* information, documenting symptoms to understand the patients clinical state (S). Next, the clinician records *objective* information, signs, quantifiable data and scientific evidence experienced by the patient (O). The clinician records *assessment* information, unifying and critically evaluating subjective and objective information to formulate a differential diagnoses (A). Finally, the clinician reports *plan* information, prescribing treatments for controlling the underlying disease (P). We had two objectives for this study 1) determine how well the SOAP

framework applies to emergency department reports, and 2) determine the types of features that support successful automated SOAP classification.

Can a clinical discourse framework be annotated with high agreement by annotators?

We constructed SOAP class definitions through both literature review and a pilot annotation study. For our purposes, we defined *subjective* as background or historical information relevant to understanding the patients current or future clinical state and *objective* as observable, measurable and quantifiable information. We did not instruct annotators to use the source of the information, patient or care provider, as a major source for their determination. We defined *assessment* as expressions of a diagnosis, impression or differential diagnosis and *plans* as any reporting of planned or implemented treatment actions, education or follow-up procedures. We conducted an initial pilot study on 10 emergency department reports (n=734 sentences) not used in this study. From the pilot study, we clarified our definitions based on annotator feedback and agreement.

Next, we conducted an annotation study to address our first objective. We randomly selected 50 emergency department (ED) reports from the University of Pittsburgh Medical Center (UPMC) aggregated from visits occurring from December 1990 through September 2003. We had trained two annotators, a registered health information administrator and registered nurse, to individually annotate the sentences from the first 25 reports from the dataset. The annotators were provided a 13-page instruction guide and annotated each sentence in the report with all SOAP classes that applied. Agreement was evaluated after the first five reports and again after the 25th report. At that point, we found agreement was consistently sufficient (kappa coefficient above 0.70 for all classes) for only the second annotator to annotate the remaining 25 reports. Disagreements in the first 25 reports were settled by randomly selecting one of the annotators answers for all classes. The annotations were collected with a web-based annotation tool built using the Django infrastructure written in Python. For each SOAP class, every sentence in the dataset was labeled as a positive or

negative instance.

We evaluated their agreement with one another using several types of agreement metrics - Observed Agreement, Positive Specific, Negative Specific, Chance Corrected, and Prevalence Corrected. Figure 5 shows formulas for computing these agreement metrics. We determined that our annotators had agreement over 0.70 for all classes; therefore, one annotator annotate the remaining sentences from 25 reports. We used the class annotations produced as our reference standard.

Agreement Metrics	Formulas
Observed	A_o=(TP +TN)/(TP+TN+FP+FN)
Positive Specific	Pos=2(TP)/(2*TP+FP+FN)
Negative Specific	Neg=2(TN)/(2*TN+FP+FN)
Cohen's kappa (chance-corrected)	$k(\text{chance}) = \frac{A_o - A_e}{1 - A_e}$
Kappa (prevalence-corrected)	$k(\text{prevalence}) = 2(A_o) - 1$

Figure 5: Definitions of Agreement Measures

Can the clinical discourse framework be automatically annotated from clinical text?

We investigated the following baselines, supervised algorithms and feature groups for creating and evaluating automated SOAP classifiers.

Baselines: To determine the complexity of the task, we initially developed simple baseline classifiers. The first baseline assigned the target class for every sentence in the reference standard (i.e., the class *objective* for the *objective* classifier, etc.). The second baseline assigned the majority class to every sentence. The third baseline used a conditional probability distribution to identify the most likely SOAP class for each section in the report. For each sentence, this classifier assigned the most likely SOAP class with the highest conditional probability, e.g., if the "disposition" section type was most likely to be assigned the *plan* class, all sentences in the "disposition" section were classified as *plan*. Sections were tagged

using SecTag, an automated section tagger [99], and conditional probabilities were calculated using the Natural Language Tool Kit (NLTK) and the pilot dataset. Specifically, for each sentence, this classifier assigned the SOAP class with the highest posterior probability, e.g., if the "disposition" section type was more likely to be assigned as a *plan* class in the pilot set, all sentences in the "disposition" section in the test set were classified as *plan*. Table 14 contains examples of section header types correlated to SOAP classes.

SOAP class	**SecTag section header types**
Subjective	allergies_and_adverse_reactions, back_review, chief_complaint, family_and_social_history, family_medical_history, history_present_illness, hospital_course, medications, past_medical_history, past_personal_and_social_history, review_of_systems, risk_factors, substance_use, tobacco_use
Objective	abdominal_exam, chest_exam, counts, derm_exam, extremity_exam, general_exam, genitourinary_exam, head_neck_exam, heart_rate, heent_course, hematology_exam, laboratory_and_radiology_data, laboratory_data, pelvis_exam
Assessment	admission_diagnosis, diagnoses, discharge_condition, discharge_diagnosis
Plan	discharge_medications, disposition_plan, ear_nose_throat_exam, follow_up

Table 14: Example sections probabilistically associated with SOAP classes

We created SOAP classifiers using a variety of feature groups and support vector machines, respectively. We chose a supervised learning approach, a linear-kernel support vector machine. We included a variety of features, including many designed to collapse similar features into a smaller set of values to reduce the feature space.

Lexical: Lexical features comprise tokens found in the report. We used the natural language toolkit (NLTK) to identify all **unigrams** and **bigrams**. In "The patient has a history of stroke" the full set of lexical features include <s>, The, patient, has, a, history, of, stroke, .,</s>, <s> The, The patient, patient has, has a, a history, history of, of stroke, stroke ., .</s>, where <s> and </s> indicate the start and end of the sentence, respectively.

Syntactic: Syntactic features consist of Penn Treebank tags [132] encoded by the Stanford part of speech tagger (09/28/2009) [133] and corrected for common tagging errors that occur in clinical narratives using seven rules that were learned by applying Brills transformation based tagger to a previous set of clinical reports [134]. For example, one of the rules states that if a token with the tag "CD" is followed by the token ".", change the tag "CD" to the tag "LS", indicating that the number is part of a numbered list. We identified the **part of speech** and **word/part of speech pair** (word/POS) for each lexical feature as a crude attempt at word sense disambiguation. For instance, discharge (NN) often indicates a clinical finding, whereas "discharge" (VB) indicates being released from the hospital.

