ASSISTED ANNOTATION OF BIOMEDICAL TEXT USING RapTAT, AN ONLINE LEARNING-BASED TOOL

By

Glenn T. Gobbel

Thesis

Submitted to the Faculty of the
Graduate School of Vanderbilt
University in partial fulfillment
of the requirements for the
degree of

MASTER OF SCIENCE

in

Biomedical Informatics

December, 2013

Nashville, Tennessee

Approved:

Michael E. Matheny, M.D., M.S., M.P.H.

Steven H. Brown, M.D., M.S.

Dario Giuse, Dr. Ing.

ACKNOWLEDGMENTS

This work would not have been possible without the support of the Department of Veterans Affairs Medical Informatics Fellowship Program (Sponsored by Office of Academic Affiliations, Office of Health Information, and HSR&D). This work was also supported by grants HIR 09-001 and HIR 09-003 from the VA Consortium for Health Informatics Research (CHIR) group.

I am grateful to all the faculty, students, and staff within both the Vanderbilt University Department of Biomedical Informatics and the Department of Veterans Affairs Tennessee Valley Health System. It has honestly been an honor and inspiration to work with everyone there. I am particularly appreciative of the members of my Thesis Committee, Dr. Michael Matheny, Dr. Steve Brown, and Dr. Dario Giuse, who provided training, support and guidance throughout my graduate work. Dr. Michael Matheny, the chairman of my committee, has provided exceptional guidance, feedback and opportunities to work in areas of exciting new areas of biomedical informatics. Dr. Steve Brown provided me with the opportunity to change the course of my career and work with an exceptional team of scientists in the field of informatics. Dr. Dario Giuse provided critical training with regard to clinical information systems and inspired me by sharing the steps required in generating the health information system created using his knowledge and vision. I would also like to thank the two VA Medical Informatics Fellows who went through the training program with me, Dr. Ruth Reeves and Dr. Diane Montella. They created an exciting and collegial atmosphere that inspired multiple creative discussions that were instrumental in developing the work within this thesis. Dr. Ted Speroff provided continued feedback and guidance on all stages of my work, and I sincerely thank him for his assistance.

Lastly, I would like to thank Julie Ness for her love and support throughout this process. She helped me to maintain the needed balance between life and academic pursuits that allowed me to carry out this work.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

ABBREVIATIONS

ACE – angiotensin converting enzyme; AHA – American Heart Association; ARB – angiotensin II receptor blocker; BOW – bag-of-words; CHF – congestive heart failure; CI – confidence interval; CRF – conditional random field; CSV – comma-separated value; EHR – electronic health record; FN – false negative; FP – false positive; GATE – General Architecture for Text Engineering; IAA – inter-annotator agreement; IML – interactive machine learning; IQV – index of qualitative variation; LVG – lexical variant generator; MCVS – Multi-threaded Clinical Vocabulary Server; NLP – natural language processing; Opt – optimism; Perf – performance; RapTAT – Rapid Text Annotation Tool; SNOMED-CT – Systematized Nomenclature of Medicine-Clinical Terms; SVM – support vector machine; TOS – token order specific; TP – true positive; TVHS – Tennessee Valley Healthcare System ; UIMA – Unstructured Information Management Architecture; UMLS – Unified Medical Language System; VA – Department of Veterans Affairs; VHA – Veterans Health Administration; VINCI - VA Informatics and Computing Infrastructure

CHAPTER 1

INTRODUCTION AND BACKGROUND

INTRODUCTION

Medical providers commonly record descriptions of patient encounters and treatments using free text narratives [1-3]. This method of clinical documentation generates a historical record that assists with continuity of care and provides justification for diagnostic tests and therapeutic interventions [4-6]. However, annotating free text to convert the data it contains into a structured form useful for computational analysis is expensive [7-9], which limits the use of such data in activities that can improve healthcare, such as biosurveillance and evaluations of medical practice guidelines.

Natural language processing (NLP) systems can often automate and thus reduce the expense of extracting structured data from free text, but their use in medicine is still limited [2, 10, 11]. One reason for this may be that existing NLP systems typically require modification before they can function reliably within a clinical domain or handle a specific extraction task. Modifying existing NLP systems generally requires a substantial investment of time and expertise. Additionally, such systems may depend upon the availability of significant computational power and time to accurately extract data, which may limit their use to relatively small-scale operations and processes that are not dependent on rapid feedback.

This thesis provides background regarding why providers continue to use free text to describe medical encounters and decisions and the implications of its continued use. It also explores the literature regarding hurdles that have prevented the wider adoption of NLP within medicine. Existing NLP-related systems are described as well as how some systems are addressing barriers to clinical adoption. The thesis framework then describes potential approaches to overcoming these barriers, and it outlines how the approach was selected that resulted in the development and implementation of a novel application, the Rapid Text Annotation Tool (RapTAT). Finally, this work includes two studies evaluating the RapTAT system and describes features of the system that would benefit from further development and ones that could prove useful in future applications.

BACKGROUND

Structure versus Unstructured Data Entry

Unstructured and structured data entry represent the two general approaches to documenting medical care [5, 11]. When electronic health record (EHR) systems use a fully structured approach, they ensure that data entry complies with specific formats that are machine readable and interpretable. As an example, such systems might limit the entry of patient symptoms to items that exist within a pre-defined set of terms. Structured entry systems may also require the user to supply certain information, such as not only the dose of a drug but also the duration, name of the provider that ordered the medication, and justification. This requirement helps to ensure that the data is complete with respect to the needs for not only continuity of care but also data processing and analysis.

In contrast to structured entry methods, unstructured data entry provides users with substantial latitude with respect to how best to record patient encounters and care. As long as the text entered meets legal and clinical requirements, medical providers using these approaches are free to employ whatever terms, abbreviations and linguistic constructs they choose. One substantial downside to this method of data entry is that there often numerous ways to describe even a single medical detail. Use of different synonyms (cardiac/heart, liver/hepatic), inflections of the same term (sneeze/sneezing, run/ran), and word order variants in medical phrases provide multiple alternatives for expressing essentially the same medical concept [12]. For example, there are at least 15 different phrases that might be used to describe the frequency of administering a drug (**Table 1**). Combining the expression of drug frequency with dose and duration further increases the number of phrase variants that might be used and the complexity associated with subsequently identifying and extracting the data, which may be further obscured due to the data being embedded within surrounding text. Further complications arise from abbreviations and homographic words with multiple meanings (e.g., 'patient' as a description of a personal trait *versus* a 'patient' being treated by a physician). Their use in medical free text can hinder interpretation, normalization, and mapping of such terms to standardized medical concepts. As a result, it can be difficult to reliably detect and track

Table 1. Phrase variants used to describe twice a day drug administration

twice a day	*twice daily*	*2X daily*
2X per day	*2X/d*	*2x/day*
every 12 hours	*every twelve hours*	*q12h*
bis in die	*b.i.d.*	*bid*
bis die	*b.d.*	*bd*

concepts expressed using such terms. Free text data entry may also increase the potential for data loss. Without requirements or prompts for data element entry, providers may not record certain elements and increase the difficulty of analysis and interpretation.

Despite the limitations of unstructured data entry, it does have several advantages over structured data. One is flexibility, which allows providers to enter data in the form and order that best matches their clinical workflow. Forcing clinicians to search for the required entry methods for documenting patient encounters or clinical orders can increase the cognitive burden associated with clinical care, introduce delays, and thus decrease efficiency [11, 13-16]. A recent report described the impact on nursing staff of introducing two EHR systems into 9 different elderly care facilities[16]. The most frequently reported adverse consequences were difficulty with data entry and/or information retrieval, which were reported by 43% of the staff. The third most common complaint, reported by 31% of the staff, was that the systems increased the complexity of managing information. Unstructured data entry also allows for greater expressivity than structured entry, so the provider can document subtle aspects of a clinical event, such as the certainty of a clinical impression. A clinician can also include aspects of an encounter that would otherwise be difficult to document, such as the reasoning process used to formulate a diagnostic plan. Until those advantages are matched by structured data entry methods, the use of unstructured data entry in EHRs is likely to continue and necessitate using manual annotation or NLP for data extraction.

Role of Natural Language Processing in Clinical Care

A number of existing NLP systems are capable of performing the general task of identifying concepts within medical narratives and mapping them to standard terminologies or ontologies, such as the Unified Medical Language System (UMLS) and/or the Systematized Nomenclature of Medicine – Clinical Terms (SNOMED-CT). Examples of these systems include HITEx [17], MedLEE [18], MCVS [19], cTAKES [20], KnowledgeMap [21, 22], MetaMap [23]and YTEX [24] among others. Because of the large number of concept occurrences that can be automatically identified within medical narratives and recorded by NLP systems, clinicians could use the data provided by these systems in a variety of ways to improve medical care, such as detecting novel associations between patient characteristics

and responses to a therapeutic intervention. Multiple studies have now demonstrated that NLP is effective for a wide-range of specific clinical uses. Examples include extracting medication information for visualization and reconciliation [25], as a screening tool for identifying patients suffering from traumatic injuries [26], automated identification of post-operative complications [27], automated detection of signs of infection and cases of influenza for use in biosurveillance [28, 29], quantitation of the use of evidence-based psychotherapy for treating post-traumatic stress disorder patients within the Department of Veterans Affairs [30], extraction of clinical decision support information from research reports included within the PubMed database [31], detection of patients who should be screened for colorectal cancer [32, 33], and determination of the efficacy of using influenza vaccines to prevent pneumonia in patients with existing influenza [34]. Given that NLP can reduce the need for providers to enter only structured data, NLP might also reduce barriers to adoption of EHR systems. In fact, sixteen years ago, McDonald identified two key reasons for lack of greater adoption, and one of the two was "we have not quite figured out how to capture the data from the physician in a structured and computer understandable form [14]."

Barriers to Clinical Adoption of NLP

Despite the great promise of NLP with regard to converting unstructured medical free text to structured data, its overall clinical usage remains limited [35]. Even when used, the functionality of NLP has typically been focused on rather specific tasks, some of which are noted above. Identifying the barriers to usage and developing methods to overcome them could increase the use of NLP, which could, in turn, increase the availability of data for evaluating and improving medical care. Chapman and colleagues identified a number of potential reasons for the slow adaptation of NLP for clinical use [10]. One of the possible barriers noted was the difficulty in reproducing results generated by a system in a particular domain or institution. NLP systems commonly consist of manually generated rules and/or statistical models generated based on machine learning [12]. Tailoring these rules and statistical models to perform well on training data can lead to over-fitting, and such systems may not perform well on test data or generalize to other medical domains or institutions. Adapting an existing NLP system to a new task may even require developing

novel NLP applications or iterative modification of the existing algorithms to match the document types under review and the environment [10, 33, 36, 37]. Furthermore, medical care, terminology, and patient populations are not static but evolve over time, and maintaining system accuracy over time may require system testing along with rule modification and re-training.

Limited system scalability may also reduce clinical adoption. NLP systems are commonly used for automating tasks and providing more rapid or less costly analysis than provided by manual review of free text. However, the demands of a large healthcare system may exceed the spatial and temporal capabilities of NLP systems that have been generated for academic studies with corpora of limited size. Data from a large health system, such as the Veterans Health Administration (VHA), can provide an appreciation of the potential scale required. Within the Veterans Health Administration in 2012, there were 83.6 million outpatient visits and 703,500 inpatient admissions with an average length of stay of 5.2 days [38, 39]. Assuming that the number of documents directly related to medical care generated by each outpatient visit or day of inpatient care is four (a bare minimum), there are at least 334 million medical care documents generated per year in the VHA, or roughly 1 million documents per day. MedLEE, one of the first and most extensive clinical NLP systems [40], was reported to require approximately 8 seconds to process a discharge summary and 0.4 seconds to process a radiology report [18]. Assuming an average processing time of 4.2 seconds, a single NLP system such as MedLEE could analyze approximately 20,000 documents per day. Parallelization may assist with the need for faster throughput, and there are open source software frameworks that can assist in the creation of parallel pipelines for NLP processing, such as the Unstructured Information Management Architecture (UIMA) [41] and the General Architecture for Text Engineering (GATE) [42, 43] systems. However, although such systems do allow for simultaneous processing of multiple documents, they do not address clinical use cases that may require real- or near-real-time NLP processing and feedback. Examples include analysis of free text and feedback at the point of care regarding medication errors or contraindications and notification of untreated incidental findings.

Although generalizability, adaptability, and scalability are important factors affecting the clinical use of NLP, the greatest barrier may be the size and technical proficiency of the

team required for implementation [10]. Despite the availability of NLP frameworks such as GATE and UIMA to assist with the creation of new and better tools, use of these tools still requires advanced computer programming skills [12]. Also, tailoring an existing NLP system to perform optimally within a specific medical facility or field often calls for substantial linguistic expertise and domain knowledge. This is a critical barrier because a key ingredient needed for successful integration of a new EHR system or module is the presence of a respected and technologically knowledgeable clinician, a "champion," who can spearhead the adoption [44-46]. This individual can provide a model for other clinicians regarding the initial and continued operation of the system and the potential benefits. Given the complexity of designing and adapting an NLP tool for use within an EHR system, identifying a clinician with the technical expertise and commitment needed to lead the implementation process may be difficult.

Need for Improved Text Annotation Tools

As the above discussion indicates, a tool that assists in adapting NLP systems to new domains would provide clinicians greater access to the advantages provided by these systems. This thesis describes and evaluates a tool developed to assist with a critical aspect of the adaptation process, namely the generation of training and testing data. The limited availability of such data can be a substantial hurdle to system development [9, 10]. A training or test corpus generally consists of hundreds to thousands of documents from the medical domain of interest in which the concepts of interest occurring within each document have been identified or "annotated." During creation of an NLP system, developers use the annotated documents for generating rules for concept identification or training the systems to identify concepts based on probabilistic algorithms. It is critical that the annotations be accurate in order to produce a worthwhile NLP system. However, creation of precisely annotated corpora is costly, particularly for clinical documents; it generally requires manual review by annotators familiar enough with the medical domain to identify the concepts of interest [47, 48]. Each annotator must examine all free text within each document in the corpus, identify the sequences of free text that relate to a concept of interest, and classify the sequence according to its associated concept. To produce the corpus generally involves three independent reviewers. Two reviewers act as annotators, and the third adjudicates any disagreements between the two.

A technique that can reduce annotation cost is pre-annotation [7, 49-51]. This approach employs an automated method to identify likely annotations prior to manual review, and it can reduce the work of a manual reviewer by decreasing the time spent searching for phrases or mapping them to concepts [50, 51]. The task of an annotator using this approach consists largely of removing incorrect and adding missing pre-annotations. However, there are potential downsides. The presence of numerous incorrect pre-annotations may increase rather than decrease the total workload of a reviewer. Also, pre-annotations may bias a reviewer by influencing the phrases chosen for annotation.

This thesis focuses on generating a system that reduces the burden of annotation by automatically adding pre-annotations without introducing bias and evaluates system performance in terms of accuracy and introduction of bias. The system to be described accomplishes pre-annotation through the use of a technique commonly referred to as interactive machine learning (IML). Generating training samples, such as annotated documents in the case of NLP, for machine learning system tends to be costly, and the general goal of interactive learning is to reduce the cost by either reducing the number of samples needed for training or decreasing the time required to annotate a corpus of given size. The process is interactive in that, unlike other forms of machine learning, the human user provides feedback to the machine learning system throughout training.

Two major categories of interactive machine learning are active learning and online learning, and the tool described in this thesis employs an online and not an active form of interactive machine learning. The reason for this is that the aim of active learning is generally to reduce the number of samples needed to produce a trained system. The system itself actively selects the next training sample to be manually labeled. One active learning strategy is to select the sample that, once labeled, will be most informative and thus provide the greatest increase in machine learning system accuracy [52]. The goal of the tool described herein is not to reduce sample size but to reduce the time and associated cost of annotating a document corpus of a given size, which can then be used to train external NLP systems. In contrast to active learning, online learning provides a way to reduce the time required to annotate a training corpus of set size. The approach is referred to as online because the training samples are provided sequentially, and the machine learning system updates its learning algorithms after the user supplies each training

sample. When using conventional machine learning, training samples are provided as a batch; the training process is separate and occurs "offline," after generating all samples used for training. The reason for hypothesizing that online learning may reduce annotation time is that the process can gradually train a system to predict how a reviewer will annotate. The process involves the system pre-annotating a document based on the training it has received up to that point, submitting the document to the annotator for review and correction, and then using the corrected annotations for further system training. This process gradually shifts the task of the reviewer from one of pure manual annotation to one of review and correction. This thesis will test both whether that review and correction process can reduce the time taken for annotation, and evaluate whether the pre-annotations provided by the system training influence the annotations selected by the reviewer.

OVERVIEW OF INCLUDED STUDIES

During design of the assisted annotation system described in this thesis, we recognized that pre-annotation by a machine learning system consists of two essential steps. The first step is to identify stretches of free text or "phrases" that the manual reviewer is likely to annotate. We refer to this step as "phrase identification." The second step is to determine the concept to which that phrase corresponds. We refer to this step as "concept mapping," and it is equivalent to a process referred to as concept recognition, normalization, or grounding in NLP systems [53-55]. Multiple NLP systems have been developed that are capable of concept mapping of clinical text [12]; examples include the Mayo Clinic Autocoder [56], the SNOMED Categorizer (SNOCat) [57], cTAKES [20], IndexFinder [58], KnowledgeMap [21], MedLEE [59], HITex [17], MedEx [60], MetaMap [23], Metaphrase [61], MicroMeSH [62], MTERMS [63], PhraseX [64], SAPHIRE [65], and SENSE [66]. These tools map phrases into a variety of user-defined lexicons, terminologies, and ontologies. In the studies supporting this thesis, we describe an alternative system that can support online machine learning and optimize the mapping of concepts within a given domain to an existing terminology or user-defined schema.

We will present the topics of phrase identification and concept mapping in this thesis in an order that is the reverse of that carried out during pre-annotation. The reason for

presenting these two topics in this order is that it was more straightforward to first develop concept-mapping methods using pre-defined phrases mapped to concepts; the order of presentation mirrors chronology of development. We will first describe our efforts to generate and evaluate a system that can be interactively trained to accurately map existing free text phrases to various concepts of interest contained within a set of defined concepts. This concept mapping system was then combined with a phrase identification system to generate a complete, machine-learning based pre-annotation system, and that system and its performance with regard to an assisted annotation task will be described and evaluated.

CHAPTER 2

CONCEPT MAPPING WITH RapTAT

BACKGROUND

Multiple computational models exist to support machine learning, including support vector machines (SVMs), logistic regression, neural networks, and Bayesian networks among others. All of these models use a set of features to make a prediction regarding an outcome. In this study, we were interested in predicting the likelihood of a particular concept mapping given a set of features consisting of a sequence of words that form a textual phrase. We used one type of Bayesian network, the naïve Bayes classifier, as our machine learning method. Many of the concept recognition tools described in the previous chapter have relied on matching of text strings to descriptions within an existing database of concepts [21, 23, 58, 59, 61, 62, 64-66]. A number of rules hand-coded within these systems were then applied to features within the text to select from among the set of matched concepts. One limitation of these approaches is that specific rules and identified features helpful in classifying text in one domain may not be generalizable to other domains. Another is that string matching within large databases can slow analyses and prevent the use of NLP systems for real-time analyses [67].

Probability-based, machine learning methods may provide a way to generate phrase-to-concept mappers tailored to the domain of interest. To the best of our knowledge, only two tools have been described that use probabilistic rather than a largely rule-based approach to map free text to medical concepts [57, 68]. The SNOCat tool combines regular expression searches and a vector space model based on term frequency and inverse document frequency to label free text with SNOMED CT concepts [57]. The Autocoder tool uses a database of previous classifications together with a naïve Bayes classifier for medical concept recognition; it maps lists of clinical diagnoses to codes within an ICD-8 based coding system [68]. The implementation treats text as a "bag-of-words" with respect to token frequency and disregards token position. Such an approach could remove important classification information and reduce accuracy. Naïve Bayes classifiers rely on conditional probability distributions of tokens given a particular classification, and those distributions may be position dependent. We therefore included in our concept mapping system not just

the tokens themselves but also their positions within a phrase of text as part of the machine learning process. We refer to this system as a token-order-specific naïve Bayes model.

