AUTOMATIC DOMAIN ADAPTATION OF WORD SENSE DISAMBIGUATION BASED ON SUBLANGUAGE SEMANTIC SCHEMATA APPLIED TO CLINICAL NARRATIVE

by

Olga Patterson

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Biomedical Informatics

The University of Utah

May 2012

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Olga Patterson**

has been approved by the following supervisory committee members:

John F. Hurdle	, Chair	**3/14/2012** Date Approved
Bruce Bray	, Member	**3/21/2012** Date Approved
Lewis Frey	, Member	**3/22/2012** Date Approved
Stephane Meystre	, Member	Date Approved
Ellen Riloff	, Member	**3/22/2012** Date Approved

and by **Joyce A. Mitchell** , Chair of the Department of **Biomedical Informatics**

and by Charles A. Wight, Dean of The Graduate School.

ABSTRACT

Domain adaptation of natural language processing systems is challenging because it requires human expertise. While manual effort is effective in creating a high quality knowledge base, it is expensive and time consuming. Clinical text adds another layer of complexity to the task due to privacy and confidentiality restrictions that hinder the ability to share training corpora among different research groups. Semantic ambiguity is a major barrier for effective and accurate concept recognition by natural language processing systems.

In my research I propose an automated domain adaptation method that utilizes sublanguage semantic schema for all-word word sense disambiguation of clinical narrative. According to the sublanguage theory developed by Zellig Harris, domain-specific language is characterized by a relatively small set of semantic classes that combine into a small number of sentence types. Previous research relied on manual analysis to create language models that could be used for more effective natural language processing. Building on previous semantic type disambiguation research, I propose a method of resolving semantic ambiguity utilizing automatically acquired semantic type disambiguation rules applied on clinical text ambiguously mapped to a standard set of concepts.

This research aims to provide an automatic method to acquire Sublanguage Semantic Schema (S3) and apply this model to disambiguate terms that map to more than one concept with different semantic types. The research is conducted using unmodified MetaMap version 2009, a concept recognition system provided by the National Library of Medicine, applied on a large set of clinical text. The project includes creating and comparing models, which are based on unambiguous concept mappings found in seventeen clinical note types. The effectiveness of the final application was validated through a manual review of a subset of processed clinical notes using recall, precision and F-score metrics.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Words cannot express the depth of my gratitude to the people who made this work possible, so I am humbly saying thank you to:

John F. Hurdle - my graduate advisor, for his active mentorship, patience, and countless hours spent discussing my project.

Bruce Bray, Ellen Riloff, Stephane Meystre, Peter Haug, Dina Demner-Fushman and Lewis Frey - the members of my graduate committee, for their time and expertise that they provided over the years.

Sean Igo - for his programming support and cheerful acceptance of any programming challenges that I had to throw at him as I was building my system prototype.

Denise Beaudoin - for her time and effort adjudicating the reference standard.

Jenifer Williams and Tyler Forbush - for their annotation work on creating the reference standard.

Jason Patterson - my husband, for supporting me and providing financially and emotionally to our family while I was busy pursuing my dreams.

Larisa and Viktor Yatsenko - my parents, for being my inspiration and serving as examples that anything is possible as long as you put enough effort in achieving your goals.

This research has been supported by the NLM under grants T15LM007124 (fellowship), 5R21LM009967- 02, and 3R21LM009967-01S1(ARRA). An allocation of computer time from the Center for High Performance Computing at the University of Utah is gratefully acknowledged.

CHAPTER 1

INTRODUCTION

Imagine a clinical world in which clinicians dictate all patient information using natural speech into an Electronic Medical Record (EMR) system; the speech is automatically parsed into a structured form and the meaningful data is stored as database entries. Unfortunately, such a world is still in the realm of science fiction. The main reason such a world has not materialized despite decades of research is that phonetic, lexical, syntactic, and semantic ambiguity is characteristic of natural speech. Advances have been made to resolve each of these types of ambiguities, and simpler subproblems have been solved at a satisfactory level [1,2]. However, an accurate, general-purpose, adaptable concept recognition system is still a hope for the future.

This dissertation project tackles the problem of semantic ambiguity of the natural text found in clinical notes. Building on previous research on semantic type disambiguation, I propose a method called the Sublanguage Semantic Schema (S3) to resolve semantic ambiguity utilizing automatically acquired semantic type disambiguation rules applied to clinical text, which was ambiguously mapped to concepts from a standard terminology. MetaMap, a powerful system designed to map terms from text to UMLS Metathesaurus, is used to illustrate the feasibility of a practical implementation of my proposed method.

1.1 Problem Statement

Clinical language is complex. It is inconsistent at first look. It is unstructured and often ungrammatical. Individual clinicians have their personal opinions on what should and should not be noted about the patient in the medical record. The content and structure of the narrative depends on the type of service provided, kind of document, clinical setting, author's role, and subject matter domain [3]. In the current research, I am focusing on the document type *clinical note*. Regardless of the purpose of a clinical note, it potentially contains clinically relevant information in the free text that is created in the process of patient care [4]. Most EMR systems have structured forms and checklists that physicians use

to record patient data. However, the healthcare environment is broad, often unpredictable, and nuanced. As of today, there are no "off-the-shelf" systems that are able to provide a user-friendly way to report all potentially important information about all patients, or the care that they received [5]. Such lack of essential functionality is the reason why free text persists as a method of keeping clinical records complete. Forcing clinicians to use structured forms that enable computer-friendly data entry has not been successful and usually causes strong user resistance [6].

Since the early years following the introduction of electronic medical record systems in clinical practice in the 1960s, information technology researchers have approached the clinical narrative as gold ore ready to be processed. Similar to gold mining, extracting meaningful information from clinical narratives has been a labor intensive process. Previous research efforts have identified "golden nuggets" using manually created rules for specific research questions [2]. As the technology and algorithms improve, developing new, more general, more precise, and more accurate methods is becoming more difficult. The latest wave of NLP research is directed to finding new ways of learning new information utilizing existing knowledge sources and technologies. Since the EMR systems have been introduced, large repositories of clinical text have been accumulated. These repositories can be used for information extraction through Natural Language Processing (NLP). Such language processing methods often rely on human annotated text.

In the general language processing world, several sets of annotated texts have been created and available to researchers for shared use [7, 8]. However, in the clinical world this approach is complicated by the sensitive nature of the texts. Clinical texts often contain identifiable data about a patient that are covered by security and confidentiality requirements such as the Health Insurance Portability and Accountability Act (HIPAA) of 1996. In this environment, only a small number of people can have access to the texts. Such access restrictions make obtaining manual annotations difficult, because the process cannot be outsourced to a third party. Each organization that attempts annotation projects is challenged with finding trusted human resources within the organization. When an NLP system is developed, it is optimized for the text that was used to create its knowledge base. Therefore, when a system is transferred into a new clinical environment, the knowledge base has to be adjusted in order to achieve the highest level of performance. Since the knowledge base acquisition is labor intensive, system adaptation is a costly and time consuming task. The knowledge-acquisition bottleneck is the major barrier for clinical NLP system

implementation [9]. Unable to adapt an existing system, large organizations develop their own in-house systems that are not shared across organizations; whereas, smaller medical facilities are forced to use off-the-shelf systems that have limited applicability and are not optimized for the organization's specific setting. Therefore, there is a clear need to develop methods of automatic knowledge base acquisition in order to enable system portability into a new clinical setting.

1.2 Main Objectives

Information retrieval, information extraction, question answering, machine translation, and most other natural language processing tasks rely on accurate concept recognition. However, clinical text is highly ambiguous; it challenges existing concept recognition systems. There is a clear need for a concept recognition system that serves a general purpose and is highly accurate. Enabling a fast, accurate and economical method of domain adaptation of a concept recognition system is the main vision of the current research.

Clinical NLP experts have always assumed that clinical language is not homogenous but varies depending on the clinical setting. However, this assumption has never been tested on a wide range of clinical text. As the first step in this project, I show that even within the same organization, the clinical language varies.

1.2.1 Aim I

Demonstrate language variability across various clinical settings within the same organization.

Research question 1.1 Is there a clear sublanguage variation among various clinical sublanguages?

Research question 1.2 Does the language variation depend on the clinical setting or a specific clinical subdomain?

When addressing Aim I, I identified the natural grouping of clinical text that resulted from unsupervised clustering of documents of different types, as described in Chapter 3.

1.2.2 Aim II

Design and validate a tool that automatically acquires and uses sublanguage characteristics for word sense disambiguation through semantic type disambiguation of terms mapped to multiple concepts with different semantic types.

Research question 2.1 Does the developed system work well for clinical term disambiguation in a range of clinical note types as compared to a manually annotated test set?

Research question 2.2 Does the system perform better than a baseline method such as MetaMap?

When addressing Aim II, I designed a system prototype and evaluated its performance on a set of manually annotated sentences extracted from clinical notes of four note types, as described in Chapter 4.

1.2.3 Aim III

Identify performance-improving steps on a range of clinical notes.

Research question 3.1 Can the feature space be substantially decreased without a significant loss of accuracy of the classification model?

Research question 3.2: Does the sublanguage feature space derived for those terms that were unambiguously mapped differ from the feature space of ambiguous terms?

When addressing Aim III, I define preprocessing and postprocessing steps that could potentially lead to improved system performance, as described in Chapter 5.

1.3 Relevance to Biomedical Informatics

According to Bernstam and colleagues, one of the main goals of biomedical informatics is to bridge the gap between the human information needs and the capabilities of the current technology [10]. Clinical informatics is a major part of a larger field of biomedical informatics. At this time, most of the information entered into a patient record in free text format is unreachable for computerized processing. Accurate, robust, and fast natural language processing would enable a vast range of possible uses of data from decision support at the point of care to reporting, surveillance, and research. My research advances the current language processing technology by enabling automatic domain adaptation of a computerized concept identification system. Improving portability of existing systems would promote collaboration between facilities and research groups.

CHAPTER 2

BACKGROUND

2.1 Natural Language Processing

NLP has a long history as research projects and practical implementations. It is traditionally defined as computerized processing of text. NLP is a very broad field that incorporates a large variety of tasks that differ in their complexity and specificity [2]. The term "natural language processing" is often equated to computational linguistics; however, these terms are not interchangeable [11]. Unlike computational linguistics, NLP approaches text as a source of data about the state of the world. It is characterized by developing and applying computational methods for a particular task and to achieve a practical purpose. The possibility of fully automatic language processing was first suggested in Weaver's memorandum that introduced the idea of machine translation [12]. Since then the field of NLP has grown to include a variety of methods and tasks of different levels of computational complexity and scope.

The list of NLP tasks ranges from low level general tasks, such as tokenization and sentence segmentation, to problem-specific tasks such as information extraction and question answering. High-level tasks rely on accurate performance of lower level tasks. For example, the accuracy of information extraction depends on correct parsing (which in turn depends on morphological segmentation, tokenization, sentence segmentation, and part of speech tagging), named entity recognition, concept recognition (which relies on word sense disambiguation), co-reference resolution, and relationship extraction. The current state-of-the-art systems that solve low level tasks achieve high accuracy; and, when used in a limited domain, are comparable to performance of a human annotator [13]. Word sense disambiguation as one of the components of an accurate concept recognition is the focus of the current project.

2.1.1 Word Sense Disambiguation

Concept recognition (or term identification) is regarded as a single most important factor in accurate information extraction [14]. Traditional linguistic theory studies the language

form and meaning as two separate though related elements of language. It recognizes that the same meaning can be expressed in various physical forms. It also posits that the same physical form can express a number of meanings. Therefore, determining correct semantic interpretation that is implied by a specific textual representation (the physical form) in specific context is an intermediary step in the process of concept recognition. The computational approach to this task is called Word Sense Disambiguation (WSD). The WSD process involves selecting one meaning out of a discrete number of known possible senses for a specific term [9, 15, 16]. The need for WSD arises as a consequence of semantic ambiguity that is characteristic to human language.

The difficulty of the WSD task varies depending on the availability of an electronic dictionary that is used to create the sense inventory for each term, as well as on the similarity of the possible senses. If the available dictionary does not have the true meaning of the term as one of the possible definitions, it is impossible for any WSD algorithm to identify the correct sense because it will be missing from the sense inventory. Similar to humans, a computerized algorithm has a hard time differentiating between similar concepts, therefore, the subtler the difference between meanings is, the more errors will result from disambiguation [15]. The most common steps to perform WSD are as follows:

1. Identify a list of specific target words for the system;
2. Create a sense inventory for each target word;
3. Extract (or create) examples that use the target word in one of the identified senses;
4. Label each instance of the target words with one of the senses from the sense inventory using manual annotation;
5. Employ machine learning or statistical approach to learn WSD rules using sample sentences;
6. Measure the system performance by applying it to another manually annotated set of examples [17].

The approaches to WSD can be grouped depending on several factors:

1. Based on the method of disambiguation model acquisition, a WSD algorithm can be: *a*) rule-based, *b*) example-based [18], or *c*) statistical [19, 20].
2. Based on the scope of disambiguation, a WSD algorithm targets: *a*) a restricted target word set, or *b*) all words.
3. Based on the extent of manual annotation performed to create the disambiguation model: *a*) supervised, *b*) unsupervised, *c*) knowledge-based, or *d*) hybrid.

Rule based systems derive their knowledge base from the manually created rules for disambiguation. *Example based* systems use example databases that contain example of sentences that use a target word in one of the identified senses. Disambiguation is performed by finding the most similar sentence example. A variety of methods can be employed to select the most informative examples [18]. *Statistical* WSD systems rely on computational algorithms to build disambiguation models using lexical features of the target word context.

Supervised WSD is one of the widely used approaches for development of WSD systems. It is based on supervised machine learning applied to a manually sense-annotated text and then uses the resulting model to perform word sense disambiguation on new text. The main disadvantage of such an approach is a high cost of manual annotation [21]. This approach is especially problematic in "all-words" WSD, where the system analyses all ambiguous words in text and not just a specific limited subset. *Unsupervised* methods are often called word sense discrimination [22] or sense discovery [23] because they aim to distinguish the word senses by clustering them in groups based on the context in which the word appears. The main disadvantage of such approaches is that after word sense discrimination is performed, human review is required to label the word sense clusters with the correct sense.

Semi-supervised or minimally supervised WSD methods use a small, manually-annotated text in combination with a large untagged corpus. Two variations of semisupervised approaches are bootstrapping and active learning methods. *Bootstrapping* uses a small corpus that was manually selected and tagged to learn the initial model and then utilizes the large untagged corpus to improve this model [9, 24]. The bootstrapping method as implemented by Yarowsky is a minimally supervised method that relies on *one sense per collocation* and *one sense per discourse* principles [25]. *Active learning* methods identify the most informative examples from the large untagged corpus and present them to a human expert for disambiguation [26, 27]. Supervised and unsupervised WSD methods are also called *corpus-based* methods because they use language models learned from a training text dataset [28].

Knowledge based WSD methods identify the word senses using external knowledge resources such as dictionaries, thesauri, or ontologies; or manually created disambiguation rules [29,30]. These methods include identifying the most likely meaning of the word using *a*) selectional preferences that restrict the semantic type of the word sense based on the context [31]; *b*) information formats with slots for specific type of information [32]; *c*) the context using unambiguous meanings of the neighboring terms [33]; *d*) semantic similarity

calculated using an ontology of semantic network [34]; or *e*) unambiguous meanings of the word in a different language using a parallel corpus [35].

Hybrid methods use variations of the approaches described above. Some examples of hybrid systems include Durham and SenseLearner. The Durham system utilizes word sense frequencies calculated using a manually annotated text and applies word collocations as well as WordNet contextual scores [36]. The SenseLearner system uses word collocations learned from a small manually-annotated corpus enhanced by the WordNet taxonomy [37].

A wide availability of large general English lexical databases, such as WordNet, and specialized ontologies, such as GeneOntology (GO) and the Unified Medical Language System (UMLS), makes it possible to develop a hybrid approach that combines knowledge based methods and supervised learning algorithms. For example, the A-CUI algorithm created by McInnes calculates similarity of the target word feature vector based on the word's surrounding context and the concept feature vector for each candidate concept extracted from the UMLS [38]. As the size and quality of the existing knowledge repositories increase, such hybrid approaches have a great potential for solving the problem of word sense disambiguation. The approach presented in this dissertation is a hybrid method because it takes advantage the UMLS Metathesaurus as a knowledge repository for the purposes of identifying training examples for the language modeling as well as a controlled vocabulary to determine the sense inventory for terms to be disambiguated.

2.2 Sublanguage Theory

Human language is very flexible to accommodate a wide range of communication purposes, including fairy tales and entertaining riddles. Many words can take a large number of meanings, making computerized language processing challenging. Zellig Harris, an American linguist, observed that the restricted use of language in the discourse of specialized domains placed strict limitations on the distribution of word classes and their co-occurrences. Harris determined that knowing these distributions can aid in determining the most appropriate meaning of terms within the boundaries of a specific domain.

Previous research has established that semantic and syntactic rules differ for narratives that come from different specialized domains. Such closed-matter subjects are characterized by a limited vocabulary, a relatively small set of word classes, and word-class sequences integrated as a sublanguage [39]. Although the specific word-recurrences in the successive sentences of a discourse are unique to that discourse, various types of co-occurrence patterns

seem to characterize various types of discourses. The various types of word co-occurrence are worth studying as the inherent carriers of various information types. And the particular pattern of word co-occurrence in a given discourse or section is useful as a framework of the particular information and information processing in that discourse [40]. Since sublanguage theory was first introduced, there have been multiple attempts to implement sublanguage principles in computer applications. The distinguishing characteristics of such an approach are performing WSD through semantic type disambiguation, which involves identifying a word class (or semantic type) for each ambiguous term and then selecting a concept that belongs to that word class out of a list of potential concepts [41]. This approach is based on selectional preferences or restrictions [9, 31, 42, 43].

A number of domains have been analyzed via sublanguage models, such as trouble tickets [44], technical maintenance manuals [45], stock market reports [46], and weather reports [47]. The work to produce the first computerized application based on sublanguage theory started in 1965 and resulted in the Linguistic String Project (LSP) [48]. That project is based on the information formats for the content of text in a given domain. It started as an attempt at computerized processing of scientific text, based on the algorithm developed by Sager (N. Sager, Procedure for left-to-right recognition of sentence structure, T.D.A.P. No. 27, University of Pennsylvania, 1960) and theoretically grounded in Linguistic String Theory suggested by Zellig Harris [49]. This theory states that any sentence can be built from the center string by adjunction, conjunction, and replacement. The center string is a sequence of *noun+tensed_verb* or *noun+tensed_verb+noun*. However, not all combinations of word categories result in a valid sentence due to a number of restrictions. The earliest full-text accessible article about LSP is by Grishman [50]. He states that in 1973, LSP was under development for 8 years and the current version at the time was version 3. The grammar used by the parser consisted of:

1. a Backus-Naur Form context-free language grammar implemented as a set of elementary strings together with rules for combining them to form sentence strings,
2. a set of restrictions on those strings; and
3. a word dictionary, listing the categories and subcategories for each word.

Another early publication is by Sager in 1975 [51], where she discussed the hypothesis that the literature of science domains has certain restrictions on language usage. These restrictions were formalized as information formats, which are repeating patterns of the word classes (also called semantic types, term classes, or word categories) and word class

relations in sentences of the text. Word classes were obtained by grouping words or phrases that occur within similar grammatical relations. The information formats contained slots for particular types of information. Sentences of the text of specific domain were identified as instances of the corresponding format. The set of formats was considered to be a sublanguage grammar. Each slot had a fixed informational content and the sentences of certain format carried specific types of information. The slots were based on the hierarchy of grammatical operators and operands; they were not determined solely on the linear sequence of words in sentences.

The main premise of sublanguage grammar is that narrow domain grammar rules are more restrictive than English grammar rules. A sentence may be well formed in general English but not well formed as a sentence in the specific domain. In the beginning of LSP, the researchers established that the semantic classes of words do not have to be specified a priori but can be extracted through the process of grouping terms that appear in the same co-occurrence patterns.

2.3 NLP in Clinical and Biomedical Domains

Since the early years of research in the natural language processing of English, newspaper and scientific literature have been the primary target languages. As a result of multiple research projects, a large number of disambiguation methods have been proposed and a large body of language samples have been annotated and made available for shared use. Availability of shared annotated corpora enabled new system evaluation and algorithm comparison. Despite such advances in main-stream NLP technology, its penetration into the clinical domain has been limited to a few research projects and a handful of commercial systems. The reason for this situation is the difference among lexical, syntactic, and semantic characteristics of clinical text and general language.

Clinical language shares some of the features of other telegraphic sublanguages, such as ill-formed and reduced sentences, lack of internal consistency, abundance of overloaded abbreviations and acronyms, misspellings, and extra linguistically-meaningless tokens resulting from local and individual practices [1, 52]. The language of biomedical literature shares some characteristics of clinical language, such as a large vocabulary of terms that are virtually exclusive to the medical domain, but it also resembles general language, because of the use of proper grammar and wide availability of shared corpora. Because of these differences, clinical language and the language used in biomedical literature are distinct

sublanguages [53]. A number of systems have been created to process narrative stored in electric format. Some of them are general and some of them are project-specific, created either as a research project or implemented in a single organization.

2.3.1 Systems Based on Sublanguage Principles

Over the last 50 years the sublanguage theory has been used as the theoretical framework for a number of different systems that have been developed and implemented for a clinical and biomedical domain.