For every verb phrase in the sentence, we encoded the **tense** of the first verb in each verb phrase as *past*, *present* or *future*. For example, we classified "She had developed a severe cough" as *past*, and "she will return if she develops a severe cough" as *future* and *present*, respectively.

Semantic: We used the Unified Medical Language System (UMLS) Metathesaurus (version.2.4.C release) courtesy of the National Library of Medicine to tag the **semantic type** and **cui** (concept unique identifier) for each token in the sentence found in the UMLS. For example, in the phrase "Lungs are clear", "Lungs" maps to the semantic type *Body Part, Organ, or Organ Component* and CUI: **C0024109**, and "clear" maps to semantic type *Idea or Concept* and CUI: **C1550016**. We also captured the **position of each semantic type** in the sentence as *Beginning*, *Middle*, or *End*, based on character counts within the sentence. We applied a feature reduction strategy [135] to encode whether a **digit type** was being

used as a *date*, *list*, *anatomical location*, *medication*, *result*, or *age*. We used simple regular expressions and heuristics to assign the digit type. For example, "1. aspirin" - *list*:, "cranial nerves II through XII are grossly intact" - *anatomic location*, "20 mg" - *medication*, and "Temp 98.6" - *result*. The emergency department reports were de-identified according to the HIPAA criteria by DE-ID software (version 5.10). We used the **de-id tags** as features representing patient sensitive or service facility information: *name*, *date*, *device-id*, or *institution*.

We identified **state of mind terms** as shallow predictors of mental postulation suggestive of medical decision making and **hedge terms** from [136] suggestive of uncertainty and speculation. For example, in "I think he has viral meningitis," "think" was encoded as a **state of mind term**. Similarly, in "She likely has the flu," "likely" was encoded as a **hedge term**. Finally, we included **trigger terms** applied by the ConText algorithm, which indicate that a problem mention in the sentence is *historical* (e.g., "history of"), *conditional* (e.g., "if"), *absent* (e.g., "denies"), or *experienced by someone other than the patient* (e.g., "family history").

Contextual: We defined the contextual information about the sentence with respect to the structure of the clinical narrative. We used the SecTag tagger to identify the **section type** for each sentence found in the report. For example, "Cardiovascular: The patient has chest pain" maps to a section type *cardiovascular_review*. SecTag defines 16,036 possible section tags.

Because emergency department report structure may follow chronological ordering similar to ideal progress notes (i.e., Subjective, Objective, Assessment, Plan), we included a feature encoding the **position of the sentence** in the report in quartiles. We also included **length of the sentence** in number of tokens. For instance, "Chief Complaint: headache" is in the *1st quartile* of the report and has *length of 6* including sentence start and end markers.

Heuristic: We developed an unsupervised method for mining **high-precision terms** from a corpus of de-identified emergency department reports (200,000 sentences from 3,577

reports) from the University of Pittsburgh NLP Repository [130]. We used an initial seed set of 5-6 terms to predict the SOAP class for each sentence by assuming all sentences that contained the seed terms belong to that class. From these tagged sentences, we used a simple conditional probability to learn new terms as good predictors for a SOAP class. For example, if "alcohol" is a *subjective* seed and tags the sentence "patient drinks one glass of alcohol a day", the conditional probability may learn "drinks" as a correlated term for *subjective*. Additionally, we conducted an error analysis on our pilot data to identify phrases we thought would be indicative of each class. For every sentence in the corpus, we created a vector of features with binary values to indicate whether or not that feature was present in the sentence. Features representing words or classes from the text (e.g., unigrams or UMLS semantic type) were generated from the pilot set so a feature not present in the pilot set was not applied to this dataset.

All features were automatically generated with programs we implemented in Python version 2.5. For each sentence, we encoded the feature value as "1" if the feature was found in the sentence and "0" otherwise. Each set of similar features was mapped to one of five feature groups: lexical, syntactic, semantic, contextual or heuristic. For each SOAP class, we trained and tested two SVMs. The first was trained on all features. The second was trained on only those features that were included by a Chi-square feature selection with a significance threshold of $p<0.05$. We used this subset to train the support vector machine to classify sentences for each SOAP class using 10-fold cross validation. We compared the output of all classifiers against the manual reference standard to address four questions: *(1) How well does a classifier perform when trained on all feature groups?* (2) *How well does a classifier perform when trained on a subset of features selected through a feature selection algorithm?* (3) *How much does each feature group contribute to performance on the classification task?* (4) *Which feature group is most informative to the classification task as a whole?* To answer question (1) we trained an inclusive classifier using all feature groups, (2) we trained a classifier using a subset of features selected with a feature selection algorithm, (3) we trained classifiers using each feature group individually, and (4) we trained classifiers by leaving out one feature group at a time (ablation study).

To evaluate how well each classifier identified each SOAP class, we used standard evaluation metrics: accuracy, recall, and precision. We computed the F1 score, which represents the harmonic mean between recall and precision and used the F1 score to select the best performing classifier. We used McNemar's test to evaluate whether the classifier errors were statistically significantly different for classifiers trained on all feature groups and classifiers trained after feature selection. We applied Yates correction (0.50) when one cell in the contingency table was less than or equal to 5 [137]. The presence (positive instances) and absence (negative instances) of the SOAP class were used to define true and false positives and true and false negatives between the reference standard and the automated classifier - see Table 15.

	Reference Standard	**Automated Classification**
True Positive (TP)	present	present
True Negative (TN)	absent	absent
False Positive (FP)	absent	present
False Negative (FN)	present	absent

Table 15: Definitions for Agreement and Performance Metrics

B.1.4 Results

We measured inter-annotator agreement of expert annotators applying the SOAP model to ED reports and developed SOAP classifiers using a diverse number of features. We observed the following results.

Can a clinical discourse framework be annotated with high agreement by annotators?

Our dataset of 50 reports was comprised of 4,130 sentences in which the number of sentences per document ranged from 32 to 198, with an average of 82.6 sentences per document. Prevalence and frequency of SOAP classes in the 4,130 sentences was as follows: 35.5% (*subjective*; n=1468), 44.0% (*objective*; n=1818), 5.5% (*assessment*; n=227), 11.3% (*plan*; n=465), and 8.1% (*not applicable*; n=335). Inter-annotator agreement for all classes exceeded the threshold for adequate agreement (0.70) - see Table 16. The most prevalent classes, *subjective* and *objective*, demonstrated greater than 0.90 agreement across all agreement metrics. Agreement was lowest for assessment with a Cohens kappa of 0.76; however, once corrected for prevalence, the kappa value increased to 0.940.