An advantage to using a token-order-specific, naïve Bayes model for mapping is that there are only modest computational demands relative to other machine learning methods. Like the bag-of-words naïve Bayes classifier, the token-order-specific naïve Bayes classifier is based on the simplifying assumption that the tokens in a phrase of text are conditionally independent. In other words, if a phrase is used to express particular concept, the presence of a particular token in the phrase does not alter the probability of any other token occurring in the phrase. Given that the tokens are conditionally independent, a system trying to map phrases of text to concepts based on tokens alone does not need to store joint probabilities reflecting the likelihood of two or more tokens occurring in the same phrase. Storing such probabilities would increase spatial demands and could become intractable for corpora with large vocabularies. Under the assumption of independence, the system only needs to store the probabilities of individual tokens in the training corpus occurring in phrases mapped to a particular concept. This assumption thus reduces the spatial requirements for calculating the most likely concept associated with a given phrase. The spatial efficiency afforded by this approach should also improve temporal efficiency. Probabilities can be stored in rapidly accessible, computer memory, potentially capable of supporting real-time concept mapping.

This first study describes the initial development and evaluation of the concept mapping part of the RapTAT. We hypothesized that machine learning based on a token-order-specific, naïve Bayes classifier could be used to create a system capable of accurately and efficiently mapping phrases within free text to medical concepts. This study determines the impact of including token order as a feature on the ability of a machine learning system to accurately reproduce the phrase-to-concept mappings of an existing NLP system from discharge summaries to SNOMED concepts. Specifically, it compares the performance of this token-order-specific classifier relative to a bag-of-words-based, naïve Bayes classifier, and it defines the impact of phrase variability and concept ambiguity on performance. Furthermore, the study evaluates the accuracy and temporal efficiency of this implementation relative to a more basic system that maps phrases of text to concept-based string look-ups within a disk-based database. Finally, the dataset from the 2010 i2b2

challenge is employed to evaluate the ability of the system to map manually annotated phrases within clinical notes to user-defined concepts.

METHODS

Sampling and Population

The main document corpus, subsequently referred to as the VA data set, was a random sample of 2860 discharge summaries collected between fiscal years 1999 and 2006 within the Tennessee Valley Healthcare System (TVHS) VA Hospital. The TVHS institutional review board and research and development committee approved the study and granted a waiver regarding the need to obtain informed consent and HIPAA authorization before using patient data. The document corpus had been previously annotated using the Multi-threaded Clinical Vocabulary Server (MCVS) NLP tool to identify noun, verb, adjective, and prepositional phrases and map them to concepts within SNOMED CT [27]. For the named entity recognition part of the task, MCVS gives preference to concepts within SNOMED CT that contain a greater number of content terms (i.e., non-stop words). The method uses word normalization, word and term level synonymy and a word order independent method for concept recognition. Phrase identification is based on a set of heuristics that use concept type as a method to perform a set of rule based combinatorial algorithms. The technique is a backward and forward chaining algorithm and takes into account the assertion value of the concept. An earlier report using a predecessor of the MCVS tool demonstrated that, after accounting for missing synonyms within SNOMED CT, its sensitivity and specificity were 0.997 and 0.979, respectively, for the mapping of entries within a clinical problem list to the ontology [19]. In a study examining the ability of MCVS to detect symptoms related to tuberculosis, acute hepatitis, and influenza within VA clinical notes, precision and recall over all symptoms evaluated were 0.91 and 0.84, respectively [29].

The MCVS tool was responsible for all pre-processing of the free text, including document parsing, sentence splitting, tokenization, and identification of phrases, which were then mapped to SNOMED CT concepts by MCVS. For the purposes of this study, we defined a phrase as an ordered sequence of tokens formulated by the author of a medical note to

express a concept. Tokens were generally words but also included other elements such as numbers, units of measurements, and dosages [19, 69]. The MCVS-processed data were provided to RapTAT as an idealized set of phrase-to-SNOMED CT concept mappings for tool development and testing. The aims for this part of the study were to evaluate the ability of RapTAT to learn to reproduce the MCVS mappings and to determine the factors that can affect tool performance.

All sequences were limited to a maximum of 7 tokens, the maximum phrase length identified by MCVS. All token characters were converted to lower case for training and evaluation. There were 567,520 phrases (22,994 unique) within the document corpus, and each phrase was mapped to one of 12,056 unique concepts by the MCVS tool. These annotated documents provided a working environment for training and evaluation of RapTAT, and the phrase-to-SNOMED CT concept mappings generated by the MCVS tool served as the reference standard for the purposes of this study. The data were stored in comma-separated value (CSV) files with each row containing a single phrase and the associated MCVS-mapped concept.

The study also used the data available from the 2010 i2b2 challenge for evaluating the performance of the RapTAT tool with regard to its ability to map manually annotated phrases to concepts within a defined schema [70]. The annotated corpus consisted of discharge summaries and progress notes from 3 institutions, University of Pittsburgh Medical Center, Beth Israel Deaconess Medical Center, and Partners Healthcare. The schema contained three concepts (problem, test, and treatment), and all annotated phrases were mapped to one of those three. The training corpus contained 170 documents, and 16,526 phrases within the corpus were manually annotated and mapped to one of the schema concepts. The test corpus contained 256 documents, and the i2b2 reviewers had mapped 31,162 phrases from that corpus to the schema concepts. RapTAT performance on the test set was evaluated with respect to how closely its mapping of phrases to one of the three i2b2 concepts matched those determined by the i2b2 organization. The assertion and relation classifications of the phrases, which were provided with the i2b2 training and test data, were not used in our study.

Machine Learning Algorithm

We used the Java programming language to generate both bag-of-words based and token-order-specific, naïve Bayes classifiers for mapping free-text phrases to SNOMED CT concepts within the RapTAT application. The system first imported the MCVS-determined phrase-to-concept mappings to establish both prior and conditional probabilities, which were then used to identify the most likely phrase-to-concept mapping within the test data. Prior probabilities for the classifier were determined based on the frequency of concept occurrences within the training data. Conditional probabilities were calculated based on the likelihood of a given token occurring within a phrase given a particular concept. In the case of the token-order-specific implementation, the RapTAT tool generated a separate conditional probability table for each of the 7 potential token positions within a phrase during training. For the bag-of-words, tokens from all positions within a phrase were used to generate a single distribution.

By treating the probability of a particular token as independent of the occurrence of all other tokens in a phrase, we were able to use the naïve Bayes equation to generate the likelihood estimate, P, of the tokens mapping to a given concept, C_i [71]. The form of that equation is

$$P(C_i|\boldsymbol{T}) = \frac{P(C_i) \bullet P(\boldsymbol{T}|C_i)}{P(\boldsymbol{T})} \quad (1)$$

where $\boldsymbol{T}$ represents the vector of tokens that make up the phrase. Using this equation reduces the task of the system to identifying the particular concept, C_i, which maximizes the right side of the equation. For a given phrase, the denominator, $P(\boldsymbol{T})$, is constant, so that only the numerator needs to be considered when determining the most likely token sequence to concept mapping. For the bag-of-words classifier,

$$P(\boldsymbol{T}|C_i) = \prod_{j=1}^{n} P(T_{ij})^{x_j} \bullet \left[1 - P\left(T_{ij}\right)\right]^{x_j} \quad (2)$$

where n is the number of unique tokens over all phrases, and x_j is one if token T_j occurs in

the phrase and zero otherwise [72]. The estimate of $P(T_{ij})$ is given by

$$P(T_{ij}) = \frac{Occurrences\ of\ Token\ T_j\ within\ Sequence\ Mapping\ to\ C_i}{Occurrences\ of\ C_i} \quad (3)$$

The token-order-specific implementation corresponds to a multinomial naïve Bayes model in which positions within a mapped token sequence represent features, and the tokens represent the assigned values of the features. For that model, the conditional probability of the phrase $\boldsymbol{T}$ of length m is

$$P(\boldsymbol{T}|C_i) = P(T_{ij1}) \bullet P(T_{ik2}) \bullet \cdots \bullet P(T_{ilm}) \quad (4)$$

where $P(T_{ijk})$ is estimated as

$$P(T_{ijk}) = \frac{Occurrences\ of\ Token\ T_j\ at\ Position\ k\ when\ Sequence\ Maps\ to\ C_i}{Occurrences\ of\ C_i} \quad (5)$$

One difficulty with using Bayes equation is that conditional probabilities can be zero for rare tokens absent from the training data. We therefore used Laplace smoothing to adjust all probabilities [73].

Hash tables stored the number of times each token was associated with a concept within the training data. In the case of the token-order-specific implementation, there was a separate hash table for each of the 7 potential positions of tokens within a phrase. For example, if the phrase "acute myocardial ischemia" occurred in the test data, RapTAT would use "acute" as a key for the hash table corresponding to the first position in a token phrase. The key would return a set containing all concepts for which acute was the first word in a phrase mapping to one of the concepts. Within the set, each concept would be associated with an integer indicating the number of times a phrase mapped to that concept and contained "acute" as the first token. All input and output data and the data structures used by the tool were maintained in random access memory during processing.

The RapTAT application is available at http://code.google.com/p/raptat/. RapTAT was developed independently and does not contain source code of any form from MCVS or other NLP system. The data structures generated based on the MCVS annotated text were created only for the purposes of testing and evaluation of the RapTAT tool. Those data

structures contain potentially identifiable patient health information and cannot be distributed or reused.

Evaluation Measures

Performance was based on the number of true positives (TP), false negatives (FN), and false positives (FP). A TP was attributed to a concept when both the RapTAT system and the reference standard mapped a phrase to that exact same concept. When RapTAT mapped the phrase to a different concept than the one identified by the reference standard, a FP was attributed to the RapTAT concept. A FN was attributed to the reference concept when RapTAT was unable to map the phrase or identified a concept different from the reference standard. The tool itself scored TPs, FPs, and FNs and calculated precision, recall, and F-measure according to the equations

$$Precision = TP / (TP + FP) \quad (6)$$

$$Recall = TP / (TP + FN) \quad (7)$$

$$F\text{-}Measure = 2 \bullet Precision \bullet Recall / (Precision + Recall) \quad (8)$$

For determining the accuracy and efficiency of string matching-based concept mapping, we generated a separate Java tool to sequentially match each of 290,741 randomly selected token sequences from our experimental data to terms in the MCVS SNOMED CT database. The tool used repeated SQL queries for matching phrase strings to terms, and phrases and their matched concepts were cached in memory during processing. Memory caching consisted of dynamically building a hash table mapping each phrase used in a SQL query to the identified concept. This improved the processing rate by eliminating repeated SQL queries on the same phrase; the more efficient hashing process was used if the phrase reoccurred in the data. Matching was carried out using both approximate and exact string matching. Approximate matching was carried out by placing wildcard characters ("%") at the beginning and end of each evaluated phrase. Queries were of the form "SELECT '*id*' FROM '*db_table*' WHERE '*name*' LIKE '*phrase*'," where '*id*' referred to the SNOMED concept identifier, '*db_table*' was a table in a local database in which SNOMED concept identifiers ('*id*') and fully qualified names of the concepts ('*name*') were columns in the table, and '*phrase*' was one of the phrases identified by MCVS. When multiple concepts were

returned from a query, only the first one was retained and tested for correspondence to the reference standard.

Primary Analysis

We used bootstrap evaluation to estimate precision, recall, and F-measure of the RapTAT-generated phrase-to-concept mappings over the entire, 2860 document corpus. The evaluation method was automated by the RapTAT system and implemented consistent with bootstrapping methods used in risk prediction modeling [74]. The analysis consisted of 1000 training and testing iterations, and each iteration began with creation of a training set generated by random sampling with the sample size equal to the original number of documents. Sampling was done with replacement; selected documents could be chosen more than once, and each training set might contain 0, 1, or multiple copies of a single document (and its associated phrases and concepts). The system did not include phrase-concept associations from previous iterations in the probability calculations. Estimated performance ($Perf_{Est}$) with respect to precision, recall, and F-measure was calculated as

$$Perf_{Est} = Perf_{App} - Opt \quad (9)$$

where $Perf_{App}$ referred to apparent performance on the entire data set when trained on the entire data set. Optimism (Opt) represents the degree to which $Perf_{App}$ overestimates performance when training and testing are done on the same data set. It is calculated by training on the bootstrap set and then measuring the difference in performance, averaged for each concept across all iterations, when testing is done on the bootstrap *versus* the entire data set. In the case of concepts with low prevalence, the training set may have limited or no training on which to base concept mapping. Under these conditions, TP and FP may both be zero so that precision is undefined, or TP and FN may both be zero so that recall is undefined. When this occurs for a given iteration, there is no calculable estimate for optimism. Based on a similar situation that can happen during cross-validation, we evaluated three different approaches for handling this issue during bootstrapping: 1) skip the iteration and do not include it in the calculation for the concept; 2) assume optimism is zero; and 3) assume optimism is one [75]. In addition, because cross-validation is more commonly used than bootstrapping for estimating the accuracy of machine learning-based models, we compared the performance measures estimated using bootstrapping to values

obtained using “leave one out” cross-validation. The leave one out evaluation consisted of an iterative process, using one of the documents in the corpus for testing and the remaining documents for training. This was done iteratively until every document had been used once for testing. To minimize bias in the estimated performance measures, TP, FP, and FN were summed over all iterations to give a total precision, recall, and F-measure for each concept [75].

Concepts included in the study were grouped within each of the top-level concepts within the SNOMED CT ontology [75] (**Table 2**). These top-level concepts form the roots of 19 hierarchical trees within SNOMED CT, and we refer to these as ‘conceptual groups.’ The grouping was done to illustrate general system performance over the many concepts present in the corpus while still allowing for detection of performance differences among groups. Macro-averages were generated by taking the average performance score for each concept determined by bootstrapping and calculating the ‘average of the averages’ for all concepts within a conceptual group, so both rare and commonly occurring concepts contributed equally.

Evaluation of Performance on i2b2 Data

The phrase tokens within the i2b2 data underwent additional processing before training of RapTAT. Initial pre-processing consisted of phrase tokenization, retention of only the first 7 tokens for each phrase, and conversion of all characters to lower case. Subsequent pre-processing consisted of tagging tokens with their parts-of-speech (POS tagging), removal of stop words (“a,” “an,” "and," “by,” “for,” “in,” “nos,” “of,” “on,” “the,” “to,” and “with”), token lemmatization, and/or inversion of token order. The influence of each of these pre-processing steps with regard to F-measure of the tool using each of these pre-processing steps was evaluated using the training set and bootstrapping as described above. The combination of steps that produced the highest F-measure was used for pre-processing tokens within the test set.

Tokenization and POS tagging was carried using the OpenNLP libraries (Apache Software Foundation) and trained maximum entropy POS tagger. Lemmatization, which converts multiple inflections of a word into a single form, such as the conversion of the both “runs”

Table 2. Macro-averaged precision, recall, and F-measure for mapping phrases to conceptual groups within SNOMED-CT using a token-order-specific classifier. The averages are calculated using only concepts that appeared 5 or more times in the corpus.

SNOMED CT Conceptual Group	Concepts Within Group	Phrases Within Group	Macro-Averaged Performance (95 % Confidence Interval)		
			Precision	Recall	F-Measure
Body Structure	350	24365	0.94 (0.92-0.96)	0.93 (0.91-0.95)	0.92 (0.90-0.94)
Clinical Finding	724	34147	0.95 (0.94-0.96)	0.95 (0.94-0.96)	0.94 (0.93-0.95)
Event	10	367	1.00 (1.00-1.00)	0.92 (0.87-0.97)	0.95 (0.92-0.98)
Linkage Concept	114	33759	0.95 (0.91-0.98)	0.95 (0.91-0.98)	0.94 (0.90-0.98)
Location	40	7536	0.98 (0.95-1.00)	1.00 (0.99-1.00)	0.98 (0.97-1.00)
Observable Entity	158	9242	0.95 (0.94-0.97)	0.96 (0.93-0.98)	0.95 (0.92-0.97)
Organism	26	523	0.92 (0.83-1.00)	0.94 (0.85-1.00)	0.92 (0.84-1.00)
Physical Force	4	423	1.00 (0.99-1.00)	0.97 (0.87-1.00)	0.98 (0.93-1.00)
Physical Object	106	4155	0.96 (0.94-0.99)	0.96 (0.94-0.98)	0.96 (0.94-0.98)
Procedure	397	23570	0.96 (0.95-0.97)	0.96 (0.95-0.97)	0.95 (0.94-0.96)
Product	203	7227	0.97 (0.95-0.99)	0.97 (0.95-0.99)	0.96 (0.95-0.98)
Qualifier Value	902	108619	0.96 (0.95-0.97)	0.94 (0.93-0.96)	0.94 (0.93-0.96)
Record Artifact	6	758	1.00 (1.00-1.00)	1.00 (0.99-1.00)	1.00 (1.00-1.00)
Situation	27	1090	0.98 (0.95-1.00)	0.91 (0.87-0.95)	0.93 (0.91-0.96)
Social Context	62	13606	0.96 (0.93-1.00)	0.97 (0.94-1.00)	0.96 (0.93-1.00)
Special Concept	2	203	0.98 (0.72-1.00)	1.00 (1.00-1.00)	0.99 (0.85-1.00)
Specimen	3	3936	1.00 (1.00-1.00)	0.99 (0.96-1.00)	0.99 (0.98-1.00)
Staging & Scales	3	51	0.99 (0.97-1.00)	0.95 (0.77-1.00)	0.97 (0.86-1.00)
Substance	165	5511	0.95 (0.92-0.98)	0.95 (0.92-0.98)	0.95 (0.92-0.97)

and “ran” to “run”, was carried out using the lexical variant generator (LVG) library from the National Library of Medicine [76], Token order inversion consisted of putting the tokens into the hash tables used for probability calculations in reverse order. The last token in an English phrase commonly constitutes the “headword” of the phrase and may strongly influence phrase interpretation [77]. We therefore hypothesized that inverting token order might improve mapping performance by aligning phrases along the last token. For example, without inversion, “acute” and “ischemia” would go into the two hash tables corresponding to positions one and two during training. When testing the system on a phrase such as “chronic myocardial ischemia,” the previous occurrence of the token “ischemia” in the second position would not directly influence the likelihood of the tested phrase mapping to the same concept as “acute ischemia” because of differences in position of “ischemia” in the two phrases. In contrast, if the two phrases were inverted, “ischemia” would be the first token in both.

Learning, Training, and Mapping Rates

To evaluate the learning rate of the system, we randomly divided the original VA corpus into training and test set with 50% of the documents. The tool was first trained using 12 random subsets ranging in size from 100 phrases to the full training set of 283,760 phrases ; precision, recall and F-measure were determined on the entire test set after training on each data subset. To minimize the bias introduced when concepts within the test set are absent from the training set, performance was measured based on the total TP, FP, and FN concept matches summed over all concepts [75]. Evaluation of system speed (phrases processed per second) was accomplished using the same training and test sets. Both training and testing consisted of the system sequentially reading in and processing each row of data from the CSV data file, and the system reported the time to complete every 10,000 rows. We also evaluated the speed of the SQL query-based, string-matching application. For all speed evaluations, text processing did not include tokenization, POS tagging, lemmatization, or token order inversion.

All training and testing of the RapTAT system and baseline evaluation were run using a standard desktop personal computer containing 1.98 GB of RAM, an Intel Core 2 Duo processor running at 2.99 GHz, and a 7200 RPM hard drive with a SATA-300 interface. The

operating system was Windows XP Professional with Service Pack 3. The SQL database used for string matching was created using Microsoft SQL Server 2008 maintained on a server containing 11.9 GB of RAM and Intel Xeon CPU running at 2.66 GHz with the Microsoft Windows Server 2003 R2 operating system. The server database was maintained on an array of fifteen, 450 GB, 15K RPM hard drives maintained in a RAID5 configuration. The SQL table used for querying was sorted and indexed using clustering based on the SNOMED CT term. The application accessed the server through the local VA intranet.