The Medical Language Processing (MLP) system is the first attempt to apply the LSP parser to medical text, initially reported in 1976 [32]. This effort was conducted by a research team that included Naomi Sager, Ralph Grishman, Ngo Nhan, and Carol Friedman. The target corpus included x-ray reports on patients with breast cancer. For that system, word classes and information formats were derived through a distributional analysis on the parsed sentences to obtain word classes and on the word classes to define formats. The distributional analysis starts with identifying words frequently occurring in the same syntactically defined environments. These words become the core of the new class. Then the environments of these words are enumerated and new words for the class are identified. Only one format was initially determined. The final version of the system had additional 11 formats. During the first attempt for MLP, 176 out of 188 sentences (94%) were successfully formatted. The MLP system is designed to perform linguistic string analysis to determine the sentence structure, regularization of the sentence structures though general English transformations, and mapping of transformed parsed sentences into format slots.

Another major concept-mapping system based on the sublanguage theory is Medical Language Extraction and Encoding system (MedLEE) developed by a team led by Carol Friedman. [54]. The grammar rules employed by MedLEE were developed manually, based on the distributional analysis of clinical notes of a specific note type - chest x-ray reports. Modifying the knowledge base to accommodate new domains required a substantial human effort. So resolving the knowledge base acquisition bottleneck by making the process automatic would simplify domain adaptation natural language processing systems that use sublanguage restriction rules for semantic disambiguation. Friedman first described the conceptual model for the MedLEE project in 1994 [55]. The model was designed by analyzing chest x-ray reports generated at Columbia Presbyterian Medical Center (CPMC). The model included four conceptual levels: *a*) the structure of the report; *b*) the findings

in the report; *c*) the structure of the medical concepts that make up the findings; *d*) the lexical information associated with individual words and multi-word phrases.

The clinical terms and their semantic types were defined in the Medical Entities Dictionary developed at CPMC. The initial analysis included 8000 chest x-ray reports. Once the conceptual levels have been defined, the prototype was implemented [55]. Initially, the semantic lexicon contained 3120 terms with associated semantic types. The semantic grammar contained 350 grammar rules. A first *proof-of-concept* study used 230 reports. Two person years were required to create the first semantic grammar. Later the semantic grammar was extended to cover mammography, discharge summaries, all of radiology, electrocardiography, and pathology [56]. During the system adaptation project, the MedLEE developers concluded that creating a system that can be equally effective on text of different domains required obtaining additional rules that would enable and disable other grammar rules based on the target clinical subdomain. As the number of covered subdomains grows, maintaining the rules might become cost prohibitive.

2.3.2 Biomedical Language Processing Systems

Domain specific vocabulary and a limited set of word categories as main characteristics of sublanguages have been successfully applied for word sense disambiguation in the biomedical domain. UMLS has been the knowledge base of choice for most NLP systems. The UMLS Metathesaurus provides a large vocabulary of medically relevant concepts and the UMLS Semantic Network provides a relatively small list of word classes (or semantic types) that are applicable to the biomedical domain. Similar to UMLS Metathesaurus, another commonly used controlled vocabulary is Medical Subject Headings (MeSH). To aid the development of new medical NLP systems, the National Library of Medicine (NLM) sponsored development of a manually-annotated text collection for the purposes of training and testing word sense disambiguation [57].

A number of projects focused on processing biomedical text. Rindflesch and Aronson developed a set of rules that determined the semantic type of the term depending on the patterns of neighboring words and semantic types within the sentence. This set of rules was applied to a small set of instances and achieved 78% disambiguation accuracy [58]. Expanding on Rindflesch and Aronson's idea, Krauthammer and Nenadic suggested performing word sense disambiguation as a two step process - term classification and term mapping. The goal of term classification is to label the term of interest with one of a small

number of semantic categories using a machine learning model, which was acquired using annotated text. Once the semantic category is identified, the term mapping step arrives to the final match between the term and a concept from a controlled vocabulary such as UMLS Metathesaurus [14]. Similarly to Krauthammer and Nenadic's approach [14], Fan and Friedman successfully exploited UMLS resources to perform word sense disambiguation through semantic type classification [41].

The idea of semantic type labeling as a step to concept recognition is further developed by Humphrey and colleagues [59]. They used Journal Descriptor Indexing (JDI) as a straightforward way to identify sublanguages within biomedical domains. The main assumption is that publications with the same JDI belong to the same sublanguage. Semantic type labeling is implemented by adjusting the likelihood of occurrence of a concept with a specific semantic type depending on the set of journal descriptors that are associated with the neighboring words. The average disambiguation precision was reported at around 78%.

Stevenson and Guo developed a hybrid WSD system that combined lexical features (such as lemmas of ambiguous words), syntactic features (such as part of speech), collocation features (such as combination of other features in ngrams), and knowledge-based features (such as UMLS identifiers and MeSH terms). Using those features, the Naive Bayes and Support Vector Machine models were tested on the NLM test collection and term disambiguation accuracy of 89.7% was achieved [60]. Similarly, a system developed by Liu et al. [61] creates a disambiguation model by learning a Naive Bayes classifier using a feature space consisting of stemmed words that appear in each evaluated abstract. The method used a one meaning per discourse assumption to aid disambiguation.

2.3.3 Clinical Language Processing Systems

Sublanguage approach is not the only method used for NLP of clinical text. Several commercial, open access and research applications have been developed. One of the earliest systems was the special purpose radiology understanding system (SPRUS) designed to encode salient features from chest X-ray reports and implemented as a module in the Health Evaluation through Logical Processing Hospital Information (HELP) medical expert system at the LDS Hospital in Salt Lake City, UT [62]. Another NLP tool developed within the same organization is SymText, which uses Bayesian networks to model the context of radiological reports in order to automate coding tasks [63]. Chest radiology reports were also the initial target domain of another HELP module, a probabilistic medical language

understanding system called MPLUS [64]. It uses Bayesian networks to represent the basic semantic types and relations in order to infer the most probable concepts consistent with the words found in a sentence. Using MPLUS as the starting point, the Automated Problem List (APL) system was designed to extract medical problems from electronic free-text documents [65].

Mayo *clinical Text Analysis and Knowledge Extraction System (cTAKES)* is a pipeline system designed by Savova for the purpose of phenotype extraction from clinical notes [66]. It was built on publicly available technologies, such as UIMA framework, OpenNLP and the SPECIALIST Lexical Tools. The system annotates text with several clinical named entities, such as drugs, diseases/disorders, signs/symptoms, anatomical sites, and procedures. Each named entity has attributes for the text span, the ontology mapping code, whether the named entity is negated, and the context (family history of, history of, probable). The system has been submitted to the Open Health Natural Language Processing Consortium (OHNLP) and can be freely downloaded.

Medical Knowledge Analysis Tool (MedKAT/P) is another freely available tool donated to the OHNLP by IBM [67]. This modular and flexible system based on UIMA framework is designed to extract structured information from narrative text in the clinical pathology domain such as pathology reports, clinical notes, discharge summaries and medical literature. The system labels text with concepts such as primary tumor or lymph node status and a number of cancer-specific characteristics such as histology, anatomical site, nodes dimensions and sizes, number of positive and excised nodes. MedKAT/P incorporates NegEx algorithm developed by Chapman to identify negation status of the concepts [68].

Health Information Text Extraction (HITEx) was initially specific to a research study on airway diseases such as asthma and chronic obstructive pulmonary disease. Now it is used as a general purpose NLP "cell" module in the i2b2 "hive" architecture. The main functionality of the system is to extract principal diagnoses, co-morbidities, and smoking status. The knowledge base for the system includes a set of manually designed regular expressions, as well as machine learning models trained on a corpus consisting of discharge summaries of the patients that had one or more related admission diagnoses defined by ICD9 codes [69].

Most of these systems have been developed within a single organization. Informatics for Integrating Biology and the Bedside (i2b2), an NIH-funded National Center for Biomedical Computing (NCBC) based at Partners HealthCare System in Boston, has promoted collab-

oration by organizing a series of NLP challenges and shared tasks. These challenges tackled the problems of de-identification [70], obesity and co-morbidities extraction [71], smoking status [72], and clinical concept extraction from clinical text [73, 74].

2.4 Project Statement

Using semantic type information has been successful in aiding the word sense disambiguation process as applied to both biomedical literature and clinical text. Similarly, limiting the scope of the NLP system also has been shown to boost the system's performance by limiting language variability. In combination, the limited system scope and semantic grammar rules have a potential to enable language processing of even the most irregular and idiosyncratic language. However, the knowledge base acquisition for such a system would be challenging due to the lack of training data. A successful WSD tool, once created, produces satisfactory results on text that is similar in syntactic and semantic characteristics to the text that was used to build it. However, performance of even the best WSD tool will inevitably decrease if the tool is applied to a text with syntactic and semantic characteristics that are different from the source text. Improving tool performance on a new text often involves either adding new semantic rules or retraining the statistical model on a new set of annotated texts. The process of new domain adaptation of a WSD tool is expensive and time consuming because it involves human experts [75]. Making the process of WSD domain adaptation automatic would increase the tool's portability across domains. The current project demonstrates the variability of clinical language and suggests a model of dealing with this variability automatically using available knowledge resources.

2.5 Resources

2.5.1 Computational Resources

This project involves analysis of a large number of original clinical notes that have not been de-identified. Due to the amount of processing that was required, as well as in order to comply with privacy and security regulations, a powerful and secure environment was needed. I utilized a new secure, HIPAA-compliant, high-performance compute cluster located at the Center of High Performance Computing.

2.5.2 Corpus

The complete set of all clinical narrative types at our medical center (a large tertiary care teaching hospital) in use during the period January 2007-December 2008 was analyzed

by a clinical expert to determine a study subset that was diverse across domains. Note types that consisted mostly of templated information, scanned hand-written documentation, or nonclinical documents were excluded. As a result, a set of 17 representative note types were selected for this study. These note types represented a cross-section of clinical narratives created by clinical personnel that varied by clinical role (physicians, nurses), specialty (cardiology, dermatology, ob/gyn, oncology, etc.), and clinical environment (ED, inpatient, outpatient). A set of 683,061 notes was extracted from the University Hospital Electronic Data Warehouse. Files that were less than 100 bytes in length were excluded because most of them did not contain clinically relevant information. The remaining 559,029 files were processed by the MetaMap. Only 557,571 of those files were successfully processed. In addition to the clinical narratives, a random set of 35,000 MEDLINE abstracts published between 2000-2008 was selected. To ensure a valid comparison to the clinical texts, abstracts less than 100 bytes and those that failed to be processed were excluded. The full list of note types and their file counts is presented in Table 2.1.

Table 2.1: The note types and the corresponding file counts used in this project.

Note Type	**Abbr.**	**File count**	**Attempted to process**	**Processed successfully**
Admission HP	AHP	51,721	43,142	42,911
Ambulatory Nursing Note	ANN	77,542	73,196	73,167
Burn Clinic Note	BCN	13,430	13,428	13,428
Cardiology Clinic Note	CCN	24,366	24,306	24,302
Case Mgmt Dschg Plan Note	CMD	30,213	30,141	30,046
Dermatology Clinic Note	DCN	6,251	6,250	6,249
Discharge Summary	DIS	65,256	65,220	64,530
Emergency Dept Report	EDR	106,250	685	685
Family Practice Clinic Note	FPC	11,626	11,270	11,233
Hematology Oncology Clinic Note	HOC	36,785	36,769	36,760
Neurology Clinic Note	NCN	24,137	23,944	23,634
Obstetrics Gynecology Clinic Note	OGC	9,355	9,289	9,277
Operative Report	OPR	76,593	76,556	76,552
Orthopaedic Clinic Note	OCN	119,094	115,655	115,654
Plastic Surgery Clinic Note	PSC	4,375	4,371	4,371
Rheumatology Clinic Note	RCN	22,647	21,393	21,358
Social Service Note IP	SSN	3,420	3,414	3,414
Total number of files:		683,061	559,029	557,571

2.5.3 MetaMap

MetaMap is a powerful concept recognition system developed by a team led by Aronson at the National Library of Medicine. Its primary aim is to map terms found in abstracts of MEDLINE citations, as well as user queries to concepts in the UMLS Metathesaurus. A recent comprehensive overview of MetaMap system is presented elsewhere [76].

Its comprehensiveness, robustness, free availability, and regular updates with the latest version of the UMLS make MetaMap very attractive for potential NLP users. However, in spite of the good coverage of the clinical domain by the UMLS Metathesaurus, MetaMap has not been applied widely to the clinical domain beyond a few research projects. The main deterrent to broad application of MetaMap on clinical narratives is its failure to perform accurate word sense disambiguation. When a term from free text matches multiple UMLS concepts, MetaMap returns a list of all mappings, making information extraction ineffective. If MetaMap's WSD algorithm is used on clinical narratives, it often selects the wrong concept because it was trained on biomedical text. The result of MetaMap processing is an XML output file that specifies sentence, phrase, syntax unit, and token boundaries, the part of speech and syntax type of each syntax unit, as well as a combination of concepts from UMLS Metahesaurus. Along with the UMLS concept identifier, the XML file has UMLS preferred concept name, and one or more corresponding Semantic Types (STs) for each concept. For my project I used the version that was the latest at the time when I started processing the data - MetaMap binary 2009 V.2 [77].

CHAPTER 3

SUBLANGUAGE

Natural Language Processing systems employed in the clinical domain operate under one of two main assumptions about clinical language: 1) the narrative of patient notes constitutes one sublanguage, or 2) each clinical subdomain imposes its special set of selectional restrictions that aid concept recognition. The examples of the systems built with the first assumption are cTAKES and HITEx. The design of cTAKES is based on the reference standard that included 273 manually annotated clinical notes of a range of note types - consult notes, discharge summary, educational visit, general medical examination, limited exam, multisystem evaluation, reports, specialty evaluation, dismissal summary, subsequent visit, therapy, and notes of general category - miscellaneous. The goal of creating such a mixture of text samples was to ensure that all areas of the clinical domain were covered by the disambiguation model [66].

As opposed to cTAKES, the HITEx system uses a reference standard corpus of 150 discharge summaries because the language and topic variability is believed to be an accurate representation of the variability across all subdomains [69]. A commercially available system, LifeCode, targets a large variety of notes but manages its performance by limiting the tasks that it can perform [78]. Another system, called KnowledgeMap, incorporates a range of clinical notes and medical textbook text in an attempt to create a general purpose knowledge base [79]. The main benefit of treating clinical language as a unified sublanguage is the relative speed of system development.

A very different approach to clinical language system development is designing systems for a specific clinical subdomain or note type. The Medical Language Extraction and Encoding System (MedLEE) is a successful and widely used general purpose system built with the assumption of language variability by note type. The initial system design was based on chest x-ray reports. When the system's use was expanded to include other types of clinical notes, the system's knowledge base was modified. However, instead of simply expanding the knowledge base to include additional semantic categories and terminology, a

set of context-dependent switches was developed that would turn on or off certain grammar rules determined by the clinical subdomain where the system is used [53]. This setup made the system highly accurate for concept recognition in some clinical subdomains, but increased the financial and time cost of system adaptation to a new target subdomain. Similar to MedLEE an array of research projects and system development efforts were conducted under an assumption of language variability. The most common way to deal with the idiosyncrasies of the narrative across domains is to specify the exact task or the clinical subdomain that the system targets and avoid making assumptions of possible system performance outside of those boundaries. The developers of Medical Information Extraction System (MedEx) describe the purpose of the system to be mapping of medication information into a structured representation. Even though the system was trained using only Discharge Summaries, the authors claim wide applicability of the system due to the narrow scope of the task [80]. Both of these assumptions are largely untested. Therefore, as the first step in my research project, I am attempting to identify the boundaries of clinical sublanguages. The purpose of such analysis is to inform future system developers when they are making a decision about their system's scope and coverage. For example, if precision in the system's performance is the top priority, then the developers will be compelled to limit the coverage of their system to only one specific note type. Without additional knowledge about the sublanguage boundaries, it would be impossible to predict the potential system performance degradation when it is applied in a different setting.

3.1 Methods

According to the traditional definition of sublanguage grammar outlined by Harris, the languages of different narrow domains differ in their lexical component - vocabulary, and in their semantic component - semantic types and semantic type patterns distributions [40]. Therefore, I approach the sublanguage boundary definition at two levels - lexical and semantic. To show that clinical language is not homogeneous at a lexical level, I use unsupervised document clustering analysis (reported in [81]). The semantic level variability is demonstrated through semantic type pattern distribution analysis (reported in [82]).

3.1.1 Document Clustering

Document clustering is a common unsupervised machine learning method of binning analyzed documents into groups treating each document as a single entity. Other approaches

that I considered for the task of identifying sublanguage boundaries were document classification, topic discovery, and latent semantic analysis.

Document classification is a supervised method of organizing documents by assigning one of several predefined categories to each document. The steps for such analysis would include manual annotation of a set of representative documents that would then be used for training of a machine learning classifier. Once a classifier is acquired, it can be used to label a test set of documents with a category. The output of this analysis would indicate whether all notes of the same note type fall into the same category or not [83]. This method assumes that human experts can correctly discern language variations to inform the classifier. However, due to natural capacity limitations, humans are not able to perceive hidden patterns from large amounts of data, and therefore, besides the obvious cost associated with human annotations, the supervised approach is inferior to computational methods of pattern discovery in data.

Similar to document clustering, *topic discovery* methods identify documents that share a similar content using unsupervised techniques. However, topic discovery labels documents with a set of possible topics, whereas, document clustering labels each document with a single label, thus simplifying the grouping structure [84].

Like document clustering, *Latent Semantic Analysis* uses term frequency weights for words used in each analyzed document. Also, as with document clustering, LSA evaluates word usage patterns for all words across all analyzed documents. However, unlike document clustering, LSA focuses on individual words and their meaning, thus the LSA hierarchical model is directed at learning the relationships between words, rather than documents. Hierarchical LSA methods allow visualizing word relatedness, rather than document relatedness, which is beneficial for word sense disambiguation but not as a sublanguage similarity measure [85].

After considering other options, I chose document clustering as a method of grouping related documents because it does not require human annotation to inform the algorithm, and because hierarchical document clustering methods allow not only identifying what types of sublanguages are present in the corpus, but also the strength of sublanguage similarity. Document clustering is a commonly used unsupervised text mining technique that has been used for a range of natural language processing tasks such as information retrieval, question answering and others. The goal of document clustering is to find a set of "natural" patterns within a large amount of unlabeled data inside the documents and then to organize similar documents into groups using some measure of similarity [86]. Cluster analysis typically

consists of *a*) feature selection and extraction; *b*) selection or design of a clustering algorithm; and *c*) cluster quality evaluation [87].

The most popular data set format for document clustering is a bag-of-words vector-space model. This method represents the entire set of documents as a $T \times D$ matrix, where T is the size of the vocabulary used in the document set; and D is the total number of documents in the data set [88]. Each document is represented as a vector of length T, and since most terms do not appear in any given document, these vectors are sparse. Typically, each value in these vectors represents the importance of the particular term (t) in the particular document (d). In order to minimize the effect of the document size and extremes in the frequency of a specific term, the well known "term-frequency inverse-document-frequency (tf-idf)" measure is often used as the weight of each term in a document vector [89, 90]. This measure takes into account how frequent a specific term is within a specific document as well as the term distribution across all documents in the analyzed corpus. Thus, terms that appear only in a few documents have higher weights, but terms that appear in most documents will have lower weights.

The goal of cluster analysis is to place each document into one of K disjoint or overlapping clusters. Each cluster usually is defined by its centroid, which is the most representative vector in the cluster. Depending on the clustering algorithm used, the centroid can be either the average point for each dimension of the feature vector, or an actual point in the data set that is the closest to the average point. Most clustering algorithms have three main components: *a*) similarity measure, used to measure vector relatedness; *b*) clustering method, the computational approach taken during the clustering process; and *c*) clustering criterion function, which is used for the optimization of the final clustering solution [91].

Similarity between two vectors can be measured by calculating a Euclidean distance, a cosine distance, or a correlation coefficient. The general clustering method can be either partitional, agglomerative, density-based, or grid-based. Depending on the final specific solution desired, the clustering methods can be either hierarchical or nonhierarchical [92]. The simplest and most widely used clustering method is K-means. Prior research concluded that a bisecting, K-means algorithm performs quite well despite its simplicity and lower computational complexity [93]. This hierarchical algorithm iteratively splits the data set until the predefined number of clusters is reached. Selection of a clustering criterion function influences the final clustering solution by putting more emphasis on cohesion or on separation of the resulting clusters.

The measure of cluster quality can be classified as either internal or external. Internal measures of cluster quality aim to assess how closely the elements in each cluster are related to each other, evaluating "cohesion" and "separation" of the clustering results. Cohesion can be measured as the average similarity of the members of the cluster to each other or to the cluster centroid. Separation evaluates the average dissimilarity of the members of a particular cluster to all other elements in the data set.

The external measures rely on knowing a true label of each of the documents. Clustering output can be measured externally in terms of purity and F-score. Purity is the proportion of each cluster that consists of the majority class. F-score evaluates precision and recall of each document type with respect to its cluster assignment. In evaluation of document clustering output, precision for each document type compares the largest number of documents that are assigned to a specific cluster to the total number of documents assigned to that cluster.

Recall for each document type compares the fraction of the largest group assigned to the same cluster to the total number of documents of that type. The F-score is a harmonic mean of precision and recall. An optimal clustering solution will have 100% purity, which means that each cluster contains elements that belong to a single class [91]. Such purity can be achieved trivially when the number of clusters is equal to the number of elements in the data set. On the other hand, the perfect F-score will be achieved only if all documents of each type are grouped into a single cluster (100% recall) and no document types share a cluster label (100% precision). Using the note type as the true class labels, I exploit purity and F-score measures in our analysis below.