SOAP Class	**Observed Agreement**	**Positive Specific**	**Negative Specific**	**Chance Corrected**	**Prevalence Corrected**
Subjective	0.97	0.96	0.98	0.94	0.94
Objective	0.95	0.95	0.96	0.91	0.91
Assessment	0.97	0.78	0.98	0.76	0.94
Plan	0.96	0.83	0.98	0.80	0.92

Table 16: Agreement between two annotators for sentences from 25 reports

Can the clinical discourse framework be automatically annotated from clinical text?

Table 17 shows predictive performance of all SOAP classifiers. Overall, most supervised classifiers outperformed the baseline classifiers. As expected, the Positive class baseline did not have adequate precision, resulting in poor F1 scores for the less prevalent classes, *assessment* (11.0) and *plan* (22.5). The Majority class baseline did not predict the SOAP class, but reflected the imbalanced class distribution in the dataset. The Section classifier performed quite well with high F1 scores for *subjective* (88.2) and *objective* (70.2); however, it produced moderate performance for the less prevalent classes of *assessment* (54.4) and *plan* (70.2). The Section classifier performed with low recall on *assessment* (50.0) and *plan* (20.6) classes.

The SOAP classifier without feature selection (w/o FS) outperformed the Section classifier baseline by increasing the points of F1 scores for all classes – 6.2 (*subjective*), 23.2 (*objective*), 7.7 (*assessment*) and 42.6 (*plan*). The improved F1 scores can be explained by increased coverage for most classes with recall gains of 5.3 points (*subjective*), 34.8 points (*objective*) and 47.0 points (*plan*). We observed these gains at no expense of precision, but instead with modest to substantial point increases of 7.4 (*subjective*), 4.8 (*objective*), 29.6 (*assessment*) and 3.8 (*plan*).

We applied feature selection (w/FS) to reduce the feature space and determine if we could further improve the F1 scores. Feature selection improved performance for most classes, showing gains ranging from 1.1 to 13.6 points with the exception of the *subjective* class, which dropped by 0.5 points. In evaluating how well each feature group performed individually, we found that no single feature group individually produced greater F1 scores than the SOAP classifiers w/FS. Finally, we assessed how informative each feature group was to SOAP class prediction using an ablation study design. For each class, we observed a reduction of F1 scores by removing the contextual feature group, which was largely due to decreases in recall without the contextual features. This finding indicates that contextual features are important to SOAP classification.

Classifiers	**S**				**O**				**A**				**P**			
	Acc	F1	Prec	Rec	Acc	F1	Prec	Rec	Acc	F1	Prec	Rec	Acc	F1	Prec	Rec
Positive	35.5	70.8	35.5	100	44.0	87.6	44.0	100	5.5	11.0	5.5	100	11.3	22.5	11.3	100
Majority	65.5	0	0	0	56.0	0	0	0	94.5	0	0	0	88.7	0	0	0
Section	91.8	88.2	90.2	86.2	78.4	70.2	89.1	58.0	95.4	54.4	60.1	50.0	90.6	33.0	82.0	20.6
w/o FS	96.2	94.4	97.6	91.5	94.2	93.4	93.9	92.8	96.8	62.1	89.7	47.5	95.2	75.6	85.8	67.6
w/FS	95.7	93.9	94.7	93.1	95.2	94.5	94.5	94.5	97.8	75.7	95.9	62.6	95.5	77.0	90.7	66.9
Lex only	93.5	90.6	92.9	88.5	93.5	92.6	92.6	92.6	97.3	69.3	94.7	54.6	94.9	73.0	91.3	60.9
Syn only	91.5	87.7	91.0	84.5	93.0	92.0	92.8	91.2	96.7	60.1	92.7	44.5	94.8	73.4	85.9	64.1
Sem only	88.4	82.0	91.0	74.7	89.5	87.1	95.0	80.4	96.5	58.0	83.5	44.5	93.4	65.0	81.3	54.2
Con only	91.5	87.5	91.4	83.9	86.2	84.7	83.3	86.1	95.4	40.0	71.6	27.8	91.0	46.7	70.7	34.8
Heur only	68.9	31.4	73.0	20.0	55.9	1.4	44.8	0.70	94.5	0	0	0	88.7	0	0	0
Sans Lex	96.0	94.4	95.5	93.3	94.8	94.1	94.3	93.8	97.8	76.6	92.0	65.6	95.7	78.9	88.5	71.2
Sans Syn	95.9	94.1	94.9	93.4	95.4	94.8	94.8	94.8	97.5	72.7	92.5	59.9	95.6	77.5	92.0	66.9
Sans Sem	95.9	94.1	95.2	93.1	95.1	94.4	94.5	94.4	97.7	74.1	95.8	60.4	95.6	77.2	91.7	66.7
Sans Con	94.1	91.5	94.2	89.0	94.0	93.2	93.4	93.1	97.6	72.5	96.4	58.1	95.0	74.0	88.9	63.4
Sans Heur	95.8	94.0	94.8	93.2	95.0	94.4	94.4	94.4	97.8	76.2	95.4	63.4	95.6	77.2	91.7	66.7

Table 17: SOAP Classifiers including baselines (positive and majority class and section), all feature groups with and without feature selection (FS), each individual feature group (Lex=lexical, Syn=syntactic, Sem=semantic, Con=contextual, Heur=heuristic) and ablation arms (sans or leave-onegroup-out).

From the feature selection algorithm, we identified the most informative features for predicting each of the four SOAP classes. Section categories, CUIs, unigrams, bigrams and word/POS pairs were among the feature values with the highest weights in Table 18.

Subjective	**Objective**	**Assessment**	**Plan**
past_medical_history	rectal_exam	discharge_condition	date_transcribed
history_present_illness	cardiovascular_exam	admission_diagnosis	reviewed_VBN
allergies_and_adverse_rxns	heent_exam	discharge_diagnosis	discharge_VB
review_of_systems	abdominal_exam	C0042029	follow_VB
ear_review	extremity_exam	C0851827	“reviewed with”
cardiovascular_review	general_exam	weakness_NNP	“admitted”
gastrointestinal_review	neurological_exam	assessment_NNP	“the plan”
neurologic_review	C0015385	GI_RB	“this plan”
medications	C0205307	assessment_NN	“<s> follow”
C1301808	C0007012	dehydrated_VBN	“evaluated by”
C0027497	elevated_VBN	noninsulin-dependent_JJ	“a lumbar”
C0030450	CO2_NNP	“confusion.”	“puncture without”
C0332272	not_RB	“:confusion”	“examination findings”
“: negative”	he_states	“to micu”	“a bit”
“: no”	“sent.”	“micu for”	“<s> I”

Table 18: Feature values with the 15 highest weights for each SOAP class.