Phrase Variance Analysis and Concept Ambiguity

To examine the impact of variations in the set of tokens mapping to a given concept on the corresponding F-measure, phrase variance for each concept was quantified using the index of qualitative variation, IQV, defined as

$$\text{IQV} = \frac{K}{K-1} \bullet \left(1 - \sum_{i=1}^{K} p_i^2\right) \quad (10)$$

where K is the number of unique phrases mapping to a given concept, and p_i is the proportion of phrases that map to a given concept accounted for by the ith phrase [78]. The IQV provides a measure of the variability of the phrases mapping to a single concept. It reaches a maximal value of one when the phrases mapping to a given concept are evenly distributed among two or more possibilities; its value approaches zero for a concept associated almost exclusively with a single phrase. The IQV is undefined when K is one, so concepts corresponding to a single, unique phrase (2957) were not included in the phrase variance analyses.

We define and computed a measure of "concept ambiguity" to formalize the relationship between mapping uncertainty and tool performance as well as the impact of classifier type on that relationship. This measure quantifies the uncertainty associated with phrase-to-concept mapping due to the use of the same token among phrases used to express distinct concepts. For some phrases, the tokens themselves and/or their sequence may uniquely identify a concept. For example, MCVS mapped the single token phrase, "keflex," to a single SNOMED CT concept, *cephalexin (product)*. Training a system like RapTAT to accurately

reproduce such a mapping is trivial. In contrast, when two or more concepts employ the same tokens for expression, the probability of mapping a given token sequence to a particular concept may be greater than zero for multiple candidate concepts, resulting in greater mapping uncertainty. For example, "abdominal" was a token in the phrases "abdominal hernia," "abdominal distention," and "left lower abdominal quadrant" as well as a number of others within our VA document corpus, and each of those phrases was mapped to a different SNOMED CT concept within the MCVS reference standard.

To quantify concept ambiguity (**Appendix A**), we first calculated the phrase-to-concept "mapping ambiguity" for each phrase that mapped to a given concept according to the reference standard. Mapping ambiguity for a phrase was based on the number of "similar" phrases in the dataset; its magnitude correlated with the number of concepts to which a phrase might map. In the case of the token-order-specific classifier, similar phrases were defined as all those having the same token at the identical position as the given phrase. In the case of the bag-of-words classifier, they were defined as all phrases having the same token as the one in the given phrase at any position. Potential mapping concepts for a particular token were defined as all concepts associated with the given or similar phrases. Mapping ambiguity for a given phrase was calculated as the size of the set of potential mapping concepts for all tokens within the phrase normalized to phrase length. Concept ambiguity was calculated as the logarithm of the average mapping ambiguity for all phrases associated with the concept based on the reference standard. As a result of using this method of calculation, when concept ambiguity approached zero, the probability of correctly mapping all the phrases associated with that concept approached one.

Statistical Analysis

Simple linear regression was used to evaluate the relationship between precision, recall, and F-measure and the number of documents containing a concept. Paired *t*-tests were used to compare performance across concepts between the token-order-specific and bag-of-words naïve Bayes classifiers. All statistical analyses were carried out using Static/IC 11.2 for Mac (Stata Corp., College Station, TX), and *p*-values of less than 0.05 were considered significant.

RESULTS

When the VA corpus was divided into separate training and test sets, recall and F-measure increased steadily for the first 10,000 training phrases regardless of the machine learning-basis of the classifier (**Figure 1**). For the token-order-specific, naïve Bayes-based classifier, all performance measures were ≥0.88 after training on phrase-to-concept mappings from 50,000 phrases. With further training, all performance measures appeared to continue to increase, reaching a precision, recall, and F-measure of approximately 0.92 using the entire set of training phrases. In comparison, the performance measures for the bag-of-words-based classifier reached a plateau in the range of 0.80-0.82 after training on 50,000 phrases or more. Increases in performance with additional training were 0.003 or less.

Although cross-validation has been used more frequently than bootstrapping for statistical validation of NLP models, bootstrapping may provide more accurate estimates of precision, recall, and F-measure. To compare performance estimation by bootstrapping to that of cross-validation, our study used both approaches to calculate the average F-measure at the concept level for the token-order-specific classifier. Because low prevalence concepts can bias both approaches [75] and should occur more frequently in smaller document corpora, our study estimated performance as a function of the number of documents containing a concept. When analyzing concepts with low prevalence (present in < 5 documents), leave-one-out cross-validation estimated a lower F-measure than the other evaluation methods (**Figure 2**). When the system used bootstrapping and set the optimism to one when it could not otherwise be calculated, it also estimated a lower value of the F-measure for low prevalence concepts than did the other bootstrapping techniques. When the system assumed an optimism of zero or skipped estimation altogether for low prevalence concepts, less bias was apparent, but there did appear to be a slight overestimation of the F-measure relative to higher prevalence concepts. For concepts in 5 or more documents, the F-measure was similar across all methods. Because of this finding, all subsequent analyses were confined to concepts occurring in at least 5 documents to minimize bias, and performance measures were calculated using bootstrapping. This reduced the VA data set to 3302 concepts and 279,088 phrases. When a concept was absent from the bootstrap set, performance measures were calculated assuming an optimism of both one and zero, and the calculated values were averaged to generate a final estimate. To quantify phrase

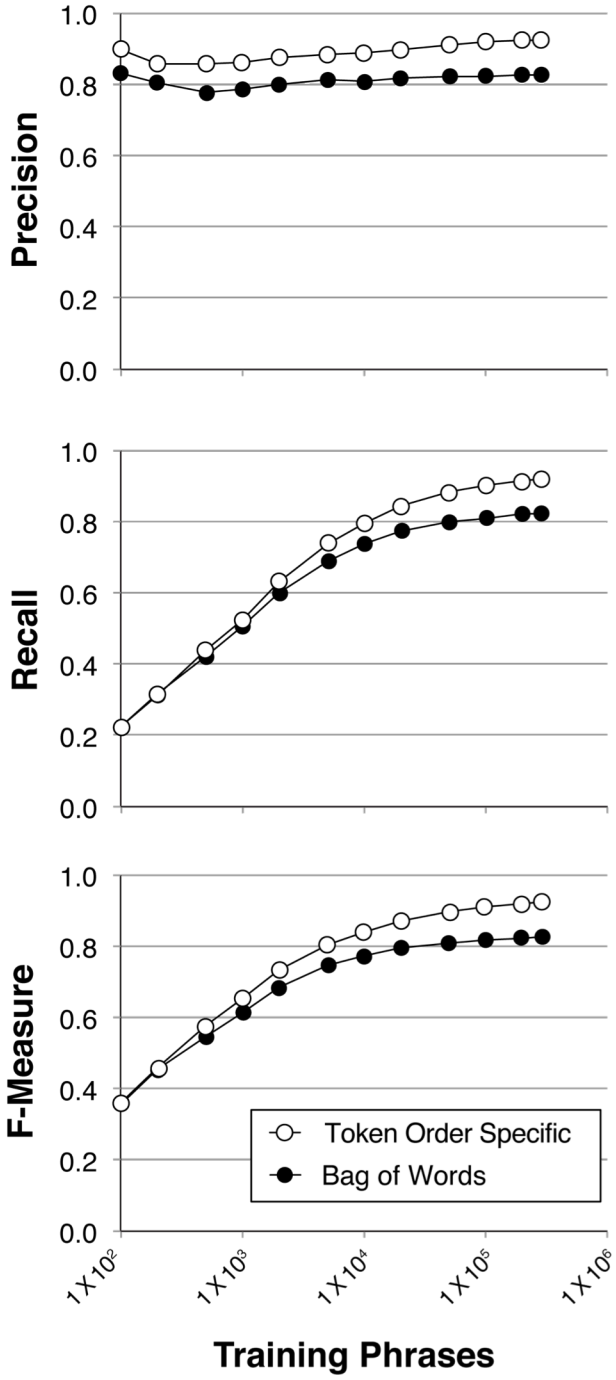


Figure 1. Learning rate of phrase mapping as a function of the number of documents used for training. The graph demonstrates the impact of language model (TOS – token-order-specific naïve Bayes, open symbols; BOW – bag-of-words, closed symbols) on precision (top), recall (middle), and F-measure (bottom). Each point represents the precision, recall, or F-measures based on the total true positives, false positives, and false negatives across all concepts in the test set (n=673 documents).

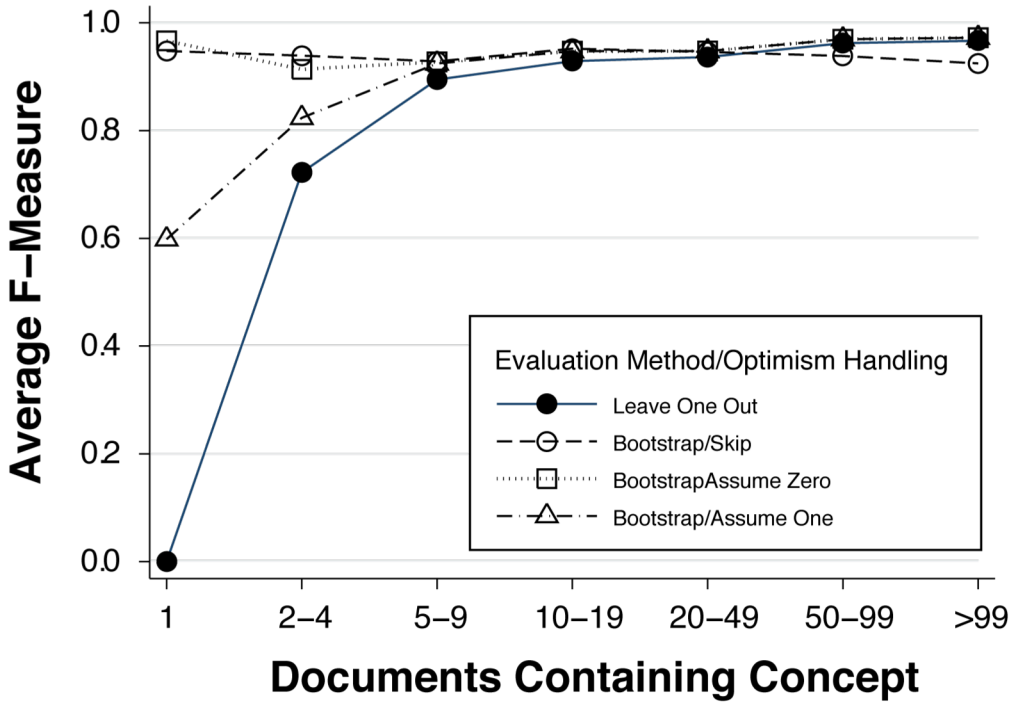


Figure 2. F-measure as evaluated by cross-validation or bootstrapping as a function of the number of the documents containing a concept. The graph also demonstrates how varying the approach to handling optimism alters the estimated F-measure relative to concept prevalence. Concepts were assigned to groups (x-axis) based on the number of documents in which they occurred, and each point represents the average over all concepts within the group.

complexity within the reduced VA data set, the number of tokens per phrase was calculated and found to be 1.5 ± 0.8 (mean ± SD) across concepts occurring in 5 or more documents with a median of 1 (interquartile range = 1 – 2). The average number of unique phrases mapping to a concept was 1.9 ± 4.3 with a median of 1 (interquartile range 1 – 2).

Using the bootstrapping technique described above to estimate performance, there was a significant increase in precision, recall and F-measure for both classifier models when the number of documents containing a concept and thus available for training the system to map to the concept increased beyond 5 (**Figure 3**). In the case of the bag-of-words classifier, precision, recall, and F-measure were in the range of 0.80-0.86 after training on 100 or more documents compared to 0.97-0.98 for the token-order-specific classifier (**Figure 3**). With respect to variability of the phrases mapping to a concept, as measured by IQV, F-measure decreased for both classifiers with increasing IQV, and the greatest decrease was associated with the bag-of-words classifier (**Figure 4, top**). Similar decreases in performance occurred as concept ambiguity increased, and the most substantial decreases were associated with the bag-of-words classifier (**Figure 4, bottom**).

The highest scoring system in terms of precision, recall, and F-measure averaged over all concepts was the token-order-specific classifier (**Table 3**). The bag-of-words classifier and exact string matching showed intermediate performance, and the approximate string matching had the lowest performance. Histogram-based analysis of F-measures for the naïve Bayes classifiers demonstrated a strong bivariate distribution for phrase mapping using the bag-of-words classifier (**Figure 5**). For that classifier, >29% of the concepts had an F-measure of 0.05 or below, whereas <2% of the concepts mapped with the token-order-specific classifier had such a low F-measure. To determine whether the low performance concepts were alone responsible for the reduced F-measures attained using the bag-of-words classifier (**Figure 3**), we restricted the analysis for concepts that had performance measures above 0.5 for both classifiers. Although this restriction did reduce the disparity between the two classifiers, precision, recall, and F-measure were still significantly higher for the token-order-specific (0.99 ± 0.04, 0.99 ± 0.04, and 0.985 ± 0.09, respectively; mean ± SD) relative to the bag-of-words classifier (0.94 ± 0.12, 0.96 ± 0.11, and 0.95 ± 0.10, respectively.

Table 3. Precision, recall, and F-measure using four different methods of phase-to-concept mapping. The analysis was restricted to only concepts that occurred in at least 5 of the documents within the corpus.

Mapping Method	Average Performance (95% Confidence Interval)		
	Precision	**Recall**	**F-Measure**
Token-Order-Specific Classifier	0.96 ±0.15 (0.95-0.96)	0.95 ± 0.16 (0.94-0.95)	0.95 ± 0.15 (0.94-0.95)
Bag-of-Words Classifier	0.63 ± 0.45 (0.62-0.65)	0.62 ± 0.46 (0.61-0.64)	0.61 ± 0.45 (0.60-0.63)
Exact String Matching	0.76 ± 0.41 (0.75-0.78)	0.56 ± 0.44 (0.54-0.57)	0.59 ± 0.44 (0.57-0.60)
Approximate String Matching	0.14 ± 0.34 (0.13-0.15)	0.12 ± 0.29 (0.11-0.13)	0.12 ± 0.30 (0.11-0.13)

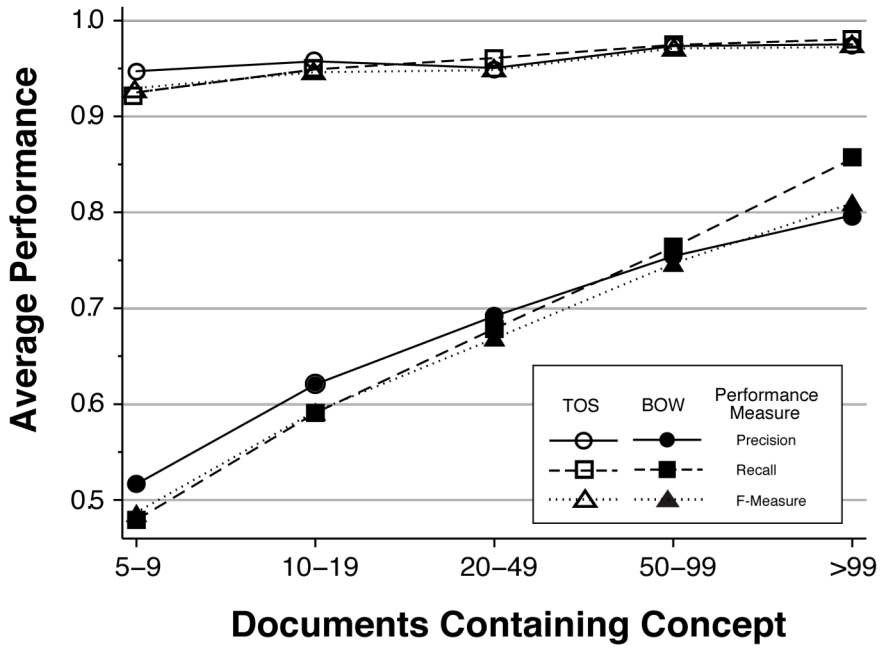


Figure 3. Impact of the number of documents containing a concept and naïve Bayes classifier type (TOS – token-order-specific naïve Bayes, open symbols; BOW – bag-of-words, closed symbols) on precision(circles with solid line), recall (squares with dashed line), and F-measure (triangles with dotted line). Concepts were assigned to groups (x-axis) based on the number of documents in which they occurred, and each point represents the average over all concepts within each group.

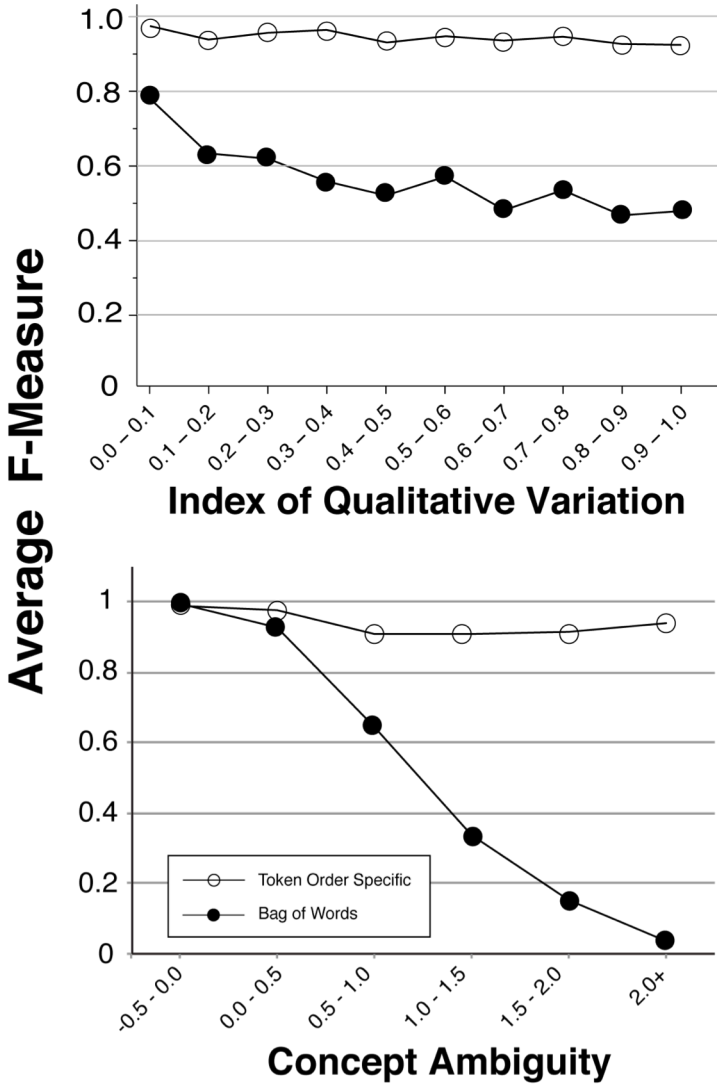


Figure 4. F-measure as a function of the variability of the phrases mapping to that concept as measured by the index of qualitative variation (IQV; top). Each point represents the average over all concepts having an IQV greater than the lower limit of the range given by the x-axis and equal or less than the upper limit. F-measure is also plotted as a function of concept ambiguity (bottom). Each point represents the average F-measure for all phrases that potentially map to the number of concepts within the range indicated along the x-axis. F-measure with respect to both the token-order-specific (TOS; open symbols) and bag-of-words (BOW; closed symbols) classifiers is included. Precision and recall closely matched the F-measure and have been excluded for the sake of clarity.