A feature vector file was created where each note was represented by the tf-idf value for each term that MetaMap matched to at least one UMLS concept. To decrease the feature space, multiword phrases were split into individual tokens and the base form of all tokens was obtained from SPECIALIST lexicon using the Norm tool [94]. In addition to the lexical attributes, semantic types of those terms that were unambiguously mapped to a UMLS concept by MetaMap were also used as attributes. The derivation of what constitutes an unambiguously mapped term is more complex than simply choosing those terms with only one MetaMap semantic mapping. Those terms can be enriched with an algorithm that exploits the mapping scores provided by MetaMap as described in Section 4.2.1.1. Using only those terms that MetaMap successfully mapped to at least one concept minimizes the size of the feature vector and focuses on only those tokens that are potentially relevant in the clinical setting, thus excluding misspellings, unrecognized locally specific abbreviations,

and other language characteristics, which are artifacts of the local practices rather than being typical of the clinical subdomain.

To perform clustering I chose bisecting k-means clustering using a cosine similarity measure with the "internal criterion function," which maximizes similarity between each document and the centroid of the cluster that it is assigned to. The clustering tool I chose was the CLUTO clustering toolkit [91]. This software package offers a set of clustering algorithms that approach clustering as an optimization process aiming to minimize or maximize the selected clustering criterion function. It is written in C, and thus is quite fast. It also manages memory well. The Java-based Weka cluster toolset was unable to process the full feature space, and was too slow to be practical for even small subsets. The selected clustering algorithm requires the number for clusters to be specified a priori. In the current study, each clustering experiment used the same number of clusters as the number of the analyzed note types. The full available corpus contained a variable number of documents for each note type. Since the selected algorithm is the most accurate when the number of documents in each class is the same, the corpus was reduced to 3000 randomly selected documents of each note type, except Emergency Department Reports that had only 685 documents available.

My initial experiment using a subset of 685 documents of each type (i.e., the size of the smallest note type, Emergency Department Reports) clustered into 18 clusters resulted in 74.8% average cluster purity. Analysis of the most descriptive and discriminating features (produced optionally by CLUTO) showed that several provider names in one type produced an unwarranted impact on clustering. After these names were identified, the feature vectors were recalculated and new clusters were analyzed.

Review of the most important features showed that clinically irrelevant words, such as "phone" and "fax" were responsible for inflated cluster purity for Case Management Discharge Plan, thus skewing the clustering results. The results of these two experiments led me to a conclusion that in order to acquire the most reasonable clusters, I had to exclude the lexical noise that resulted from the artifacts of the local practices and templates. So I manually designed a short stopword list that consisted of the most frequent person names and also the words "phone" and "fax." This stopword list also included five semantic types that were identified as the most common for all note types [82]. These semantic type are: Findings, Temporal Concept, Qualitative Concept, Quantitative Concept, and Functional Concept.

After those stopwords were excluded, the new data set was analyzed and the average cluster purity of the resulting solution dropped to 73.3%. This confirmed that the artifact terms were artificially improving the clustering for some note types, for example, terms that occurred frequently in section headers. To eliminate noisy terms more systematically, I calculated an additional set of stopwords that aimed to reduce the lexical artifacts for all the note types. The new stopword list excluded any term in a specific note type if that term appeared in more than 95% of all documents of this note type. These terms were eliminated from the feature set for that particular note type but not for the other note types.

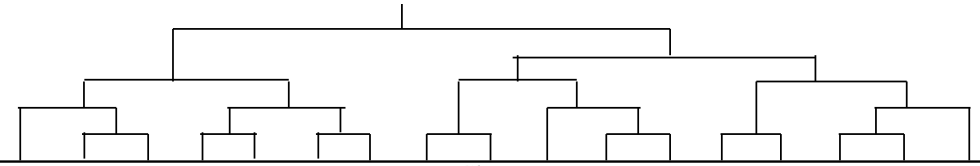
Processing the new data set resulted in an even lower average purity of 70.0%. Even though eliminating artifacts of the local practices resulted in lower cluster purity, I believe that by doing so I achieved clustering that more faithfully reflects the lexical patterns of the analyzed clinical *subdomains* rather than lexical noise due to local practice. Purity is calculated in terms of the majority class for each cluster and reflects how well each cluster is represented by one of the document classes. Lower purity indicates that the cluster contains notes of different classes, thus showing that those document classes have some documents that are lexically related among each other. For example, Table 3.1 shows that cluster 13 mostly has documents from three note types - Ambulatory Nursing Notes, Case Management Discharge Plan, and Emergency Department Reports. On the other hand, cluster 6 is mostly represented by documents of a single note type - Rheumatology Clinic notes. When comparing the cluster assignment for Discharge Summaries and Admission History and Physical, it is notable that out of 18, clusters 8 have similar counts of these note types. This is indicative of the large overlap in the lexical and semantic patterns appearing in the documents of these note types. The next set of experiments evaluated the effect of larger sample size on clustering. Emergency Department Reports had only 685 notes available to us, so they were excluded from further processing. The feature vectors representing the remaining sixteen note types and MEDLINE abstracts with 3,000 documents in each set were clustered into seventeen groups. As Table 3.2 illustrates, most document types were grouped each in its own cluster.

Several note types are shown to be more general than others, such as, Admission History and Physical, Ambulatory Nursing Notes, Discharge Summary, Family Practice Clinic Notes and, not surprisingly, MEDLINE abstracts. Case Management Discharge Plan, Dermatology Clinic, and Plastic Surgery Notes exhibited a dichotomy in the lexical patterns. As the cluster hierarchy shows, despite such a split, each pair of the clusters are

Table 3.1: Clustering results of the data set consisting of 685 documents per document type. Values that represent less than 1% of the total note type count were excluded for visual clarity.

Document Type	Clusters																		Total	Recall	Precision	F-score
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17				
Admission History Physical				15		13				103	59				46	112	39	277	685	0.17	0.4	0.24
Ambulatory Nursing Note		56	295		12	16							13	158			84	18	685	0.93	0.43	0.59
Burn Clinic Note						669										7			685	0.85	0.98	0.91
Cardiology Clinic Note				300						375									685	0.43	0.55	0.48
Case Mgmt Dschg Plan Note			8		496								11	166					685	0.92	0.72	0.81
Dermatology Clinic Note	196							468											685	0.92	0.68	0.79
Discharge Summary				14		13			14	60	91		11		49	94	132	197	685	0.12	0.29	0.17
Emergency Dept Reports						60		7		270	12			207	9	72		21	685	0.31	0.39	0.35
Family Practice Clinic Note	7							9		15	37		8	10		41	529		685	0.62	0.77	0.69
Hematology Oncology Clinic									7						14			650	685	0.4	0.95	0.57
MEDLINE abstracts		11		10	12				9	33	20	15	45		65	14	18	415	685	0.26	0.61	0.36
Neurology Clinic Note															666				685	0.76	0.97	0.85
Orthopaedic Clinic Note												8				655			685	0.6	0.96	0.74
Obstetrics Gynecology Clinic											600	7		43				16	685	0.71	0.88	0.79
Operative Report		111							9		20	532							685	0.89	0.78	0.83
Plastic Surgery Clinic Note									579					14		73			685	0.91	0.85	0.88
Rheumatology Clinic Note						9	661												685	0.96	0.97	0.96
Social Service Note IP													641				37		685	0.86	0.94	0.89
Cluster size	222	178	318	349	540	791	689	507	636	869	843	600	750	608	874	1085	855	1616	12330	0.65	0.73	0.66
Cluster purity	0.88	0.62	0.93	0.86	0.92	0.85	0.96	0.92	0.91	0.43	0.71	0.89	0.86	0.34	0.76	0.60	0.62	0.40	0.70	--	--	--

Table 3.2: Results of hierarchical cluster analysis of the set of 17 document types (n=3,000 notes per set). The values are the counts of all documents of the particular type that were grouped into each of the clusters. Values that represent less than 1% of the total note type count were excluded for visual clarity.



Document types	Cluster ID 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Total	Precision	Recall	F-score
Admission History/								316	315	322	150	1367			469			3000	0.22	0.46	0.29
Ambulatory Nursing Note	120	72	289	353	1580				77			282		61		46		3000	0.96	0.53	0.68
Burn Clinic Note												38		2877	39			3000	0.94	0.96	0.95
Cardiology Clinic Note										2963								3000	0.80	0.99	0.88
Case Mgmt Dschg Plan	33	1729	1194															3000	0.90	0.58	0.70
Dermatology Clinic Note						1988	919											3000	0.95	0.66	0.78
Discharge Summary	36							408	137	221	80	1729			331			3000	0.27	0.58	0.37
Family Practice Clinic	38					32		166			39	2446			167			3000	0.39	0.82	0.52
Hematology Oncology									2907		49							3000	0.55	0.97	0.70
MEDLINE abstracts	131							102	1718	115	365	254			63	116		3000	0.33	0.57	0.41
Neurology Clinic Note	37										2887							3000	0.80	0.96	0.87
Orthopaedic Clinic Note															2879			3000	0.64	0.96	0.77
Obstetrics Gynecology								2752	63			63						3000	0.71	0.92	0.80
Operative Report								81							37	2790	52	3000	0.89	0.93	0.91
Plastic Surgery Clinic														55	412		2425	3000	0.93	0.81	0.87
Rheumatology Clinic Note												31	2891					3000	0.97	0.96	0.97
Social Service Note IP	2926	53																3000	0.86	0.98	0.92
Cluster size	3387	1913	1568	368	1645	2099	993	3854	5282	3706	3626	6325	2971	3077	4466	3124	2596	51000	0.71	0.80	0.73
Cluster purity	0.86	0.90	0.76	0.96	0.96	0.95	0.93	0.71	0.55	0.80	0.80	0.39	0.97	0.94	0.64	0.89	0.93	0.76	--	--	--

closely related, indicating similarity between the clusters. Increased sample size and removal of a more general document set (Emergency Department Reports) resulted in increase of the average purity to 76%.

The general note types, which are not specific to any clinical subdomain, span different topics and were excluded for the next experiment. I processed the new data set consisting of the documents of those 12 note types, which are more focused on a specific clinical subdomain. The resulting 12 clusters had an impressive level of purity, 95.5%. Average F-score was also 95.5% (Table 3.3). This indicates that the overwhelming majority of the notes of each note type exhibit lexical patterns that are characteristic of that note type. Analysis of a slightly lower F-score for Orthopedic (OCN) and Plastic Surgery Clinic Notes (PSC) and Operative Report (OPR) indicated a topic overlap for a portion of these notes as pointed out by the descriptive features for cluster 12 (Table 3.3), which are {fracture, orthopedics, motion, knee, splint, radiographic}.

3.1.2 Semantic Pattern Distribution

A domain specific set of sentence types is one of the main characteristics of sublanguage grammar definition outlined by Harris [95]. According to Harris, the more specialized a domain, the smaller the set of semantic type structures that are common in the narrative of that domain and that are designed to carry a specific type of information. Harris's sublanguage definition of semantic sentence structure links the semantic role relationships between words in sentences, such as predicate-argument relationships, with the semantic types of the words. For example, in a statement "Patient reported pain" the word "patient" has semantic type "Patient group" and the word "pain" is of the "Sign or Symptom" semantic type. In terms of semantic roles, the predicate is the verb "reported", "patient" is the subject, and "pain" is the object of the sentence. Thus, the semantic structure that can be derived from this statement for verb "reported" is that the object is "Patient group" and the subject is "Sign or Symptom." All semantic structures have a set of paraphrastic patterns, because the same information can be carried out in various physical forms. Therefore, the same semantic sentence structure can be expressed by different linear word sequences. (Such as "Patient reported pain" and "Pain was reported by patient"). The full set of form and content relations in sentences of a specific domain can be expressed as a distribution of linear sequences of semantic types (or semantic type patterns) in sentences within that domain. For the purposes of such analysis, I created a set of semantic pattern

Table 3.3: Results of clustering 12 note types with 3000 documents of each type. Values that represent less than 1% of the total note type count were excluded for clarity.

Document Types	Clusters												Total	Recall	Precision	F-score
	1	2	3	4	5	6	7	8	9	10	11	12				
Burn Clinic Note	2897						41						3000	0.97	0.97	0.97
Cardiology Clinic Note		2977											3000	0.98	0.99	0.99
Case Mgmt Dschg Plan Note			2934									43	3000	0.98	0.98	0.98
Dermatology Clinic Note				2895									3000	0.99	0.97	0.98
Hematology Oncology Clinic					2921	50							3000	0.96	0.97	0.97
Neurology Clinic Note						2901						37	3000	0.97	0.97	0.97
Orthopaedic Clinic Note							2902						3000	0.82	0.97	0.89
Obstetrics Gynecology Clinic					53			2860					3000	0.97	0.95	0.96
Operative Report							106	64	2741	51			3000	0.95	0.91	0.93
Plastic Surgery Clinic Note	43						431			2450			3000	0.96	0.82	0.88
Rheumatology Clinic Note											2914		3000	0.98	0.97	0.98
Social Service Note IP												2980	3000	0.95	0.99	0.97
Cluster size	2991	3037	2994	2933	3053	3006	3526	2943	2880	2548	2962	3127	36000	0.96	0.95	0.95
Cluster purity	0.97	0.98	0.98	0.99	0.96	0.97	0.82	0.97	0.95	0.96	0.98	0.95	0.95			

distributions for each note type, according to the method described below, and compared the patterns across note types.

The semantic type patterns consist of linear sequences of semantic types within the predefined window. Since the patterns are not defined in terms of simply co-occurrence in a sentence, the relative position of the terms participating in a pattern is meaningful. Therefore, the list of potential patterns is a cross-product of the semantic type set and positions relative to the term of interest. Initial measurement of semantic type frequency revealed that almost all note types had an average of between two and three mappings per sentence. I concluded that due to sparsity evaluating patterns of more than three mappings would fail to produce useful patterns. Therefore, in order to minimize the number of patterns, each pattern consists of the term of interest and two other terms within the predefined window.

Each position within the window was numbered according to the relative position from the term of interest, such that the term of interest was numbered (0); the term after the term of interest was numbered (1); the term before the term of interest was numbered (-1), and so on. The different combinations of positions of mappings within the window were grouped into fifteen formats as outlined in Table 3.4. Thus, a semantic type co-occurrence format is an abstract sequence of mapping positions relative to the center that corresponds to the position of the term of interest. Table 3.4 gives examples of patterns derived from a sentence "The patient reported severe upper quadrant abdominal pain" with the term "upper" as the term of interest, assuming the following semantic types for each of the mappings (with their four-letter abbreviations, which are also described in Appendix A):

- patient - Patient or Disabled Group - podg
- reported - Health Care Activity - acty
- severe - Qualitative Concept - qlco
- upper - Spatial Concept - spco
- quadrant -Spatial Concept - spco
- abdominal - Body Location or Region - blor
- pain - Sign or Symptom - sosy

Relative frequency of observed sequences of mappings that fell within the evaluation window were calculated. Those patterns that occurred only once were treated as outliers and were excluded from the analysis. Ambiguously mapped terms were counted as unmapped. Only unambiguously mapped terms were used in the patterns. Therefore, patterns of only

Table 3.4: The formats of the patterns found within the window of size 3. The example evaluates sentence "The patient[*podg*] reported[*acty*] severe[*qlco*] upper[*spco*] quadrant[*spco*] abdominal[*blor*] pain[*sosy*]" with the term "upper" as the term of interest.

Format number	**Format structure**	**Examples of patterns**	**Corresponding terms**
Format 1	(-3) (-2) (0)	podg_acty_spco	patient_reported_upper
Format 2	(-3) (-1) (0)	podg_qlco_spco	patient_severe_upper
Format 3	(-3) (0) (1)	podg_spco_spco	patient_upper_quadrant
Format 4	(-3) (0) (2)	podg_spco_blor	patient_upper_abdominal
Format 5	(-3) (0) (3)	podg_spco_sosy	patient_upper_pain
Format 6	(-2) (-1) (0)	acty_qlco_spco	reported_severe_upper
Format 7	(-2) (0) (1)	acty_spco_spco	reported_upper_quadrant
Format 8	(-2) (0) (2)	acty_spco_blor	reported_upper_abdominal
Format 9	(-2) (0) (3)	acty_spco_sosy	reported_upper_pain
Format 10	(-1) (0) (1)	qlco_spco_spco	severe_upper_quadrant
Format 11	(-1) (0) (2)	qlco_spco_blor	severe_upper_abdominal
Format 12	(-1) (0) (3)	qlco_spco_sosy	severe_upper_pain
Format 13	(0) (1) (2)	spco_spco_blor	upper_quadrant_abdominal
Format 14	(0) (1) (3)	spco_spco_sosy	upper_quadrant_pain
Format 15	(0) (2) (3)	spco_blor_sosy	upper_abdominal_pain

some of the fifteen formats can be extracted from any given sentence. The evaluated co-occurrence patterns represented linear sequences of mappings and other terms found within the text of each clinical note type. The most common format was Format 8 where a mapping alternated with another term in a sequence. Second most common format types where those where a mapping was followed by another term and then two adjacent mappings such as Formats 2, 7 and 15. Formats that represented mappings separated by two other terms, such as Format 3 and 12 were not as common across all analyzed note types.

The sublanguage theory states that a specialized domain puts restrictions on the number of semantic types and semantic type patterns that are used in the sublanguage. So to evaluate whether languages of different clinical note types exhibit sublanguage characteristics, I compared the relative frequency of the collected patterns across all available note types. When pattern frequencies are sorted in reverse order and ranked starting with the most frequent pattern, cumulative frequency can be visualized as illustrated in Figure 3.1 and Figure 3.2. These curves show how restricted are sublanguages used in each of the note types. The steeper the curve is, the more constrained is the language. Comparing the cumulative relative frequency of the top most frequent semantic types for MEDLINE

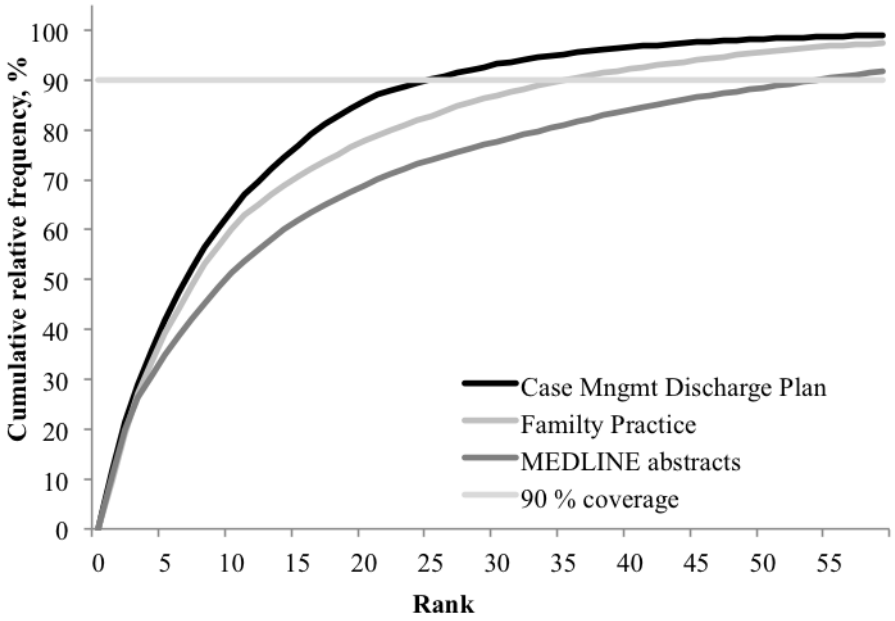


Figure 3.1: Cumulative relative frequency of the different semantic types used in Case Management Discharge Plan (CMD), Family Practice Clinic notes (FPC), and MEDLINE abstracts (MLN). The curves of other clinical note types fell between CMD and FPC lines and were excluded from the figure for visual clarity.

abstracts and clinical note types indicates that the language of the clinical notes is more restricted. As Figure 3.1 shows, a much larger number of semantic types is required to cover 90% of the unambiguously mapped concepts found in MEDLINE abstracts then in the analyzed clinical notes. For clinical notes, that number fell between 25 and 35 semantic types, whereas biomedical literature actively employed 57 semantic types. Semantic type patterns also indicate that clinical notes exhibit sublanguage characteristics. According to the sublanguage theory, semantic type patterns indicate the type of information structures that are used in the text. Thus, the smaller number of semantic type patterns is, the more restricted is the sublanguage they describe. Figure 3.2 presents a further evidence that the language used in the biomedical literature is more general than the language of clinical notes because the cumulative relative frequency curve has a gradual incline that does not go much flatter until it reaches full coverage. The shape of MEDLINE abstracts' cumulative relative frequency curve suggests that increasing the analyzed sample size would lead to discovery of more patterns. On the other hand, Ambulatory Nursing Notes are exhibiting characteristics of a very constrained sublanguage because the curve rises quickly and plateaus at almost

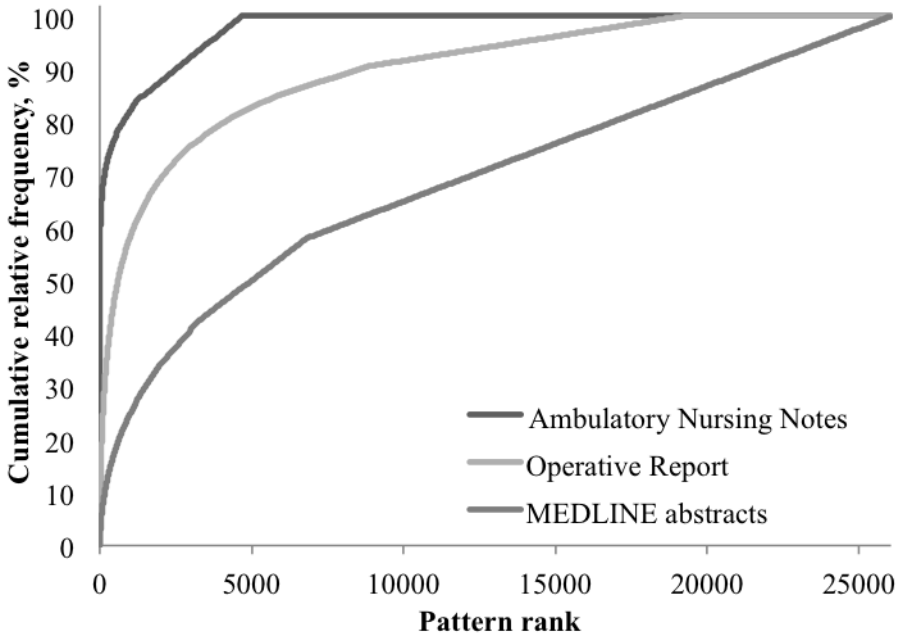


Figure 3.2: Cumulative relative frequency of patterns of format 8 for Ambulatory Nursing Notes (ANN), Operative Report (OPR), and MEDLINE abstracts(MLN). The curves of other clinical note types fell between ANN and OPR lines and were excluded from the figure for visual clarity

100%. Thus, the curve indicates that a smaller number of sentences is needed to illustrate all possible types of information that are used in the text of those clinical notes.