B.1.5 Discussion

The objectives of this study were to (1) assess the applicability of the SOAP model for ED reports and (2) determine which features contribute to accurate SOAP classification.

Can a clinical discourse framework be annotated with high agreement by annotators?

The SOAP model applied to 3,836 (92.9%) sentences in our dataset. All sentences that were not assigned a SOAP class by annotators either served administrative purposes, such as "Signed by: **NAME[AAA XXX GGG], MD," or were incorrectly segmented sentences, such as a section heading like "PHYSICAL EXAM:" segmented as a sentence. There were several sentences with more than one class assigned (3.99%). In rare cases, multiple class assignment was due to incorrect sentence segmentation in which two sentences were segmented as one. Most sentences with multiple SOAP classifications represented descriptions of clinical reasoning relating, for example, an *objective* measurement to a *plan* or a *plan* to an *assessment*. For instance, "She will be discharged in good condition with impression of viral illness" consists of both *plan* (patient will be discharged) and *assessment* (impression that she has a viral illness).

Annotators showed high agreement on the SOAP annotation task. The coverage of SOAP classes and high agreement for expert annotation suggests that the SOAP framework is applicable to ED reports and that the annotation schema for SOAP classes was well defined.

Performing the pilot study was helpful in evolving a schema for human experts. We also suspect that giving annotators flexibility of assigning all classes that apply to a single sentence was important for eliciting good agreement. Disagreements between annotators occurred most often when a statement was an *assessment*. This was consistent with our finding during the pilot study and was often a point of disagreement during training. For example, one annotator consistently labeled some sentences containing a problem mention as

an *assessment* even when the context indicated a *plan*, like "UTI" in "He was given printed instructions about UTI, Pyridium, and ciprofloxacin."

Can the clinical discourse framework be automatically annotated from clinical text?

Prevalence of SOAP classifications in our dataset varied from 6% for assessment to 44% for *objective classes*. As expected, performance of the SVM classifier was higher for more prevalent classes, with F1-scores of 0.62 (*assessment*), 0.76 (*plan*), 0.94 (*subjective*), and 0.93 (*objective*). Precision was higher than recall for all classes (the lowest precision score was 0.86 for *plan*), suggesting that false positive classifications were less of a problem than false negative classification and that more training data could further improve performance. Feature selection tended to improve classification performance, especially for the two less prevalent classes, which showed 1.4 point (*plan*) and 13.6 point (*assessment*) increases. From 32,215 original features, the number of features was reduced by a range of 82.7% (*plan*) to 94.1% (*assessment*), indicating that most features were not needed for accurate classification or that features we included had overlapping information. For some features, the number of values was reduced, including unigrams, bigrams, UMLS CUIs, section tags, and part-of-speech tags. For other features, all values for that feature were eliminated, including UMLS semantic type and its position in the sentence and the ConText lexicon. Eliminated features all belonged to the semantic feature type, but many were probably too broad to be discriminatory. For example, concepts with the UMLS semantic type "Body Part, Organ, or Organ Component" can occur in a description of review of systems, which is *subjective*, and in a description of a physical exam, which is *objective*, and therefore may not distinguish between the two classes. Semantic features that were not eliminated conveyed more specific information about the concept, such as the UMLS CUI **C0015385:** *limbs*, or about the reasoning process of the physician, such as hedge terms or words indicating state of mind. Features with the highest positive weights included section headings (discharge_diagnosis for *assessment* and abdominal_exam for *objective*), and predictive unigrams and bigrams (: negative for *subjective* and admitted for *plan*), and UMLS CUIs (**C0205307:** *normal* for

objective and **C0851827:** *diabetes mellitus* for *assessment*).

We evaluated the contribution of different types of features to classification accuracy. Assigning a SOAP class based only on the section in which a sentence was found was less accurate than a classifier using all features, but was quite accurate for *subjective* (F1 score 0.88) and *objective* (F1 score 0.70) classes. Recall was especially low with the section classifier for all classes, but *subjective* (0.86 *subjective*, 0.58 *objective*, 0.50 *assessment*, 0.21 *plan*). This finding is consistent with our experience in other classification tasks [64], showing that section is a critical factor in interpreting the context of a clinical problem, but is not reliable enough to be the sole factor in classification. Good performance of the section-based SOAP classifier suggests that our map from sections to SOAP classes was effective and that SecTag performed well at automatically tagging sections. For example, the ability to distinguish the section medications from discharge_medications was critical to accurate assignment of *subjective* and *plan* classes. Reports from other institutions and other report types may be less amenable to automated section tagging.

No classifier trained on an individual feature group produced an F1 score better than the classifier using all feature groups w/FS. However, performances of the syntactic feature group on *objective* and *plan* classification and of lexical features on *objective* classification were not statistically different from performance when using all groups. When we removed individual feature groups in the ablation studies, performance generally did not decrease significantly. Comparable performance may be due to overlap in feature values. For instance, the presence of lexical features such as unigrams and bigrams may provide enough information to discriminate the class when other features like state of mind and hedge terms were held out. Removing contextual features, such as sentence length and quartile position in a report, significantly decreased F1 scores for classifying *subjective* and *assessment* classes suggests that location within the structure of a document is meaningful. One interesting and unexpected finding was that removing lexical features produced a higher F1 score for classifying *assessment* and *plan* sentences. It may be that not relying on the words in the text can result in better performance when there is sparse training data.

During the pilot study, we performed a detailed error analysis on less prevalent classes and identified phrases and terms we thought would improve classification performance. We included the hand-crafted phrases as the heuristic feature group in this study and found that they did not provide any useful knowledge for predicting *assessment* and *plan*. This result may be due to the fact that the hand-crafted phrases are not very frequent, or it may be that the phrases are an indication of overfitting to our pilot data.