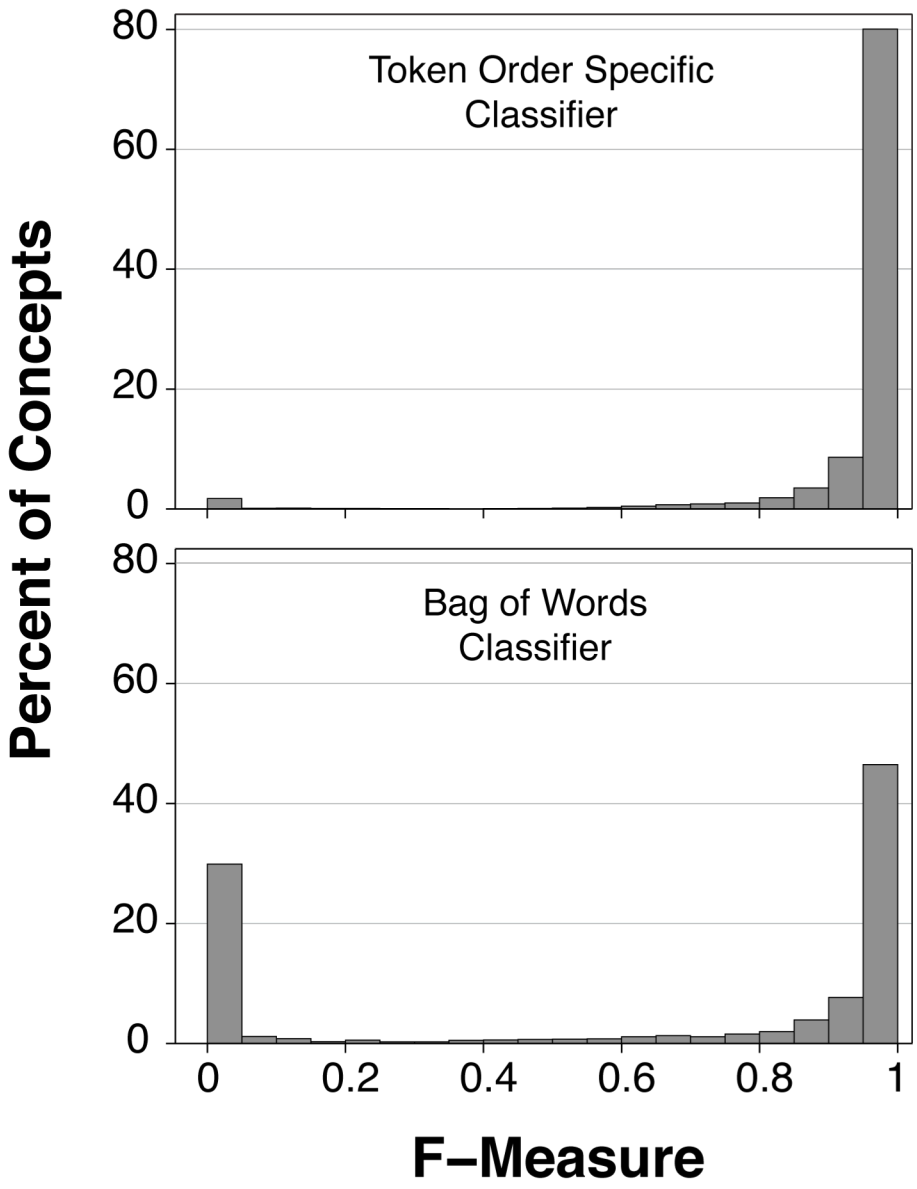


Figure 5. Histogram demonstrating the impact of classifier type on the distribution of concepts with respect to F-measure for phrase-to-concept mapping.

Precision, recall, and F-measure for the token-order-specific classifier were generally similar across the 19 SNOMED CT conceptual groups included in the study. Average performance measures were >0.91 for all groups (**Table 2**), and the majority of groups had an average precision (18 of 19), recall (14 of 19), and F-measure (13 of 19) exceeding 0.95.

When the token-order specific classifier was applied to the i2b2 training set, pre-processing of phrases by removing stop words, POS tagging, or reversal of token order improved F-measure, which was estimated using bootstrapping (**Table 4**). Combining all three of these pre-processing steps was associated with the greatest F-measure. Lemmatization had no measurable effect or reduced performance. When these pre-processing steps were used during training of RapTAT on the training set and evaluating it on the test set, measured precision and F-measure were generally in the 0.87-0.94 range depending on the concept, and recall was in the 0.87-0.91 range (**Table 5**). Performance increased continuously as the number of training documents increased. This is demonstrated by the strong log-linear relationship between the number of training phrases and precision, recall, and F-measure based on regression analysis (r^2>0.96 and p<0.001 for all performance measures) (**Figure 6**).

Training and evaluation rates for the two, naïve Bayes-based classifiers were similar. The token-order-specific and bag-of-words classifiers were able to process 149.4 and 128.0 phrases per millisecond during training (data not shown), respectively, and they mapped 29.6 and 24.4 phrases per millisecond during evaluation (**Figure 7**). This rate did not include any lexical processing, which was done by the MCVS tool before generating the CSV file imported into RapTAT. There were no detectable changes in the rate during training or mapping. With respect to string matching using SQL queries, memory caching gradually increased the rate of phrase processing using exact and approximate matching. Both methods reached a plateau rate after processing approximately 1 X 10^5 phrases (**Figure 7**). The mapping method used by the naïve Bayes-based classifiers was approximately 5-fold faster than the plateau rate of the exact string matching method. After reaching the plateau, the rate of string matching per millisecond remained within the range of 4-8 for exact phrase matches and 0.10-0.25 for approximate phrase matches.

Table 4. Impact of token pre-processing on RapTAT performance. Performance with respect to concept mapping on the 2010 i2b2 training data was assessed using bootstrapping to estimate the F-measure when phrase tokens were subject to stop word removal, part-of-speech (POS) tagging, reversal of the order in which tokens were entered into hash tables or evaluated, conversion of the tokens into their lemmas, or stop word removal, POS tagging, and token order reversal combined.

Concept	Pre-Processing					
	None	Stop Word Removal	POS Tagging	Token Order Reversal	Use Lemmas	Stop Word Removal/ POS Tagging/ Token Order Reversal
Problem	0.93	0.93	0.93	0.94	0.93	0.94
Test	0.91	0.92	0.92	0.92	0.91	0.92
Treatment	0.90	0.91	0.91	0.92	0.90	0.92

Table 5. Phrase mapping performance of RapTAT on the i2b2 test data. Performance on each of the schema concepts is shown. Average performance over all concepts (last row) was computed by weighting each performance measure based on the number of times the concept occurred in the test data.

Concept	Performance Measure		
	Precision	Recall	F-Measure
Test	0.91	0.89	0.90
Problem	0.93	0.87	0.90
Treatment	0.94	0.91	0.87
All	0.93	0.86	0.89

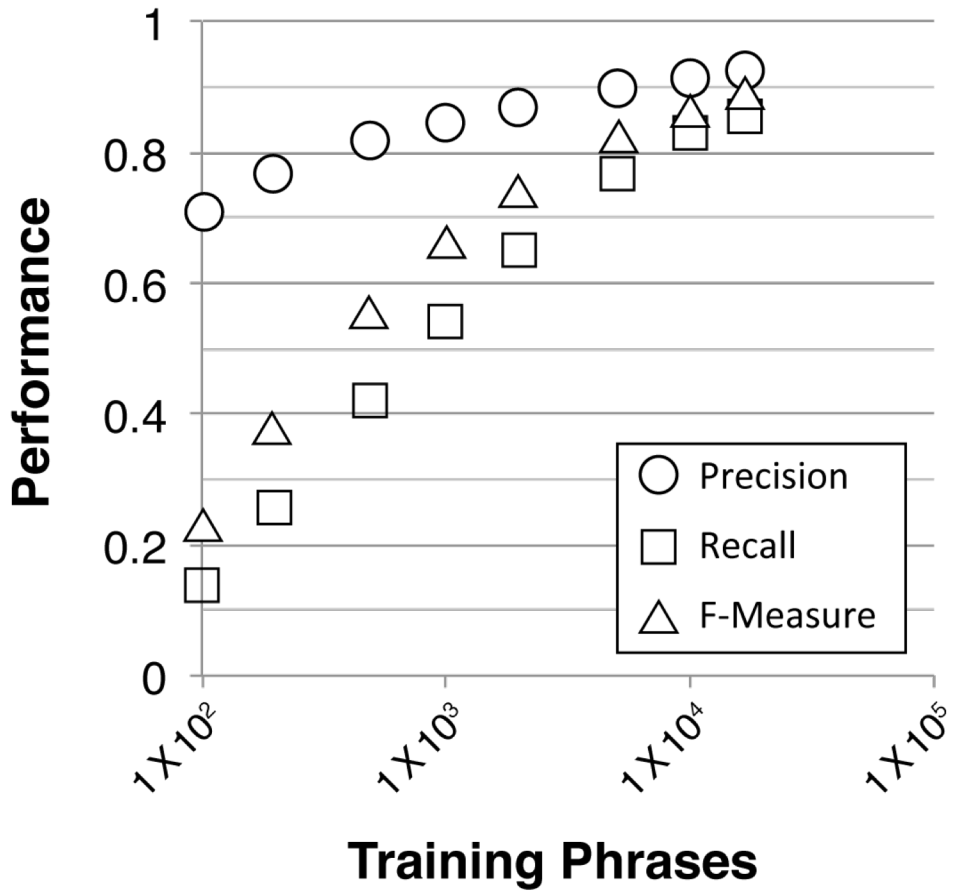


Figure 6. Precision, recall, and F-measure of the RapTAT tool on the i2b2 test data relative to the number of phrases used for training.

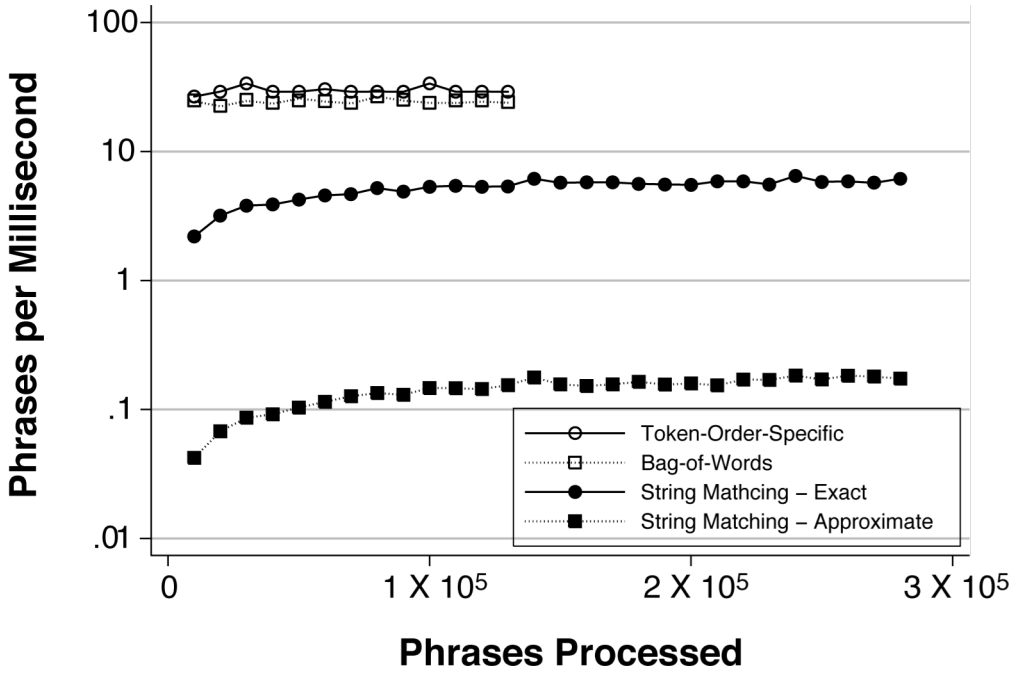


Figure 7. Rate of phrase-to-concept mapping relative to the number of phrases processed. Rates for the token-order-specific and bag-of-words classifiers as well as exact and approximate string-matching classifiers are included. The rates for the machine learning-based classifiers (token-order-specific and bag-of-words) end after approximately 140,000 phrases because the remaining phrases were used for training exclusively. No training was required for the string matching-based classifiers, so the analysis included all phrases.

DISCUSSION

Our study tested the feasibility of using a token-order-specific, naïve Bayes classifier-based, machine learning system to quickly analyze free-text phrases and accurately map them to associated concepts. The generalizability of the tool is suggested by its ability to accurately reproduce the mappings of both automated and manual annotations. In addition, the results demonstrate that the tool can rapidly process phrases both during training and during mapping of phrases to concepts. The discharge summaries used for our study contained, on average, 260 words and 216 mapped phrases. Our findings suggest that our tool requires just 7.3 milliseconds to map all the phrases from a single discharge summary. Such a tool should be able to support mapping of such summaries for near-real-time NLP, which could be useful in assisting annotators with regard to concept mapping, enhancing existing NLP tools by improving concept mapping efficiency, or providing rapid, interactive feedback following free-text entry. Temporal performance did not decrease during testing, which suggests that the tool should be scalable for mapping purposes. Also, the tool dealt with over 15,000 unique phrases using a 256 MB memory partition for the Java virtual machine, so there is potential for expansion based on current desktop computer configurations.

The use of hash tables likely contributed to system performance. This design choice provided high temporal efficiency during mapping because the approach constrained searches for maximizing equation 1 to only concepts associated with at least one of the phrase tokens. It also reduced spatial requirements because the system only stored actual associations between tokens and concepts in the training data rather than, for example, creating a matrix of all potential associations between every concept and token.

Averaged F-measures for the token-order-specific classifier were generally high with regard to the MCVS phrase mappings, being at least 0.92 for the top-level SNOMED CT conceptual groups and above 0.95 in the majority of cases. The F-measures were somewhat lower for the manual annotations from the i2b2 data set but still above 0.89 overall. The lower performance of RapTAT on the manual annotations was expected given that performance was evaluated on a test set rather than estimated using bootstrapping. Furthermore, it appears that performance was limited somewhat by the size of the training

set based on the steady increase of the F-measure learning curves (**Figure 6**). Finally, we would expect mapping consistency to be higher for automated than manual annotations, which would likely effect system performance.

The apparently high F-measures achieved by the RapTAT system are somewhat surprising in view of the implicit assumption by naïve Bayes models that all phrase tokens are conditionally independent given the mapped concept. The multinomial naïve Bayes model, such as the token-order-specific classifier used here, maintains this independence assumption; the probability of a given word occurring at a particular position within a sequence is assumed to be independent of the presence of other tokens at other positions within the sequence. A phrase such as "ischemic myocardial infarction," may violate the assumption, because the occurrence of "ischemic myocardial" would seem to increase the probability of the next word being "infarction." However, naïve Bayes classifiers often perform well even when the independence assumption is violated [79].

There have been numerous biomedical challenge tasks that evaluate performance with regard to the combination of named entity recognition and concept mapping, and the results of these challenges provide standards for evaluating new computational systems. A report on the 2010 i2b2 challenge indicated that the best system achieved an F-measure of 0.85 for exact matches and 0.92 for inexact matches [70, 80], while RapTAT achieved comparable F-measures ranging from 0.87 to 0.94. Because the i2b2 challenge systems were responsible for both named entity recognition and mapping, direct comparison of RapTAT to the systems tested in the i2b2 challenge is not possible. Unfortunately, few challenges, if any, have focused on concept normalization alone [53], but the i2b2 results suggest that RapTAT performs this particular task relatively well. A recent report did focus on normalization of diseases found in the Arizona disease corpus [53, 81]. The investigators enhanced two existing biomedical concept mapping systems, Peregrine and MetaMap, with rules to handle issues such as term variation and abbreviations [53]. The maximum F-score achieved in that study with regard to concept mapping was 0.736. However, even though the study focused on concept normalization, mapping still relied on automation of named entity recognition to identify phrases. Named entity recognition performance in that study, which achieved a maximum F-measure of 0.854, undoubtedly limited concept-mapping accuracy.

Our comparison of the token-order-specific to the bag-of-words and string-matching classifiers suggests that the former provided substantial advantages over the other methods. Performance of the token-order-specific classifier was considerably higher after training on phrases within 5 to over 100 documents (**Figure 3**). The performance of the bag-of-words classifier reported here actually exceeds that reported by Pakhomov, Buntrock, and Chute, who used a bag-of-words classifier to categorize medical diagnosis and found that precision, recall, and F-measure were 0.59, 0.51, and 0.54, respectively [68]. That study narrowed the use of the bag-of-words classifier to potentially difficult phrases, ones that could not be readily mapped using previous classifications, which may have diminished performance. Differences in the performance measures may also be related to the amount of classifier training, and further training might reduce the differences in precision, recall, and F-measure between the two classifiers (**Figure 3**). Nevertheless, our data also indicate that the token-order-specific classifier would likely outperform the bag-of-words classifier if training sets were limited in size.

One reason that performance can be diminished by a bag-of-words classifier is demonstrated by the phrase, "hemodynamically stable," which was present in 75 documents within the corpus and was mapped to a SNOMED-CT concept of the same name within the conceptual group "clinical finding." The word "stable" alone occurred much more frequently as a phrase than "hemodynamically stable" and was mapped to a concept within the "qualifier value" conceptual group by MCVS. So, when token position was not included as a feature by the bag-of-words classifier, the mapping of the phrase was strongly influenced by the high probability of association of "stable" with the qualifier value concept. In contrast, when "stable" was the second word in a phrase, it always mapped to "hemodynamically stable," and when token position was used as a feature by the token-order-specific classifier, it correctly mapped the phrase. Although increased computational accuracy often requires additional calculations leading to a decrease in processing speed, this did not appear to be the case in the present study. Processing speed for the token-order-specific classifier appeared to be similar to and possibly slightly faster than the bag-of-words classifier (**Figure 7**). The reason for this is likely that token sequence length is one determinant of classifier speed, and both classifiers must evaluate each of the tokens in a phrase during mapping. However, the bag-of-words classifier must

also evaluate the probabilities for tokens that are not in a phrase under evaluation (*cf*. Equation 2 *versus* 4).

Another advantage of the token-order-specific classifier is that it diminished the impact of phrase variability and concept ambiguity on performance (**Figure 4**). This could be important if one was planning to train the tool to reproduce human-generated phrase-to-concept mappings, which might be done if using the tool to automate or assist with manual annotations. Both inter- and intra-reviewer discrepancies during manual annotation can produce substantial variation in the phrases identified and the mappings selected. The amount of variation would likely be greater than that produced by an NLP application such as the MCVS tool used in the current study.

The speed and accuracy of the RapTAT tool is encouraging, but the present system has limitations. For one, the tool only selects the most probable concept mapping when provided with a phrase; it does not identify which raw text phrases should be annotated. However, previous work suggests that such a task is feasible. D'Avolio *et al* reported on use of the ARC tool to identify noun phrases within raw text [82]. Using a conditional random field classifier as the basis of machine learning, ARC automated retrieval of three different concepts and their associated phrases from an i2b2 document set, generating micro-averaged F-measures in the range of 0.80 – 0.83. In addition, there are a number of pre-processing steps that need to be carried out before the phrase-to-concept mapping carried out by RapTAT. These steps will add to the overall document processing time, but our analysis of the time required for database lookups, even when memory caching is used, suggest that the temporal performance of the RapTAT system could help existing NLP systems to move closer to near-real-time processing. Furthermore, because the system is trained using existing phrase-to-concept mappings, it will reproduce any inaccuracies generated by the system that created the initial mappings. Also, the tool needs to process an adequate number of training examples to accurately map phrase to concepts, and the number required increases as phrase variability and concept ambiguity increase. However, synthetic phrases similar to those likely to be found in the domain could be added to boost the training for rare or ambiguous concepts. The tool does not determine the assertion value of concepts, so currently there is no way to distinguish among positive, negative, and uncertain concepts. Similarly, the current version of the tool does not do any semantic

analysis. Therefore, once tool training is completed, phrases that might map to more granular concepts and abbreviations whose meaning depends on context only map to a single concept. In addition, it will not map abbreviations that were not included in the training data. Finally, RapTAT cannot combine simple concepts to generate compositional expressions, which are needed to more fully encode documented medical phrases [83]. Despite the limitations of RapTAT, the tool does provide a systematic method for learning and accurately reproducing both established and novel phrase-to-concept mappings, such as might be needed when applying NLP to a new domain. Current efforts to further develop the tool may allow users to train it to determine assertion values and delineate the concepts that might be combined to form compositional expressions.

In addition to addressing the current limitations discussed above, our future plans include using the system to generate an assistive annotation tool. Current manual annotation systems require a reviewer to first identify a phrase of interest and then select from a list of concepts for mapping. By training RapTAT to reproduce concepts selected by an annotator, it should be possible to automate the mapping process or limit the concepts presented to an annotator to only those with high probability. The RapTAT concept-mapping module has been incorporated into plug-ins and components of the GATE and Unstructured Information Management application (UIMA) NLP frameworks. Using this approach, RapTAT can also be used to systematically generate rapid concept mappers that are tailored to different domains and tasks and that can be used within larger, existing NLP systems.