3.2 Discussion

The original aims for this step of my research focused on identifying the sublanguage boundaries among the notes that originated in various clinical subdomains and settings. Applying document clustering to a large set of clinical narratives allowed me to expose the differences in the lexical and semantic patterns used within different clinical environments as well as among different author types. This broad, systematic survey formally establishes what many clinical NLP researchers have suspected for a long time, namely that clinicians in different subdomains use language in a highly idiosyncratic way. Clustering also showed that contrary to the commonly held belief, the clinical setting does not carry as much weight in determining a clinical sublanguage boundary. The semantic pattern distribution curves indicate how restricted sentence semantic structures are, which is a clear evidence that the language of different note types meets the requirements to be regarded as proper

sublanguages. Together with the document clustering results, semantic pattern distributions indicate that the clinical language is not homogeneous, but rather is a collection of separate, though related, sublanguages. It is reasonable to expect that NLP systems that rely on statistical measures will perform differently on narratives that come from different clinical subdomains.

CHAPTER 4

SUBLANGUAGE SEMANTIC SCHEMA SYSTEM

The sublanguage theory proposed and developed by Zellig Harris became the theoretical basis for my project [39]. In order to implement sublanguage principles for word sense disambiguation, I defined Sublanguage Semantic Schema (S3) and implemented it as the Sublanguage Semantic Schema System (S3 System). For clarity, the following definitions will be used in the remainder of this text:

token - the smallest lexical unit analyzed by MetaMap. Includes words, numbers, and punctuation.

supporting tokens - tokens marked by MetaMap with one of the following parts of speech: auxiliary verb, complement, conjunction, determiner, modal verb, preposition, pronoun, and punctuation. These tokens are skipped by MetaMap algorithm during mapping.

term - one or more semantically linked tokens identified by MetaMap.

concept - a UMLS concept identified by MetaMap.

candidate - one of the concepts that represent the sense inventory of the mapped term. MetaMap identifies multiple candidates that are combined into a candidate set for each phrase. Disambiguation of the candidates is a task required for accurate mapping.

mapped term - a term that was mapped by MetaMap to at least one candidate.

mapping - a term that was *unambiguously* mapped to a UMLS concept using the rules of unambiguity I developed. The mapping has a UMLS concept identifier and a semantic type associated with it.

ambiguous term - a term that was mapped to multiple candidates.

ST - the semantic type of the UMLS concept associated with the mapping.

These definitions are also listed in the Appendix B for reference.

4.1 Sublanguage Semantic Schema

Previous research operationalized the sublanguage grammar as Domain Information Schema (DIS) [96]. DIS consisted of a set of semantic classes, the words and phrases that

belong to these classes, and the predicate-argument relationships among the members of these classes specific to the domain. I analyzed the applicability of this approach to the clinical domain and realized that this definition of the sublanguage structure is not feasible due to the difficulty of obtaining the most integral part of such schema - the predicate and argument labeling of terms. An accurate parser adapted to clinical text is rarely available in practice. Therefore, to make the approach more generalizable, I redefined the sublanguage structure and I propose a slightly different interpretation of a sublanguage. Instead of patterns based on predication, I decided to use linear sequence of semantic types as a manifestation of semantic type patterns. For the purposes of this research, the Sublanguage Semantic Schema (S3) is defined as a semantic grammar that describes a sublanguage. S3 consists of:

- A set of semantic types and corresponding conditional probabilities of these semantic types in a sentence;
- A set of semantic type patterns and corresponding conditional probabilities of these patterns in a sentence;
- A semantic type classification model resulted from a machine learning algorithm.

4.2 System Design

To demonstrate the feasibility of a sublanguage based approach to word sense disambiguation, I created a system prototype and called it Sublanguage Semantic Schema System (S3 System). The system requirements included the following specifications:

1. General purpose - The system has to be able to learn the disambiguation model for all clinically relevant words in the text.
2. Unsupervised learning - System adaptation to a new clinical subdomain should not require clinical expertise and manual annotations.
3. Real time disambiguation - The system has to be able to provide real time processing during disambiguation. This requirement arises as a result of the ultimate vision of creating a real time language processing system for clinical domain. It is acceptable for the training phase to be computationally intensive.
4. Easy component upgrade and replacement - Since UMLS, the selected knowledge base, and MetaMap, the concept mapping engine, undergo yearly updates, it is essential that the S3 System provides a simple way to replace these components with newer versions without requiring extensive system modifications.

The first two requirements are the two main distinguishing features of the current project. WSD systems that satisfy these two requirements usually struggle to achieve high accuracy. The system has two main parts - training module and application module. The complete prototype consists of the code that I developed to perform data manipulation, as well as the MetaMap engine to perform text mapping to UMLS Metathesaurus, and the MegaM algorithm to learn and apply logistic regression model. I used Groovy language for all programming, which I chose because it is a powerful, agile, and dynamic language that is based on Java virtual machine and incorporates Java code and libraries. Thus, an application written in Groovy can be used with any operating system.

4.2.1 Training Module

Similar to most supervised statistical corpus-based disambiguation methods, the S3 System relies on a sample of text as a training data for deriving a statistical model. Unlike supervised methods, the S3 System acquires annotated text automatically by taking advantage of the existing manually curated knowledge repository. The training module consists of the following parts:

1. Text parsing and concept mapping - The raw text is sent through MetaMap engine to arrive to an automatically annotated text as described in Section 2.5.2.
2. Feature vector extraction - The XML files resulted from MetaMap processing of the training corpus are processed and a set of feature vector is extracted as described in Section 4.2.1.1.
3. Pattern extraction - For each note type a set of linear sequences of semantic types is extracted as described in Sections 3.1.2 and corresponding probabilities are calculated as described in Section 4.2.2.
4. Machine learning model training - A logistic regression model is obtained using MegaM package as described in Section 4.2.3.

The general data flow for the training module is presented in Figure 4.1. A set of clinical notes of a specific note type are fed into the S3 System. The notes are then sent to MetaMap for processing. Once MetaMap completes mapping, the system modifies the MetaMap output into feature vectors and applies a machine learning algorithm to acquire semantic type classifier. MetaMap output is also processed to identify semantic type patterns for this dataset. As the final output of the training module, the patterns and classifier are stored for future use.

The S3 method is not limited to MetaMap. Any method of querying a large vocabulary can be used as long as it is powerful enough to perform mapping quickly. Similarly, logistic regression is not the only machine learning algorithm that can be used for obtaining an accurate classifier as long as it can handle multiclass data and is able to incorporate discrete or binary features.

4.2.1.1 Feature Vector Extraction

In the current implementation, the feature vector was created for each unambiguously mapped term, so the first step in the feature extraction process is to identify what a term is and which terms are unambiguous. The MetaMap output was delivered in XML format. A parsing module was written to parse the XML files and identify potential terms. Previous research with biomedical texts has used a simple definition of unambiguous mappings: those phrases that mapped to a single concept [41].

My initial calculation of the proportion of phrases mapped to a single concept in the clinical documents compared to MedLINE abstracts showed that that proportion is two to three times higher in biomedical text than clinical text. Therefore, I concluded that the single-candidate definition is not appropriate for clinical text because of its high level of ambiguity, which results in extremely limited mappings. After reviewing a large number of clinical text mappings produced by MetaMap, I derived additional rules of term boundaries and term unambiguity. These rules rely on the evaluation metric generated by MetaMap

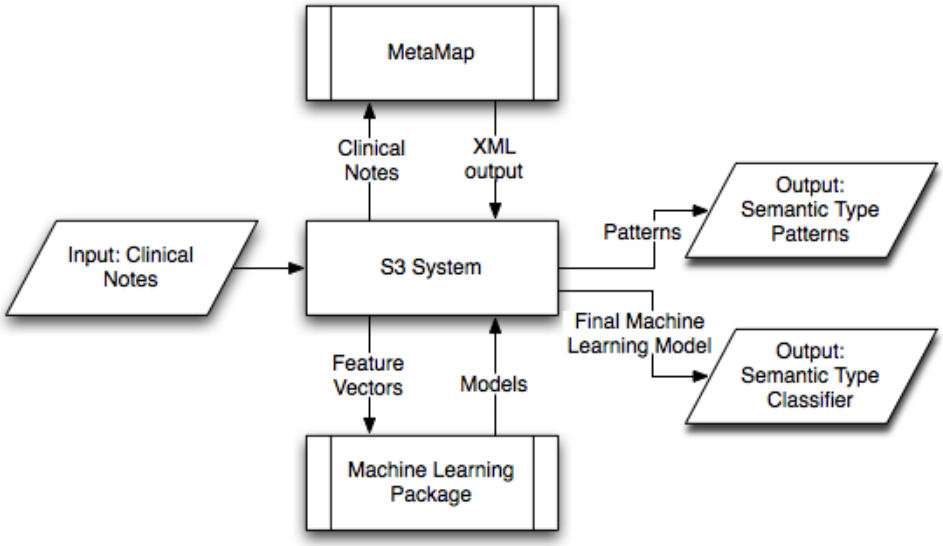


Figure 4.1: S3 System training data flow

to measure the quality of the match between the term in the analyzed phrase and a Metathesaurus concept [97]. Manual disambiguation of a number of sentences resulted in the development of the following text processing heuristics:

- Candidates that cover multiple tokens separated by one or more tokens other than supporting tokens, are excluded from further analysis, because in most reviewed cases the correct meaning was represented by contiguous tokens (as opposed to disjoint tokens).
- The phrase chunking into terms is done starting from the last token of each phrase and determining the longest contiguous term right-to-left, because in English, the head of a phrase is generally the last word of the phrase [98].
- If one or more numeric tokens belong to the same phrase and if the phrase does not have any candidates, such sequence of tokens is treated as a single token. This heuristic came from the fact that MetaMap does not have a mechanism to recognize dates and phone numbers as a single lexical unit.
- For the purposes of this research, punctuation tokens are ignored and are not considered in pattern and feature vectors. This heuristic resulted from observation that clinical text is full of implicit tables and other formatting done by authors to improve human readability of the documents, and the fact that inconsistent use of various punctuation marks is widespread.

These heuristics help to define which chunks of text represent terms and what candidates are included in the sense inventory for each term. Once each term is matched to a set of candidates, the following conditions are used to identify unambiguous term:

1. MetaMap produced only one candidate for the term even if variant generation was required to find this match (variants reduce the MetaMap mapping score, but here the mapping is still considered to be unambiguous).
2. MetaMap produced a single identical match except for spelling variation, capitalization, NOS suffixes and inversions such as Cancer, Lung vs. Lung Cancer.
3. MetaMap produced multiple matches, but all of the candidates have the same semantic type.
4. MetaMap produced a single match for the term such that the match evaluation score is either over 900 or, if no mappings over 900 are found, over 800.

Once an unambiguous term is identified, a set of features is collected from the MetaMap output. The exact feature set depends on the availability of information. The necessary

component is the semantic type of the analyzed terms. Other than that, any additional information about the term is potentially useful. In the current research, MetaMap functionality determined what can be potentially used as the context information for word sense disambiguation.

As I contemplated what features would be more important and what features could be ignored, I made a decision to incorporate all available information into the language model and to evaluate the importance of each feature as a subsequent system optimization analysis. MetaMap provides the following information about each sentence: *a*) utterance boundaries; *b*) phrase boundaries; *c*) syntax units; *d*) tokens; *e*) lexical category part of speech; *f*) syntax type; *g*) UMLS concept identifier (UMLS CUI); *h*) UMLS preferred name; *i*) sources vocabularies and terminologies that were the original sources of the concept.

Previous research based on a similar method identified that a feature set extracted using a larger window size does not consistently yield better accuracy than a set based on window size 3. [41] Additionally, a full corpus analysis showed that the average number of phrases within sentences identified by MetaMap ranged between 5 and 8 depending on the note type. Therefore, for the full feature list, a window size of 3 within the sentence boundaries is selected. Thus, the full feature set has features for the term of interest as well as for the three terms prior and after the term of interest. Seven terms are included in each feature vector.

The feature subset that directly describes the term of interest includes the part of speech and the syntax type as the features. For all other terms within the window, the feature subsets include the normalized tokens, part of speech, and syntax type. If the term was unambiguously mapped, the feature subset for that term also includes the semantic type, the UMLS preferred term, and a set of binary attributes that indicate whether the term is included in a specific terminology. UMLS Metathesaurus combines concepts from more than a hundred different source vocabularies.

In addition to the described features, the initially extracted vectors also have several metadata that are potentially beneficial for cross-checking the extractor accuracy. These additional data items include the file name, the sentence number and the term number of the term of interest within the sentence.

4.2.2 Patterns

At the time of model training, conditional probabilities for each semantic type are defined based on the presence of other semantic types in the predefined positions for each format. In

addition to the formats described in section 3.1.2, conditional probabilities were calculated for each semantic type separately, and for formats that contain only one other mapping within the predefined window. So based on the number of participating mappings in the pattern, the formats have the following three levels:

Level 0 - the patterns of this format represent only the term of interest that is at position 0. There is only one Level 0 Format, because it represents the conditional probability of occurrence of a specific semantic type in notes of a specific note type regardless of the surrounding mappings.

Level 1 - the patterns of the Level 1 Formats include the term of interest and one other mapping within the predefined window. There are a total of 6 formats of Level 1.

Level 2 - the patterns of the Level 2 Formats include the term of interest and two other mappings within the predefined window. Note that the analysis in Section 3.1.2 is based on patterns of this format level.

The Bayes' rule was used for the calculations of conditional probability. For example, for the pattern of Level 2 Format 1 from the example in Table 3.4, conditional probability is calculated using this formula:

$$P(spco_{(0)}|podg_{(-3)}acty_{(-2)}) = \frac{P(podg_{(-3)}\text{-}acty_{(-2)}\text{-}spco_{(0)})}{P(podg_{(-3)}acty_{(-2)})}$$

In this example, the output is the probability to see a "spco" semantic type in position 0, if the term in position -3 is unambiguously mapped to a concept with a semantic type "podg" and the term in position -2 has a mapping to a concept with semantic type "acty". These conditional probabilities are calculated for all linear sequences of unambiguously mapped terms in the training corpus. [1]

4.2.3 Semantic Type Classification Model

4.2.3.1 Sparse File Format

The full set of feature vectors extracted for each analyzed note types included a large number of feature vectors with a large number of features, some of which are binary and some of which are categorical. In order to acquire a classification model, I needed to find an

[1] My initial system design included creating and populating a MySQL database designed for the purposes of easy feature frequency calculation and feature analysis. After working with this database for some time I realized that in order to make querying and other processing fast, I needed to flatten the relational database into a single table, which would have made the table extremely large (over 100 million rows and over 1000 columns). Therefore, I abandoned that idea and wrote all data into a series of smaller text files processed by a set of Groovy modules specifically designed for each type of processing.

existing software or classification algorithm implementation that could handle such a large dataset. I considered several software packages including a widely used Weka data mining package [99]. Some algorithms are able to handle sparse data formats, so the S3 System module that deals with semantic type classification model acquisition includes converting full feature vectors (so called dense vectors) into sparse vectors.

The output of sparse conversion is a file in the sparse format where each discrete feature of the dense file is converted in a set of binary features. The sparse conversion module also outputs a full dictionary that links the sparse feature name to the feature in the dense dataset and its specific value. The sparse file format decreases the size of the file containing the dataset because only those features that are present in the feature vector are included. The Figures 4.2a and 4.2b present examples of a dense and sparse vector.

4.2.3.2 Machine Learning Algorithm

When selecting a machine learning algorithm to perform semantic type classification, I used the following guidelines:

- The algorithm has to be able to provide multiclass classification;
- The algorithm has be be able to handle categorical or binary features;
- The algorithm hast to be scalable and be able to process large data sets;
- The output of the algorithm should include not only the final prediction, but also a set of probabilities for other classes;
- The specific implementation of the algorithm has to be robust and relatively fast.

```
filename.txt, 1, 4, idcn, continue, verb, verb,
    5, null, to, adv, adv, null,
    3, null, will, modal, modal, null,
    6, inpr, follow, verb, verb, NCI SNOMEDCT,
    2, ocdi, social work, head, noun, AOD MSH MTH SNOMEDCT,
    7, null, this, det, det, null,
    1, zzzz, plan, verb, verb, null
```

(a) Dense feature vector.

```
114 MM142 MM143 MM1095 MM1096 MM1103 MM1135 MM1166
    MM1167 MM1218 MM1219 MM1432 MM1485 MM2141 MM2670
    MM11341 MM11342
```

(b) Sparse feature vector in MegaM format.

Figure 4.2: Examples of feature vectors.

After considering a number of different algorithms, I decided to use logistic regression as implemented by Hal Daume III, called MegaM [100]. MegaM satisfies all these requirements. This tool is based on maximum likelihood and maximum a posterior optimization of the maximum entropy models. By performing multiple iterations and weights adjustments, MegaM arrives to the optimal set of regression coefficients. MegaM accepts a sparse data file where each row represents as a set of features that describe an unambiguous term. The output of MegaM processing is a logistical regression model that gives a weight to each feature.

Even though MegaM is quite robust, it runs out of memory when the dataset size exceeds the system capacity. So I had to decrease the data file to make it manageable by MegaM. After some experimentation I found that a sample size of 50,000 was small enough for MegaM to process but large enough to produce a stable model. Therefore, the semantic type classification model was created using a reduced data set. Logistic regression models are prone to overfitting [101]. In order to mitigate this issue the training records were selected randomly from the full data set.

4.3 System Application

After the sublanguage semantic schema is obtained, it can be automatically applied for run-time disambiguation. In order to disambiguate all terms in a specific text segment, a set of steps is performed (see Figure 4.3). First, MetaMap processes the text and produces an XML file that contains the full set of concepts mapped to the terms found in the text

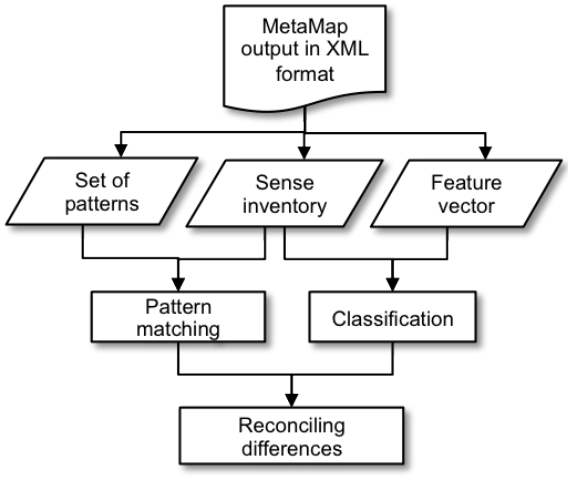


Figure 4.3: S3 System word sense disambiguation flow.

passage. The S3 system then analyzes the XML output and for each ambiguous term in the file it creates three components: 1) a feature vector according to the description presented in Section 4.2.1.1; 2) a set of semantic type sequences (patterns) taking into account the neighboring unambiguous mappings; and 3) a list of the corresponding UMLS concepts and semantic types. The latter serves as the sense inventory for the term.

Once sparse feature vectors and patterns are extracted, the S3 system uses classification and pattern matching to arrive to two semantic type predictions. Applying the previously acquired classification model to each feature vector, the S3 System classifies the terms with the most likely semantic types. Along with the most likely semantic type, the classification model also provides a list of probabilities of each semantic type in the sense inventory. It is possible that the most likely semantic type is actually not in the sense inventory.

The next step analyzes the patterns extracted for the term of interest and matches them with the patterns in the sublanguage semantic schema. This pattern matching step assigns probabilities to the semantic types from the sense inventories. The semantic type with the highest probability is selected as the pattern matching prediction. If none of the potential patterns matched any of those contained in the patterns set, then the term is marked as failed disambiguation.

After the previous steps are performed, each term has a sense inventory consisting of the semantic types, classifier-predicted semantic type, and pattern matched semantic type. If the semantic type prediction produced by the logistic regression classifier differs from the one produced by the pattern matching, the classifier and pattern predictions have to be reconciled. There are several possible outcomes. If the pattern matching did not fail disambiguation, the most likely semantic type is chosen as the final prediction. If the pattern matching failed to find a probable semantic type, the classification semantic type is determined to be the final semantic type.

The final step is word sense disambiguation. The semantic type disambiguation results in the most probable semantic type. The sense inventory is reviewed and the UMLS concept that has the most probable semantic type is selected. If more than one concept has the same semantic type, the concept with the highest mapping score is selected. If all concepts with the specified semantic type have the same mapping score, they are assumed to be synonymous and all of the concepts are returned as the final concept selection.