We reviewed the most heavily-weighted features for each of the target SOAP classes. For the *subjective* class, the most predictive features included sections that describe past and recent history, subsections of the review of systems, as well as CUIs attributed to "Signs and Symptoms", "Qualitative Concept", and "Geographic Locations." We would expect this, since physicians often describe the symptoms of the patient in terms of quality, severity and onset. In contrast, sections attributed to physical examination and CUIs associated with "Body Part, Organ, or Organ Component", "Functional Group", and "Biologically Active Substance, Inorganic Chemical" were predictive of *objective* sentences, which is consistent with our intuition that physicians describe findings and observations for each of the body systems and describe results from diagnostic tests and laboratories. Many of the features most predictive of *assessment* included diagnosis sections and CUIs describing "Population Groups" and "Disease or Syndrome." For the *plan* class, section tags were not highly predictive. We suspect this can be explained by the fact that physicians tend not to adhere to document structure as strictly at the end of a report as they do in the initial portion of the report, i.e., Plan and Assessment tend to become condensed into the ED Course as reports of implemented treatments, medical decision making and potential plans for follow-up. We also found that the word sense of a unigram is important for determining if a statement is a *plan*, such as discharge_VB versus discharge_NN.

B.1.6 Conclusion

We determined 1) the SOAP framework could be annotated with high agreement and 2) a SOAP classifier could be trained with moderate to high performance. The diverse features we used resulted in accurate automated assignment of *subjective* and *objective* classes and of fair assignment of *assessment* and *plan* classes. There is a tradeoff between the cost of acquiring syntactic and semantic features and the modest improvement over lexical features. SOAP classification of sentences could be a useful feature in other NLP tasks and could help localize information in reports for use in visualization and assessment of clinical care.

We developed an automated SOAP classifier that could be applied to clinical texts to help identify and structure the context of described problem mentions including potentially annotated treatments and tests related to pertinent patient problems.

BIBLIOGRAPHY

[1] S Visweswaran, P Hanbury, M Saul, and GF Cooper. Detecting adverse drug events in discharge summaries using variations on the simple Bayes model. In *AMIA Annual Symp Proc*, pages 689–93, 2003.

[2] QT Zeng, S Goryachev, S Weiss, M Sordo, SN Murphy, and R Lazurus. Extracting principle diagnosis, co-morbidity and smoking status for asthma research: Evaluation of a natural language processing system. *J Pathol Inform*, 6(30), 2010.

[3] H Harkema, WW Chapman, M Saul, ES Dellon, RE Schoen, and A Mehrotra. Developing a natural language processing application for measuring the quality of colonoscopy procedures. *J Am Med Inform Assoc*, 18:i150–i156, 2011.

[4] WW Chapman, LM Christensen, MM Wagner, PJ Haug, O Ivanov, JN Dowling, and RT Olszewski. Classifying free-text triage chief complaints into syndromic categories with natural language processing. *Artif Intell Med.*, 33(1):31–40, 2005.

[5] C Holmes. The Problem list beyond meaningful use. *J of AHIMA*, pages 30–35, 2011.

[6] S Meystre and P Haug. Randomized controlled trial of an automated problem list with improved sensitivity. *International Journal of Medical Informatics*, 77:602–12, 2008.

[7] JD Carpenter and PN Gorman. Using medication list problem list mismatches as markers of potential error. In *AMIA Annu Symp Proc*, pages 106–110, 2002.

[8] V Bahyam, W Hsu, E Watt, A Bui, H Kangarloo, and RK Taira. Problem-centric organization and visualization of patient imaging and clinical data. *Radiographics*, 29:331–343, 2009.

[9] TC Sibanda. Was the patient cured? understanding semantic categories and their relationships in patient records. Technical report, MIT, 2006.

[10] B Webber, M Egg, and V Kordoni. Discourse structure and language technology. *Natural Language Engineering*, (1):1–56, 2011.

[11] L Weed. *Medical Records, Medical Education and Patient Care: The Problem-Oriented Record as a Basic Tool.* Medical Publishers: Press of Case Western Reserve University, Cleveland: Year Book, 1970.

[12] Center for Medicare and Medicaid Services. EHR Incentive Programs-Maintain Problem List. http://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/downloads//3_Maintain_Problem_List.pdf.

[13] American Health Information Management Association. Appendix A: Definitions of Problem Lists from Authoritative Sources.

[14] L Carlo, Chase HS, and Weng C. Aligning structured and unstructured medical problems using UMLS. In *AMIA Annu Symp Proc.*, pages 91–95, 2010.

[15] D Jurafsky and JH Martin. *Speech and Language Processing.* Prentice-Hall, Englewood Cliffs, NJ, 2008.

[16] O Bodenreider and A McCray. Exploring semantic groups through visual approaches. *J Biomed Inform*, 36(6):414–32, 2003.

[17] A McCray, A Burgun, and O Bodenreider. Aggregating UMLS semantic types for reducing conceptual complexity. *Med Info*, pages 216–20, 2001.

[18] O Bodenreider and A Burgen. *Medical Informatics: Advances in Knowledge Management and Data Mining in Biomedicine*. Springer-Verlag, 2005.

[19] GA Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[20] B Smith and C Fellbaum. Medical WordNet: a new methodology for the construction and validation of information resources for consumer health. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04. Association for Computational Linguistics, 2004.

[21] CF Baker, CJ Fillmore, and JB Lowe. The Berkeley FrameNet Project. In *COLING-ACL '98: Proceedings of the Conference*, pages 86–90, Montreal, Canada, 1998.

[22] J Ruppenhofer, M Ellsworth, MRL Petruck, CR Johnson, and J Scheffczyk. FrameNet II: Extended theory and practice. Technical report, 2006.

[23] M Palmer, D Gildea, and P Kingsbury. The Propositional Bank: an automated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, 2005.

[24] EH Shortliffe and JJ Cimino. *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. Springer Science + Business Media, LLC, New York, NY, 2006.

[25] PM Nadkarni, L Ohno-Machado, and WW Chapman. Natural Language Processing: an Introduction. *J Am Med Inform Assoc*, 18:544–551, 2011.

[26] S Russell and P Norvig. *Artificial Intelligence: a Modern Approach*. Prentice Hall, 1995.

[27] JR Quinlan. *Machine Learning*. Kluwer Academic Publishers, 1986.