CONCLUSION

Because RapTAT can accurately map phrases to a large repertoire of concepts distributed widely across an existing ontology, it could serve as an alternative mapping system within an existing NLP tool. Given more than 5 training instances, the F-measure exceeds 0.92, which should be sufficient for many tasks. In addition, with a mapping rate of ~30 phrases per millisecond, the system should be fast enough to readily support phrase-to-concept mapping within a near-real-time NLP system. Using the tool to fully automate the human annotation process will require further development in the form of identifying free text phrases for labeling.

CHAPTER 3

ASSISTED ANNOTATION USING RapTAT

BACKGROUND

Pre-annotation is one method that has been investigated for its potential ability to reduce the time and effort required to annotate a document. Its goal is to reduce the number of annotations a reviewer must add. Previous implementations used a dictionary generated specifically for the annotation task at hand or an existing NLP system. For example, Lingren *et al* created a dictionary tailored to generate pre-annotations in clinical trial announcements, focusing on the impact of pre-annotation on the ability of reviewers to label disease and symptom-related concepts. Pre-annotation decreased the time needed for review by 14-21% compared to fully manual annotation [7]. Investigations using existing NLP systems for pre-annotation of non-medical documents have reported decreases in annotation time of 50-58% for named entity recognition, part-of-speech tagging and parsing within non-medical documents [49-51].

Despite the reported benefits of pre-annotation, there are some potentially important considerations regarding its use. Inaccurate pre-annotations may require deletion or correction, and evidence indicates that time savings correlate with pre-annotation accuracy [95, 96]. For some tasks, pre-annotation may not alter annotation time [97], and the presence of multiple, inaccurate pre-annotations may actually increase annotation time [95, 98]. Also, pre-trained systems capable of pre-annotating for a specific task or medical realm either may not exist or be sufficiently accurate when used within a new domain. Although it is possible to create task-specific pre-annotation systems [7], doing so may require substantial effort and offset the time-savings afforded by pre-annotation. Furthermore, although some studies have found no evidence to suggest that pre-annotation induces bias or reduces quality of annotating text for biomedical concepts or part-of-speech [7, 8, 99], Fort and Sagot suggest that pre-annotations can induce bias leading to decreases in random errors but increases in systematic errors by reviewers [95].

The present study describes the design and evaluation of the full, RapTAT assisted annotation tool, which may provide as an alternative approach to previously described methods of pre-annotation. The concept mapping system described in the previous chapter is combined with a phrase identification system to generate a complete, machine-learning based pre-annotation system. In the study, we assess the impact of generating pre-annotations interactively using online, iterative machine learning as implemented in RapTAT on annotation burden. Specifically, the study evaluates whether RapTAT can support interactive, assisted annotation and reduce the time required for annotation without negatively affecting inter-annotator consistency or inducing annotation bias relative to manual review. The system is tested with regard to its ability to support the development of an annotated reference corpus that will eventually be used to train and test an external machine-learning based NLP system. The goal of that NLP tool is to detect clinical signs and treatments that can reveal the consistency with which providers adhere to American Heart Association (AHA) guidelines for CHF care. Following AHA guidelines has been shown to reduce hospital admissions, improve quality of life, and decrease mortality of CHF patients[84, 85], so rapidly identifying discrepancies between guidelines and care can help to mitigate decreases in care quality. To accomplish its mission, the NLP system will need to identify 7 concepts within clinical notes: 1) mentions of angiotensin converting enzyme (ACE) inhibitor administration; 2) mentions of angiotensin II receptor blocker (ARB) administration; 3) mentions of ejection fraction; 4) quantitative measures of ejection fraction; 4) mentions of left ventricular systolic function; 6) qualitative measures of left ventricular systolic function; and 7) documented reasons for not administering ACE inhibitors or ARBs when otherwise indicated.

METHODS

Sampling and Population

The study corpus consisted of CHF patient notes including discharge summaries, emergency department triage and nursing notes, internal medicine attending notes, neurology resident notes, physician discharge notes, physician history and physical notes, and primary care outpatient notes. Documents were selected from a larger corpus

consisting of a random sample of documents generated from September of 2007 to September of 2008 by 6 independent VA medical centers from the western U.S. Patients were excluded if they: a) participated in trials related to angiotensin-converting enzyme inhibitors or angiotensin receptor blockers; b) had comfort measure advanced directives; c) were fitted with heart assist devices (excepting pacemakers or defibrillators); or d) had a heart or heart/lung transplant. The final study corpus contained 404 documents from 171 patients. The Tennessee Valley and Salt Lake City Health System VA and University of Utah institutional review boards and research and development committees approved the study and granted a waiver regarding the need to obtain informed consent and HIPAA authorization.

Schema Development

A cardiology expert and 3 experienced annotators designed the annotation schema using an iterative process involving schema generation, annotation of a document sample, review of the annotations, and schema revision. The schema development process defined the key concepts that occur within the medical record and that relate to clinical care guidelines for CHF patients. According to the guidelines, patients in systolic heart failure with an ejection fraction of ≤40% should be treated with ACE inhibitors or, alternatively, ARBs [100]. The schema was designed to provide annotations so that the NLP tool could identify 1) evidence of heart failure, 2) whether the patient was receiving ACE inhibitors or ARBs, and 3) if a reason was provided for not prescribing ACE inhibitors or ARBs to heart failure patients. The final schema contained 7 concepts (**Table 6**), and the task of annotators was to identify phrases in the text that express those concepts.

Annotator Training

Four reviewers, all experienced in clinical note annotation, were responsible for annotation. All annotators were provided with annotation guidelines specific to the schema. Two were responsible only for manual annotations, and the remaining two carried out only RapTAT-assisted annotation. To train all reviewers with respect to the annotation schema, the creators of the schema used consensus annotation to generate a training set of 30

Table 6. Schema demonstrating the seven concepts annotated within the corpus used for assisted annotation. Text samples demonstrating phrases that should be annotated are also included.

Concept	Number of Documents Containing Concept	Number of Patients with Concept	*Sample Text (Annotated Phrases in Bold)
Angiotensin Converting Enzyme Inhibitor	272	132	"**ACEI**," "**ACE inhibitor**", "**Altace**", "**Vaseretic**," "**Captopril**," "**Lisinopril**"
Angiotensin II Receptor Blocker	107	53	"**ARB**," "**Angiotensin receptor blocker**," "**Sartans**," "**Losartan**"
Ejection Fraction	201	118	"**Estimated ejection fraction**," "**EF**", **"LVEF",** "**Ejection fraction**"
Ejection Fraction Quantitation	197	116	"EF=**60-70%**," "EF is **about 30%**," "Ejection fraction in the range of **40 to 50%**"
Left Ventricular Systolic Function/Dysfunction	79	51	"**LV systolic function**," "**Systolic dysfunction**," "**LV function**," "Normal LV size and **function**,"
Left Ventricular Systolic Function Value	76	48	"**Mild** systolic dysfunction," "Systolic function is **borderline normal**"
Reason Not on ACE Inhibitor/ARB	40	26	"**Elevated creatinine** levels," "**Developing sepsis**," **Patient refuses to take ACEI**", "**Renal disease"**

***** Examples corresponding to each concept were provided to reviewers as part of the annotation guidelines, but they were not meant to comprehensively represent all phrases that might refer to a given concept. For the concept "Reason not on ACEI/ARB," reviewers were instructed to annotate a phrase only when it was provided as an explicit reason for not prescribing one of the drugs.

documents distinct from the study corpus. Reviewers annotated the training set in batches of 10 using the Knowtator annotation tool (**Figure 8**) [101]. They were required to achieve an agreement score exceeding 80% between their annotations and the adjudicated training set before proceeding with review of documents in the study corpus, where

$$Agreement = \frac{Matches}{Matches + Non\text{–}Matches} \quad (1)$$

Annotation of the Study Corpus

Each document in the corpus was randomly assigned to one of 20 batches, and each batch contained 19-21 documents (**Figure 9**). The batches were used as units of analysis for statistical purposes and to identify document sets for training RapTAT during assisted annotation. Assisted reviewers annotated the first document batch without any pre-annotation to provide the initial training of the machine learning algorithms within RapTAT. The next batch was pre-annotated by RapTAT based on this training, displayed within Knowtator for review and correction by the assisted annotators, and the corrected annotations were entered into RapTAT to update its training before pre-annotating the subsequent batch. This iterative process of pre-annotation, correction, and updating of RapTAT training was carried out by separate instances of RapTAT for each of the two assisted reviewers, and it continued until the final batch had been corrected following pre-annotation. Manual annotators also used Knowtator for annotating each batch, but the documents were not pre-annotated. An adjudicator who was neither a manual nor assisted annotator reviewed the manual annotations to produce the reference standard. Inter-annotator agreement (IAA) was calculated using Equation 1.

Text Processing

RapTAT learns to pre-annotate documents with the likely annotations of a reviewer based on iterative feedback from that same reviewer. The tool used two different probabilistic models to estimate the likelihood of a reviewer 1) annotating a particular phrase and 2) mapping that phrase to a particular schema concept (**Table 6**). For both models, we

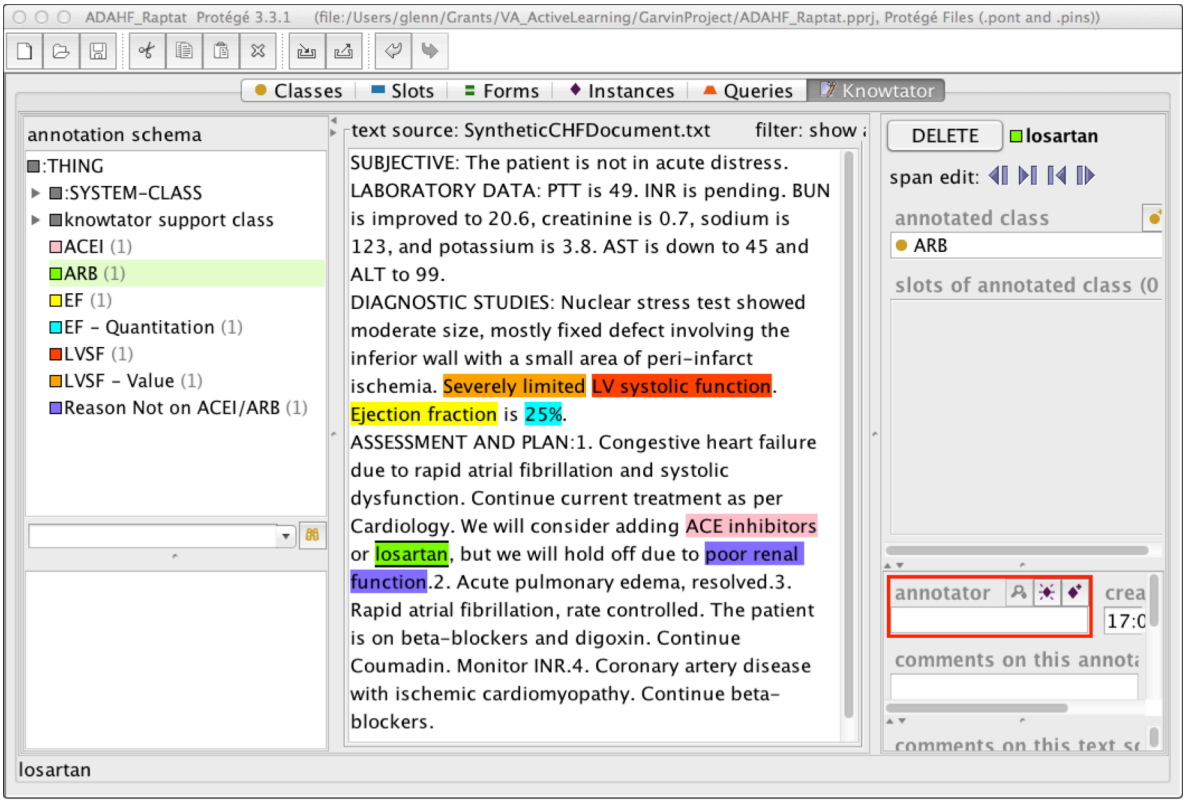


Figure 8 Screen capture of the Knowtator annotation plug-in within the Protégé application. The displayed document is synthetic but contains text representative of that found within the study corpus. Schema concepts are listed on the left. For each corpus document, reviewers use the input device of the computer to highlight all phrases mapping to one of the schema concepts and to select the concept associated with each highlighted phrase.

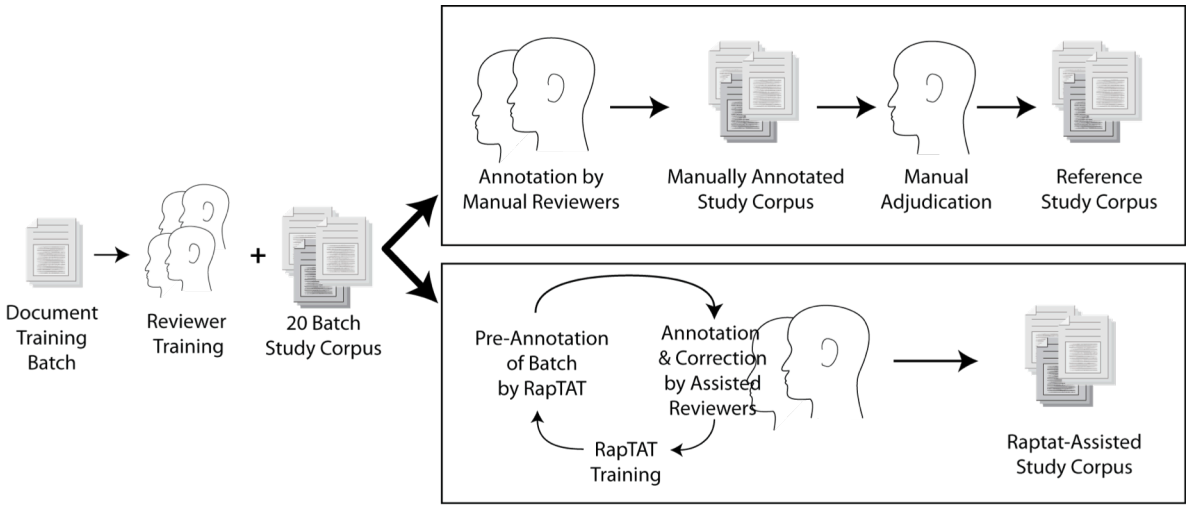


Figure 9 Document flow for generating the annotated study corpus using either manual review and adjudication or RapTAT-assisted review.

defined a token as a contiguous group of characters that corresponded to a word, value, or unit of measure, and a phrase as a contiguous sequence of one or more tokens that is representative of one of the schema concepts. Considering only token sequences (S) in a phrase without regard to context, the probability of annotation (A) of a given sequence is

$$P(A|S) = \frac{Number\ of\ Annotations\ of\ S}{Number\ of\ Occurrences\ of\ S} \quad (2)$$

We modified this equation for use in RapTAT because, by using this simple phrase identification model, subsequences shorter than the complete annotated phrase do not enter into probability calculations. For example, if "high fever of unknown origin" were annotated, the probability of annotating the subsequence "high fever" would not increase. Such a model could reduce recall by underestimating the probability of annotating token sequences that occur infrequently as complete annotated phrases even though they might occur frequently as subsequences. We therefore adjusted RapTAT to give partial credit to subsequences (**Table 7**). Each subsequence within an annotated phrase of length *i* in a sequence of length *j* was credited with an annotation count of *i*/*j* (numerator, Equation 2). Thus, the credited count was lower for sub-sequences that were particularly short relative to the length of the complete annotated phrase. All token sequences whose first token was not the first token in an annotated phrase were considered unlabeled and contributed equally to the number of sequence occurrences (denominator, Equation 2). Estimating the likelihood of mapping a phrase to a concept was accomplished using a multinomial naïve Bayes classifier. The classifier calculated the most probable concept for a given phrase, using the equation

$$P(C_i|T_1,\cdots,T_k) = \frac{P(C_i)\bullet P(T_1|C_i)\bullet\cdots\bullet P(T_k|C_i)}{P(T_1,\cdots,T_k)} \quad (3)$$

where $P(C_i)$ refers to the probability of occurrence of the *i*th concept, *k* is the number of tokens in the phrase and T_k refers to the token at the *k*th position in the sequence. The value of $P(T_k|C_i)$ is provided by the equation

Because the denominator in Equation 3, $P(T_1,\cdots,T_k)$, is constant when mapping a given phrase, finding the most probable concept for mapping is reduced to identifying the one that maximizes the numerator. Laplace smoothing adjusted for the occurrence of tokens

Table 7. Examples demonstrating how annotated phrases and their subsequences are counted during training, where *n* represents the number of tokens in the phrase.

Sequence Length	Phrase	Tokens	Number of Annotations Credited to Sequence
Full, Annotated Phrase	"LV systolic function"	3	1.0
n-1 Subsequence	"LV systolic"	2	0.67
n-2 Subsequence	"LV"	1	0.33
Full, Annotated Phrase	"Renal disease"	2	1.0
n-1 Subsequence	"Renal"	1	0.5

missing from the training data [73]. Multiple studies exist that have used this multinomial naïve Bayes models for text classification [102], although, to the best of our knowledge, the use of token position as a feature for medical concept mapping is unique to RapTAT.

RapTAT System Design

The RapTAT system was programmed in Java, and consisted of one module that determined the likelihood of phrase annotation and a second that determined the likelihood of a given phrase mapping to a particular concept (**Figure 10**). Phrases analyzed by the system were limited to contiguous sequences of ≤ 7 tokens. Before analysis by the two RapTAT modules, the text was pre-processed, which consisted of detecting sentence boundaries, dividing each sentence into tokens, removing "stop word" tokens ("and," "by," "for," "in," "nos," "of," "on," "the," "to," and "with"), and identifying and adding the appropriate part of speech to the token as a suffix. The pre-processing steps were carried out using the OpenNLP libraries (Apache Software Foundation). All versions of RapTAT are available at http://code.google.com/p/raptat/, and version 0.6a was used for this study.

Evaluation Measures

RapTAT was evaluated based on the number of true positives (TP), false negatives (FN), and false positives (FP) within the pre-annotations. Precision, recall, and F-measure provided measures of performance of the RapTAT tool and were calculated with respect to both the corrected annotations from the RapTAT-assisted annotators and the reference standard described above. A TP was defined as an overlap of one or more tokens between the RapTAT-generated and reference standard that mapped to the same concept. RapTAT automatically scored TPs, FPs, and FNs and calculated precision, recall, and F-measure according to the equations

$$Precision = TP / (TP + FP) \quad (4)$$

$$Recall = TP / (TP + FN) \quad (5)$$

$$F\text{-}Measure = 2 \bullet Precision \bullet Recall / (Precision + Recall) \quad (6)$$

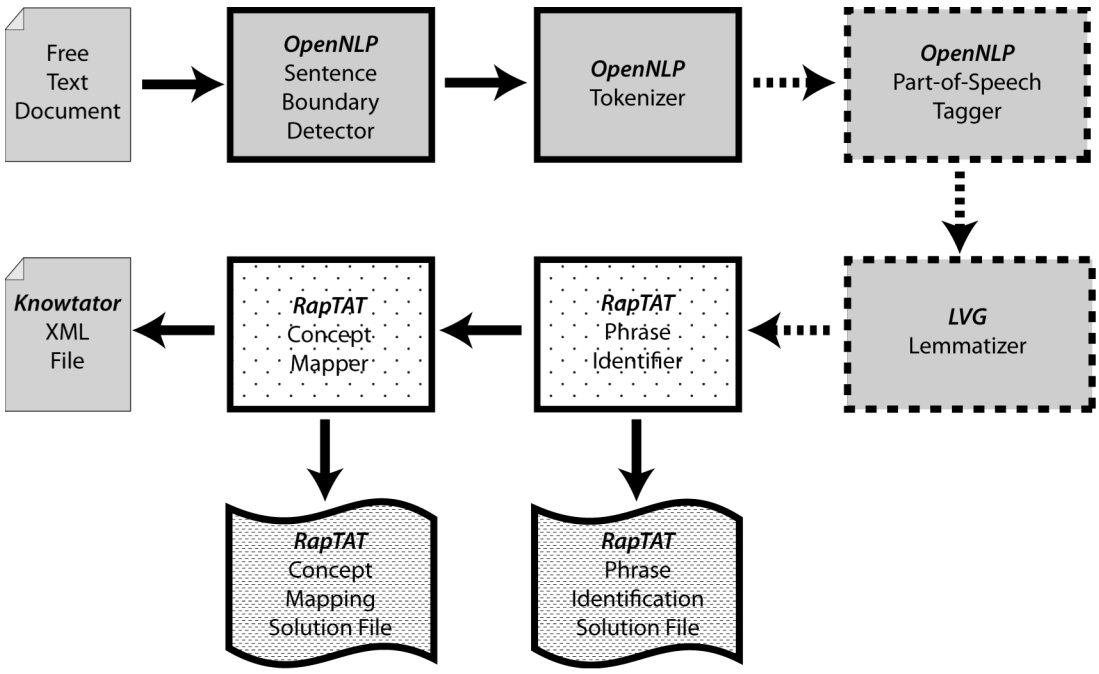


Figure 10 Data flow during training and pre-annotation by the RapTAT machine learning system. Dotted lines and arrows represent optional parts of the system that are available but were not used in this study, such as LVG lemmatization. Stippled patterns represent RapTAT specific modules (light stippling) and files (dense stippling).