Figure 4.4 presents the overview of the data flow during the application phase. The system accepts a raw clinical text, either as a single sentence or a full clinical note. The

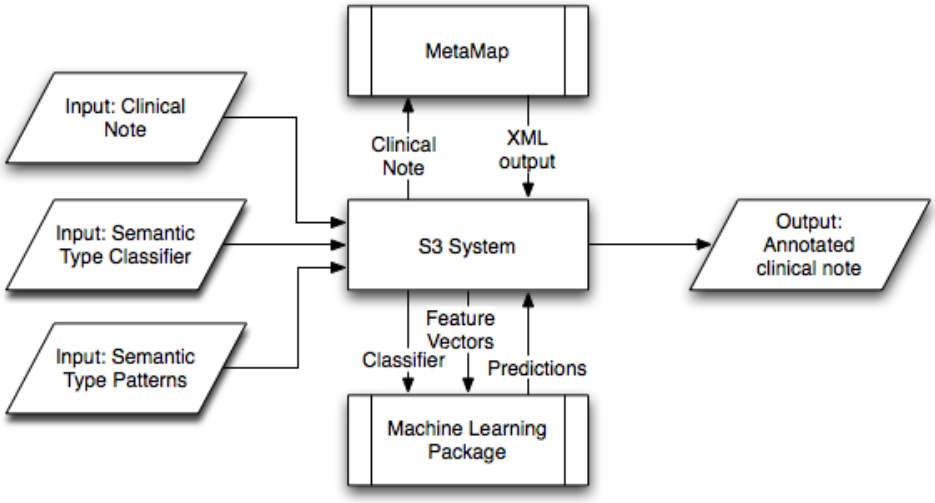


Figure 4.4: S3 System application data flow

current prototype does not have a functionality to determine which sublanguage semantic schema is applicable to the input text, so the operator has to specify that for the purposes of disambiguation. The S3 System then performs the processing steps outlined above and the final output is the clinical note, where each meaningful term is annotated with semantic type and UMLS concept.

The pattern set for the sublanguage contains patterns of three format levels. Calculating the most probable semantic type starts with patterns of the most restrictive format level - Level 2. If no patterns of that format level match the patterns found for the term of interest, patterns of less restrictive format levels are applied. One of the parameters of the application phase is the lowest pattern format level to be applied. If the level parameter is set to 0, then the S3 System will disambiguate all terms. If the level parameter is set to 3, then the S3 System will not perform pattern matching because 3 is larger than the lowest available format level.

Another parameter that can be manipulated during disambiguation is the lowest acceptable probability predicted by MetaM. If the most likely semantic type is not one of the semantic types from the sense inventory, the semantic types in the sense inventory are sorted by their probability. If the probability of the most likely semantic type is below the probability threshold, then the prediction is rejected and disambiguation is deemed to have failed. If the probability threshold is set to 1.0 or higher, then classification prediction will not be considered for disambiguation, because it is higher than possible probability. If the

probability threshold is set to 0.0, then S3 System will disambiguate all terms simply based on the most likely semantic type.

Varying these two parameters, the S3 System operator can balance the number of false positives with the number of terms that the system fails to disambiguate. Thus, manipulation of the parameters changes precision and recall of the system.

4.4 Validation

The S3 System is an implementation of the Sublanguage Semantic Schema approach to word sense disambiguation. Once the system prototype is completed, validation is needed. The validation provides answers to Aim II research questions:

Research question 2.1 – Does the developed system work well for clinical term disambiguation in a range of clinical note types as compared to a manually annotated test set?

Research question 2.2 – Does the system perform better than a baseline method such as MetaMap and the majority sense method?

To answer these research questions, I created a manually annotated corpus that can serve as a reference standard for the S3 System performance evaluation. In order to evaluate performance, the following definitions of the metrics are used in the analysis:

Reference Standard Positive – Total number of terms that were mapped by MetaMap to at least one UMLS concept that reflected their true meaning. These terms will be referred to as properly mapped terms.

Reference Standard Negative – Total number of terms that were mapped by MetaMap to at least one UMLS concept, but none of the concepts reflected the true meaning of the terms. These terms will be referred to as mismapped terms.

Total Positive – The number of terms that the S3 System was able to process and produce a semantic type prediction. These terms will be referred to as disambiguated terms.

Total Negative – The number of terms that the S3 System failed to disambiguate.

True Positive – The number of disambiguated terms that were disambiguated correctly. The correctly disambiguated term is the one what has the semantic type assigned by the system, which is the same as the one assigned by human annotator.

True Negative – The number of mismapped terms that the S3 System was not able to disambiguate.

Accuracy is the proportion of the terms in the reference standard that were disambiguated correctly.

Recall is the proportion of the properly mapped terms in the reference standard that were correctly disambiguated by the S3 System.

Precision – The proportion of the disambiguated terms that were disambiguated correctly.

4.4.1 Annotations

The most common way to conduct a rigorous system performance testing is to compare the system output against a reference standard [15]. The final accuracy of the S3 System depends on the performance of its components: MedPost tagger, which performs sentence segmentation and part of speech tagging; and MetaMap concept recognition engine, which performs mapping of terms to the concepts in UMLS Metathesaurus. The S3 System performs word sense disambiguation on the sense inventories resulted from MetaMap processing. To ensure that the evaluation targets the component that I developed, the reference standard had to be created accounting for the limitations of other components. The annotators task was limited to manual word sense disambiguation. The annotators were not asked to determine the sentence or phrase boundaries, or the intended meaning of the text beyond the sense inventory identified by MetaMap.

4.4.1.1 Sample Selection

Notes of seventeen note types are available to me; however, evaluating the S3 System performance on all note types is not feasible due to financial and time constraints. Therefore, four note types were selected for evaluation purposes: 1) Admissions History and Physical Notes; 2) Discharge Summaries; 3) Cardiology Clinic Notes; and 4) Social Service Notes.

The model employed by the S3 System performs disambiguation relying on the context information at a sentence level and disregards the larger context. Therefore, the annotators were presented with individual sentences for annotations. Each term in the corpus identified by MetaMap as a potentially relevant concept is an opportunity to fail. Therefore, the size is defined as a number of terms in the sample. Sample size is generally determined using Cochran's sample size formula.

$$Sample\ Size = \frac{(Z^2 p(1-p))}{c^2}$$

where Z is the Z value corresponding to a specific confidence level (for 95% confidence level Z=1.96); p is the proportion of units in relation to the true proportion that one could expect (this value is assumed to be 0.5); and c is the degree of precision or confidence interval.

The sample size calculations assuming confidence level of 95% and confidence interval of 3%, the sample size is determined to be 1067 terms for each note type. The method employed by the S3 System relies on the sentence context for disambiguation and is optimal if at least 3 or more terms are present in the sentence. Therefore, the sample for annotation was created by extracting a random set of sentences that contain at least three terms. Prior analysis estimated that such sentences have on average 6.7 terms per sentence. Simple calculation $\frac{1067}{6.7}$ results in a sample size of 160 sentences per note type or 640 total sentences. The sample size calculations for comparison of two proportions with a confidence interval of 3% and a power of 90% for proportions 75% and 80% results in sample size of 1193, or approximately 1200 terms. Calculating the number of sentences that would be needed to get at least 1200 terms is approximately 180. Based on these calculations, I decided that the sample size of 200 sentences for each note type would provide enough power to accurately analyze the S3 System performance across all note types.

4.4.1.2 Annotation Process

Due to the strict privacy and security requirements directed by HIPAA, the notes are stored on a cluster specifically designed to store and process clinical data. The annotators were required to access the data remotely passing through two levels of authentication. They were not allowed to copy the clinical text to their personal computers. To access the data, the annotators had to first login into the CHPC via Virtual Private Network client. Then they had to use a remote access application such as X Window System. A Windows server was set up with Samba access to the data and an annotation application.

The extensible Human Oracle Suite of Tools (eHOST) application is an annotation tool developed by Brett South and a research team of the Office of Information & Technology (OI&T) at the Veterans Affairs [102]. It can be used to annotate text by creating an annotation object represented by a span of text, the associated concept according to an annotation schema. The distinguishing feature of eHost is its ability to accept annotations created outside of the application, called preannotations.

Since the goal of the annotation step is to create a reference standard to test performance of the S3 System, the annotation task is quite limited. For each term identified by MetaMap, the annotators had to choose one of the candidates that resulted from the MetaMap processing, or to specify that none of the candidates reflected the intended meaning of the term. For that purpose, each term had an associated sense inventory of UMLS concepts

and corresponding semantic type. The preannotations were loaded to eHost and presented to the annotators for disambiguation. The UMLS semantic network contains 135 semantic types, which a large number. To simplify the annotation scheme, only two markables were used: Concept and None. The markable type Concept was designed to contain information about the UMLS concept that MetaMap mapped to the term. The markable type None was designed to handle the case when none of the candidates represent the meaning of the corresponding term correctly. Each annotation object specified the span of each term, which was derived from the MetaMap output, and the concept description, which was a combination of the following MetaMap output elements that identify a specific UMLS concept: CandidateCUI, CandidateMatched, CandidatePreferred, and SemType. Two annotators were recruited to perform annotations of this project. They both have had previous experience with annotations and have used eHost. Both of them are affiliated with the University of Utah as current and former employees. Both of the annotators completed Human Research training through CITI. The annotators' task was to review the sense inventory for each term and select that concept that reflected the meaning of the term most accurately. Some of the terms were mislabeled by MetaMap, resulting in a set of annotations for a term, none of which reflect the meaning of the term. In those cases the annotators are asked to mark the term as markable type None. A total of 5430 terms were disambiguated by the annotators. 374 of those terms were marked as "None of the above", indicating that MetaMap failed to produce at least one concept for the term that reflects the intended meaning of the work in text (see Table 4.1).

In addition, the annotators disagreed on whether one or none of the candidates represent the actual meaning for 653 terms, and 451 annotations were concepts with different semantic types, which indicates that a large portion of the concepts are so vague that their actual meaning might be interpreted in multiple ways. If two conflicting annotations had the same semantic type, these annotations were deemed as equal and one of the annotations

Table 4.1: Annotated corpus description.

Note Type	AHP	CCN	DIS	SSN	Total
Properly mapped terms	1252	1231	1329	1244	5056
Mismapped terms	54	113	70	137	374
Total term count	1306	1344	1399	1381	5430
Pair-wise agreement	82.5%	84.5%	84.3%	80.0%	

was selected randomly as the reference standard. For those terms that were marked with concepts that had different semantic types, another clinically trained person performed adjudication. The average annotator pair-wise agreement was 82.8%.

4.5 Measuring Performance

To evaluate whether the S3 System performs well on clinical text, I compared the semantic types assigned to terms in the reference standard by the S3 System to those assigned by the human annotators. Following the steps outlined in Section 4.2.1, I trained models for the four note types using the full set of notes available to me. The full description of the training corpus for validation is presented in Table 4.2. After the models were acquired, I applied them to the manually annotated corpus and received the results are reflected in Table 4.3.

4.5.1 Model Comparison

Previous analysis suggested that notes of different types belong to similar or dissimilar sublanguages. As an example, the hierarchical clustering tree in Table 3.2 showed that

Table 4.2: Full data description for the four note types that were used in validation.

Note Type	AHP	CCN	DIS	SSN
Number of processed files	42,911	24,302	64,530	3,414
Total number of unambiguous terms	10,973,008	3,861,766	13,372,050	262,102
Number of unique patterns extracted from the training corpus	105,455	57,693	106,908	24,529

Table 4.3: Accuracy of S3 System as tested on a manually annotated set of sentences with format level threshold of 2 and classification probability threshold of 0.1.

Note Type	AHP	CCN	DIS	SSN	Average
Match	900	903	959	854	
Mis-match	186	186	203	199	
WSD Failed	166	142	167	191	
Total Terms	1252	1231	1329	1244	
Recall	0.719	0.734	0.722	0.686	0.715
Precision	0.829	0.829	0.825	0.811	0.824
F-score	0.770	0.778	0.770	0.744	0.765

Discharge Summaries and Admission History and Physical notes are relatively similar and can be assumed to be of the related sublanguages. On the other hand, Cardiology Clinic Notes and Social Service Notes have a distant relationship and can be assumed to be of different sublanguages. To illustrate the benefit of rapid sublanguage modeling with the S3 System, I compared disambiguation accuracy of sublanguage semantic schemata trained using one note type and applied to a different note type. Comparing the performance of the model that was trained on Discharge Summaries and applied to the same note type to the performance of such a model when it is applied to Admission History and Physical notes indicates the relative performance of S3 approach trained and applied to test of similar sublanguages. As Table 4.4 indicates, the models, trained on either Admission History and Physical or Discharge Summaries and applied to notes of either of these types, perform similarly. Neither recall nor precision proportions were determined to be significantly different when models are cross-applied. This finding confirms the assumption that the language used in these two note types can be considered the same sublanguage. On the other hand, comparing the performance of the Semantic Sublanguage Schema trained on one note type and applied to the same note type to the performance when it is applied to notes of a different note type, indicates the change in the model performance when source and target corpus come from a distantly related subdomain. Two-sample tests of

Table 4.4: Comparison of accuracy of S3 System on Admission History and Physical and Discharge Summaries. Disambiguation was performed with pattern format Level 2 and classification probability threshold of 0.1.

		Tested on	
Trained on		AHP	DIS
AHP	True Positive	900	966
	Mismatch	186	209
	Failed Disambiguation	166	154
	Total Terms	1252	1329
	Recall	0.719 (±0.025)	0.727 (±0.024)
	Precision	0.829 (±0.022)	0.822 (±0.022)
DIS	True Positive	862	959
	Mismatch	183	203
	Failed Disambiguation	270	167
	Total Terms	1252	1329
	Recall	0.688(±0.026)	0.722(±0.024)
	Precision	0.825(±0.023)	0.825(±0.022)

recall proportions achieved by a model trained on Cardiology Clinic Notes and tested on the same compared to such a model tested on Social Service notes resulted in $p < 0.001$. Table 4.5 illustrates that the average recall of models trained on Cardiology Clinic Notes and applied to Social Service Notes differs significantly because 95% confidence intervals do not intersect. Table 4.6 outlines the results of applying MetaMap on the validation test with the assumption that terms with the same semantic type are synonymous. Comparing the S3 system performance to the unambiguous mappings presented by MetaMap outlined in Table 4.3, we can see that while applying the S3 system to disambiguate terms did not improve the F-score, the recall of disambiguation is higher when the S3 System is applied.

4.6 Discussion

The aims for this step of the research focused on the design and prototype implementation of a system based on sublanguage semantic schema. For that purpose, I designed an application that illustrates how such a system can be implemented as a stand-alone package. This proof-of-concept prototype achieved a level of recall and precision of concept identification that is comparable to other approaches that involve all-word word sense disambiguation. Further optimization can potentially bring the accuracy to the level that

Table 4.5: Comparison of accuracy of the S3 System on Cardiology Clinic Notes (CCN) and Social Service Notes (SSN). Disambiguation was performed with pattern format Level 2 and classification probability threshold of 0.1. The value in parentheses represent the 95% confidence interval.

		Tested on	
Trained on		CCN	SSN
CCN	True Positive	903	826
	Mismatch	186	188
	Failed Disambiguation	142	230
	Total Terms	1231	1244
	Recall	0.734 (±0.025)	0.664 (±0.026)
	Precision	0.829 (±0.021)	0.815 (±0.022)
SSN	True Positive	840	854
	Mismatch	148	199
	Failed Disambiguation	243	191
	Total Terms	1231	1244
	Recall	0.682(±0.026)	0.686(±0.026)
	Precision	0.850(±0.020)	0.811(±0.022)

Table 4.6: MetaMap performance as applied to the manually annotated set.

Note Type	AHP	CCN	DIS	SSN	Average
Match	776	806	847	805	
Mis-match	30	53	46	72	
WSD Failed	446	372	436	367	
Total Terms	1252	1231	1329	1244	
Recall	0.620	0.655	0.637	0.647	0.640
Precision	0.963	0.938	0.948	0.938	0.941
F-score	0.754	0.771	0.762	0.759	0.762

would be acceptable for practical clinical purposes. The current prototype meets the basic system requirements specified in Section 4.2. First of all, this system is general purpose because it does not limit a set of words that it is able to disambiguate. Second, it is able to acquire a language model using an unsupervised method. Third, the speed of disambiguating clinically relevant concepts in a short narrative is close to real-time, and can be further improved by providing more powerful computational resources. And the last requirement of easy component upgrade and replacement is satisfied by the pipe-line system architecture that is easily adjustable to accept data in a different format.

CHAPTER 5

SYSTEM IMPROVEMENT

5.1 Error Analysis

The preliminary S3 System validation demonstrated that automatic domain adaptation of a concept recognition system is feasible in terms of time and human expert involvement. However, the initial accuracy level achieved by the current design required improvement. In order to identify the system elements that can be improved, I performed an error analysis comparing the human annotated corpus to the S3 System output.

When performing an error analysis, I had to consider the possibility that the feature space of unambiguous terms differs from the feature space of ambiguous terms. If this were the case, the classification model obtained on the unambiguous terms would be inadequate for ambiguous terms. To test this possibility, I used a bisecting K-means clustering algorithm to compare the feature space of ambiguous and unambiguous terms. Applied to the terms found in the validation data set, clustering to various cluster numbers arrived at similar results. The clustering results for 10 clusters are presented in Table 5.1.

Clustering purity that is not significantly different from the proportion of the majority class (in this case, unambiguous mappings were the majority) indicates that the clustering split does not depend on whether the record represents ambiguous or unambiguous term. Therefore, I concluded that the feature space of unambiguous terms effectively represents the feature space of ambiguous terms.

Table 5.1: Clustering purity for all terms in the reference standard corpus when grouping into 10 clusters.

Note Type	AHP	DIS	CCN	SSN
Term Count	1306	1399	1344	1381
Unambiguous Mappings	761	851	825	867
Ambiguous Mappings	545	548	519	514
Proportion of Unambiguous Mappings	0.583	0.608	0.614	0.628
Clustering Purity	0.583	0.628	0.614	0.628

After the overall approach was determined to be adequate, I compared the mappings selected by human annotators to those concepts selected by the S3 System. Table 5.2 shows the overall accuracy of the S3 System. For this error analysis, I used the output from the S3 System processing of the validation corpus, using a format level threshold of 2 and a classification probability threshold of 0.1. These thresholds are rather conservative, emphasizing precision over recall. The unambiguous mappings were determined using the heuristics outlined in Section 4.2.1.1.

As indicated in the beginning of Section 4.4, accuracy depends on the number of the correctly disambiguated terms and the number of mismapped terms that the S3 System failed to disambiguate. This definition of accuracy is premised on the assumption that those terms that are mismapped by MetaMap are linked to concepts whose semantic types do not conform to the semantic grammar of the sublanguage, and therefore, all of the candidates would be rejected by the S3 System as being of low probability. Thus, those cases where the S3 System fails to disambiguate a term should be understood as being outside of the sublanguage semantic grammar extracted from the training examples. As expected, the proportion of mismapped terms among those terms that the S3 System

Table 5.2: Accuracy of S3 System as tested on a manually annotated set of sentences with format level threshold of 2 and classification probability threshold of 0.1.

Note Type	AHP	CCN	DIS	SSN
Total Terms	1306	1344	1399	1381
Reference Standard Positive	1252	1231	1329	1244
Reference Standard Negative	54(4.1%)	113(8.4%)	70(5.0%)	137(9.9%)
Unambiguous mappings total	761	825	851	867
matched	702(92.2%)	728(88.2%)	780(91.7%)	732(84.4%)
mismapped	36(4.7%)	57(6.9%)	39(4.6%)	80(9.2%)
mismatched	23(3.0%)	40(4.9%)	32(3.8%)	55(6.4%)
Disambiguated total	1129	1163	1219	1143
matched	900(79.7%)	903(77.6%)	957(78.7%)	852(74.5%)
mismapped	43(3.9%)	74(6.4%)	57(4.7%)	90(7.9%)
mismatched	186(16.5%)	186(16.0%)	205(16.7%)	201(17.6%)
WSD Failed Total	177	181	180	238
mapped	166(93.8%)	142(78.5%)	167(92.8%)	191(80.3%)
mismapped	11(6.2%)	39(21.5%)	13(7.2%)	47(19.7%)
Accuracy (average 68.6%)	69.8%	70.1%	69.5%	65.2%

failed to disambiguate is significantly higher for Cardiology Clinic and Social Service Notes ($p < 0.0001$ for both) and marginally higher for Admissions History and Physical and Discharge Summaries ($p = 0.068$ and $p = 0.076$ respectively) as compared to the proportion of mismapped terms that were disambiguated.

As seen in Table 5.2, a relatively large proportion of terms that were unambiguously mapped to a UMLS concept were mapped erroneously. In these cases, the concept suggested by MetaMap did not represent the correct meaning of the term. This type of error introduced noise into the training corpus and resulted in misleading semantic patterns and a faulty classification model. Reviewing the mismapped terms revealed that those terms mostly represent general English words that are not specific to clinical text (Table 5.3). This illustrates the limitations of the underlying knowledge base - UMLS Metathesaurus - that is not designed to be a comprehensive vocabulary of general English, but rather a specialized biomedical terminology.

In addition to mismapped cases, other sources of errors are unambiguously mapped terms that did not match the human annotator selections. These occurred when the correct sense in the sense inventory was represented by a candidate with a lower mapping score. For example, the validation corpus for Admissions History and Physical contains a sentence "She is unable to walk without assistance." The term "assistance" had the following concepts in the sense inventory:

- C0018896 Helping Behavior (socb)
- C0557034 Patient assistance (hlca)
- C1269765 Assisted (fndg)
- C1515950 American Stop Smoking Intervention for Cancer Prevention (hcro)

The annotators selected "Patient assistance", semantic type "Healthcare Activity," as the correct concept. However, the word *patient* is not specified in the sentence creating partial concept match that leads to a lower mapping score. On the other hand, mapping assistance to social behavior was performed through simple matching and it received the highest score. Though these cases are relatively rare, resolving the problem of mismatched unambiguous terms would lead to noise reduction in the training set, which in turn would increase the sublanguage model accuracy.