[28] R Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[29] R Kohavi. Scaling up the accuracy of NaiveBayes Classifiers: a DecisionTree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207. AAAI Press, 1996.

[30] T Bayes and R Price. An essay towards solving a problem in the doctrine of chances. *Phil Trans*, 53:370–418, 1963.

[31] R Duda and P Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[32] C Cortes and V Vapnik. *Machine Learning*. Kluwer Academic Publishers, 1995.

[33] J Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. 1998.

[34] LA Liu, D Wei, and H Lei. *Handbook of Research on Computational and Systems Biology: Interdisciplinary Applications*. IGI Global, 2011.

[35] G Bordage and M Lemieux. Semantic structures and diagnostic thinking of experts and novices. *Acad Med*, 66:70–72, 1991.

[36] Center for Disease Control. Classification of Diseases, Functioning, and Disability. http://www.cdc.gov/nchs/icd/icd9cm.htm.

[37] A Aronson. MetaMap: Mapping Text to the UMLS Metathesaurus. Technical report, National Library of Medicine, 2006.

[38] National Library of Medicine. The CORE Problem List Subset of SNOMED-CT. Unified Medical Language System 2011. http://www.nlm.nih.gov/research/umls/SNOMED-CT/core_subset.html.

[39] JR Campbell and TH Payne. A Comparison of four schemes for codification of problem lists. In *AMIA Annu Symp Proc.*, pages 201–205, 1994.

[40] D Lee, F Lau, and H Quan. A method for encoding clinical datasets with SNOMED CT. *BMC Medical Informatics and Decision Making*, 10(53):1–12, 2010.

[41] H. Wasserman and J. Wang. An applied evaluation of SNOMED CT as a clinical vocabulary for the computerized diagnosis and problem list. In *AMIA Annual Symp Proc*, pages 699–703, 2003.

[42] P Elkin, D Brown, C Husser, B Bauer, D Wahner-Roedler, and S Rosenbloom. Evaluation of the content coverage of SNOMED CT: Ability of SNOMED clinical terms to

represent clinical problem lists. In *Mayo Clinic Proceedings.*, volume 81, pages 741–8, 2006.

[43] KW Fung, C McDonald, and S Srinivasan. The UMLS-CORE Project: a study of the problem list terminologies used in large healthcare institutions. *J Am Med Inform Assoc*, 17:675–680, 2010.

[44] A Wright, J Flebowitz, AB McCoy, and DF Sittig. Comparative analysis of the VA/Kaiser and NLM CORE Problem Subsets: An empirical study based on problem frequency. In *AMIA Annual Symp Proc.*, pages 1532–40, 2011.

[45] JR Campbell, J Xu, and KW Fung. Can SNOMED CT fulfill the vision of a compositional terminology? analyzing the use case for problem list. In *AMIA Annu Symp Proc.*, pages 181–8, 2011.

[46] AR Aronson and Lang FM. An Overview of MetaMap: Historical prospective and recent advances. *J Am Med Inform Assoc*, 17:229–236, 2010.

[47] Q Zou, WW Chu, C Morioka, GH Leazer, and H Kanarloo. IndexFinder: A method of extracting key concepts from clinical texts from indexing. pages 171–180, 2008.

[48] JC Denny, JD Smithers, A Spickard, and RA Miller. A new tool to identify key biomedical concepts in text documents, with special application to curriculum content. page 1007, 2002.

[49] AR Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proc AMIA Symp*, pages 17–21, 2001.

[50] S Meystre and P Haug. Evaluation of medical problem extraction from electronic clinical documents using MetaMap Transfer (mmtx). *Stud Health Technol Inform*, 116:823–828, 2005.

[51] I Solti, B Aaronson, G Fletcher, M Solti, JH Gennari, M Cooper, and T Payne. Building an automated problem list based on natural language processing: Lessons learned in the early phase of development. pages 687–691, 2008.

[52] Ö Uzuner, BR South, S Shen, and SL DuVall. 2010 i2B2/VA challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*, 18(5):552–556, 2011.

[53] JP Pestian, C Brew, P Matykiewicz, DJ Hovermale, N Johnson, KB Cohen, and W Duch. A shared task involving multi-label classification of clinical free text. In *BioNLP 2007: Biological, translational, and clinical language processing*, pages 97–104, 2007.

[54] WW Chapman, W Bridewell, P Hanbury, GF Cooper, and BG Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform*, 2001:34–301, 2001.

[55] W Chapman, W Bridewell, P Hanbury, G Cooper, and B Buchanan. Evaluation of negation phrases in narrative clinical reports. In *Proceedings AMIA*, pages 105–109, 2001.

[56] A Goryachev, M Sordo, L Ngo, and QT Zeng. Implementation and evaluation of four different methods of negation detection. Technical report, 2006.

[57] Ö Uzuner, X Zhang, and T Sibanda. Machine learning and rule-based approaches to assertion classification. *J Am Med Inform Assoc*, 16(1):552–556, 2009.

[58] BE Chapman, S Lee, HP Kang, and WW Chapman. Document-level classification of CT pulmonary angiography reports based on an extension of the ConText algorithm. *J Biomed Inform*, 44(5):728–737, 2011.

[59] C Clark, J Aberdeen, M Coarr, D Tresner-Kirsh, B Wellner, A Yeh, and L Hirschman. MITRE system for clinical assertion status classification. *J Am Med Inform Assoc*, 11(18):563–567, 2011.

[60] WW Chapman, D Chu, and JN Dowling. ConText: an algorithm for identifying contextual features from clinical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, BioNLP '07, pages 81–88, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[61] H Harkema, JN Dowling, T Thornblade, and WW Chapman. ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *J. of Biomedical Informatics*, 42(5):839–851, 2009.

[62] RA Wilson, WW Chapman, SJ DeFries, MJ Becich, and BE Chapman. Automated ancillary cancer history classification from mesothelioma patients from free-text clinical reports. *J Pathol Inform*, pages 1–24, 2010.

[63] S Goryachev, H Kim, and Q Zeng-Treitler. Identification and extraction of family history information from clinical reports. In *AMIA Annu Symp Proc.*, pages 247–251, 2008.

[64] DL Mowery, H Harkema, JN Dowling, JL Lustgarten, and WW Chapman. Distinguishing historical from current problems in clinical reports: Which textual features help? In *Proceedings of the Workshop on BioNLP*, pages 10–18, 2009.