We used leave-one-out cross-validation to estimate the performance of RapTAT with respect to each of the schema concepts. Cross-validation consisted of training RapTAT using all but one of the annotated documents from a given reviewer; RapTAT then generated annotations for the "left-out" document, which were compared to those of the reference standard. This process was repeated for each document and reviewer. Precision, recall, and F-measure for a given concept were calculated by combining the TPs, FPs, and FNs for that concept.

Reviewer Annotation Time and Rate

To assess batch-to-batch changes in annotation time, each RapTAT-assisted reviewer recorded the time required to review each document. Time per batch was normalized to batch size in kilobytes. Because correct pre-annotations might decrease and incorrect annotations might increase annotation time, we also calculated annotation rate of both manual and assisted reviewers with respect to only the annotations that were added or corrected. Correction was defined either modifying the beginning or end offsets of the annotation or changing the concept to which the phrase mapped. We defined the annotation rate as the number of annotations added or corrected per minute based on timestamps generated by Knowtator for each annotation. Because rates were not normally distributed, we determined the median rates for each reviewer and batch, and that data was used for statistical evaluations of the change in annotation rate as a function of batch number.

RapTAT System Training and Annotation Rates

To evaluate the training rate of the RapTAT system, we measured the time required to process the first 10 document batches. To evaluate the annotation rate, the corpus was divided into two independent training and test groups with 10 batches of documents in each. After processing the training documents, annotation rate of RapTAT was calculated based on the time spent pre-annotating the test documents. Times were normalized to document corpus size in kilobytes. Time required to read the corpus from disk into computer memory and read and write data structures before and after training was

excluded from all rate calculations. Heap size of the Java Virtual Machine was ≤1 GB. Training and testing were carried out on the VA Informatics and Computing Infrastructure (VINCI) server, which ran on an Intel Xeon quad-core processor running at 2.27 GHz and supplied with 128 GB of RAM. The operating system was Windows Server, 2008 R2 Enterprise.

Statistical Analysis

The study used simple linear regression to evaluate the statistical significance of changes in F-measure, annotation rate, and fraction of annotations correctly provided by RapTAT as a function of document batch. A "correct" RapTAT annotation was defined as a pre-annotation was neither added nor corrected by the reviewer. To compare the similarity of RapTAT-generated pre- annotations to the assisted and reference standard annotations, we ran paired *t*-tests on estimates of precision, recall, and F-measure across all batches. A Student's t-test was used to compare the number of annotations added or corrected by assisted *versus* manual reviewers. A two-sample proportion test was employed to identify statistical differences for single measures of IAA. All statistical analyses were carried out using Stata/IC 11.2 for Mac (Stata Corp., College Station, TX), and *p*-values of less than 0.05 were considered significant.

RESULTS

There was a notable decrease in annotation time from batch to batch for the RapTAT-assisted reviewers, especially over the first 6-7 batches, followed by a slower apparent decrease over batches 14-20 (**Figure 11; top**). Annotation time decreased by approximately 50% from the first to the last batch. Part of this decrease may be accounted for by the gradual decrease in the number of annotations that had to be added or corrected by the annotators over the course of annotation (**Figure 11; bottom**). Averaged over the entire corpus, the two manual annotators added 100 ± 18 (mean ± SD) annotations per batch. The assisted annotators added or corrected significantly less, 78 ± 12 annotations per batch, and 21 ± 9 annotations per batch were generated as pre-annotations by RapTAT during assisted annotation and did not require correction. To determine if the decrease in

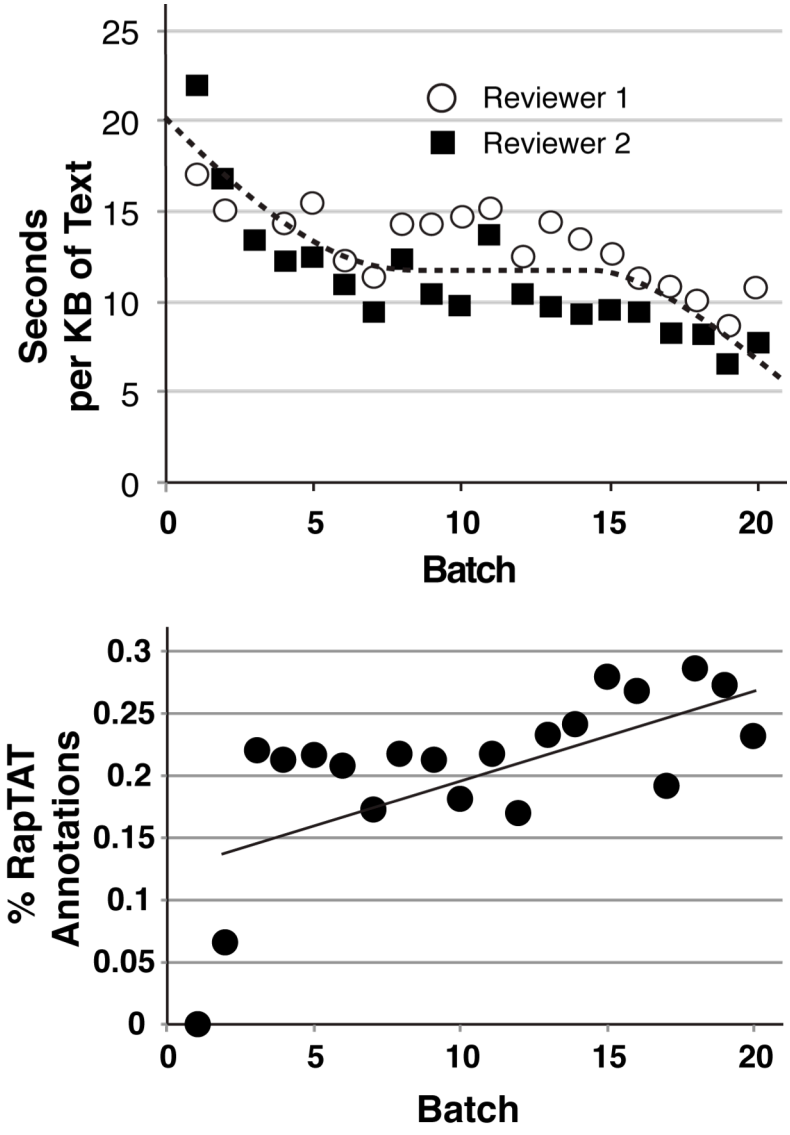


Figure 11 Time required to annotate text as a function of the number of document batches reviewed (top), and the fraction of all annotations that were uncorrected by reviewers and added only by RapTAT. For the annotation time plot (top), each symbol represents the time taken by a single RapTAT-assisted annotator for a particular batch of documents from the study corpus, and the dashed line represents the apparent, batch-to-batch trend in annotation time. For the plot of the fraction of annotations generated by RapTAT alone with no correction by the annotators (bottom), each symbol represents the total number of uncorrected annotations generated by RapTAT for each batch divided by the total number of annotated phrases in the batch; the least squares line of regression is also included, and the slope is significantly different from zero (p<0.01).

annotation number alone accounted for the marked decrease in annotation time (**Figure 11; top**), the rate of only adding or correcting annotations while excluding correct annotations generated by RapTAT was evaluated. The annotation rate of the assisted reviewers significantly increased over the course of annotation (+0.145 added or corrected annotations per minute per batch; 95% CI = 0.07 – 0.22) and approximately doubled from the first to the last batch (**Figure 12**). In contrast, the batch-to-batch change in annotation rate for the manual reviewers was significantly lower than that of the assisted annotators and did not change significantly over the course of annotation (+0.022 annotations per minute per batch; 95% CI = -0.004 to 0.048).

F-measure of the RapTAT pre-annotations relative to the assisted reviewer annotations steeply increased over the initial 5-6 batches (**Figure 13; left**). After a single batch of training, F-measure was 0.5-0.6 and increased over 0.80 after three batches. Precision and recall increased similarly. Linear regression analysis of the performance scores after the initial 5 batches revealed a non-significant trend toward a continuing increase in F-measure ($p=0.0623$ for slope > 0 by linear regression analysis). There was no evidence that pre-annotation had an adverse effect on annotation quality. Although the RapTAT pre– annotations were more similar to the annotations of the assisted reviewers than the reference standard based on significantly increased precision, recall and F-measure across all batches (paired *t*-test; $p<0.05$), the average increases were generally slight (≤0.046).

Also, differences in precision, recall and F-measure between pre-annotations measured relative to the reference standard and pre-annotations measured relative to the reviewer annotations (**Figure 13; left *versus* right**) remained roughly the same throughout the course of annotation. In addition, there was no evidence that pre-annotation adversely affected IAA, which was significantly greater for assisted than manual annotation for three individual concepts as well as overall (**Table 8**).

The performance of RapTAT with respect to its ability to accurately annotate phrases was concept dependent (**Table 9**). The four highest F-measures ranged from 0.80 to 0.97 and corresponded to the most highly prevalent concepts in the corpus, and the lowest

Table 8. Inter-annotator agreement (IAA) between the two manual and between the two RapTAT-assisted reviewers.

Concept	Average IAA (95% Confidence Interval)	
	Manual	Assisted
Angiotensin Converting Enzyme Inhibitor	0.89 (0.86-0.93)	***0.93**** (0.91-0.96)
Angiotensin II Receptor Blocker	0.81 (0.72-0.89)	***0.97**** (0.95-1.00)
Ejection Fraction	0.86 (0.80-0.93)	***0.97**** (0.95-1.00)
Ejection Fraction Quantitation	0.90 (0.85-0.94)	0.88 (0.83-0.92)
Left Ventricular Systolic Function/Dysfunction	0.82 (0.73-0.91)	0.76 (0.62-0.89)
Left Ventricular Systolic Function Value	0.85 (0.78-0.93)	0.77 (0.64-0.90)
Reason Not on ACE Inhibitor/ARB	0.58 (0.46-0.70)	0.54 (0.45-0.64)
Total (Combined Over All Concepts)	0.85 (0.81-0.88)	***0.89**** (0.87-0.91)

* indicates significant difference when comparing the IAA of the manual reviewers *versus* that of the assisted reviewers.

Table 9. Precision, recall, and F-measure of RapTAT for each schema concept.

Concept	Performance Measure		
	Precision	*Recall*	*F*
Angiotensin Converting Enzyme Inhibitor	0.97	0.94	0.95
Angiotensin II Receptor Blocker	0.99	0.96	0.97
Ejection Fraction	0.96	0.95	0.96
Ejection Fraction Quantitation	0.77	0.82	0.80
Left Ventricular Systolic Function/Dysfunction	0.61	0.82	0.70
Left Ventricular Systolic Function Value	0.83	0.37	0.51
Reason Not on ACE Inhibitor/ARB	0.36	0.12	0.18
Total (Combined Over All Concepts)	0.87	0.82	0.85

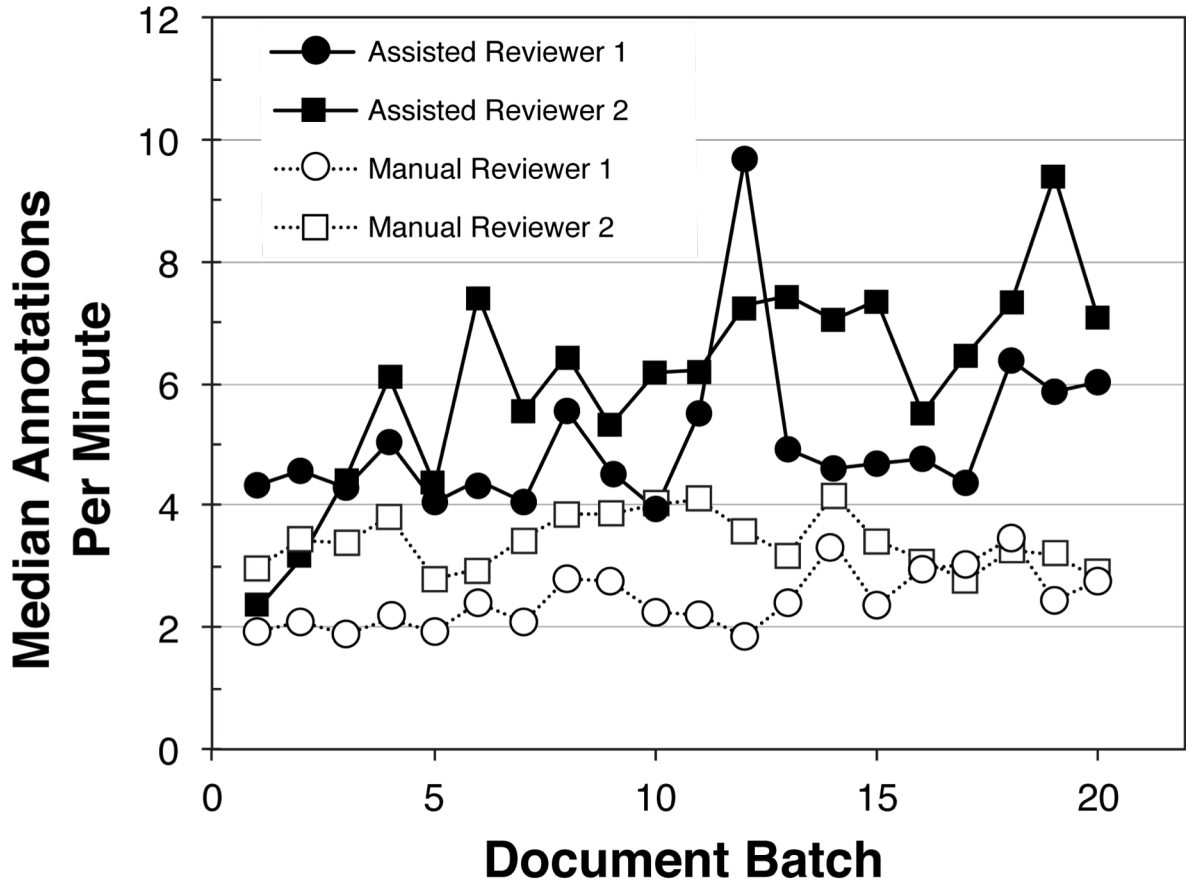


Figure 12 Annotation rate as a function of the number of document batches reviewed. Each symbol represents the rate for a single reviewer for a particular batch of documents from the study corpus. The rate represents the inverse of the time between adding or correcting annotations for both manual (open symbols) and RapTAT-assisted (closed symbols) reviewers.

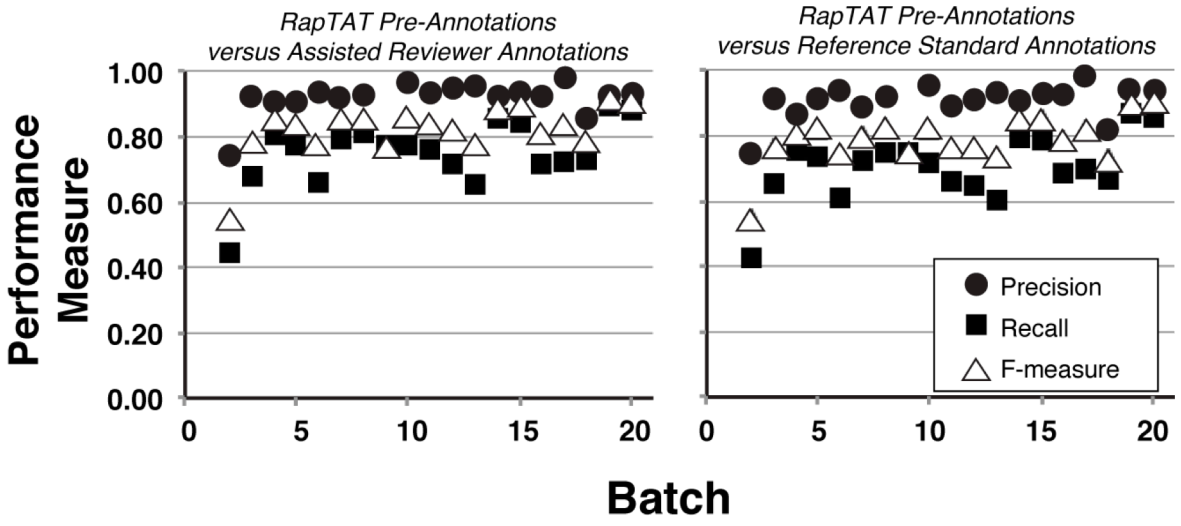


Figure 13 Precision, recall, and F-measure of the RapTAT tool as a function of the number of document batches used for training. Pre-annotations provided by the RapTAT tool were scored for performance *versus* either the assisted reviewer annotations (left) or the reference standard annotations (right).

F- measure was for the least prevalent concept, "Reason Not on ACE Inhibitor/ARB" concept (**Table 6**).

Processing speed of the RapTAT tool during annotation was 132.0 msec per kilobyte of text. "Pre-processing," which we define as sentence boundary detection, tokenization, part-of-speech detection, and stop word removal, took most of the time (123 msec); only 9 msec were required for training once the text was read into computer memory and pre-processed. Annotation rate by the tool was 116.6 msec, which consisted of 116 msec for pre-processing and 0.55 msec for phrase identification and concept mapping.

DISCUSSION

This study demonstrates that pre-annotation based on interactive, iterative machine learning can reduce the burden associated with creating an annotated corpus. Considering the annotation time and rate of the two assisted reviewers compared to the manual reviewers, we estimate that using assisted rather than manual annotation would have saved each reviewer roughly 16 h for annotation of the entire 404-document corpus. Also, our study did not find any evidence to suggest that pre-annotation introduces bias. Before the study, we were concerned that the closed feedback loop between RapTAT and each reviewer might induce a drift in the annotations, so that pre-annotations might closely match annotations of each reviewer but increasingly deviate from those of the reference or other annotator over the course of annotation. However, the IAA for the assisted annotators was actually equal to or higher than the manual annotators. Also, the precision, recall, and F-measure for the pre-annotations relative to the assisted annotations and for the pre-annotations relative to the reference standard remained similar throughout the course of annotation.

The F-measure of the pre-annotations relative to the assisted reviewer annotations was below 0.8 for the first few annotation batches, so the tool may only provide slight assistance in the early stages. This is a limitation of the iterative training needed for RapTAT compared to prior approaches that initially pre-annotate all documents using existing tools or ones created for the task. As RapTAT learns and improves, the number of

annotations that must be added or corrected decreases and the annotation time of the reviewers correspondingly decreases. Fort and Sagot examined the impact of pre-annotation accuracy on annotation time and found that increasing accuracy from 66.5% to 81.6% was associated with an ~50% decrease in annotation time [95]. In our study, F-measure reached 81% after 3 document batches, which suggests that approximately 60 documents may be required for training RapTAT to a level of accuracy such that its pre-annotations substantially reduce annotation time. The impact of training on F-measure was concept dependent, which may be partially related to concept prevalence, so the rate of increase in annotation speed as a function of the number of documents annotated may be slower for infrequent concepts.