Reviewing the cases in which a term is mapped and is ambiguous revealed that a large proportion of mismatches involves terms that refer to an activity or healthcare activity such as *care*, *transfer*, *admit*, *discharge*, *document*, *report* and others as represented by verbs. It

Table 5.3: List of mismapped terms found in the validation corpus. Italicized terms were mapped unambiguously.

Note Type	Terms	Total Instance Count
Admission History and Physical	*number*, arrangement, base, bear, *calm*, *check*, *cocaine use*, demonstrate, *dramatic*, *english*, essentially, ext, *f*, *five day*, *follow*, *go*, *gravity*, i, jp, *last*, *mdi*, *mission*, move, numb, *o*, *other*, *over*, *own*, period, *plt*, receive, *russell*, *sandy*, *sat*, *service*, *show*, *study*, *su*, support, *switch*, tip, *v d*, *want*, *wish*, *x3*	54
Cardiology Clinic Notes	*number*, *address*, back, *bid*, bpm, check, *clear*, couple, *crisp*, crt, *ddd*, *diagonal*, *dm*, *exhaust*, *fairly*, *fall*, feel, fib, *follow*, *go*, i, *interrogate*, joint, ken, *kt*, *last*, life, lv, *mark*, *may*, met, *nfm*, *note*, *other*, *otis*, *pace*, particularly, *qt*, radiate, raise, *rck*, reasonable, *ride*, *right*, *rule*, *rv*, s2, sander, *saw*, *see*, *service*, shift, *show*, solution, spring, *strange*, *switch*, *tee*, *tell*, turn, *warner*, *wish*, work	113
Discharge Summary	*number/number*, back, birch, *chem*, *corner*, *dc*, dr, *ely*, face, feel, fill, *five day*, *follow*, *gaf*, gbs, h, i, level, lim, line, *love*, *mark*, *mom*, mra, *o*, *other*, p2, *p4*, pass, provide, radiate, *remarkable*, right, *robert*, *rule*, *russell*, *rvr*, *serial*, set, *settle*, *show*, sing, *smith*, solve, *spencer*, *stand*, tram, *transition*, *trial*, *up to*, *wish*, work	70
Social Service Notes	*number/number*, accommodation, *address*, *amy*, *angel*, *aspen*, back, bear, *bradley*, *calm*, *carina*, *center*, *challenge*, *check*, closure, *coordinate*, couple, creek, crystal, ctp, *cynthia*, *daniel*, *deliver*, *experience*, *f*, *fall*, *far*, feel, *flow*, *follow*, *gloria*, *god*, h, *hch*, hear, i, *impact*, *last*, *ld*, *live*, *logan*, *long*, *look*, *lpc*, manage, *manner*, *mark*, *mary*, *melinda*, mill, mojave, *mom*, move, *nelson*, *note*, *o*, *oa*, *other*, *own*, *pcmc*, *personal assistance*, *prima*, *program*, provide, psych, *remarkable*, rise, *robert*, *round*, senior, shot, *slat*, *spark*, *sunrise*, *talk*, *tell*, *tom*, *trail*, *transition*, *turn*, *validation*, *washington*, *wish*, *y*	137

is interesting to note that various forms of a term *report* were used four times in Admissions History and Physical and 11 times in Social Service Notes validation corpus; the annotators disagreed on the meaning of the term in three and nine cases respectively. This indicates that in the UMLS Metathesaurus the concept definitions for different meanings of *report* are not clear enough for humans to interpret. Thus, the computational approach also fails to make that distinction easily.

Two other semantic types that contribute a relatively large proportion of erroneous matches are "Sign or Symptom" and "Disease or Syndrome." These semantic types, if mismatched, are most often categorized by the S3 System as "Finding." Since this

distinction is often difficult for humans to resolve as well, this type of error will be challenging to solve. Similarly, semantic types "Body Part, Organ, or Organ Component" and "Body Location or Region" are often confused by the S3 System as well as by human annotators.

The semantic type "Finding" appears to be so general that terms of that semantic type are often misclassified. The term *negative* is used five times in the AHP validation set and in four of those cases the annotators disagreed on the meaning of the term. The S3 System selection matched the majority vote for the term "negative" three times and only in two cases was a wrong concept selected.

One common mistake made by the S3 System is classifying a gender in a common word combination such as *the patient is a n-year old female* as a concept of "Population group" semantic type, whereas the annotators consistently selected "Organism attribute" as a semantic type for these terms.

In general, the reasons for errors made by the S3 System can be categorized as those due to the limitations of the primary knowledge base, the UMLS Metathesaurus, and those due to the limitations of the current S3 System design and implementation. The UMLS Metathesaurus limitations led to an insufficient concept definition and inadequate language coverage, which resulted in an incomplete sense inventory for terms. The UMLS Metathesaurus is under active development, so it is likely that some of the issues will be resolved over time. However, since even a highly restricted clinical sublanguage uses elements of general English language, an additional lexical knowledge base such as WordNet can be used to enhance language coverage [103].

Similarly to the UMLS Metathesaurus, WordNet categorizes concepts into semantic groups called synsets. At a higher level of granularity, these synsets are called base types and represent concept categorization similar to the UMLS Semantic Network. Thus, the lexical information from WordNet can be integrated easily into the S3 System. The main concern for such an expanded knowledge base would be reconciling the cases when the same meaning is represented by concepts in the UMLS Metathesaurus and in WordNet. Since the UMLS semantic types and WordNet base types do not have a one-to-one relationship, equating two concepts might be challenging. An automatic method of mapping UMLS concepts to WordNet concepts has been suggested and could potentially be integrated into the S3 System [104]. In addition to the issues with finding an appropriate meaning for a term, a number of lower level problems have been uncovered. These problems were encountered in the initial steps of text processing by MetaMap system. One of the assumptions of the S3

method is that the employed concept recognition system is robust enough to process a large majority of the files and is able to extract enough unambiguous concepts for training. The analysis of files that failed MetaMap processing revealed that failures occur when:

- the incoming text contains special characters, which lead to system crashes;
- the incoming text is exceptionally ambiguous, which leads to such a large number of mappings that the system is overwhelmed by endless processing.

One method to improve the concept recognition performance is to include a preprocessing step that would remove special characters that might lead to the system crashes. This preprocessing step also can replace certain tokens such as dates, patient and provider names, geographic locations, and other similar data items with specialized tokens. Such a token replacement can be implemented easily because those items are usually included in the patient record, or are otherwise accessible from the electronic medical record system. The main benefit of such a preprocessing step would be to remove those proper nouns that are homonymous with clinically relevant terms and might be confused by the concept recognition module.

5.2 Optimization

The limitations of the current S3 System design are due to shortcomings of the *pattern matching*, due to limitations of the machine learning *classification*, or due to inadequate *integration* of pattern and classification predictions.

5.2.1 Pattern Matching

The classical definition of sublanguage is based on semantic type co-occurrence patterns that are linked to predicate-argument relationships within sentences. The current implementation of the S3 System relies on a linear sequence of semantic types to acquire semantic type pattern probabilities. This deviation from the classical definition is more straightforward to implement. However, this approach results in diluted patterns because even a slight difference in word order would result in a different pattern, thus decreasing the conditional probability of certain semantic type patterns.

The main reason for using a modified semantic type pattern definition is a lack of an easily accessible accurate parser and part of speech tagger. The S3 System relies on the MedPost tagger to perform word and sentence segmentation and part of speech tagging [105]. Since the knowledge base for the tagger is derived from analyzing biomedical text, the tagger's performance on the clinical text is insufficient. One of the most common causes

of incorrect sentence segmentation is a period that is a part of an abbreviation token and does not mark a sentence boundary. Faulty sentence and phrase segmentation contributes to the number of mismappings because it decreases the number of correct phrases that potentially have matches in the UMLS.

5.2.2 Classification Accuracy Improvement

The initially created dataset for machine learning classification contained all features that could be extracted from MetaMap output. However, low initial accuracy of the machine learning algorithm (which is not to be confused with the final S3 System accuracy) necessitated feature space modification. One of the common ways to address machine learning model accuracy is to increase the size of the training data set [106]. However, when designing the initial S3 System prototype, I identified that due to the machine learning algorithm's computational limitations, the sample size had to be restricted to around 50,000 records to enable MegaM to complete processing successfully. Since the logistic regression model builds a set of weights for each feature, the larger is the number of dimensions - the larger is the computational complexity of the learning algorithm. Therefore, I concluded that decreasing the number of features would make increasing the sample size possible.

To perform feature selection, I chose the information gain algorithm as implemented in Weka v 3-6-1 [99]. This algorithm evaluates the information gain for each attribute with respect to the class. The output from such processing includes a list of features and their corresponding information gain in descending order. Using the full set of unambiguously mapped terms for each note type, I randomly selected 200,000 records and applied information gain with ranking feature selection algorithm. To find what features would provide the best feature set, I created four feature sets that include features with information gain above four different thresholds: 0.01, 0.001, 0.0001, and 0.00001. Table 5.4 lists the number of features that were selected at different information gain thresholds.

For each feature set I randomly extracted datasets with a varying number of records. Then I trained a MegaM model on each data set with a different feature set and record number combination. I then applied those models to the same randomly selected dataset of 50,000 records. The accuracy of the resulting models is presented in Figures 5.1, 5.2, 5.3, and 5.4. All processing was performed on a powerful 12-core 92 GB compute node. The time to perform training was calculated and is presented in Figures 5.5, 5.6, 5.7, and 5.8. As expected, a larger datafile with more features and more records takes longer to process.

Table 5.4: Feature counts based on different information gain thresholds.

Note Type	AHP	CCN	DIS	SSN
Threshold				
0.01	971	288	191	146
0.001	1,630	1,745	1,586	1,697
0.0001	13,012	9,945	12,362	10,724
0.00001	60,666	40,358	58,126	39,187

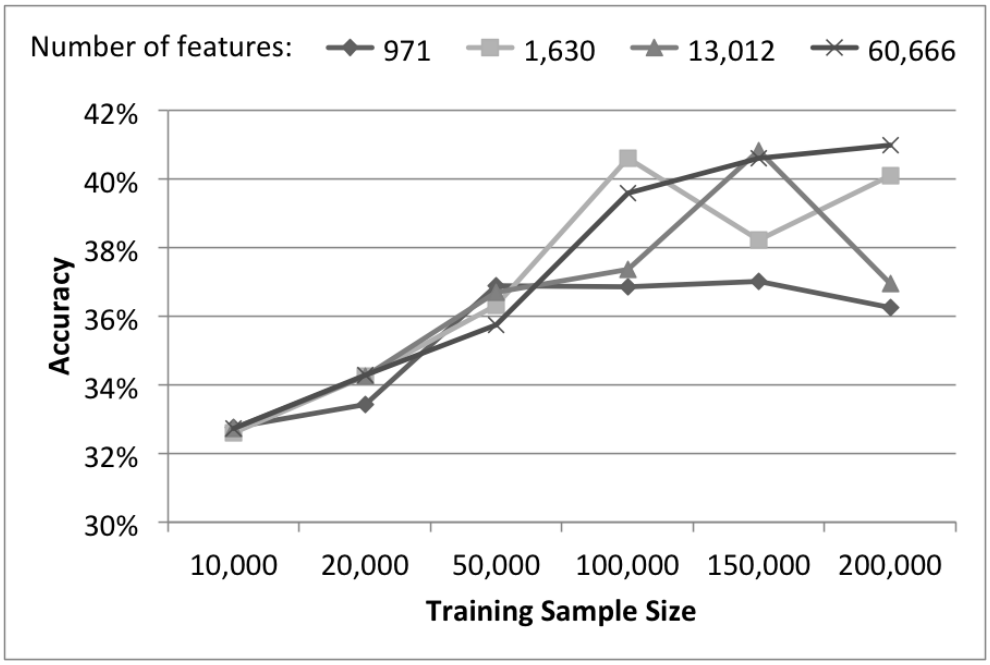


Figure 5.1: Classification accuracy as a function of the number of features and number of records for Admission History and Physical.

Similar results were obtained for all four note types. Looking at both graphs for each of the four note types, I concluded that information gain threshold of 0.001 and sample size of 100,000 is the right balance between performance speed and accuracy of the resulting models.

5.2.3 Determining Final Predictions

The initial design of the S3 System gives preference to pattern matching predictions when determining the most likely semantic type given the content of the term. This design is based on the assumption that if a specific semantic type sequence appeared in the training

Figure 5.2: Classification accuracy as a function of the number of features and number of records for Cardiology Clinical Notes.

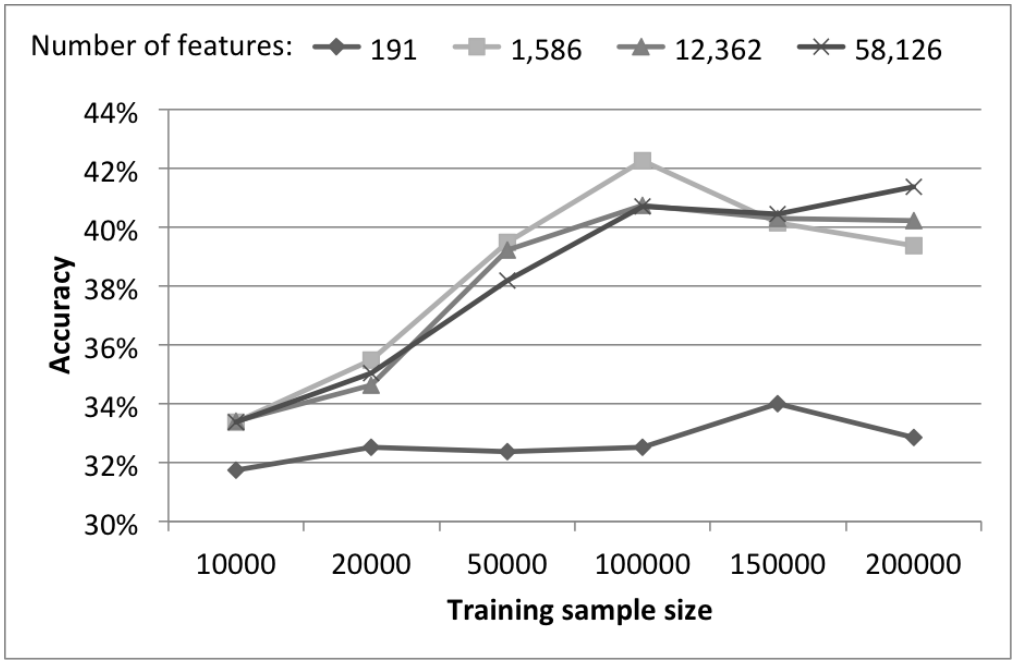


Figure 5.3: Classification accuracy as a function of the number of features and number of records for Discharge Summaries.

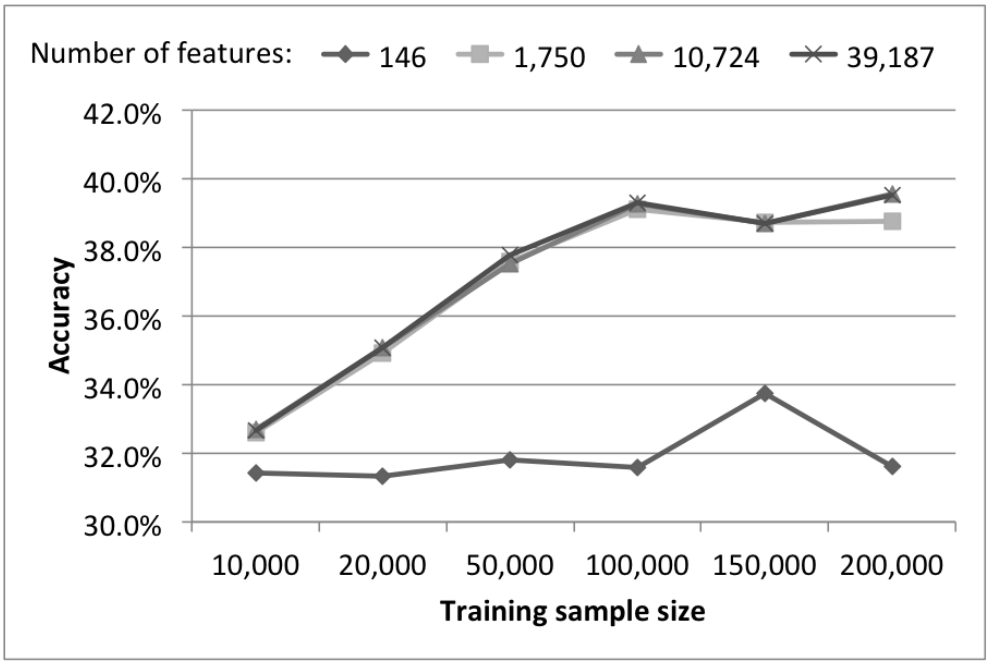


Figure 5.4: Classification accuracy as a function of the number of features and number of records for Social Service Notes.

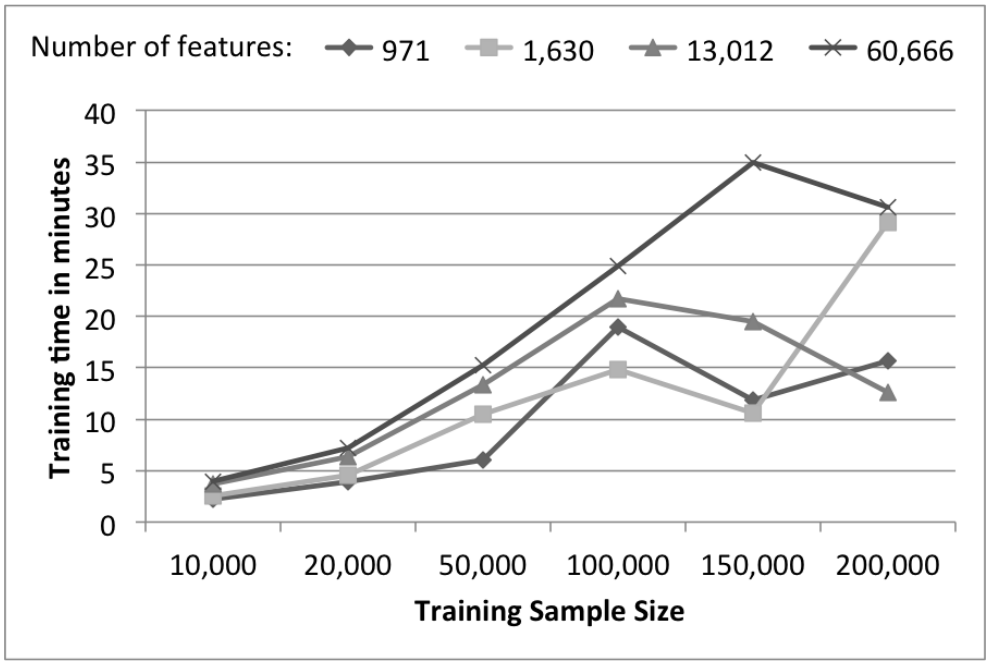


Figure 5.5: Training processing time as a function of the number of features and number of records for Admission History and Physical.

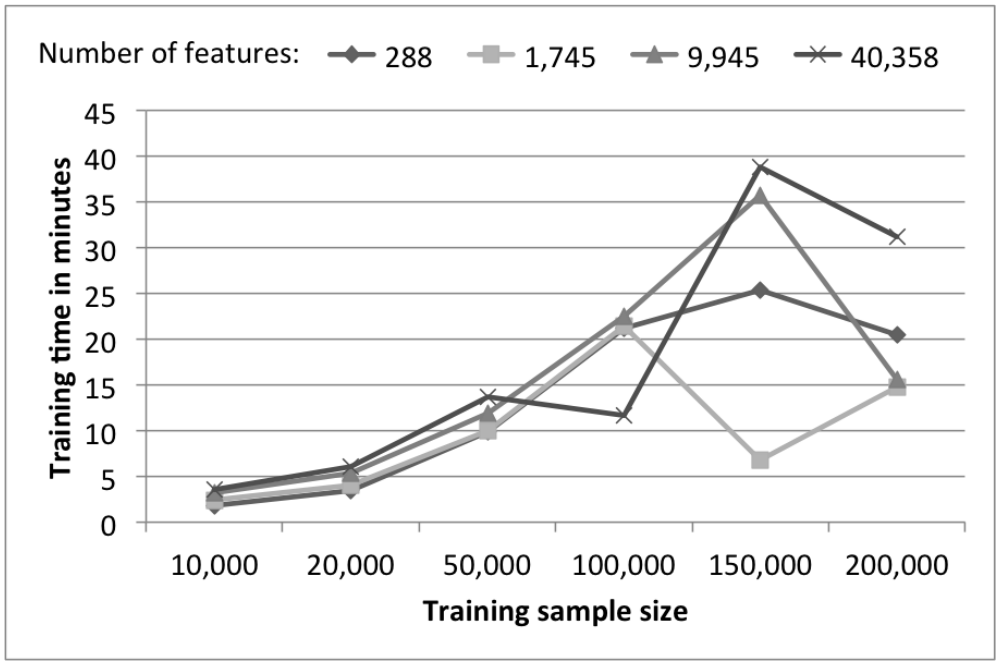


Figure 5.6: Training processing time as a function of the number of features and number of records for Cardiology Clinical Notes.