[65] J Cogley, N Stokes, J Carthy, and J Dunnion. Analyzing patient records to establish if and when a patient suffered from a medical condition. In *Workshop on Biomedical Natural Language Processing (BioNLP 2012)*, pages 38–46, 2012.

[66] AK Irvine, S Haas, and T Sullivan. TN-TIES: A system for extracting temporal information from emergency department triage notes. pages 328–332, 2008.

[67] D Mahalingam, R Medlin, D Travers, and S Haas. Temporal Information Extractor: Identifying symptom onset date from emergency department notes. page 1876, 2012.

[68] L Zhou, C Friedman, S Parsons, and G Hripcsak. System architecture for temporal information extraction, representation, and reasoning in clinical narrative reports. pages 869–873, 2005.

[69] L Zhou, G Melton, S Parsons, and G Hripcsak. A temporal constraint structure for extracting temporal information from clinical narrative. *J Biomed Inform*, 34(4):424–439, 2006.

[70] G Hripcsak, L Zhou, S Parsons, AK Das, and SB Johnson. Modeling electronic discharge summaries as a simple temporal constraint satisfaction problem. *J Am Med Inform Assoc*, 12(1):55–63, 2005.

[71] B deBuijin, C Cherry, S Kiritchenko, J Martin, and X Zhu. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2B2 2010. *JAMIA*.

[72] R Saurı, J Littman, B Knippen, R Gaizauskas, A Setzer, and J Pustejovsky. TimeML annotation guidelines. http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf.

[73] G Savova, S Bethard, W Styler, J Martin, M Palmer, J Masanz, and W Ward. Towards Temporal Relation Discovery from the Clinical Narrative. pages 568–572, 2009.

[74] A Roberts, R Gaizauskas, M Hepple, N Davis, G Demetriou, Y Guo, J Kola, I Roberts, A Setzer, A Tapuria, and B Wheeldin. The CLEF Corpus: Semantic annotation of clinical text. pages 625–629, 2007.

[75] BL Webber. Description formation and discourse model synthesis. pages 42–50.

[76] R Prasad, N Dinesh, A Lee, Ei Miltsakak, L Robaldo, A Joshi, and B Webber. The Penn Discourse TreeBank 2.0. In *In Proceedings of LREC*, 2008.

[77] R Prasad, S McRoy, N Frid, AK Joshi, and H Yu. The Biomedical Discourse Relation Bank. *BMC Bioinformatics*, 12:188, 2011.

[78] GK Savova, WW Chapman, J Zheng, and RS Crowley. Anaphoric relations in the clinical narrative: Corpus creation. *J Am Med Inform Assoc*, 18:459–65, 2011.

[79] WW Chapman, GK Savova, J Zheng, M Tharp, and RS Crowley. Anaphoric reference in clinical reports: Characteristics of an annotated corpus. *J Biomed Inform*, 45(3):507–21, 2011.

[80] R Grishman and B Sundheim. Message Understanding Conference - 6 a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, pages 466–471. Association for Computational Linguistics.

[81] Informatics for Integrating Biology and the Bedside. 2011 i2B2 Coreference Challenge. https://www.i2b2.org/NLP/DataSets/Main.php.

[82] J Zheng, WW Chapman, TA Miller, C Lin, RS Crowley, and Savova GK. A system for coreference resolution for the clinical narrative. *J Am Med Inform Assoc*, 19(4):660–667, 2012.

[83] TY He. Coreference resolution on entities and events for hospital discharge summaries. Master's thesis, Massachusetts Institute of Technology, 2007.

[84] R Mulkar-Mehta, JR Hobbs, CC Liu, and XJ Zhou. Discovering causal and temporal relations in biomedical texts. pages 74–80, 2009.

[85] WW Chapman and P Haug. Bayesian modeling for linking causally related observations in chest X-ray reports. pages 587–591, 1998.

[86] B Rink, S Harabagiu, and K Roberts. Automatic extraction of relations between medical concepts in clinical texts. *J Am Med Inform Assoc*, 18:594–600, 2011.

[87] JF. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[88] PW Jordan, DL Mowery, J Wiebe, and WW Chapman. Annotating conditions in clinical narratives to support temporal classification. 2010.

[89] H Harkema, A Setzer, R Gaizauskas, M Hepple, R Power, and J Rogers. Mining and modelling temporal clinical data. 2005.

[90] M Verhagen, R Gaizauskas, F Schilder, M Hepple, J Moszkowicz, and J Pustejovsky. The TempEval Challenge: Identifying temporal relations in text. *Language Resources and Evaluation, Special Issue on Computational Semantic Analysis of Language: SemEval-2007 and Beyond*, 43(2):161–179, 2009.

[91] J Pustejovsky, M Verhagen, X Nianwen, R Gaizauskas, M Hepple, F Schilder, G Katz, R Sauri, E Saquete, T Caselli, N Calzolari, K Lee, and S Im. TempEval-2: Evaluating events, time expressions, and temporal relations, 2009.

[92] N UzZama, H Llorens, L Derczynaksi, M Verhagen, and J Pustejovsky. TempEval-3: Evaluating events, time expressions, and temporal relations.

[93] Informatics for Integrating Biology and the Bedside. 2012 i2B2 shared-tasks and workshop on challenges in natural language processing for clinical data. https://www.i2b2.org/NLP/TemporalRelations/Call.php.

[94] H Jung, J Allen, N Blaylock, W de Beaumont, L Galescu, and M Swift. Building timelines from narrative clinical records: Initial results based-on deep natural language understanding. pages 625–629, 2011.

[95] L Zhou, S Parsons, and G Hripcsak. The evaluation of a temporal reasoning system in processing clinical discharge summaries. *J Am Med Inform Assoc*, 15(1):99–106, 2005.

[96] R Gaizauskas, H Harkema, M Hepple, and A Setzer. Task-oriented extraction of temporal information: the case of clinical narratives. In *Temporal Representation and Reasoning, 2006*, pages 188–195, 2006.

[97] P Bramsen, P Deshpande, YK Lee, and R Barzilay. Finding temporal order in discharge summaries. pages 81–85, 2006.

[98] P Raghavan, E Fosler-Lussier, and AM. Lai. Learning to temporally order medical events in clinical text. pages 70–74, 2012.