In the current study, while RapTAT was used to generate the pre-annotations, annotators added and corrected pre-annotations using the separate Knowtator tool. Our goal is to eventually embed RapTAT within an annotation tool. This will allow annotators to update the machine learning algorithms after each document and obviate the import and export of data that was required in this study. When designing RapTAT, we were concerned that existing language models, such as maximum entropy Markov and conditional random fields, might not be sufficiently rapid to support iterative training and pre-annotation in a way that would not introduce delays during annotation. We therefore used language models and worked to implement algorithms that would be fast enough to support the interactive annotation process described in this study. Based on the annotation and system-training rate determined in this study, RapTAT should be readily capable of supporting real-time, interactive annotation. The current rate-limiting factor is disk access. Since one kilobyte of text equals approximately one-half a page, the current RapTAT system should take about one second to train on 4 pages or annotate 8 pages once the documents are read from disk and stored in computer memory.

The impact of the interactive approach to pre-annotation described here on annotation time appears to be within the range reported in other studies on pre-annotation, which decreased annotation time by 14-58% [7, 8, 49-51]. Interactive, assisted pre-annotation in the current study approximately doubled the annotation rate relative to manual reviewers.

Studies examining changes in IAA due to pre-annotation have been less consistent, with some studies reporting no change and another reporting an increase of 11% [7, 8, 98]. Interactive assisted annotation in the current study improved IAA by ~27%. Although some of the decrease in annotation time in our study was expected and likely due to the increased fraction of annotations correctly labeled by RapTAT, there was an unexpected increase in annotation rate unrelated to annotation number. One possible explanation is that correcting annotations may take less time than adding missing annotations. The existence of pre-annotations may also reduce the cognitive burden by decreasing the number of annotations that have to be identified on each document or helping to delineate document sections. With respect to the increase in IAA for assisted annotators, we theorize that pre-annotation by RapTAT may help reviewers to identify and annotate phrases they might otherwise overlook, thus reducing inter-annotator discrepancies. A potential benefit of increased IAA is a decrease in the adjudication workload.

Although previous studies have suggested that pre-annotation can reduce annotation burden, the iterative, machine learning-based approach to pre-annotations described here has some important advantages. One is that there is no need to identify or create a pre-annotation system because such a system is generated during the annotation process. RapTAT can be used without the linguistic and computational experience that might otherwise be required to implement a pre-annotation system. A second advantage is that the system carrying out pre-annotation is automatically optimized for the schema and intended domain via machine learning during annotation. Considering that low pre-annotation accuracy can slow the annotation process [95, 98], correctly tailoring the pre-annotation to the domain is important, and non-optimized pre-annotation tools, such as pre-existing systems or dictionaries developed for a task, may not be sufficient.

There have been previous reports on the use of machine-learning based pre-annotations for assisted annotation. Culotta *et al* described an iterative approach similar to the one described for RapTAT for training a named entity recognition system. Using simulations, they reported that their approach reduced the number of "actions" required by an annotator by 42% [103]. The MIST tool has been used to annotate protected health

information within medical documents, and it can be trained to identify other concepts [104]. Another annotation tool, BOEMIE, is reported to include the ability to use a similar interactive approach to assist with text annotation [105]. To the best of our knowledge, the impact of using MIST or BOEMIE on annotation time and IAA and their ability to support real-time interactive annotation have not been reported.

CONCLUSION

This study demonstrates that interactive, iterative machine learning as provided by RapTAT can assist with the annotation of text by gradually learning to produce accurate pre-annotations. Doing so substantially reduces the annotation time by decreasing the number of annotations that must be added by reviewers and helping to accelerate the rate at which reviewers are able to add missing annotations and correct inaccurate ones. RapTAT also improves IAA, which should accelerate adjudication when using multiple reviewers for annotation. Integration of RapTAT or a similar system with an annotation tool could help to mitigate an important barrier to implementing NLP systems in the medical realm.

CHAPTER 4

FUTURE DIRECTIONS

The studies described in this thesis demonstrate that online machine learning can assist and accelerate annotation of text. A question that remains to be explored is whether there are refinements to the described approach that could further improve the RapTAT assisted annotation tool and reduce the annotation burden. A second question is whether, given the benefits provided here by online learning, this approach might also be used for other text classification tasks, such as document and patient classification. A third question is whether the language processing algorithms created in these studies can be used to generate other medical informatics tools for improving patient care. The following discussion examines each of these questions in turn.

LANGUAGE MODEL REFINEMENTS

A few potential modifications to the assisted annotation tool described in this thesis might further reduce the annotation burden. One potential modification would be to refine the text analysis to extract further information about the extracted concepts. Annotation of medical notes often entails more than simple concept identification. Perhaps the most straightforward example is the need to include whether a concept found in the text was negated or uncertain. Clearly, negated symptoms, such as found in the example, "the patient denies any exercise intolerance," has very different implications from an assertion such as "the patient reports extreme exhaustion after exercise." Capturing both asserted and negated symptoms and other concepts can be important in a medical record. Assuming that a narrative describing a patient encounter is accurate, text describing a negated symptom indicates that the symptom was investigated and found to be absent. In contrast, for symptoms are not even mentioned in the text, there is generally no way to determine whether such symptoms are present or absent. To determine whether a concept is negated generally requires an analysis of the context in which it appears. Several tools have been developed that use context to accurately determine the assertion status of concepts in text. Examples include NegEx, which is a rule-based system with a reported precision of 84.5% and recall of 77.8% [106], and NegScope, which uses a

machine learning-based, conditional random field model supplemented with a dictionary of negation terms to establish negation status and has a reported precision of 91.7% and recall of 92.7% [107]. Other contextual attributes of concepts that can be important in the medical realm include the concept "experiencer," which describes whether a symptom or treatment was experienced by a patient or someone else such as a family member, and concept "temporality," which describes whether the occurrence of an entity pertains to the past, present, or future. These particular attributes are now extracted by the rule-based ConText tool with a precision and recall of 74.2% and 67.4%, respectively, for temporality and 100% and 50%, respectively, for experiencer [108]. The performance of the NegEx, NegScope, and ConText tools suggests that it should be feasible to expand a tool like RapTAT to use online learning to assist with assigning attributes to concepts.

Other language models may be more accurate than the Bayesian concept mapper and phrase identification models described in our studies. Given that the speed of assisted annotation is dependent on pre-annotation accuracy [95, 98], finding a more accurate model could further increase the rate of annotation. In the 2010 i2b2 challenge, the task for groups entering the challenge was to identify medical problems, tests, and treatments in clinical text, and conditional random field (CRF) language models were found, in general, to perform best [70, 109, 110]. For the studies in this thesis, we generated a novel, probabilistic method of phrase identification because CRFs require user input to identify and define the "features" to include, which will affect model performance. Concept recognition features can be simple, such as the tokens within the text, or complex, such as the nature of multiple, surrounding tokens. Increasing the number of features can improve model accuracy but may also increase training time and lead to over-fitting. Also, identifying features for potential inclusion may require linguistic expertise that is not readily available to someone using a RapTAT-like tool. Nevertheless, because feature selection takes place during the training of CRFs, it may be possible to generate and include an extensive number of text features in the model initially and allow online training to select those that most improve accuracy. Another characteristic of CRFs that would probably have to be addressed is that they are typically trained offline using the entire training corpus. A method of incrementally training CRFs, which is required for

implementing online learning, has recently been described [111]. The investigators used a modification of stochastic gradient descent, which gradually updates the model with new training instances, and the modified optimization method obtained an order of magnitude increase in training speed relative to offline, batch training [111]. Whether this increase in speed is sufficient to support real-time updating of the language model and feedback to an assisted reviewer as done in this study remains to be established.

Although the time requirement is a substantial barrier to annotation, and RapTAT appears to lower that barrier by accelerating annotation, another important hurdle is that annotation of medical notes often requires reviewers with substantial expertise. For example, if the annotation task was to identify free text phrases signifying dehydration, a reviewer inexperienced in medical terminology might have a hard time accurately recognizing such phrases. With substantial training in the domain and the annotation task, it might be possible to use less experienced reviewers for annotation, but the time spent training reviewers itself adds to the annotation burden. An online learning system like RapTAT might be able to reduce the expertise requirement. If RapTAT or a similar system was pre-trained using the annotations of an expert reviewer, the pre-annotations provided by such a system might help to guide inexperienced reviewers regarding how to perform the annotation task. Based on the results of the studies in this thesis, 50-60 documents might be sufficient for pre-training and generating a reasonably accurate (F-measure > 0.80) pre-annotation system.

ONLINE LEARNING AND DOCUMENT CLASSIFICATION

The online learning approach used by RapTAT very likely has uses beyond those described in this thesis, such as training systems to classify medical documents. A number of machine learning-based systems have been described for carrying out document classification [112]. As with NLP, these systems must be trained using a corpus of manually classified documents, and generating such a corpus is expensive. Online learning might reduce the time spent on manual classification by switching the task from one of full document review to one of only identifying incorrect classifications. It might also reduce the time spent on manual classification by minimizing the size of the training set. In our studies, the primary

goal was to speed the annotation of corpus of given size rather than to train the RapTAT algorithms. Our intention was then to use that corpus for training other, external NLP algorithms. However, online learning can be and typically is used for direct algorithm training. Because performance is monitored continuously during the training that occurs as a part of online learning, the manual part of the process, whether it involves text annotation or document classification, can stop as soon as the system being trained reaches an acceptable level of performance. With offline batch training, there is generally no way to determine *a priori* the number of documents required to achieve adequate performance, so reviewers may examine more documents than actually needed to assure that there are enough to generate an accurate system.

The potential use of online learning for training systems with respect to document classification has been demonstrated by Borodin *et al* [113]. They used a modified, centroid-based classification algorithm for online learning to classify medical literature documents. Their purpose was not to accelerate training but to use online learning to adaptively respond to changes in documents over time and minimize potential decreases in performance. They found that the accuracy of their algorithm with regard to classifying a medical literature corpus was higher when trained using online learning than when using batch offline learning. While the Borodin study demonstrates that online learning can be used for document classification training, whether online learning can be used to accelerate such training has not been established.

NEAR REAL TIME NATURAL LANGUAGE PROCESSING

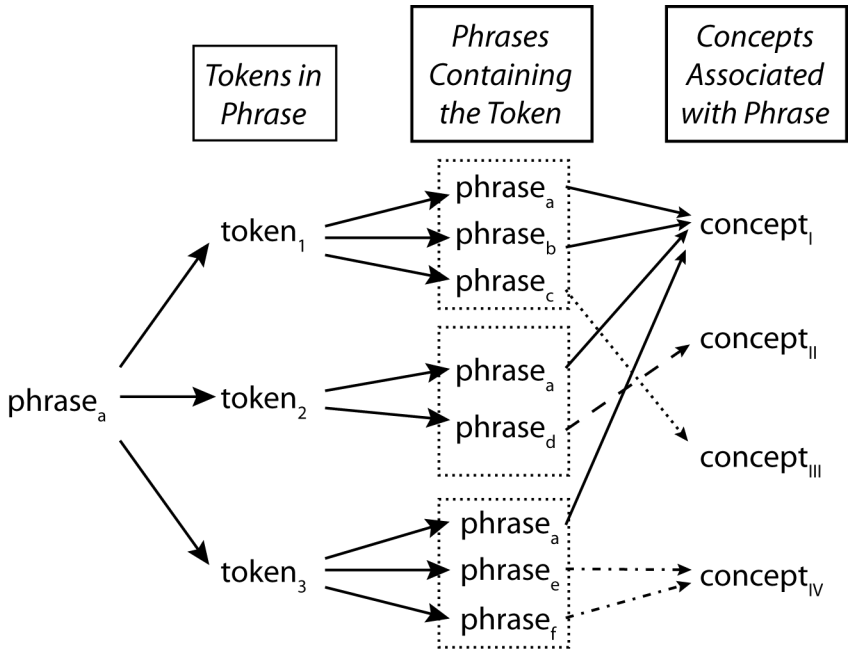
The initial purpose of developing the RapTAT algorithms was to support assisted annotation. Based on the studies included in this thesis, it appears that these algorithms can be trained to achieve sufficient accuracy to carry out common NLP tasks such as concept recognition and normalization of the phrases used for expressing a particular concept. A number of NLP systems already exist that can accurately recognize or normalize concepts, examples of which include MCVS [19], the Mayo Clinic Autocoder [68], the SNOMED Categorizer (SNOCat) [57], KnowledgeMap [21], IndexFinder [58], MedLEE [59], MetaMap [23], Metaphrase [61], MicroMeSH [62], PhraseX [64], SAPHIRE [114], and

SENSE [66]. The advantage of the RapTAT algorithms over those within existing systems appears to be the speed and scalability. To date, the largest corpus to which RapTAT has been applied consisted of 18,000 clinical notes averaging 1-2 pages in length. Processing, including file input and output, took 90 minutes, equivalent to approximately three documents each second. The rate did not change during processing, which suggests that the system may scale for processing larger corpora and is potentially capable of processing 288,000 documents per day. In comparison, a report on the well-established MedLEE NLP system suggests that it is capable of processing approximately 20,000 documents per day [40].

Two potential uses of RapTAT afforded by its scalability and speed are for medical surveillance and clinical feedback at the point of care, respectively. The ability of RapTAT to process large numbers of documents suggests that it could be used for continuous monitoring of the text within large document corpora produced daily by healthcare centers. Potential examples include monitoring of adverse effects from medical devices or medications following Federal Drug Administration approval, surveillance for unexpected clustering of disease outbreaks, and oversight of the ongoing rate of medical procedure complications. One potential use with respect to the ability of RapTAT to provide near real time feedback is the automated generation of problem lists based on the text created during a clinical encounter. Such a system might aid the clinician and his or her colleagues by assisting in the detection and explicit documentation of clinical problems. The clinician could even provide feedback to the system following problem list creation. In this way, the system could be individually tailored to the type of expressions used by the clinician and the typical problems encountered. Near real time analysis of clinical notes at the point of care could also be used for automated retrieval of knowledge relevant to the concepts present in the text. Cimino first proposed the term “infobuttons” for links within the electronic medical record to electronic information resources [115]. Not surprisingly, infobutton use is reported to increase when contextual information is used to increase the relevancy of the linked information [116]. Considering the importance of context with respect to identifying pertinent resources, the ability to quickly process the information provided within the free text could provide additional context and further increase

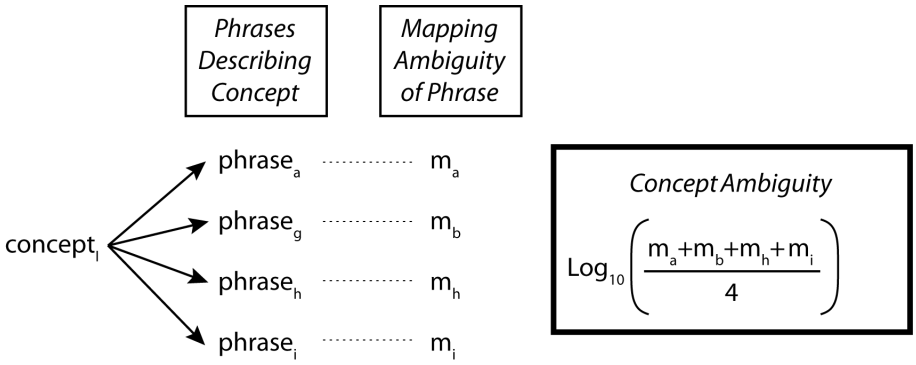
relevance. Del Fiol and Haug have shown that machine learning can significantly improve the selection of appropriate resource links [117]. The key concepts of interest within a clinical system and the resources that can provide information about those concepts can change or "drift" over time, and Del Fiol and Haugh demonstrated that periodically retraining the system to adapt to drifts over time could improve the ability of the system to select relevant resources. They also showed that they could increase system performance by tailoring the system to a particular user based on machine learning. Online learning could be used to continuously update the system to respond to conceptual and resource changes over time. It might also be used to automatically tailor the system to individual providers based on feedback regarding the utility of Infobutton links identified by the system.

An additional advantage provided by using RapTAT as a full NLP system is that it opens up the potential for also using active learning during training. As mentioned earlier, active learning can decrease the cost of annotating text by actively involving the learning algorithm in the document selection process [86]; its goal is to train the system while requiring as few samples as possible. Its implementation within RapTAT could complement the use of online learning, which speeds up the rate of annotation and tracks tool performance during training. Active learning has been applied in a wide variety of language processing tasks [87], examples of which include part-of-speech tagging [88, 89], text categorization [90, 91], named entity recognition [92, 93], and classification of assertions [94], and it has been reported to reduce the number of required training samples by 38-63% [92, 94]. To the best of our knowledge, there are no reports describing the impact of combining active with online learning in a single system on overall annotation burden, and studying such a burden could be a worthwhile are for investigation.



Mapping Ambiguity – This measure reflects the number of concepts to which a given phrase can be mapped. The above figure demonstrates how mapping ambiguity is determined for a theoretical example, $phrase_a$, which is defined by a sequence of three tokens, $token_1$, $token_2$, and $token_3$. A concrete example of such a phrase would be "acute myocardial ischemia" with the tokens being "acute," "myocardial," and "ischemia." For a token-order-specific (TOS) classifier, we consider "similar" phrases to $phrase_a$ (dotted squares in the figure) to be all phrases that contain a particular token, such as $token_1$, at the same position within the token sequence that defines the phrase. For a bag-of-words (BOW) classifier, similar phrases to $phrase_a$ would be those containing $token_1$ at any position within their token sequences. In the figure, $phrase_a$ is similar to itself as well as $phrase_b$ and $phrase_c$. For example, a phrase such as "acute myocardial ischemia" would be similar to "acute hepatitis" according to both the TOS and BOW classifiers because the token "acute" is in the first position in both phrases. It would also be similar to "relapsing acute pancreatitis" according to the BOW but not the TOS classifier. "Similar" phrases might be mapped to the same or different concept in SNOMED CT or another user-defined ontology. As shown in the figure, $phrase_a$ and $phrase_b$ are similar and map to $concept_I$. They are also similar to $phrase_c$, but it maps to a different concept, $concept_{III}$. The phrases "acute myocardial ischemia" and "acute coronary insufficiency" would be examples of similar phrases that map to the same SNOMED CT concept, while "acute hepatitis" would be a similar phrase that mapped to a distinctly different concept.

To calculate mapping ambiguity, we need to know the number of tokens in the phrase and the concepts associated with these tokens. In the theoretical example in the figure, $phrase_a$ is similar to itself, $phrase_b$, $phrase_c$, $phrase_d$, $phrase_e$, and $phrase_f$, which are associated with 4 concepts, $concept_I$, $concept_{II}$, $concept_{III}$, and $concept_{IV}$. We normalize this value to the number of tokens to offset the probability that a longer sequence with more tokens is expected to be associated with more concepts. For this example, the mapping ambiguity of $phrase_a$ is 4/3 or 1.33.



Concept Ambiguity – This measure quantifies the degree to which phrases that map to a given concept are distinctly associated with that concept. Concept ambiguity decreases as the mapping ambiguity of the phrases associated with the concept decrease. For example, in the above figure, if $phrase_a$, $phrase_b$, $phrase_c$, $phrase_d$ all have a token sequence length of one and are associated with only one concept, $concept_I$, then the mapping ambiguity of each is one, and the concept ambiguity is base 10 log of the average mapping ambiguity, which is $\log_{10}(4/4)$ or zero.

Note that it is possible for concept ambiguity to be less than zero. If a phrase has a token length, L, of greater than one but maps to only one concept, its mapping ambiguity will be 1/L, a value less than one. If the average mapping ambiguity is less than one, the log of this value and thus concept ambiguity will be a negative value.

REFERENCES

1 Xu H, Stenner SP, Doan S, et al. MedEx: a medication information extraction system for clinical narratives. *J Am Med Inform Assoc*. 2010;**17**:19-24.

2 Ohno-Machado L. Realizing the full potential of electronic health records: the role of natural language processing. *J Am Med Inform Assoc*. 2011;**18**:539.

3 Deleger L, Grouin C, Zweigenbaum P. Extracting medical information from narrative patient records: the case of medication-related information. *J Am Med Inform Assoc*. 2010;**17**:555-8.