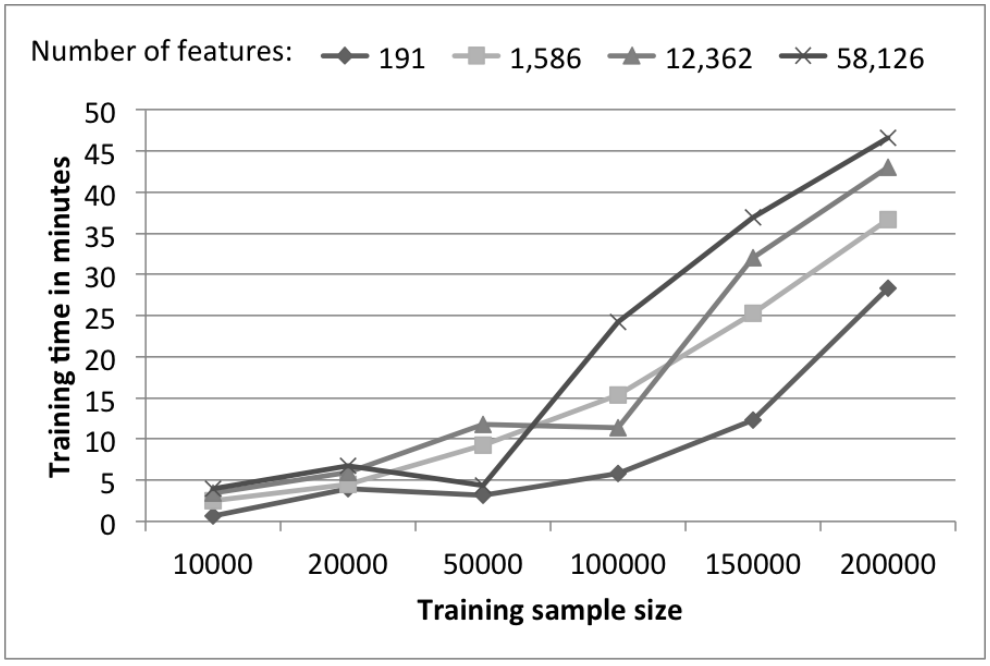


Figure 5.7: Training processing time as a function of the number of features and number of records for Discharge Summaries.

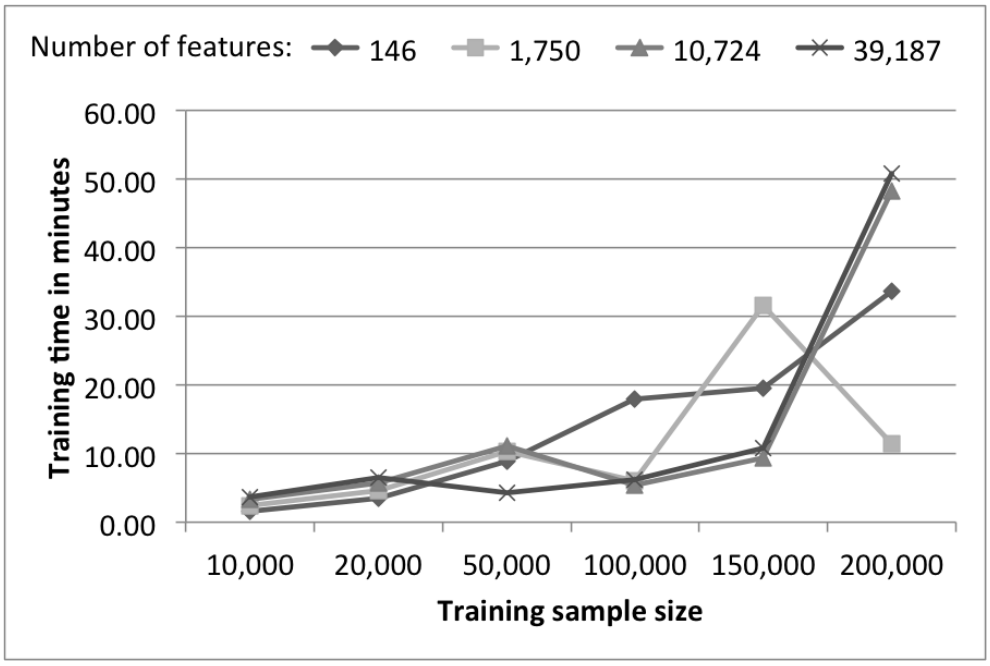


Figure 5.8: Training processing time as a function of the number of features and number of records for Social Service Notes.

corpus, it must be a legitimate semantic rule in the sublanguage grammar. If the system finds at least one matching pattern, the logistic regression classification prediction is ignored. To increase the probability that the pattern found for the term of interest is an appropriate pattern, I set the pattern format level to 2, thus requiring at least two unambiguously mapped terms to be included in the pattern. In this case, if the term of interest does not have two unambiguous neighbors within the predefined window, the semantic type predictions will be determined solely based on the classification predictions. This simple heuristic is logical and practical for initial implementation. However, a more sophisticated method of finding a balance between pattern matching output and classification predictions is needed. In machine learning, methods of combining several classification models are called "ensemble of classifiers" or "committee." Such methods are based on obtaining several models using different types of classification algorithms and then combining the predictions of these algorithm into a single final prediction [107]. Combining predictions is done by either averaging the output of all models, or selecting one of the model's output.

Another method to improve accuracy of predictions is to implement the S3 approach iteratively in order to acquire a more accurate sublanguage semantic schema. During the training phase, such a design would first learn a sublanguage model using unambiguously

mapped concepts. Then, the initially learned S3 model can be applied to disambiguate some additional terms. After that application, the S3 can be recalculated using unambiguously mapped terms as well as the disambiguated concepts, thus increasing the S3 coverage. Similarly, at the run time, the final S3 model can be used iteratively to increasingly disambiguate more terms.

5.3 Discussion

The third aim of this research focused on improving the initial S3 System performance. Firstly, I established that the feature space of unambiguous and ambiguous terms do not differ significantly. Therefore, the assumption that unambiguous examples provide sufficient language coverage and enable comprehensive semantic grammar acquisition is empirically supported.

Secondly, the error analysis revealed a number of possible system improvements that can improve the overall accuracy of a system based on S3. Some of the errors can be resolved through an improved and expanded knowledge repository and a more accurate parser and part of speech tagger. Other issues can be resolved by optimizing performance of the S3 System itself. For example, the set of experiments that I conducted in the course of my research indicated that a higher information gain threshold during the feature selection process would decrease the number of selected features. A smaller number of features produces a smaller dataset, which in turn enables the use of a larger number of training examples in order to provide a more accurate machine learning model.

The error analysis also revealed that there is a subset of errors that would be especially hard to resolve. These errors are the cases when the presented senses for a specific word are too vague or too similar for humans to agree on consistently. Thus, they represent the upper bound of a potential system performance [108, p. 267].

My work showed that while the current implementation of a system based on an automatically acquired sublanguage model does not produce perfect word sense disambiguation, a number of possible improvements can potentially make such a system highly accurate and suitable for language processing in a real-world clinical environment.

CHAPTER 6

DISCUSSION

The main goal of the current project is to suggest an automatic method of obtaining a domain specific knowledge base for word sense disambiguation and illustrate the feasibility of such a method. I achieved this goal by describing the sublanguage semantic schema method and demonstrating potential performance effectiveness of the proposed method as exemplified by the S3 System prototype. Even though the prototype is not a system that is ready to be implemented into a clinical setting, it serves as a proof-of-concept implementation that can be perfected with additional time and software development investment. The main contributions of my research are as follows:

1. The S3 method enables automatic semantic type pattern acquisition for the purposes of knowledge extraction and word sense disambiguation. This in turn makes domain adaptation of existing concept recognition systems feasible to any interested clinical organization due to significant reduction of human effort.
2. Comparison of the sublanguage semantic schemata of different clinical sublanguages informs researchers and practitioners of the sublanguage similarities, which in turn can inform terminology and ontology development.
3. Feature selection applied to all lexical features advances our knowledge on the salient clinical language features, which in turn supports development of new classification models for the purposes of information extraction and information retrieval from clinical text.

To argue the innovative nature of my S3 approach, I have to compare it to previous research that is based on similar methods and targets a similar research problem. There are several research efforts that are most closely related to the methods employed by the S3 approach:

- Similar to the S3 approach, Magnini and colleagues utilize domain information to facilitate word sense disambiguation [109]. In their research the discourse domain was determined based on the surrounding unambiguous terms. This approach relies on

both the “one domain per discourse“ and the “one sense per domain” assumptions. When the possible senses of a term belong to disjointed domains, this approach produces encouraging results. However, in the case of clinical notes, the difference between clinical domains is often quite subtle, therefore, disambiguation using their approach might not be as accurate. The “one sense per discourse” approach also fails when a word is used in multiple senses within the same discourse, which is often the case in clinical text.

- Similar to the S3 System, the selectional preference acquisition system proposed by McCarthy and colleagues has two phases: the training phase and run-time disambiguation phase [110]. Their system identifies predicate-argument pairs and obtains sense distributions using noun and verb senses from WordNet. The system employes Bayes' rule to estimate the probability of a specific verb occurring with a specific noun. Unlike the S3 System, their system requires an accurate parser and part of speech tagger, as well as anaphora resolution. The S3 System simply uses a linear sequence relationship for co-occurrence patterns because of the lack of access to an existing parser and part of speech tagger optimized for clinical text. Thus, the S3 System is more robust to the parser errors.
- Similar to my research, Sekine defined an automatic way to identify and describe a new sublanguage [111]. Unlike my approach, Sekine employed a clustering approach to group similar Wall Street Journal articles purely for the purposes of identifying similar text. Sekine's work did not result in a word sense disambiguation method.
- Similar to the S3 System, Schutze's approach attempts to acquire the knowledge base for word sense disambiguation automatically based on word co-occurrences [22]. Unlike S3 model, Schutze's approach is directed to acquire disambiguation models for each ambiguous word individually. Therefore, his approach is time consuming and is limited to only those words for which disambiguation models have been acquired.
- Similar to the S3 System, the method developed by Humphrey and colleagues relies on the UMLS semantic type of the possible senses in order to select the most appropriate concept for the term [59]. Unlike their approach, the S3 System does not rely on manually-selected sublanguage-specific keywords such as journal descriptors.
- Similar to the S3 System, CuiTools approach is unsupervised and is based on the term co-occurrences for each ambiguous word [112]. Unlike CuiTools approach, the S3 method creates one disambiguation model for all ambiguous words, whereas CuiTools

creates individual models for each word. So even though CuiTools employs an unsupervised approach, the method requires identification of ambiguous words and the creation of a set of disambiguation models, one for each ambiguous word.

- Similar to the S3, the vector approach developed by McInnes attempts to perform all-words disambiguation in an unsupervised manner using the UMLS as the knowledge base [113]. Her approach is based on comparing the text of the UMLS CUI and ST definitions and the context of the ambiguous term. Unlike the S3 approach, McInnes's method relies on computationally expensive analysis during application phase. As such, this approach is not feasible to implement as a real-time disambiguation module. In contrast, the S3 approach performs most of computations during training time, thus creating a static knowledge base that can provide near real-time concept recognition.
- Similar to the S3, a disambiguation model proposed by Stevenson and Guo employs UMLS Metathesaurus to automatically generate training examples using unambiguous terms [114]. Their approach, called monosemous relatives, identifies a set of lexical substitutes for each sense of the ambiguous word of interest such that they can be mapped unambiguously to their related sense. To generate the training dataset, the unambiguous terms are substituted with their ambiguous relatives. This approach is automatic and, therefore, different disambiguation models could be learned for each clinical domain. However, it relies on availability of a large corpus that contains instances of variable expression of similar ideas. Unlike monosemous relatives approach, the S3 approach targets sublanguages that tend to express similar ideas similarly, which is characteristic of narrow domains such as clinical domains. Also, the S3 approach can be applied even if the available raw corpus is relatively small. For example, the set of Emergency Department Reports that was extracted for this project from the University of Utah Hospital data warehouse contained only 685 reports. In addition, Stevenson and Guo's approach uses 15 most common unambiguous co-occurring concepts. The co-occurrence information used in their approach is found in MRCOC table, which is a part of UMLS Metathesaurus and is specific to Medline abstracts. Unlike their approach, the S3 method obtains co-occurrence information for each clinical domain separately, thus optimizing that information for each domain.
- Similar to the S3, the disambiguation system developed by Stevenson and Guo aims to perform word sense disambiguation applying several existing machine learning algorithms in order to learn a sense classifier using lexical and semantic features

extracted from the context of each ambiguous word [60]. Their approach focuses on biomedical text and uses UMLS CUIs, as well as MESH terms, as features. Unlike their system, the S3 approach does not rely on manually annotated text or long form expansions, which often precede abbreviations in biomedical text and which are not available in clinical narrative, for the purposes of obtaining the disambiguation model. Also, as I stated before, the S3 approach is an "all-words" disambiguation method and does not require manual identification of ambiguous words.

6.1 Limitations

The error analysis presented in Section 5.1 revealed that the S3 System performance is affected by the limitations of the used knowledge base (UMLS) and concept-recognition system (MetaMap), as well as the limitations of the implemented system design. A number of steps can be developed to improve the S3 System design shortcomings. However, the UMLS and MetaMap limitations are out of reach of the potential developers of a system based on the S3 approach. Therefore, the Sublanguage Semantic Schema approach has several limitations that are inherent to the approach rather than to a specific implementation.

First, the S3 method assumes that most unambiguously mapped concepts supply the correct sense for the mapped term. However, an error analysis indicated that 4 to 10% of all unambiguous mappings are incorrect, as seen in Table 5.2. This results in an error that the S3 approach cannot overcome regardless of how well it is optimized for the domain of interest. Such mismapping results from inadequacy of the vocabulary used in concept mapping. Identifying those mismapped terms is an important step in domain adaptation of clinical NLP systems. Automatically, these terms can be identified through the S3 System, by comparing the list of semantic types in the term's candidate set to the list of potential semantic types suggested by the S3 System. If these two sets do not overlap, the term is potentially mismapped. Adequacy of terminology can be reviewed manually by experts.

Second, the S3 method assumes that those unambiguous terms, which were missmapped, are uniformly distributed and, therefore, will not lead to strong co-occurrence patterns. This assumption can be evaluated using a manually annotated corpus. However, such a corpus would have to be relatively large in order to identify a practically useful number of mismapped terms to determine whether they would lead to strong patterns.

Third, the S3 method assumes that those terms, which are mapped to a set of concepts with the same semantic types and receive the same mapping confidence score (such as

MappingScore from MetaMap), are synonyms and, therefore, equal. As a result, the S3 approach selects one of the concepts randomly and presents it as the corresponding disambiguation. The impact of this assumption is not known at this time and can be evaluated by human experts as a future project. There have been attempts to perform disambiguation of terms with the same semantic type. A variation of one of those approaches can potentially be implemented as a part of a system based on the S3 approach [113, 115].

6.2 Opportunities for Future Work

My ultimate vision is to develop an electronic medical record system that would accept clinical notes in a form of free text and extract patient clinical data into structured information to be used for various decision making, recording, reporting, quality assurance, and surveillance purposes. I plan to evaluate sublanguage grammar variations among sections of clinical notes to identify similarities across sections of different note types. My future work will involve a tight collaboration with clinical organizations that share my vision.

While in the current design of the S3 System prototype, MetaMap and UMLS Metathesaurus limitations are not addressed, one potential improvement to the S3 System is to enrich the knowledge base with general English vocabulary. WordNet, a large, publicly available lexical database can be queried to improve the sense inventory and increase the probability that it includes the correct sense for each term. This combination of UMLS Metathesaurus and WordNet would also have a negative side effect of decreasing the number of unambiguously mapped terms.

Another major improvement of the system would be to include a sublanguage classification module that would determine what sublanguage a specific note belongs to. This module can be implemented by first clustering a training set and determining sublanguage boundaries, and then during run time, the system can analyze each incoming note and identify the cluster (and, therefore, the sublanguage), to which this note is the most similar.

Another potential system improvement step involves changing the granularity of the sublanguage definition. It has been noted that focusing on a specific section in a clinical note improves performance of an information extraction tool [116]. Therefore, breaking notes by sections and identifying sublanguage clusters using sections of documents rather than full notes might result in more focused and restricted sublanguages. In conjunction with the sublanguage classification module mentioned above, the lower granularity sublanguage definition can greatly improve the overall system performance.

CHAPTER 7

CONCLUSION

Accurate information extraction from real-world clinical texts depends on effective word sense disambiguation. Language in the clinical domain changes and expands with time. The UMLS Metathesaurus is a valuable knowledge repository that is updated on a regular basis. Any clinical language processing system that makes use of an evolving knowledge source like UMLS has a long-term advantage (i.e., it is less likely to grow stale). This work demonstrates the following:

- The language of the clinical domain is not uniform, but rather a collection of distinct sublanguages. This work demonstrates for the first time that the clinical sublanguage boundaries align with the clinical subdomains rather than clinical setting in which the narrative originated.
- Modifying a WSD tool for each new clinical subdomain improves its accuracy.
- Automatic acquisition of domain information structure will save time and financial resources as new clinical sublanguages are added or as old ones evolve.
- A hybrid disambiguation algorithm that utilizes a manually curated knowledge base (UMLS) and one that requires little effort from the user to leverage, has a potential to be highly accurate for word sense disambiguation in the long run.

Automatic domain adaptation of a concept recognition system saves time and money by creating a knowledge base for word sense disambiguation that is optimized for a specific type of narrative.

APPENDIX A

SEMANTIC TYPES

The following is a list of semantic types included in the UMLS Semantic Network.

acab — Acquired Abnormality

acty — Activity

aggp — Age Group

alga — Alga

amas — Amino Acid Sequence

aapp — Amino Acid, Peptide, or Protein

amph — Amphibian

anab — Anatomical Abnormality

anst — Anatomical Structure

anim — Animal

antb — Antibiotic

arch — Archaeon

bact — Bacterium

bhvr — Behavior

biof — Biologic Function

bacs — Biologically Active Substance

bmod — Biomedical Occupation or Discipline

bodm — Biomedical or Dental Material

bird — Bird

blor — Body Location or Region

bpoc — Body Part, Organ, or Organ Component

bsoj — Body Space or Junction

bdsu — Body Substance

bdsy — Body System

carb — Carbohydrate

crbs — Carbohydrate Sequence

cell — Cell

celc — Cell Component

celf — Cell Function

comd — Cell or Molecular Dysfunction

chem — Chemical

chvf — Chemical Viewed Functionally

chvs — Chemical Viewed Structurally

clas — Classification

clna — Clinical Attribute

clnd — Clinical Drug

cnce — Conceptual Entity

cgab — Congenital Abnormality

dora — Daily or Recreational Activity

diap — Diagnostic Procedure

dsyn — Disease or Syndrome

drdd — Drug Delivery Device

edac — Educational Activity

eico — Eicosanoid

elii — Element, Ion, or Isotope

emst — Embryonic Structure

enty — Entity

eehu — Environmental Effect of Humans

enzy — Enzyme

evnt — Event

emod — Experimental Model of Disease

famg — Family Group

fndg — Finding

fish — Fish

food — Food

ffas — Fully Formed Anatomical Structure

ftcn — Functional Concept

fngs — Fungus

gngp — Gene or Gene Product (pseudo ST for gene terminology)

gngm — Gene or Genome

genf — Genetic Function

geoa — Geographic Area

gora — Governmental or Regulatory Activity

grup — Group

grpa — Group Attribute

hops — Hazardous or Poisonous Substance

hlca — Health Care Activity

hcro — Health Care Related Organization

horm — Hormone

humn — Human

hcpp — Human-caused Phenomenon or Process

idcn — Idea or Concept

imft — Immunologic Factor

irda — Indicator, Reagent, or Diagnostic Aid

inbe — Individual Behavior

inpo — Injury or Poisoning

inch — Inorganic Chemical

inpr — Intellectual Product

invt — Invertebrate

lbpr — Laboratory Procedure

lbtr — Laboratory or Test Result

lang — Language

lipd — Lipid

mcha — Machine Activity

mamm — Mammal

mnob — Manufactured Object

medd — Medical Device

menp — Mental Process

mobd — Mental or Behavioral Dysfunction

mbrt — Molecular Biology Research Technique

moft — Molecular Function

mosq — Molecular Sequence

npop — Natural Phenomenon or Process

neop — Neoplastic Process

nsba — Neuroreactive Substance or Biogenic Amine

nnon — Nucleic Acid, Nucleoside, or Nucleotide

nusq — Nucleotide Sequence

ocdi — Occupation or Discipline

ocac — Occupational Activity

ortf — Organ or Tissue Function

orch — Organic Chemical

orgm — Organism

orga — Organism Attribute

orgf — Organism Function

orgt — Organization

opco — Organophosphorus Compound

patf — Pathologic Function

podg — Patient or Disabled Group

phsu — Pharmacologic Substance

phpr — Phenomenon or Process

phob — Physical Object

phsf — Physiologic Function

plnt — Plant

popg — Population Group

pros — Professional Society

prog — Professional or Occupational Group

qlco — Qualitative Concept

qnco — Quantitative Concept

rcpt — Receptor

rnlw — Regulation or Law

rept — Reptile

resa — Research Activity

resd — Research Device

rich — Rickettsia or Chlamydia

shro — Self-help or Relief Organization

sosy — Sign or Symptom

socb — Social Behavior

spco — Spatial Concept

strd — Steroid

sbst — Substance

tmco — Temporal Concept

topp — Therapeutic or Preventive Procedure

tisu — Tissue

vtbt — Vertebrate

virs — Virus

vita — Vitamin

APPENDIX B

INDEX

Glossary

ambiguous term a term that was mapped to multiple candidates. 33

candidate one of the concepts that represent a potential meaning of the mapped term. MetaMap identifies multiple candidates that are combined into a candidate set for each phrase. Disambiguation of the candidates is a task required for accurate mapping. 33

concept a UMLS concept identified by MetaMap. 33

mapped term a term that was mapped by MetaMap to at least one candidate. 33

mapping a term that was *unambiguously* mapped to a UMLS concept using the unambiguity heuristics. The mapping has a UMLS concept identifier and a semantic type associated with it. 33

mismapped term a term that was mapped by MetaMap to at least one candidate but none of the candidates represented the correct meaning of the term in a given context. 44, 53, 66

term One or more semantically linked tokens that MetaMap attempts to map. 33

token The smallest lexical unit analyzed by MetaMap. Includes words, numbers, and punctuation. 33

Acronyms

cTAKES clinical Text Analysis and Knowledge Extraction System. 14, 18

DIS Domain Information Schema. 33

EMR Electronic Medical Record. 1, 2

HITEx Health Information Text Extraction. 14, 18

MedEx Medical Information Extraction System. 19

MedLEE Medical Language Extraction and Encoding System. 18, 19

MeSH Medical Subject Headings. 12, 13

NLM National Library of Medicine. 12, 13

NLP Natural Language Processing. 2, 3, 5, 10, 12, 13, 15, 17, 66

S3 Sublanguage Semantic Schema. 1, 33, 34, 36, 62, 66

S3 System Sublanguage Semantic Schema System. 33–35, 40, 43–46, 48, 49

ST Semantic Type. 17, 33

UMLS Unified Medical Language System. 8, 12, 13, 17, 34

WSD Word Sense Disambiguation. 6, 7, 9, 15, 17, 35, 68

REFERENCES

[1] Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting information from textual documents in the electronic health record: a review of recent research. Yearbook of medical informatics. 2008; p. 128–44.