[99] JC Denny, A Spickard, KB Johnson, JF Peterson, and RA Miller. Evaluation of a method to identify and categorize section headers in clinical documents. *J Am Med Inform Assoc*, pages 806–15, 2009.

[100] PS Cho, RK Taira, and H Kangarloo. Automatic section segmentation of medical reports. pages 155–159, 2003.

[101] DL Mowery, JM Wiebe, S Visweswaran, H Harkema, and WW Chapman. Building an automated SOAP classifier for emergency department reports. *J Biomed Inform*, 45:71–81, 2012.

[102] L Carlson, D Marcu, and ME Okurowski. *Current Directions in Discourse and Dialogue*. Kluwer Academic Publishers, 2003.

[103] MG Core and JF Allen. Coding dialogs with the damsl annotation scheme, 1997.

[104] C Friedman, OP Alderson, HJ Austin, JJ Cimino, and BS Johnson. A general natural language text processor for clinical radiology. *J Am Med Inform Assoc*, 1(2):161–174, 1994.

[105] WR Hersh and LC Donohoe. SAPHIRE International: a tool for cross-language information retrieval. pages 673–674, 1998.

[106] SB Koehler. *Symtext: a Natural Language Understanding System for Encoding Free Text Medical Data*. PhD thesis, The University of Utah, 1998.

[107] LM Christensen, PJ Haug, and M Fiszman. MPLUS: a probabilistic medical language understanding system. pages 29–36. Association for Computational Linguistics, 2002.

[108] GK Savova, J Masanz, P Orgen, J Zheng, S Sohn, K Kipper-Schuler, and C Chute. Mayo Clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *J Am Med Inform Assoc*, 17:507–13, 2010.

[109] S Meystre and P Haug. Automation of a problem list using natural language processing. *BMC Medical Informatics and Decision Making*, 5(30), 2005.

[110] DL Mowery, H Harkema, and WW Chapman. Temporal annotation of clinical text. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, BioNLP '08, pages 106–107. Association for Computational Linguistics, 2008.

[111] WW Cohen. Fast effective rule induction. 1995.

[112] J Lustgarten. Rule learner in-house implementation. www.dbmi.pitt.edu/probe.

[113] DL Mowery, S Velupillai, and WW Chapman. Medical diagnosis lost in translation: Analysis of uncertainty and negation expressions in english and swedish clinical texts. BioNLP '12, pages 56–64, Stroudsburg, PA, USA, 2012.

[114] DL Mowery, PW Jordan, J Wiebe, H Harkema, J Dowling, and WW Chapman. Semantic annotation of clinical events for generating a problem list. pages 1032–1041, 2013.

[115] ST Wu, VC Kaggal, D Dligach, JJ Masanz, P Chen, L Becker, WW Chapman, GK Savova, H Liu, and Chute CG. A common type system for clinical natural language processing. *J Biomed Semantics*, 4(1), 2013.

[116] N Elhadad, WW Chapman, T OGorman, M Palmer, and G Savova. The ShARe schema for the syntactic and semantic annotation of clinical texts. *J Biomed Semantics*, 2013.

[117] WW Chapman, JN Dowling, and MM Wagner. Generating a reliable reference standard set for syndromic case classification. *J Am Med Inform Assoc*, 12:618–629, 2005.

[118] WW Chapman and JN Dowling. Inductive creation of an annotation schema for manually indexing clinical conditions from emergency department reports. *J Am Med Inform Assoc*, 39(2):196–208, 2006.

[119] G Hripcsak and AS Rothschild. Agreement, the F-measure, and reliability in information retrieval. *J Am Med Inform Assoc*, 12(3):296–298, 2005.

[120] R Artstein and M Poesio. Inter-coder agreement for computational linguistics. *Comput Linguist*, 34(4):555–596, 2008.

[121] DL Mowery, JM Wiebe, S Visweswaran, T Schleyer, and WW Chapman. Which NLP annotations contribute to accurate automatic problem list generation? In *NLM Training Day 2012*, 2012.

[122] M Poesio and R Vieira. A corpus-based investigation of definite description use. *Comput Linguist*, 24(2):183–216, 1998.

[123] OF Zaidan, J Eisner, and CD Piatko. Using "annotator rationales" to improve machine learning for text categorization. pages 260–267, 2007.

[124] DL Mowery, M Ross, S Velupillai, J Wiebe, SM Meystre, and WW Chapman. Generating patient problem lists from the ShARe corpus using SNOMED CT/SNOMED CT CORE problem lists. BioNLP '14, pages 54–58, Baltimore, MD, USA, 2014.

[125] M Saeed, C Lieu, G Raber, and Roger G. MIMIC II: a massive temporal icu patient database to support research in intelligent patient monitoring. *Comput Cardiol*, 2002.

[126] BR South, S Shen, J Leng, TB Forbush, SL DuVall, and WW Chapman. A prototype tool set to support machine-assisted annotation. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, BioNLP '12, pages 130–139. Association for Computational Linguistics, 2012.

[127] N Elhaded. ShARe corpus. https://sites.google.com/site/shareclefehealth/.

[128] D Mowery. ShARe/CLEF eHealth challenge 2014 - Task 2. https://sites.google.com/a/dcu.ie/clefehealth2014/task-2/2014-dataset.

[129] S Velupillai. *Shades of Certainty – Annotation and Classification of Swedish Medical Records*. Doctoral thesis, Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden, April 2012.

[130] D Mowery, H Harkema, BE Chapman, R Hwa, J Wiebe, and WW Chapman. An automated soap classifier for emergency department reports. 2010.

[131] Journal of Biomedical Informatics - Author Rights. http://www.elsevier.com/journal-authors/author-rights-and-responsibilities.

[132] B Santorini. Part-of-speech tagging guidelines for the penn treebank project, 1990.

[133] K Toutanova, D Klein, CD Manning, and Y Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. *HLT-NAACL*, pages 252–259, 2003.

[134] E Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Comput Linguist*, pages 543–565, 1995.

[135] A Khoo, Y Marom, and D Albrecht. Experiments with sentence classification. *Australasian Language Technology Workshop*, pages 18–25, 2006.

[136] RE Mercer, CD Marco, and FW. Kroon. The frequency of hedge terms in citation contexts in scientific writing. *Adv Artif Intell*, page 7588, 2004.

[137] F Yates. Contingency tables involving small numbers and the 2 test. pages 217–235, 1934.