4 Weed LL. The problem oriented record as a basic tool in medical education, patient care and clinical research. *Ann Clin Res*. 1971;**3**:131-4.

5 Rosenbloom ST, Stead WW, Denny JC, et al. Generating Clinical Notes for Electronic Health Record Systems. *Appl Clin Inform*. 2010;**1**:232-43.

6 Rosenbloom ST, Denny JC, Xu H, et al. Data from clinical notes: a perspective on the tension between structure and flexible documentation. *J Am Med Inform Assoc*. 2011;**18**:181-6.

7 Lingren T, Deleger L, Molnar K, et al. Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *J Am Med Inform Assoc*. 2013;10.1136/amiajnl-2013-001837.

8 Névéol A, Islamaj Doğan R, Lu Z. Semi-automatic semantic annotation of PubMed queries: a study on quality, efficiency, satisfaction. *J Biomed Inform*. 2011;**44**:310-8.

9 South BR, Shen S, Jones M, et al. Developing a manually annotated clinical document corpus to identify phenotypic information for inflammatory bowel disease. *BMC Bioinformatics*. 2009;**10 Suppl 9**:S12.

10 Chapman WW, Nadkarni PM, Hirschman L, et al. Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions. *J Am Med Inform Assoc*. 2011;**18**:540-3.

11 Rosenbloom ST, Denny JC, Xu H, et al. Data from clinical notes: a perspective on the tension between structure and flexible documentation. *J Am Med Inform Assoc*. 2011;**18**:181-6.

12 Nadkarni PM, Ohno-Machado L, Chapman WW. Natural language processing: an introduction. *J Am Med Inform Assoc*. 2011;**18**:544-51.

13 Ash JS, Berg M, Coiera E. Some unintended consequences of information technology in health care: the nature of patient care information system-related errors. *J Am Med Inform Assoc*. 2004;**11**:104-12.

14 McDonald CJ. The barriers to electronic medical record systems and how to overcome them. *J Am Med Inform Assoc*. 1997;**4**:213-21.

15 Walji MF, Kalenderian E, Tran D, et al. Detection and characterization of usability problems in structured data entry interfaces in dentistry. *Int J Med Inform*. 2013;**82**:128-38.

16 Yu P, Zhang Y, Gong Y, et al. Unintended adverse consequences of introducing electronic health records in residential aged care homes. *Int J Med Inform*. 2013;**82**:772-88.

17 Zeng QT, Goryachev S, Weiss S, et al. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. *BMC Med Inform Decis Mak*. 2006;**6**:30.

18 Friedman C, Shagina L, Lussier Y, et al. Automated encoding of clinical documents based on natural language processing. *J Am Med Inform Assoc*. 2004;**11**:392-402.

19 Elkin PL, Brown SH, Husser CS, et al. Evaluation of the content coverage of SNOMED CT: ability of SNOMED clinical terms to represent clinical problem lists. *Mayo Clin Proc*. 2006;**81**:741-8.

20 Savova GK, Masanz JJ, Ogren PV, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc*. 2010;**17**:507-13.

21 Denny JC, Smithers JD, Spickard A, 3rd, et al. A new tool to identify key biomedical concepts in text documents. *AMIA Annu Symp Proc*; 2002.

22 Denny JC, Smithers JD, Miller RA, et al. "Understanding" medical school curriculum content using KnowledgeMap. *J Am Med Inform Assoc*. 2003;**10**:351-62.

23 Aronson AR. Effective mapping of biomedical text to the UMLS metathesaurs: the MetaMap program. *AMIA Annu Symp Proc*; 2001.

24 Garla V, Lo Re V, 3rd, Dorey-Stein Z, et al. The Yale cTAKES extensions for document classification: architecture and application. *J Am Med Inform Assoc*. 2011;**18**:614-20.

25 Cimino JJ, Bright TJ, Li J. Medication reconciliation using natural language processing and controlled terminologies. *Stud Health Technol Inform*. 2007;**129**:679-83.

26 Day S, Christensen LM, Dalto J, et al. Identification of trauma patients at a level 1 trauma center utilizing natural language processing. *J Trauma Nurs*. 2007;**14**:79-83.

27 Murff HJ, FitzHenry F, Matheny ME, et al. Automated identification of postoperative complications within an electronic medical record using natural language processing. *JAMA*. 2011;**306**:848-55.

28 Elkin PL, Froehling DA, Wahner-Roedler DL, et al. Comparison of natural language processing biosurveillance methods for identifying influenza from encounter notes. *Ann Intern Med*. 2012;**156**:11-8.

29 Matheny ME, Fitzhenry F, Speroff T, et al. Detection of infectious symptoms from VA emergency department and primary care clinical documentation. *Int J Med Inform*. 2012;**81**:143-56.

30 Shiner B, D'Avolio LW, Nguyen TM, et al. Measuring Use of Evidence Based Psychotherapy for Posttraumatic Stress Disorder. *Adm Policy Ment Health*. 2012;10.1007/s10488-012-0421-0.

31 Workman TE, Fiszman M, Hurdle JF. Text summarization as a decision support aid. *BMC Med Inform Decis Mak*. 2012;**12**:41.

32 Denny JC, Peterson JF, Choma NN, et al. Development of a natural language processing system to identify timing and status of colonoscopy testing in electronic medical records. *AMIA Annu Symp Proc*; 2009.

33 Denny JC, Choma NN, Peterson JF, et al. Natural language processing improves identification of colorectal cancer testing in the electronic medical record. *Med Decis Making*. 2012;**32**:188-93.

34 Wahner-Roedler DL, Welsh GA, Trusko BE, et al. Using natural language processing for identification of pneumonia cases from clinical records of patients with serologically proven influenza. *AMIA Annu Symp Proc*. 2008:1165.

35 Chapman WW. Closing the gap between NLP research and clinical practice. *Methods Inf Med*. 2010;**49**:317-9.

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

37 Wang X, Hripcsak G, Markatou M, et al. Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study. *J Am Med Inform Assoc*. 2009;**16**:328-37.

38 Selected Veterans Health Administration Characteristics: FY2002 to FY2012: U.S. Department of Veterans Affairs; 2012.

39 National Center for Veterans Analysis and Statistics. 2012 VHA Facility Quality and Safety Report: Department of Veterans Affairs; 2012.

40 Friedman C. Towards a comprehensive medical language processing system: methods and issues. *AMIA Annu Symp Proc*. 1997:595-9.

41 Apache UIMA. 2013, Access Date; URL: http://uima.apache.org/index.html

42 Cunningham H, Maynard D, Bontcheva K, et al. *Developing Language Processing Components with GATE Version 7 (a User Guide)*: University of Sheffield, Dept. of Computer Science; 2013.

43 Cunningham H, Humphreys K, Gaizauskas R, et al. Software infrastructure for natural language processing. *Proceedings of the Fifth Conference on Applied natural language processing*; Washington, DC: Association for Computational Linguistics; 1997.

44 Goldstein DE. *Medical informatics 20/20 : Quality and Electronic Health Records through Collaboration, Open Solutions, and Innovation*. Sudbury, Mass.: Jones and Bartlett Publishers; 2007.

45 Lorenzi NM, Riley RT. *Managing Technological Change : Organizational Aspects of Health Informatics*. 2nd ed. New York: SpringerScience+Business Media; 2004.

46 Lorenzi NM, Kouroubali A, Detmer DE, et al. How to successfully select and implement electronic health records (EHR) in small ambulatory practice settings. *BMC Med Inform Decis Mak*. 2009;**9**:15.

47 Hsiao I, Brusilovsky P. Modeling peer review in example annotation. *ICCE, The 16th International Conference on Computers in Education*; Taipei, Taiwan: ICCE; 2008.

48 Crystal MR, Kubala F, MacIntyre R. Studies in data annotation effectiveness. *Proceedings of the DARPA Broadcast News Workshop*; Herndon, Virginia; 1999.

49 Chiou F-D, Chiang D, Palmer M. Facilitating treebank annotation using a statistical parser. *First International Conference on Human Language Technology Research*; San Diego: Association for Computational Linguistics; 2001.

50 Ganchev K, Fernando P, F., Mandel M, et al. Semi-automated named entity annotation. *Linguistic Annotation Workshop*; Prague, Czech Republic: Association for Computational Linguistics; 2007.

51 Marcus MP, Marcinkiewicz MA, Santorini B. Building a large annotated corpus of English: the penn treebank. *Comput Linguist*. 1993;**19**:313-30.

52 Hahn U, Beisswanger E, Buyko E, et al. Active Learning-based corpus annotation--the PathoJen experience. *AMIA Annu Symp Proc*. 2012;**2012**:301-10.

53 Kang N, Singh B, Afzal Z, et al. Using rule-based natural language processing to improve disease normalization in biomedical text. *J Am Med Inform Assoc*. 2013;**20**:876-81.

54 Tsuruoka Y, McNaught J, Ananiadou S. Normalizing biomedical terms by minimizing ambiguity and variability. *BMC Bioinformatics*. 2008;**9 Suppl 3**:S2.

55 Naderi N, Kappler T, Baker CJ, et al. OrganismTagger: detection, normalization and grounding of organism entities in biomedical documents. *Bioinformatics*. 2011;**27**:2721-9.

56 Pakhomov SV, Hanson PL, Bjornsen SS, et al. Automatic classification of foot examination findings using clinical notes and machine learning. *J Am Med Inform Assoc*. 2008;**15**:198-202.

57 Ruch P, Gobeill J, Lovis C, et al. Automatic medical encoding with SNOMED categories. *BMC Med Inform Decis Mak*. 2008;**8 Suppl 1**:S6.

58 Zou Q, Chu WW, Morioka C, et al. IndexFinder: a method of extracting key concepts from clinical texts for indexing. *AMIA Annu Symp Proc*; 2003.

59 Friedman C. A broad-coverage natural language processing system. *AMIA Annu Symp Proc*; 2000.

60 Xu H, Stenner SP, Doan S, et al. MedEx: a medication information extraction system for clinical narratives. *J Am Med Inform Assoc*. 2010;**17**:19-24.

61 Tuttle MS, Olson NE, Keck KD, et al. Metaphrase: an aid to the clinical conceptualization and formalization of patient problems in healthcare enterprises. *Methods Inf Med*. 1998;**37**:373-83.

62 Elkin PL, Cimino JJ, Lowe HJ, et al. Mapping to MeSH: The art of trapping MeSH equivalence from within narrative text. *Proc 12th SCAMC*; 1988.

63 Zhou L, Plasek JM, Mahoney LM, et al. Using Medical Text Extraction, Reasoning and Mapping System (MTERMS) to process medication information in outpatient clinical notes. *AMIA Annu Symp Proc*; 2011.

64 Srinivasan S, Rindflesch TC, Hole WT, et al. Finding UMLS Metathesaurus concepts in MEDLINE. *AMIA Annu Symp Proc*; 2002.

65 Hersh W, Hickam DH, Haynes RB, et al. Evaluation of SAPHIRE: an automated approach to indexing and retrieving medical literature. *Proc Annu Symp Comput Appl Med Care*; 1991.

66 Zieman YL, Bleich HL. Conceptual mapping of user's queries to medical subject headings. *Proc AMIA Annu Fall Symp*; 1997.

67 Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. *J Am Med Inform Assoc*. 2010;**17**:229-36.

68 Pakhomov SV, Buntrock JD, Chute CG. Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *J Am Med Inform Assoc*. 2006;**13**:516-25.

69 Elkin PL, Froehling D, Wahner-Roedler D, et al. NLP-based identification of pneumonia cases from free-text radiological reports. *AMIA Annu Symp Proc*; 2008.

70 Uzuner O, South BR, Shen S, et al. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*. 2011;**18**:552-6.

71 Nigam K, McCallum A, Mitchell T. Semi-supervised text classification using EM. In: Chapelle O, Schölkopf B, Zien A, editors. Semi-supervised Learning. Cambridge, Mass.: MIT Press; 2006.

72 Alpaydin E. *Introduction to Machine Learning*. Cambridge, Mass.: MIT Press; 2004.

73 Manning CD, Raghavan P, Schütze H. *Introduction to Information Retrieval*. New York: Cambridge University Press; 2008.

74 Steyberg EW, Harrell FE, Borsboom GJJM, et al. Internal validation of predictive models: efficiency of some procedures for logistic regression analysis. *J Clin Epidem*. 2001;**54**:774-81.

75 Forman G, Scholz M. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newletter*. 2010;**12**:49-57.

76 National Library of Medicine NIoH. Lexical Tools. 2013, Access Date: 2013; URL: http://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/2013/docs/userDoc/tools/lvg.html

77 Charniak E. Immediate-head parsing for language models. *Proc 39th Annl Meeting Assoc Comput Linguistics*; 2001.

78 Gibbs JP, Poston DL. The division of labor: conceptualization and Related Measures. *Social Forces*. 1975;**53**:468-76.

79 Mitchell TM. *Machine Learning*. Boston, MA: McGraw-Hill; 1997.

80 de Bruijn B, Cherry C, Kiritchenko S, et al. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *J Am Med Inform Assoc*. 2011;**18**:557-62.

81 Chowdhury MFM, Lavelli A. Disease mention recognition with specific features. *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*. 2010:83-90.

82 D'Avolio LW, Nguyen TM, Goryachev S, et al. Automated concept-level information extraction to reduce the need for custom software and rules development. *J Am Med Inform Assoc*. 2011;**18**:607-13.

83 Brown SH, Elkin PL, Bauer BA, et al. SNOMED CT: utility for a general medical evaluation template. *AMIA Annu Symp Proc*; 2006.

84 Bonow RO, Bennett S, Casey DE, Jr., et al. ACC/AHA Clinical Performance Measures for Adults with Chronic Heart Failure: a report of the American College of Cardiology/American Heart Association Task Force on Performance Measures (Writing Committee to Develop Heart Failure Clinical Performance Measures): endorsed by the Heart Failure Society of America. *Circulation*. 2005;**112**:1853-87.

85 Juckett D. A method for determining the number of documents needed for a gold standard corpus. *J Biomed Inform*. 2012;**45**:460-70.

86 Thompson CA, Califf ME, Mooney RJ. Active Learning for Natural Language Parsing and Information Extraction. *Sixteenth International Conference on Machine Learning*: Morgan Kaufmann Publishers Inc.; 1999.

87 Olsson F. A Literature Survey of Active Machine Learning in the Context of Natural Language Processing: Swedish Institute of Computer Science (SICS) Technical Report; 2009. Report No.: T2009:06.

88 Dagan I, Engleson SP. Committee-based sampling for traiining probabilistic classifiers. *Twelfth Internation Conference on Machine Learning*; Tahoe City, California: Morgan Kaufmann; 1995.

89 Ringger E, McClanahan P, Haertel R, et al. Active learning for part-of-speech tagging: accelerating corpus annotation. *Linguistic Annotation Workshop*; Prague, Czech Republic: Association for Computational Linguistics; 2007.

90 McCallum A, Nigam K. Employing EM and Pool-Based Active Learning for Text Classification. *Fifteenth International Conference on Machine Learning*: Morgan Kaufmann Publishers Inc.; 1998.

91 Lewis DD, Gale WA. A sequential algorithm for training text classifiers. *17th annual international ACM SIGIR conference on Research and development in information retrieval*; Dublin, Ireland: Springer-Verlag New York, Inc.; 1994.

92 Hachey B, Beatrice A, Becker M. Investigating the effects of selective sampling on the annotation task. *Ninth Conference on Computational Natural Language Learning*; Ann Arbor, Michigan: Association for Computational Linguistics; 2005.

93 Vlachos A. Active Annotation. *Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*; Trento, Italy; 2006.

94 Chen Y, Mani S, Xu H. Applying active learning to assertion classification of concepts in clinical text. *J Biomed Inform*. 2012;**45**:265-72.

95 Fort K, Sagot B. Influence of pre-annotation on POS-tagged corpus development. *Fourth Linguistic Annotation Workshop*; Uppsala, Sweden: Association for Computational Linguistics; 2010.

96 Ringger E, Carmen M, Haertel R, et al. Assessing the Costs of Machine-Assisted Corpus Annotation through a User Study. *Sixth International Conference on Language Resources and Evaluation (LREC'08)*; Marrakech, Morocco: European Language Resources Association (ELRA); 2008.

97 Rehbein I, Ruppenhofer J, Sporleder C. Assessing the benefits of partial automatic pre-labeling for frame-semantic annotation. *Third Linguistic Annotation Workshop*; Suntec, Singapore: Association for Computational Linguistics; 2009.

98 Ogren PV, Savova G, Chute C. Constructing evaluation corpora for automated clinical named entity recognition. *Language Resources and Evaluation Conference (LREC)*; 2008.

99 Dandapat S, Biswas P, Choudhury M, et al. Complex linguistic annotation --- no easy way out!: a case from Bangla and Hindi POS labeling tasks. *Third Linguistic Annotation Workshop*; Suntec, Singapore: Association for Computational Linguistics; 2009.

100 Yancy CW, Jessup M, Bozkurt B, et al. 2013 ACCF/AHA Guideline for the Management of Heart Failure: A Report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines. *Circulation*. 2013;10.1161/CIR.0b013e31829e8776.

101 Ogren PV. Knowtator: a Protégé plug-in for annotated corpus construction. *North American Chapter of the Association for Computational Linguistics on Human Language Technology*; New York, New York: Association for Computation Linguistics; 2006.

102 Schneider K-M. Techniques for improving the performance of naive bayes for text classification. *6th international conference on Computational Linguistics and Intelligent Text Processing*; Mexico City, Mexico: Springer-Verlag; 2005.

103 Culotta A, Kristjansson T, McCallum A, et al. Corrective feedback and persistent learning for information extraction. *Artif Intell*. 2006;**170**:1101-22.

104 Aberdeen J, Bayer S, Yeniterzi R, et al. The MITRE Identification Scrubber Toolkit: design, training, and assessment. *Int J Med Inform*. 2010;**79**:849-59.

105 Fragkou P, Petasis G, Theodorakos A, et al. Boemie ontology-based text annotation tool. *6th International Conference on Language Resources and Evaluation (LREC)*; Marrakech, Morocco; 2008.

106 Chapman WW, Bridewell W, Hanbury P, et al. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform*. 2001;**34**:301-10.

107 Agarwal S, Yu H. Biomedical negation scope detection with conditional random fields. *J Am Med Inform Assoc*. 2010;**17**:696-701.

108 Chapman WW, Chu D, Dowling JN. ConText: An algorithm for identifying contextual features from clinical text. *BioNLP 2007: Biological, translational, and clinical language processing*; Prague, Czech Republic; 2007.

109 Wallach HM. Conditional Random Fields: An Introduction. Philadelphia, PA: Department of Computer and Information Sciences Technical Reports, University of Pennsylvania; 2004. Report No.: CIS Technical Report MS-CIS-04-21.

110 Sutton C, McCallum A. Conditional Random Fields: An Introduction. *Foundations and Trends in Machine Learning*. 2011;**4**:267-373.

111 He Z, Wang H. A Comparison and Improvement of Online Learning Algorithms for Sequence Labeling. *COLING*; Mumbai, India; 2012.

112 Garla V, Lo Re V, 3rd, Dorey-Stein Z, et al. The Yale cTAKES extensions for document classification: architecture and application. *J Am Med Inform Assoc*. 2011;**18**:614-20.

113 Borodin Y, Polishchuk V, Mahmud J, et al. Live and learn from mistakes: A lightweight system for document classification. *Inf Process Manage*. 2013;**49**:83-98.

114 Hersh W, Hickam DH, Haynes RB, et al. Evaluation of SAPHIRE: an automated approach to indexing and retrieving medical literature. *Proc Annu Symp Comput Appl Med Care*. 1991:808-12.

115 Cimino JJ, Elhanan G, Zeng Q. Supporting infobuttons with terminological knowledge. *AMIA Annu Symp Proc*. 1997:528-32.

116 Cimino JJ, Friedmann BE, Jackson KM, et al. Redesign of the Columbia University Infobutton Manager. *AMIA Annu Symp Proc*. 2007:135-9.

117 Del Fiol G, Haug PJ. Infobuttons and classification models: a method for the automatic selection of on-line information resources to fulfill clinicians' information needs. *J Biomed Inform*. 2008;**41**:655-66.