[2] Nadkarni PM, Ohno-Machado L, Chapman WW. Natural language processing: an introduction. Journal of the American Medical Informatics Association. 2011 Aug; 18(5):544–551.

[3] McDonald C, Huff S, Mercer K, Hernandez JA, Vreeman DJ. Logical Observation Identifiers Names and Codes (LOINC) Users' Guide; 2011.

[4] Demner-Fushman D, Chapman WW, McDonald CJ. What can natural language processing do for clinical decision support? Journal of biomedical informatics. 2009 Oct; 42(5):760–72.

[5] Johnson SB, Bakken S, Dine D, Hyun S, Mendonça E, Morrison F, et al. An electronic health record based on structured narrative. Journal of the American Medical Informatics Association: JAMIA. 2008; 15(1):54–64.

[6] McDonald CJ. The barriers to electronic medical record systems and how to overcome them. Journal of the American Medical Informatics Association: JAMIA. 1997; 4(3):213–21.

[7] Marcus MP, Marcinkiewicz MA, Santorini B. Building a large annotated corpus of English: the penn treebank. Computational Linguistics. 1993 Jun; 19(2):313–330.

[8] Schubert L, Tong M. Extracting and evaluating general world knowledge from the Brown corpus. In: Proceedings of the HLT-NAACL 2003 workshop on Text meaning. vol. 9. Morristown, NJ, USA: Association for Computational Linguistics; 2003. p. 7–13.

[9] Navigli R. Word sense disambiguation. ACM Computing Surveys. 2009 Feb; 41(2):1–69.

[10] Bernstam EV, Smith JW, Johnson TR. What is biomedical informatics? Journal of biomedical informatics. 2010 Feb; 43(1):104–10.

[11] Kay M. A Life of Language. Computational Linguistics. 2005 Dec; 31(4):425–438.

[12] Weaver W. Translation. In: Pierce J, editor. Language and machines: computers in translation and linguistics. Publication (National Research Council (U.S.)), no. 1416. Washington, DC, USA: National Academy of Sciences, National Research Council; 1966.

[13] Mihalcea R. Performance analysis of a part of speech tagging task. In: CICLing'03 Proceedings of the 4th international conference on Computational linguistics and intelligent text processing; 2003. p. 158–167.

[14] Krauthammer M, Nenadic G. Term identification in the biomedical literature. Journal of biomedical informatics. 2004 Dec; 37(6):512–26.

[15] McCarthy D. Word Sense Disambiguation: An Overview. Language and Linguistics Compass. 2009; 3(2):537–558.

[16] Navigli R. A Quick Tour of Word Sense Disambiguation, Induction and Related Approaches. In: SOFSEM 2012: Theory and practice of computer science. vol. 7147 of Lecture Notes in Computer Science; 2012. p. 115–129.

[17] Agirre E, Edmonds P. Word Sense Disambiguation. vol. 33 of Text, Speech and Language Technology. Agirre E, Edmonds P, editors. Dordrecht: Springer Netherlands; 2007.

[18] Fujii A, Inui K, Tokunaga T, Tanaka H. Selective Sampling for Example-based Word Sense Disambiguation. Computational Linguistics. 1999 Oct; 24(4):573–597.

[19] Dunham G. 10. In: Grishman R, Kittredge RI, editors. The role of syntax in the sublanguage of medical diagnostic statements. Lawrence Erlbaum Associates; 1986. p. 175–194.

[20] Basili R, Pazienza MT, Velardi P. Acquisition of selectional patterns in sublanguages. Machine Translation. 1993 Sep; 8(3):175–201.

[21] Ng HT, Goh WB, Low KL. Feature selection, perceptron learning, and a usability case study for text categorization. In: SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM; 1997. p. 67–73.

[22] Schutze H. Automatic word sense discrimination. Comput Linguist. 1998; 24(1):97–123.

[23] Pantel P, Lin D. Discovering word senses from text. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02. New York, New York, USA: ACM Press; 2002. p. 613.

[24] Mihalcea RF, Moldovan DI. A Highly Accurate Bootstrapping Algorithm For Word Sense Disambiguation. International journal on artificial intelligence tools. 2001; 10(1-2):5–21.

[25] Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods. In: ACL '95 Proceedings of the 33rd annual meeting on Association for Computational Linguistics; 1995. .

[26] Zhu J. Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem. In: Proceedings of ACL; 2007. p. 783–790.

[27] Chan YS, Ng HT. Domain Adaptation with Active Learning for Word Sense Disambiguation. In: Proceedings of the 45th Annual Meeting of the Association of

Computational Linguistics. Prague: Association for Computational Linguistics; 2007. p. 49–56.

[28] Fan JW, Friedman C. Deriving a probabilistic syntacto-semantic grammar for biomedicine based on domain-specific terminologies. Journal of biomedical informatics. 2011 Oct; 44(5):805–14.

[29] Agirre E, Lacalle OLD, Soroa A. Knowledge-Based WSD on Specific Domains: Performing Better than Generic Supervised WSD. In: Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09); 2009.

[30] Mykowiecka A, Marciniak M, Kupść A. Rule-based information extraction from patients' clinical data. Journal of biomedical informatics. 2009 Oct; 42(5):923–36.

[31] Karambelkar S. Acquisition of selectional preferences in natural language processing [Dissertation]. University of Sheffield; 2001.

[32] Hirschman L, Grishman R, Sager N. From text to structured information. In: Proceedings of the June 7-10, 1976, national computer conference and exposition on - AFIPS '76. New York, New York, USA: ACM Press; 1976. p. 267.

[33] Lesk M. Automatic sense disambiguation using machine readable dictionaries. New York, New York, USA: ACM Press; 1986.

[34] Soler S, Montoyo A. A Proposal for WSD Using Semantic Similarity. In: Gelbukh A, editor. Computational Linguistics and Intelligent Text Processing. vol. 2276 of Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg; 2002. p. 51–67.

[35] Gliozzo AM, Ranieri M, Strapparava C. Crossing Parallel Corpora and Multilingual Lexical Databases for WSD. In: Gelbukh A, editor. Computational Linguistics and Intelligent Text Processing. vol. 3406 of Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005. p. 242–245.

[36] Hawkins P, Nettleton D. Large Scale WSD Using Learning Applied to SENSEVAL. Computers and the Humanities. 2000; 34(1):135 – 140.

[37] Mihalcea R, Csomai A. SenseLearner: word sense disambiguation for all words in unrestricted text. In: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions - ACL '05. Morristown, NJ, USA: Association for Computational Linguistics; 2005. p. 53–56.

[38] McInnes BT. Supervised and Knowledge-based Methods for Disambiguating Terms in Biomedical Text using the UMLS and MetaMap; 2009.

[39] Harris ZS. A Theory of Language and Information: A Mathematical Approach. Clarendon Press; 1991.

[40] Harris ZS. A grammar of English on mathematical principles. New York: Wiley; 1982.

[41] Fan JW, Friedman C. Word sense disambiguation via semantic type classification. AMIA Annual Symposium proceedings. 2008; p. 177–81.

[42] Wilms GJ, Boggess LC. Automated Induction of a Lexical Sublanguage Grammar Using a Hybrid System of Corpus and Knowledge-Based Techniques. Mississippi State University, Department of Computer Science; 1995.

[43] Taira RK. Natural Language Processing of Medical Reports. In: Medical Imaging Informatics. vol. 3. Springer US; 2010; p. 257–298.

[44] Symonenko S, Rowe S, Liddy ED. Illuminating Trouble Tickets with Sublanguage Theory. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL. New York, NY, USA; 2006. p. 169–172.

[45] Isabelle P, Bourbeau L. TAUM-AVIATION: its technical features and some experimental results. Comput Linguist. 1985; 11(1):18–27.

[46] Kittredge R, Mel'Cuk I. Towards a computable model of meaning-text relations within a natural sublanguage. Karlsruhe, West Germany: Morgan Kaufmann Publishers Inc.; 1983. p. 657–659.

[47] Lehrberger J. 3. In: Kittredge RI, Lehrberger J, editors. Automatic translation and the concept of sublanguage. Berlin: Walter de Gruyter; 1982. p. 81–106.

[48] Sager N, Lyman M, Nhàn NT, Tick LJ. Medical language processing: applications to patient data representation and automatic encoding. Methods of information in medicine. 1995 Mar; 34(1-2):140–6.

[49] Harris ZS. String analysis of sentence structure. Papers on formal linguistics. The Hague: Mouton; 1962.

[50] Grishman R, Sager N, Raze C, Bookchin B. The linguistic string parser. New York, New York, USA: ACM Press; 1973.

[51] Sager N, Grishman R. The restriction language for computer grammars of natural language. Commun ACM. 1975 Jul; 18(7):390–400.

[52] Fitzpatrick E, Bachenko J, Hindle D. 3. In: Grishman R, Kittredge RI, editors. The status of telegraphic sublanguages. Lawrence Erlbaum Associates; 1986. p. 39–51.

[53] Friedman C, Kra P, Rzhetsky A. Two biomedical sublanguages: a description based on the theories of Zellig Harris. Journal of biomedical informatics. 2002; 35(4):222–35.

[54] Friedman C, Cimino JJ, Johnson SB. A conceptual model for clinical radiology reports. In: Proc Annu Symp Comput Appl Med Care. Queens College of the City University of New York.. American Medical Informatics Association; 1993. p. 829–833.

[55] Friedman C, Alderson PO, Austin JH, Cimino JJ, Johnson SB. A general natural-language text processor for clinical radiology. Journal of the American Medical Informatics Association: JAMIA. 1994; 1(2):161–74.

[56] Friedman C. A broad-coverage natural language processing system. AMIA Annual Symposium proceedings. 2000; p. 270–274.

[57] Weeber M, Mork JG, Aronson AR. Developing a test collection for biomedical word sense disambiguation. AMIA Annual Symposium proceedings. 2001; p. 746–50.

[58] Rindflesch TC, Aronson AR. Ambiguity resolution while mapping free text to the UMLS Metathesaurus. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. Symposium on Computer Applications in Medical Care. National Library of Medicine, Bethesda, MD 20894; 1994; p. 240–4.

[59] Humphrey SM, Rogers WJ, Kilicoglu H, Demner-Fushman D, Rindflesch TC. Word Sense Disambiguation by Selecting the Best Semantic Type Based on Journal Descriptor Indexing: Preliminary Experiment. Journal of the American Society for Information Science and Technology (Print). 2006 Jan; 57(1):96–113.

[60] Stevenson M, Guo Y. Disambiguation in the biomedical domain: The role of ambiguity type. Journal of biomedical informatics. 2010 Sep; .

[61] Liu H, Johnson SB, Friedman C. Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS. Journal of the American Medical Informatics Association: JAMIA. 2002 Jul; 9(6):621–36.

[62] Haug PJ, Ranum DL, Frederick PR. Computerized extraction of coded findings from free-text radiologic reports. Work in progress. Radiology. 1990 Feb; 174(2):543–8.

[63] Koehler SB. Symtext: a natural language understanding system for encoding free text medical data; 1998.

[64] Christensen LM, Haug PJ, Fiszman M. MPLUS: a probabilistic medical language understanding system. In: Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain. Morristown, NJ, USA: Association for Computational Linguistics; 2002. p. 29–36.

[65] Meystre S, Haug PJ. Automation of a problem list using natural language processing. BMC Med Inform Decis Mak. 2005; 5:30.

[66] Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. Journal of the American Medical Informatics Association: JAMIA. 2010; 17(5):507–13.

[67] Coden A, Savova G, Sominsky I, Tanenblatt M, Masanz J, Schuler K, et al. Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model. Journal of biomedical informatics. 2009 Oct; 42(5):937–49.

[68] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. Evaluation of negation phrases in narrative clinical reports. AMIA Annual Symposium proceedings. 2001; p. 105–109.

[69] Zeng QT, Goryachev S, Weiss S, Sordo M, Murphy SN, Lazarus R. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. BMC medical informatics and decision making. 2006 Jan; 6:30.

[70] Uzuner O, Luo Y, Szolovits P. Evaluating the state-of-the-art in automatic de-identification. Journal of the American Medical Informatics Association: JAMIA. 2007; 14(5):550–63.

[71] Uzuner O. Second i2b2 workshop on natural language processing challenges for clinical records. AMIA Annual Symposium proceedings. 2008; p. 1252–3.

[72] Uzuner O, Goldstein I, Luo Y, Kohane I. Identifying patient smoking status from medical discharge records. J Am Med Inform Assoc. 2008; 15(1):14–24.

[73] Uzuner O, Solti I, Cadag E. Extracting medication information from clinical text. Journal of the American Medical Informatics Association: JAMIA. 2010; 17(5):514–8.

[74] Uzuner O, South BR, Shen S, Duvall SL. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. Journal of the American Medical Informatics Association: JAMIA. 2011; 18(5):552–6.

[75] Grishman R, Hirschman L, Nhan NT. Discovery procedures for sublanguage selectional patterns: initial experiments. Comput Linguist. 1986; 12(3):205–215.

[76] Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. Journal of the American Medical Informatics Association: JAMIA. 2010 May; 17(3):229–36.

[77] MetaMap; 2009. http://metamap.nlm.nih.gov/.

[78] Heinze DT, Morsch ML, Potter BC, Sheffer RE. Medical i2b2 NLP smoking challenge: the A-Life system architecture and methodology. Journal of the American Medical Informatics Association: JAMIA. 2008; 15(1):40–3.

[79] Denny JC, Irani PR, Wehbe FH, Smithers JD, Spickard A. The KnowledgeMap project: development of a concept-based medical school curriculum database. AMIA Annual Symposium proceedings. 2003; p. 195–199.

[80] Xu H, Stenner SP, Doan S, Johnson KB, Waitman LR, Denny JC. MedEx: a medication information extraction system for clinical narratives. Journal of the American Medical Informatics Association: JAMIA. 2010; 17(1):19–24.

[81] Patterson O, Hurdle JF. Document clustering of clinical narratives: a systematic study of clinical sublanguages. AMIA Annual Symposium proceedings. 2011; p. 1099–107.

[82] Patterson O, Igo S, Hurdle JF. Automatic acquisition of sublanguage semantic schema: towards the word sense disambiguation of clinical narratives. AMIA Annual Symposium proceedings. 2010; p. 612–6.

[83] Garla V, Re VL, Dorey-Stein Z, Kidwai F, Scotch M, Womack J, et al. The Yale cTAKES extensions for document classification: architecture and application. Journal of the American Medical Informatics Association: JAMIA. 2011 May; 18(5):614–20.

[84] Ponsporrata A, Berlangallavori R, Ruizshulcloper J. Topic discovery based on text mining techniques. Information Processing & Management. 2007 May; 43(3):752–768.

[85] Paaß G, Kindermann J, Leopold E. Learning Prototype Ontologies by Hierachical Latent Semantic Analysis. In: ECML/PKDD 2004 Workshop on Knowledge Discovery and Ontologies; 2004.

[86] Liu T, Liu S, Chen Z. An Evaluation on Feature Selection for Text Clustering. In: Proceedings of the Twentieth International Conference on Machine Learning. Washington, DC, USA: In ICML; 2003; p. 488–495.

[87] Xu R, Wunsch D. Survey of clustering algorithms. IEEE transactions on neural networks. IEEE Neural Networks Council. 2005 May; 16(3):645–78.

[88] Jayabharathy J, Kanmani S, Parveen AA. A survey of document clustering algorithms with topic discovery. Journal of Computing. 2011; 3(2):21–27.

[89] Sparck Jones K. Index term weighting. Information Storage and Retrieval. 1973 Nov; 9(11):619–633.

[90] Jurafsky D, Martin JH. Speech and Language Processing. 2nd ed. Pearson Prentice Hall; 2009.

[91] Zhao Y, Karypis G. Data clustering in life sciences. Molecular biotechnology. 2005 Sep; 31(1):55–80.

[92] Zaïane OR, Foss A, Lee CH, Wang W. On Data Clustering Analysis: Scalability, Constraints, and Validation. In: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining; 2002; p. 28–39.

[93] Steinbach M, Karypis G, Kumar V. A Comparison of Document Clustering Techniques — Karypis Lab. In: KDD Workshop on Text Mining; 2000.

[94] Browne AC, Divita G, Lu C, McCreedy L, Nace D. Lexical Systems; A report to the Board of Scientific Counselors; 2003.

[95] Harris ZS. The structure of science information. Journal of biomedical informatics. 2002 Aug; 35(4):215–21.

[96] Grishman R, Hirschman L, Friedman C. Natural language interfaces using limited semantic information. In: Proceedings of the 9th conference on Computational linguistics. Czechoslovakia: Academia Praha; 1982; p. 89–94.

[97] Aronson AR. Metamap: Mapping text to the UMLS Metathesaurus; 2006.

[98] O'Dwyer B. Modern English structures: form, function, and position. 2nd ed. Peterborough Ont.: Broadview Press; 2006.

[99] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software. ACM SIGKDD Explorations Newsletter. 2009 Nov; 11(1):10.

[100] Daumé H. Notes on CG and LM-BFGS Optimization of Logistic Regression; 2004.

[101] Komarek P, Moore A. Fast Robust Logistic Regression for Large Sparse Datasets with Binary Outputs. In: Ninth International Workshop on Artificial Intelligence and Statistics. Artificial Intelligence and Statistics. Society for Artificial Intelligence and Statistics; 2003.

[102] South BR, Leng J, Anderson K, Shen S, Thibault J, DuVall S. The Extensible Human Oracle Suite of Tools (eHOST) for Pre-Annotation of Clinical Narratives.

In: BioCreative: Critical Assessment of Information Extraction in Biology Annual Conference; 2010.

[103] Miller GA. WordNet: A Lexical Database for English. Communications of the ACM. 1995; 38:39–41.

[104] Mougin F, Burgun A, Bodenreider O. Using WordNet to improve the mapping of data elements to UMLS for data sources integration. AMIA Annual Symposium proceedings. 2006; p. 574–8.

[105] Smith L, Rindflesch TC, Wilbur WJ. MedPost: a part-of-speech tagger for bioMedical text. Bioinformatics. 2004 Sep; 20(14):2320–2321.

[106] Xu H, Markatou M, Dimova R, Liu H, Friedman C. Machine learning and word sense disambiguation in the biomedical domain: design and evaluation issues. BMC bioinformatics. 2006 Jan; 7:334.

[107] Bishop CM. Pattern Recognition and Machine Learning. Secaucus, NJ, USA: Springer-Verlag New York, Inc.; 2006.

[108] Clark A, Fox C, Lappin S. The Handbook of Computational Linguistics and Natural Language Processing (Google eBook). John Wiley and Sons; 2010.

[109] Magnini B, Strapparava C, Pezzulo G, Gliozzo A. The role of domain information in Word Sense Disambiguation. Natural Language Engineering. 2002 Dec; 8(4):359–373.

[110] Mccarthy D, Carroll J, Preiss J. Disambiguating Noun and Verb Senses Using Automatically Acquired Selectional Preferences. In: Proceedings of the SENSEVAL-2 Workshop at ACL/EACL'01. 2001; p. 119–122.

[111] Sekine S. Automatic Sublanguage Identification for a New Text. In: Second Annual Workshop on Very Large Corpora; 1994. p. 109–120.

[112] McInnes BT, Pedersen T, Carlis J. Using UMLS Concept Unique Identifiers (CUIs) for word sense disambiguation in the biomedical domain. AMIA Annual Symposium proceedings. 2007; p. 533–7.

[113] Mcinnes BT. An Unsupervised Vector Approach to Biomedical Term Disambiguation: Integrating UMLS and Medline; 2008.

[114] Stevenson M, Guo Y. Disambiguation of ambiguous biomedical terms using examples generated from the UMLS Metathesaurus. Journal of biomedical informatics. 2010 Oct; 43(5):762–73.

[115] Tran N, Luong T, Krauthammer M. Mapping terms to UMLS concepts of the same semantic type. AMIA Annual Symposium proceedings. 2007; p. 1136.

[116] Wang X, Chase H, Markatou M, Hripcsak G, Friedman C. Selecting information in electronic health records for knowledge acquisition. Journal of biomedical informatics. 2010 Aug; 43(4):595–601.